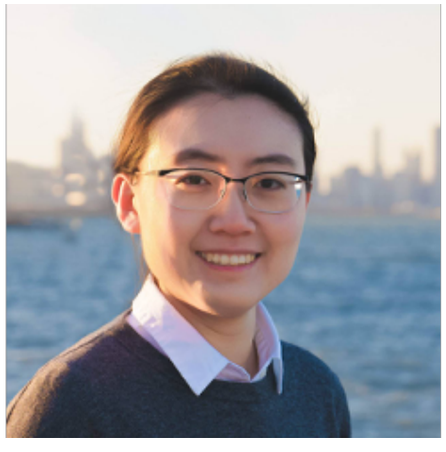
Knowledge Distillation for Language Models: Challenges and Opportunities with Sequential Data

Yuqiao Wen[1,3] Freda Shi[2,4,5] Lili Mou[1,3,5]


yq.when@gmail.com fhs@uwaterloo.ca doublepower.mou@gmail.com

[1]Dept. Computing Science, University of Alberta

[2]David R. Cheriton School of Computer Science, University of Waterloo

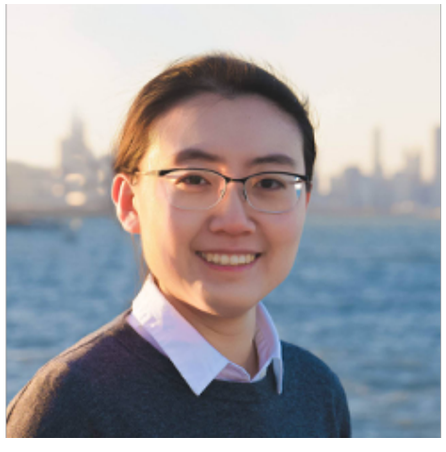
[3]Alberta Machine Intelligence Institute (Amii)

[4]Vector Institute [5]Canada CIFAR AI Chair

AAAI'26 Tutorial

Knowledge Distillation for Language Models: Challenges and Opportunities with Sequential Data

Yuqiao Wen[1,3] Freda Shi[2,4,5] **Lili Mou**[1,3,5]


yq.when@gmail.com fhs@uwaterloo.ca doublepower.mou@gmail.com

[1]Dept. Computing Science, University of Alberta

[2]David R. Cheriton School of Computer Science, University of Waterloo

[3]Alberta Machine Intelligence Institute (Amii)

[4]Vector Institute [5]Canada CIFAR AI Chair

AAAI'26 Tutorial

Lili Mou is admitting

- All-level students
  MSc, PhD, exchanging

- Visiting scholars
  RA, Postdoc

Lili Mou (LLM)’s Lab

Lili Mou (LLM)'s Lab strictly follows the **KD methodologies** discussed in this tutorial

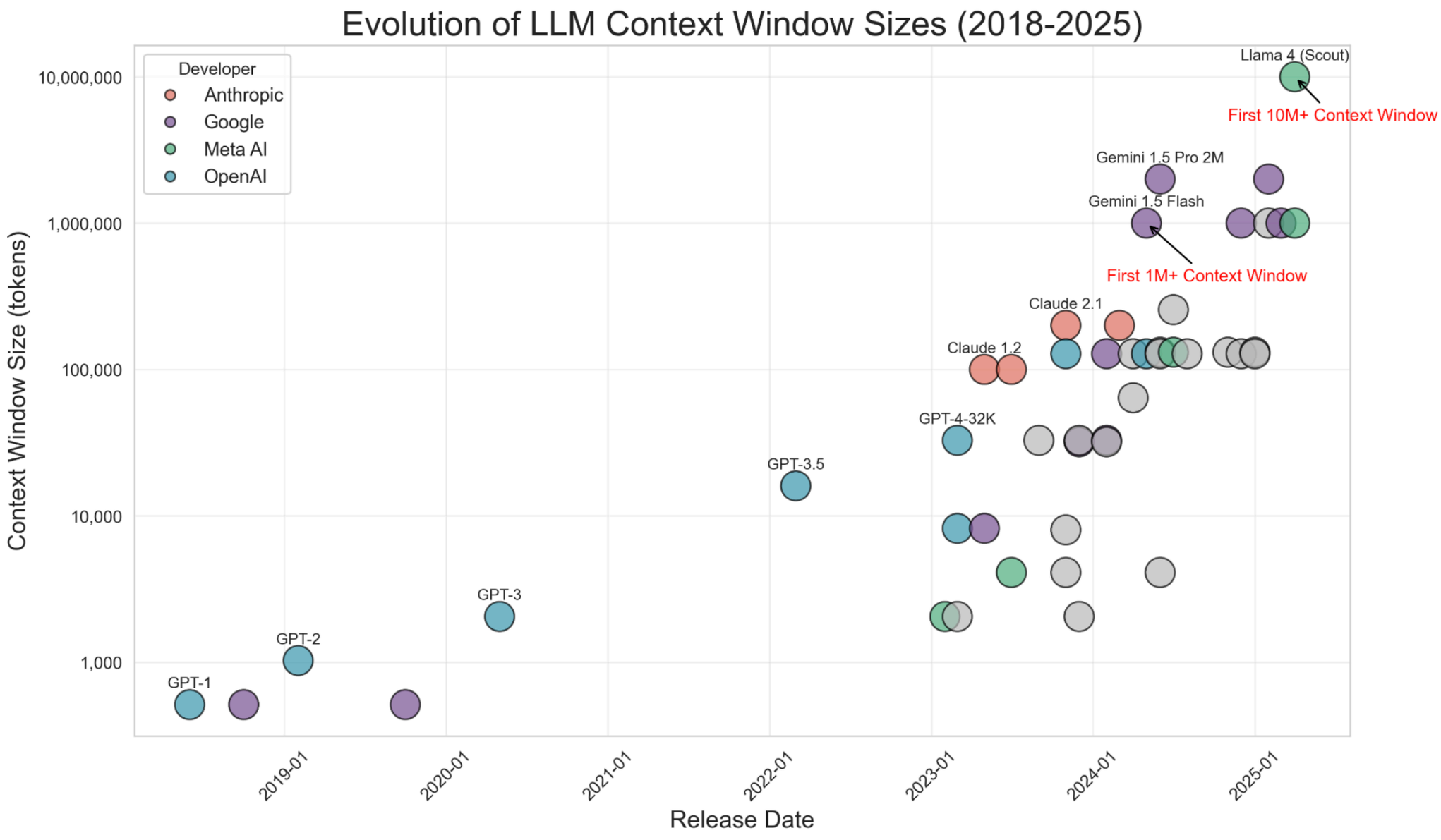
Tutorial Outline

- **Session I [45min]:**
  - **Introduction**
  - f-Divergence KD [ACL'23]
  - RL for LLM Distillation [NeurIPS'23, COLING'24]
- Break [15min]
- Session II [15min]
  - Multi-Teacher Distillation [ICLR'24, ACL'24, AAAI'25a,b]
  - KD Applications to LLMs
  - Conclusion and Future Work

UNIVERSITY OF ALBERTA · UNIVERSITY OF WATERLOO · amii · VECTOR INSTITUTE · CIFAR

Why knowledge distillation?

- Model size grows exponentially



Source: https://www.meibel.ai/post/understanding-the-impact-of-increasing-llm-context-windows

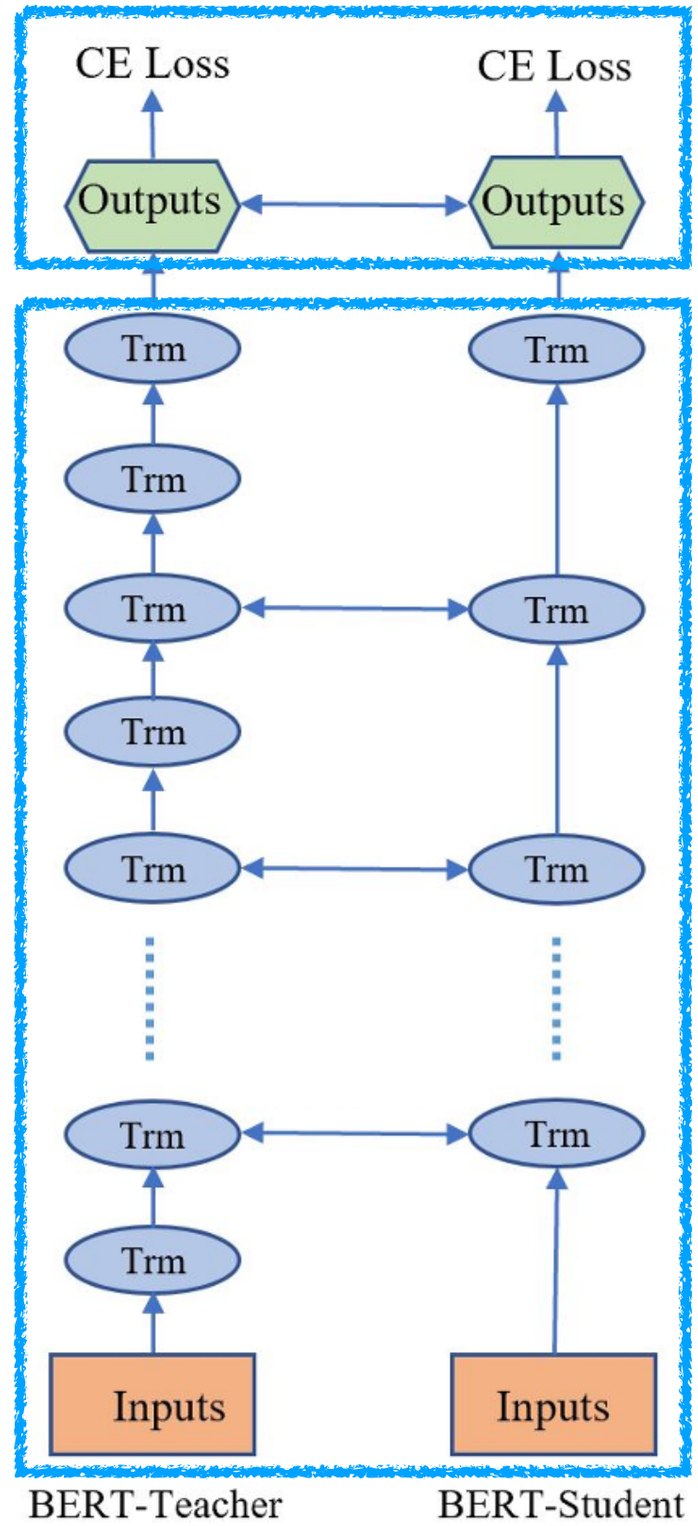
Why knowledge distillation?

- Knowledge distillation (KD): Transferring knowledge
  - From: a large teacher model
  - To: a small student model
- KD improves efficiency
  - Parameter efficiency
    - Memory, storage, preventing overfitting
  - Time efficiency
    - Fine-tuning, inference

Categorization of KD Methods

- Output matching
  - Mandatory
  - Informs student of the task

- Intermediate-layer matching
  - Student learns step by step
  - Optional but often helpful



[Figure adapted from Sun et al. (2019)]

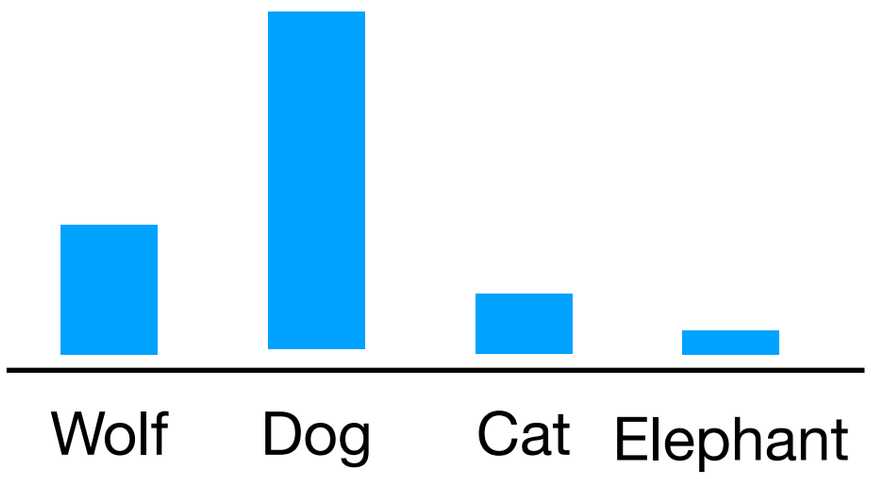
Output Matching

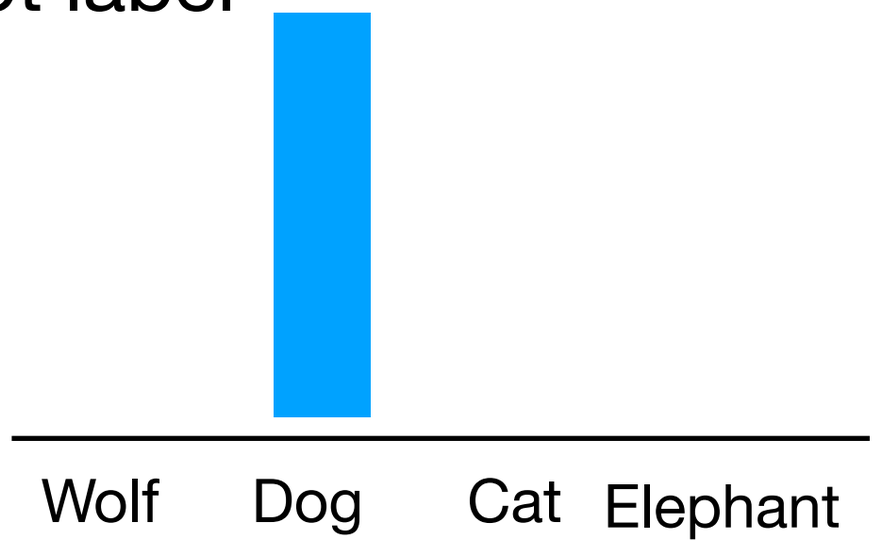
- (Soft) cross-entropy loss

$$\sum_{x} -p(x)\log q_\theta(x)$$

  - $p(x)$: Teacher distribution (may adjust temperature)
  - $q_\theta(x)$: Student distribution

- Why soft cross-entropy loss?
  - Teacher provides a full distribution
  - More informative than a one-hot label





Ranking-Based Output Matching

- Learning a student model Q to rank samples consistently with the teacher P.

$$\arg\min_{Q} \mathbb{E}_{\langle x^+, x- \rangle} \left[-\log Q(x^+) + \log Q(x^-) \right]$$
$$s.t.\ P(x^+) > P(x^-)$$

- Example application in LM alignment (Tunstall et al., 2023):
  - Learning to rank the output sequences given input.

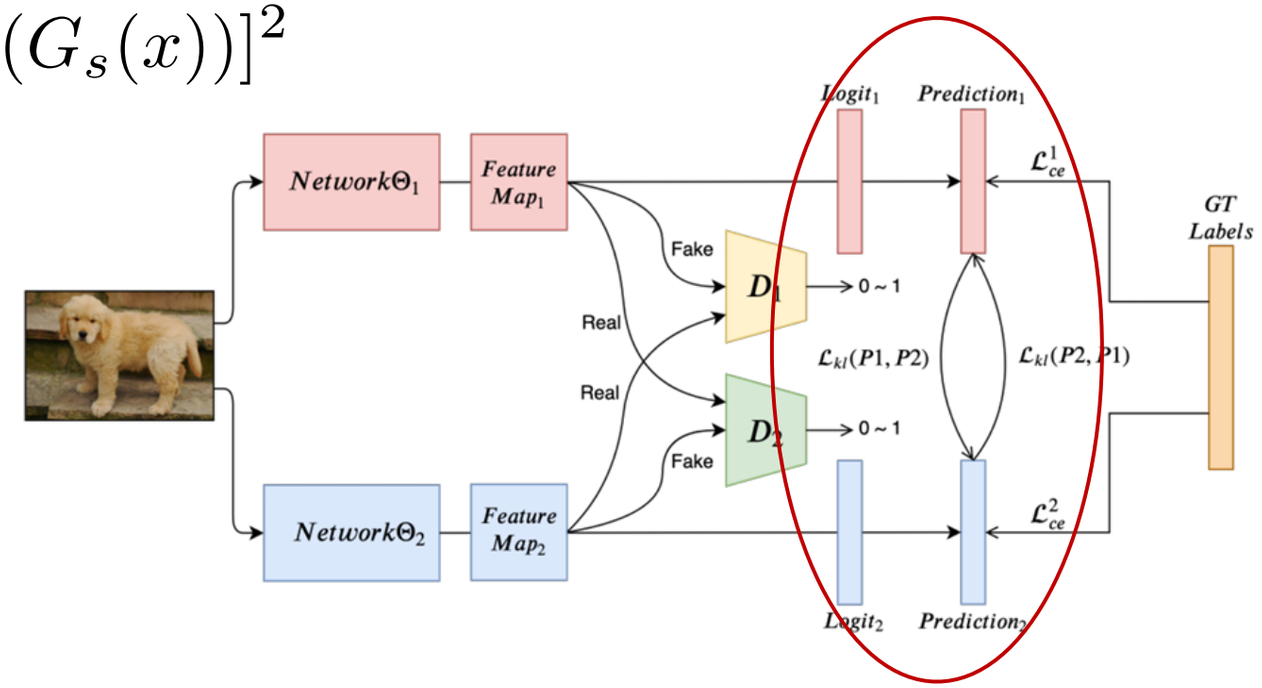
Intermediate-Layer Matching

- Guides the student to learn step by step
- L_2 loss with linear mapping (Romero et al., 2015)

$$\mathcal{L}_{\text{matching}} = \sum_{i=1}^{N} \sum_{j=1}^{L} ||\mathbf{t}_{i,j} - \alpha\left(\mathbf{W}\mathbf{s}_{i,j}\right)||_2^2$$

- Adversarial loss (Chung et al., 2020)

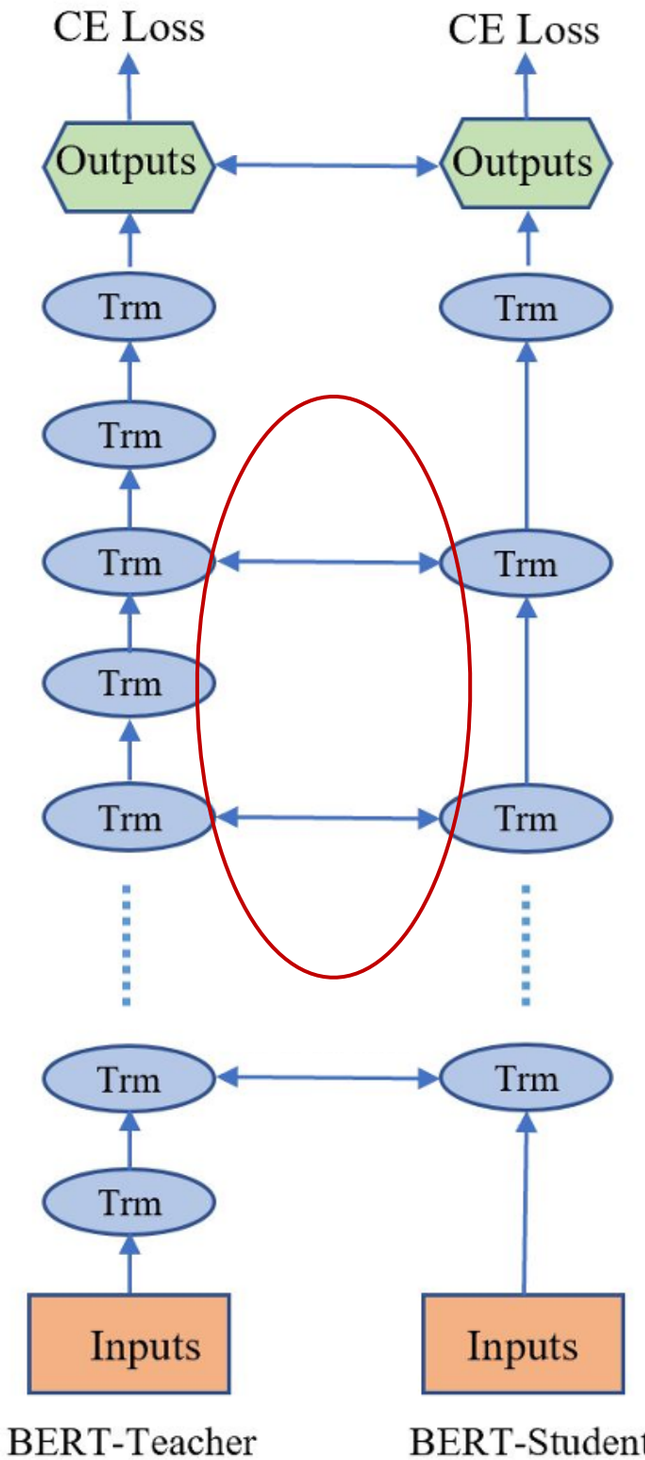
$$\mathcal{L}_{\text{disc}} = [1 - D_s(G_t(x))]^2 + [D_s(G_s(x))]^2$$
$$\mathcal{L}_{\text{gen}} = [1 - D_s(G_s(x))]^2$$



[Figure adapted from Chung et al. (2020)]

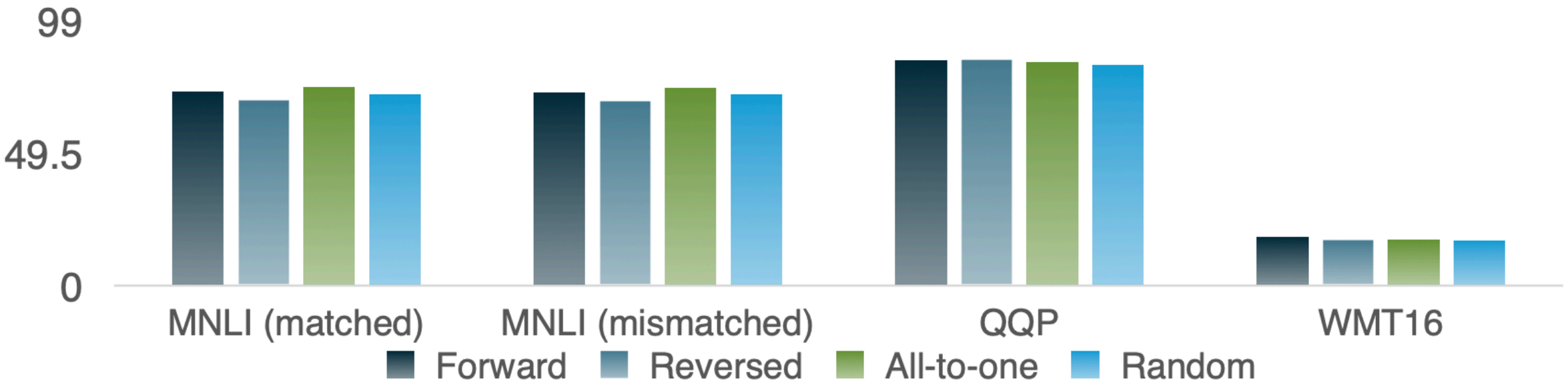
Layer Selection

- Which layer is matched to which?
- Previous strategies
  - Evenly spaced (Sun et al., 2019)
  - All-to-one (Wang et al., 2020)
  - Weighted combination (Passban et al., 2021)
- Our recent findings (Yu et al., 2025)
  - Intermediate-layer matching is important
  - Layer selection is not

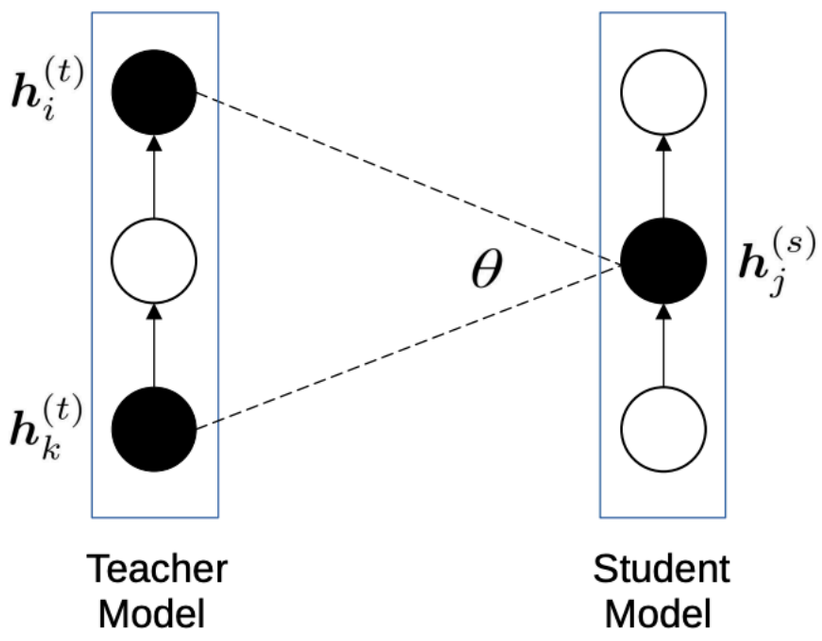


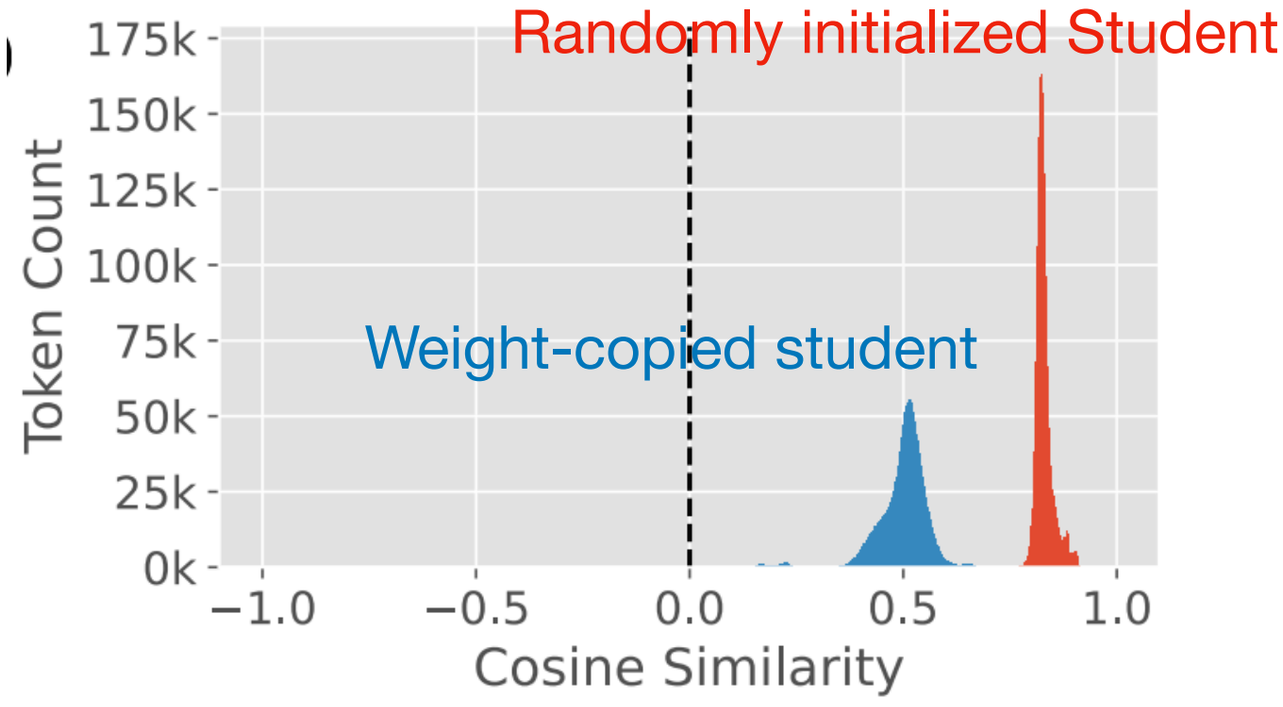
[Figure adapted from Sun et al. (2019)]

Layer Selection Performance



- A plausible explanation:
  - Between-layer distance << between-model distance

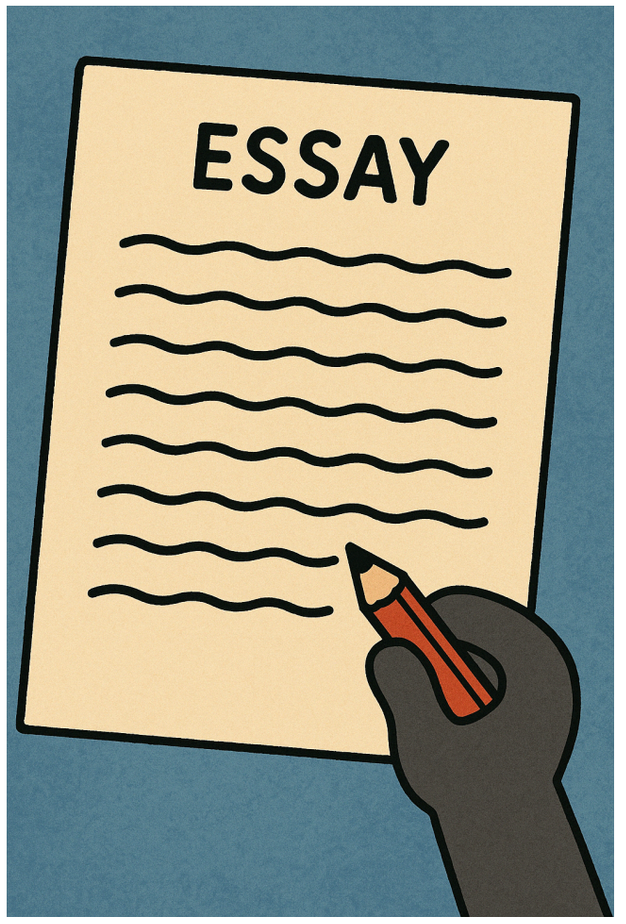




Difficulty of Sequential Generation

Classification

Generation



Cartoons generated by ChatGPT

Difficulty of Sequential Generation

- Larger search space
  - More categories (e.g., 50K tokens)
  - Exponential growth wrt length

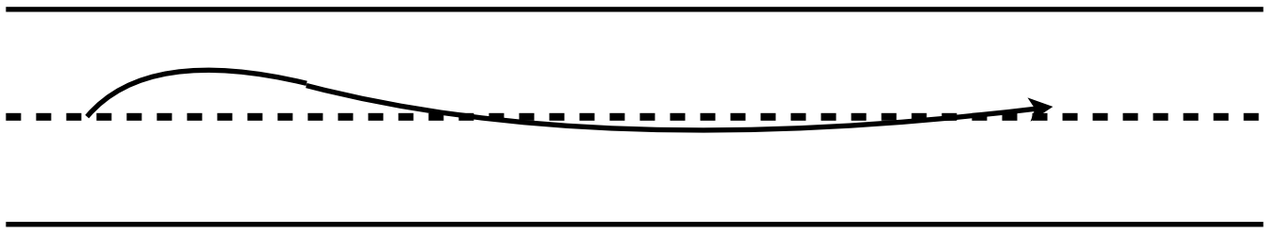
Difficulty of Sequential Generation

- Larger search space

- Exposure bias
  - Teacher-force training
  - Self-generation

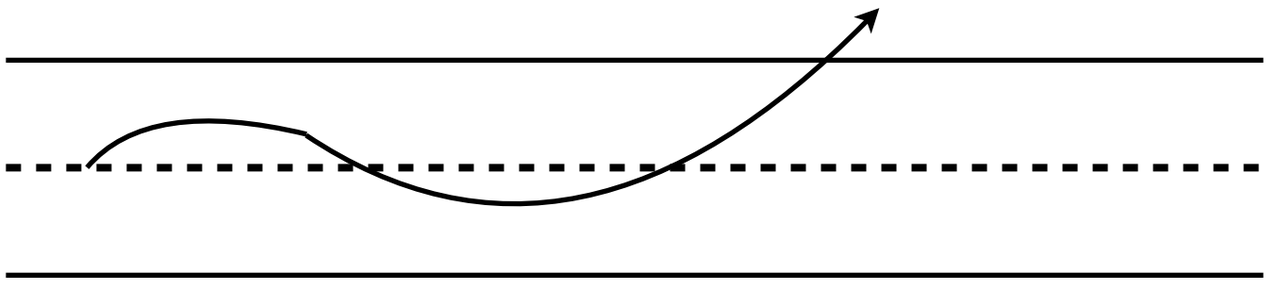
Analogy: Autonomous Driving

Training phase



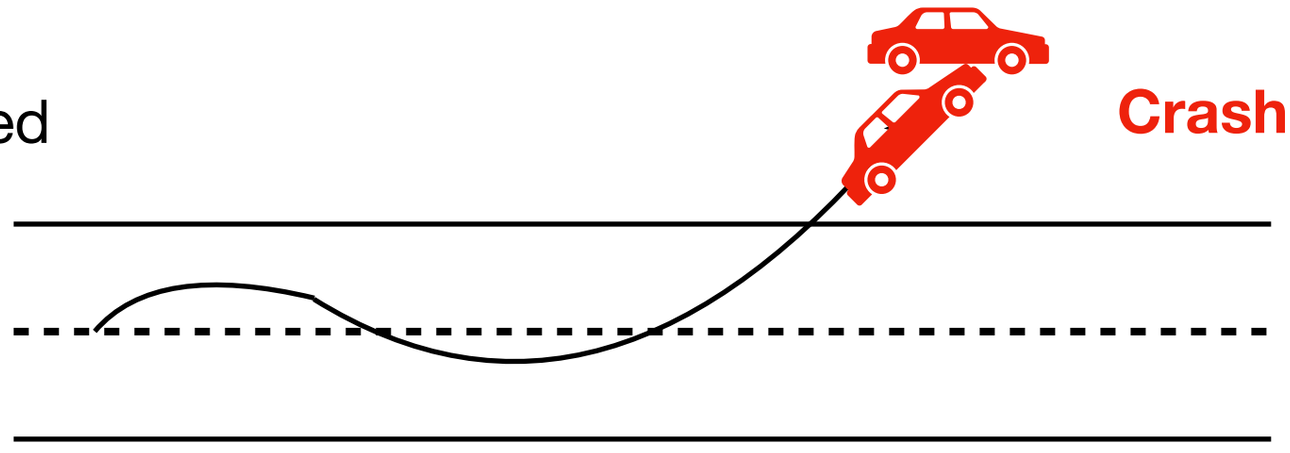
Inference phrase (deployment)

Now what?

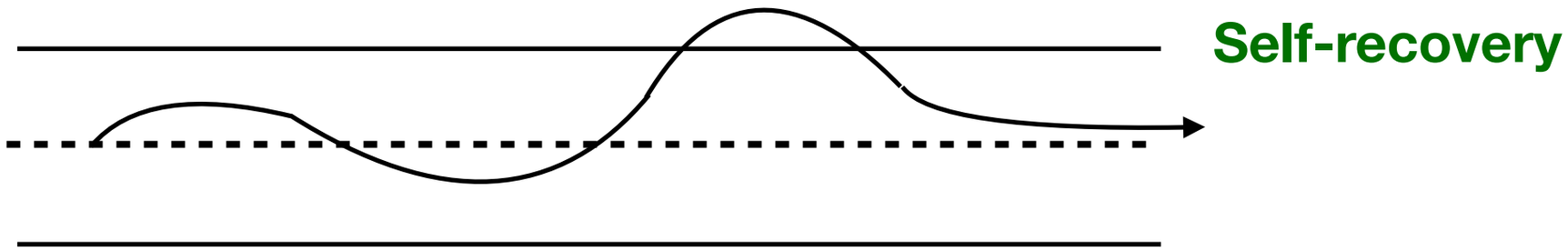


Analogy: Autonomous Driving

Undesired

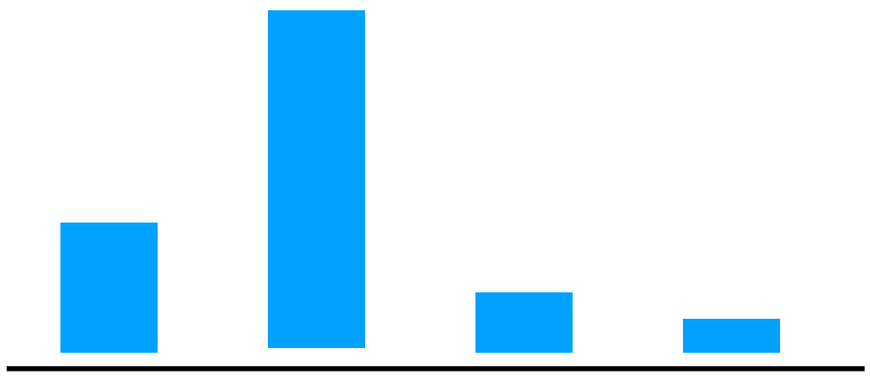


Desired

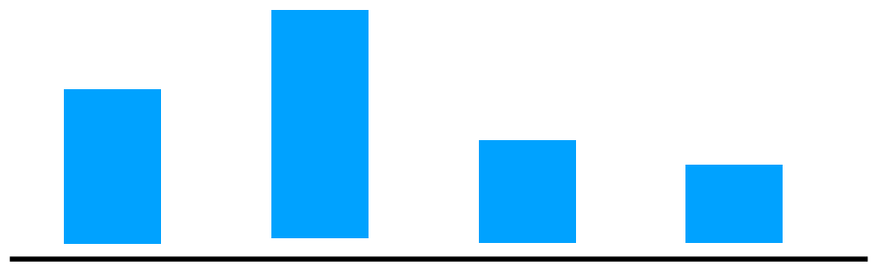


Difficulty of Sequential Generation

- Larger search space
- Exposure bias
- Peakness is important!

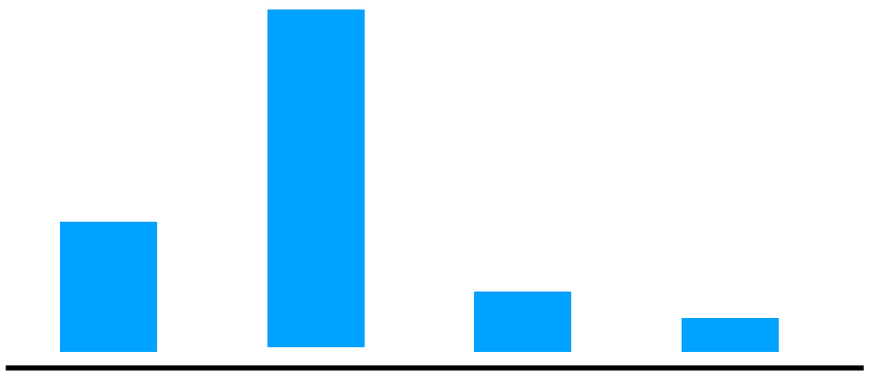


Low temperature

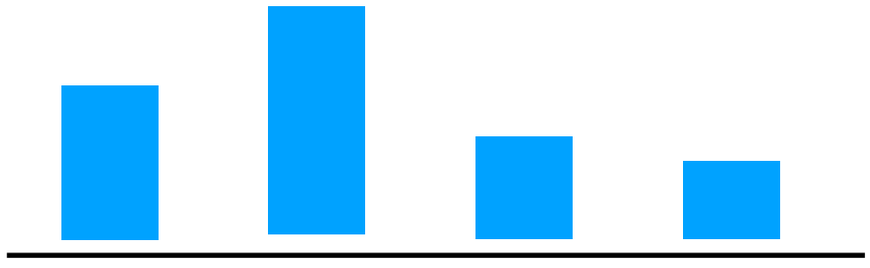


High temperature

Peakness of Prediction



Low temperature



High temperature

- Classification: argmax is unchanged
- Generation (structured prediction)

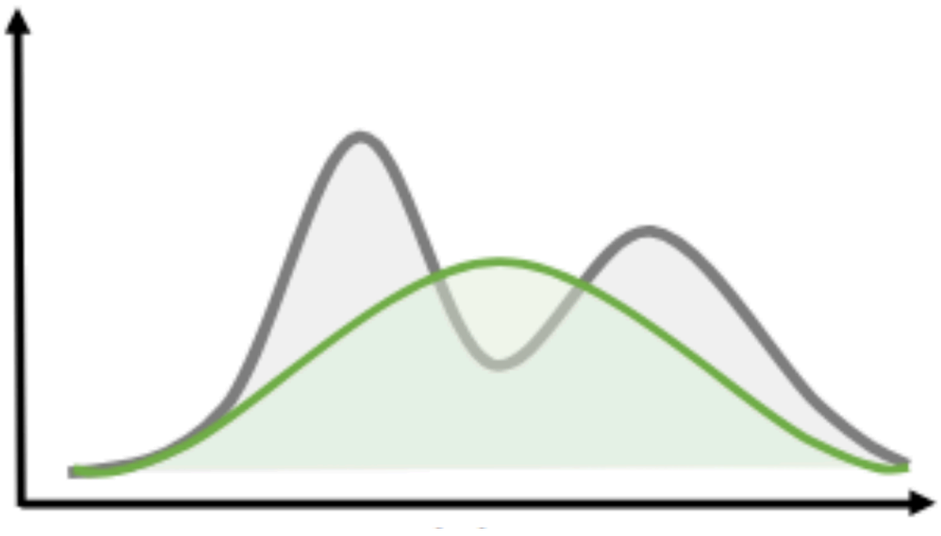
$$P(\text{structure}) = \prod P(\text{step})$$

Different steps considered jointly
Temperature affects voting power

ML Models Don’t Learn the Right Peakness

$$\mathcal{L} = \sum_{y} p(y) \log \frac{p(y)}{q_\theta(y)}$$

- If target $p(\cdot)$ is one-hot
  - Non-target labels are not learned explicitly
- Even if $p(\cdot)$ is a full distribution
  - ML tends to give overly smooth distribution



Tutorial Outline

- **Session I [45min]:**
  - Introduction
  - **f-Divergence KD [ACL'23]**
  - RL for LLM Distillation [NeurIPS'23, COLING'24]

- Break [15min]

- Session II [15min]
  - Multi-Teacher Distillation [ICLR'24, ACL'24, AAAI'25a,b]
  - KD Applications to LLMs
  - Conclusion and Future Work

Straightforward Attempt

- SeqKD [Kim and Rush, 2016]

$$J_{\text{SeqKD}} = \mathbb{E}_{\mathbf{Y}\sim p}[-\log q_\theta(\mathbf{Y})]$$

$$\approx -\sum_{t=1}^{|\mathbf{y}|} \log q_\theta(\mathrm{y}_t \mid \mathbf{y}_{<t})$$

  - Teacher predicts a sequence
  - Student learns in a pseudo-supervised way
  - Overcomes large space by hard KD
    - hard prefix / hard label

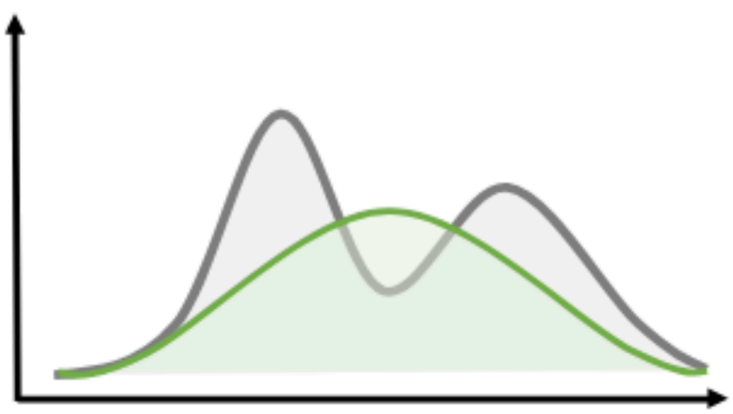
Straightforward Attempt

- SeqKD [Kim and Rush, 2016]

$$J_{\text{SeqKD}} = \mathbb{E}_{\mathbf{Y}\sim p}[-\log q_\theta(\mathbf{Y})]$$

$$\approx -\sum_{t=1}^{|\mathbf{y}|} \log q_\theta(\mathrm{y}_t \mid \mathbf{y}_{<t})$$

- Drawback



$$P(\text{structure}) = \prod P(\text{step})$$

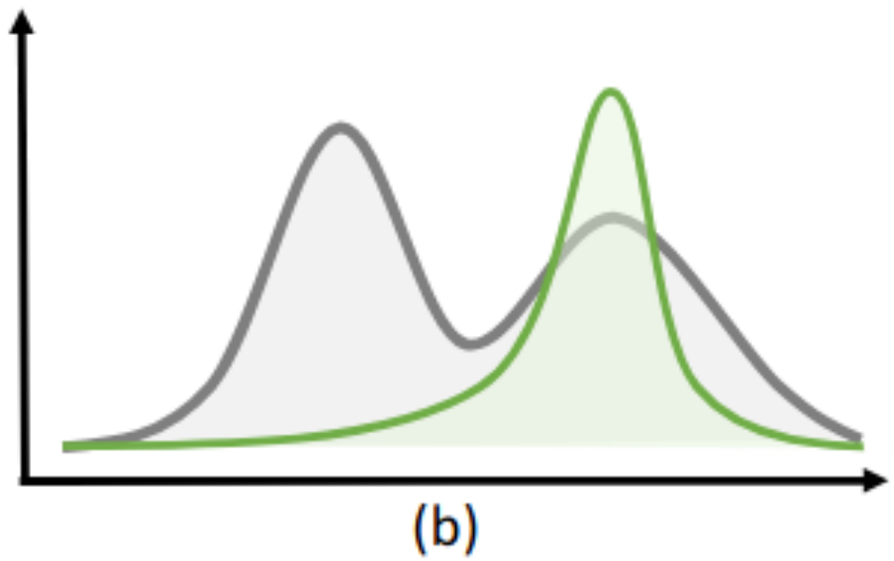
Different steps considered jointly
Temperature affects voting power

Reverse KL

- ENGINE

$$J_{\text{ENGINE}} = \mathbb{E}_{\mathbf{Y}' \sim q_\theta}[-\log p(\mathbf{Y}')] \approx -\sum_{t=1}^{|\mathbf{y}'|} \log p(\mathrm{y}'_t \,|\, \mathbf{y}'_{<t})$$

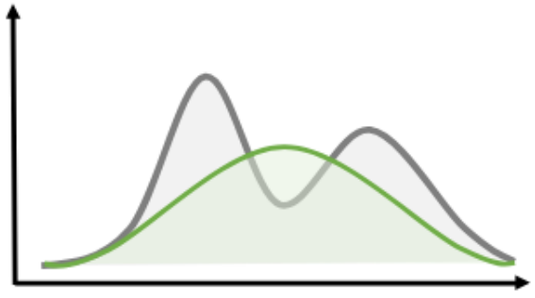
- Drawback



KL vs RKL Distillation

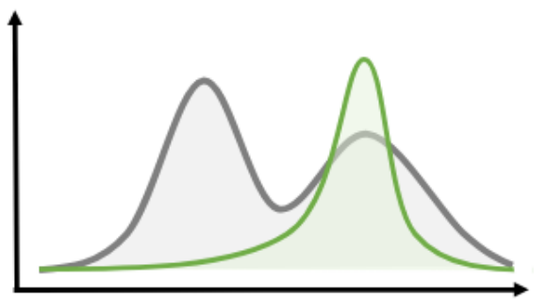
$$J_{\mathrm{KL}} = D_{\mathrm{KL}}(p \| q_\theta) = \mathbb{E}_{\mathbf{Y} \sim p}\left[\log \frac{p(\mathbf{Y})}{q_\theta(\mathbf{Y})}\right]$$

Mode averaging



$$J_{\mathrm{RKL}} = D_{\mathrm{KL}}(q_\theta \| p) = \mathbb{E}_{\mathbf{Y}' \sim q_\theta}\left[\log \frac{q_\theta(\mathbf{Y}')}{p(\mathbf{Y}')}\right]$$

Mode collapsing



Disclaimer: Discrete distributions don’t have modes. Just an analogy.

f-divergence

$$D_f(p(t)\|q(t)) = \sum_t q(t)\, f\left(\frac{p(t)}{q(t)}\right)$$

Divergence	$f(t)$
Kullback–Leibler (KL)	$t \log t$
Reverse KL (RKL)	$-\log t$
Jensen–Shannon (JS)	$-(t+1)\log(\frac{t+1}{2}) + t \log t$
Total variation distance (TVD)	$\frac{1}{2}\lvert t-1\rvert$

Symmetric Distillation

- Asymmetric distillation
  - $\mathrm{KL}(p \| q_\theta) \neq \mathrm{KL}(q_\theta \| p)$
  - $\mathrm{RKL}(p \| q_\theta) \neq \mathrm{RKL}(q_\theta \| p)$
  - Severe mode averaging or collapsing 😞
- Symmetric distillation
  - Jensen—Shannon (JS) distillation
  - Total variation distance (TVD) distillation
  - Balance between the two extremes 😊

JS Distillation

$$J_{\mathrm{JS}} = \frac{1}{2}\mathbb{E}_{\mathbf{Y}\sim p}\left[\log \frac{p(\mathbf{Y})}{m(\mathbf{Y})}\right] + \frac{1}{2}\mathbb{E}_{\mathbf{Y}'\sim q_\theta}\left[\log \frac{q_\theta(\mathbf{Y}')}{m(\mathbf{Y}')}\right]$$

- Denominator is a"Middle" distribution
- Non-zero if either $p(\cdot)$ or $q_\theta(\cdot)$ is not zero

$$m(\cdot) = \frac{1}{2}p(\cdot) + \frac{1}{2}q_\theta(\cdot)$$

JS Distillation

$$J_{\text{JS}} = \frac{1}{2}\mathbb{E}_{\mathbf{Y}\sim p}\left[\log\frac{p(\mathbf{Y})}{m(\mathbf{Y})}\right] + \frac{1}{2}\mathbb{E}_{\mathbf{Y}'\sim q_\theta}\left[\log\frac{q_\theta(\mathbf{Y}')}{m(\mathbf{Y}')}\right]$$

$$\approx \frac{1}{2}\sum_{t=1}^{|\mathbf{y}|}\sum_{\mathrm{Y}_t\in V} -p(\mathrm{Y}_t\,|\,\mathbf{y}_{<t})\log(m(\mathrm{Y}_t\,|\,\mathbf{y}_{<t}))$$

MC sampling for prefix

Step-wise decomposition

$$+\frac{1}{2}\sum_{t=1}^{|\mathbf{y}'|}\sum_{\mathrm{Y}'_t\in V}\left[q_\theta(\mathrm{Y}'_t\,|\,\mathbf{y}'_{<t})\log(q_\theta(\mathrm{Y}'_t\,|\,\mathbf{y}'_{<t})\right.$$

$$\left.-q_\theta(\mathrm{Y}'_t\,|\,\mathbf{y}'_{<t})\log(m(\mathrm{Y}'_t\,|\,\mathbf{y}'_{<t}))\right] + \text{const}$$

TVD Distillation

$$J_{\text{TVD}} = \frac{1}{2} \sum_{\mathbf{Y}} | \, q_\theta(\mathbf{Y}) - p(\mathbf{Y}) \, |$$

TVD Distillation

$$J_{\mathrm{TVD}} = \frac{1}{2} \sum_{\mathbf{Y}} | q_\theta(\mathbf{Y}) - p(\mathbf{Y}) |$$

$$\leq \frac{1}{4} \mathbb{E}_{\mathbf{Y} \sim p} \left[\sum_{t=1}^{|\mathbf{Y}|} \sum_{\mathrm{Y}_t \in V} | q_\theta(\mathrm{Y}_t | \mathbf{Y}_{<t}) - p(\mathrm{Y}_t | \mathbf{Y}_{<t}) | \right]$$

$$+ \frac{1}{4} \mathbb{E}_{\mathbf{Y}' \sim q_\theta} \left[\sum_{t=1}^{|\mathbf{Y}'|} \sum_{\mathrm{Y}'_t \in V} | q_\theta(\mathrm{Y}'_t | \mathbf{Y}'_{<t}) - p(\mathrm{Y}'_t | \mathbf{Y}'_{<t}) | \right]$$

Step-wise decomposition: Upper bound

Theorem 1. *(a) The sequence-level KL, RKL, and JS divergences can be decomposed exactly into step-wise terms. (b) The sequence-level TVD can be upper bounded by step-wise terms.*

TVD Distillation

$$J_{\text{TVD}} = \frac{1}{2} \sum_{\mathbf{Y}} |\, q_\theta(\mathbf{Y}) - p(\mathbf{Y}) \,|$$

$$\leq \frac{1}{4} \mathbb{E}_{\mathbf{Y} \sim p} \left[\sum_{t=1}^{|\mathbf{Y}|} \sum_{\mathrm{Y}_t \in V} |\, q_\theta(\mathrm{Y}_t \,|\, \mathbf{Y}_{<t}) - p(\mathrm{Y}_t \,|\, \mathbf{Y}_{<t}) \,| \right]$$

$$+ \frac{1}{4} \mathbb{E}_{\mathbf{Y}' \sim q_\theta} \left[\sum_{t=1}^{|\mathbf{Y}'|} \sum_{\mathrm{Y}'_t \in V} |\, q_\theta(\mathrm{Y}'_t \,|\, \mathbf{Y}'_{<t}) - p(\mathrm{Y}'_t \,|\, \mathbf{Y}'_{<t}) \,| \right]$$

$$\approx \frac{1}{4} \sum_{t=1}^{|\mathbf{y}|} \sum_{\mathrm{Y}_t \in V} |\, q_\theta(\mathrm{Y}_t \,|\, \mathbf{y}_{<t}) - p(\mathrm{Y}_t \,|\, \mathbf{y}_{<t}) \,|$$

MC sampling for prefix

$$+ \frac{1}{4} \sum_{t=1}^{|\mathbf{y}'|} \sum_{\mathrm{Y}'_t \in V} |\, q_\theta(\mathrm{Y}'_t \,|\, \mathbf{y}'_{<t}) - p(\mathrm{Y}'_t \,|\, \mathbf{y}'_{<t}) \,|$$

Sampling from Student

$$J_{\mathrm{JS}} = \frac{1}{2}\mathbb{E}_{\mathbf{Y}\sim p}\left[\log \frac{p(\mathbf{Y})}{m(\mathbf{Y})}\right] + \frac{1}{2}\mathbb{E}_{\mathbf{Y}'\sim q_\theta}\left[\log \frac{q_\theta(\mathbf{Y}')}{m(\mathbf{Y}')}\right]$$

$$J_{\mathrm{TVD}} = \frac{1}{2}\sum_{\mathbf{Y}} |\, q_\theta(\mathbf{Y}) - p(\mathbf{Y}) \,|$$

$$\leq \frac{1}{4}\mathbb{E}_{\mathbf{Y}\sim p}\left[\sum_{t=1}^{|\mathbf{Y}|}\sum_{\mathrm{Y}_t\in V} |\, q_\theta(\mathrm{Y}_t \,|\, \mathbf{Y}_{<t}) - p(\mathrm{Y}_t \,|\, \mathbf{Y}_{<t}) \,|\right]$$

$$+\frac{1}{4}\mathbb{E}_{\mathbf{Y}'\sim q_\theta}\left[\sum_{t=1}^{|\mathbf{Y}'|}\sum_{\mathrm{Y}'_t\in V} |\, q_\theta(\mathrm{Y}'_t \,|\, \mathbf{Y}'_{<t}) - p(\mathrm{Y}'_t \,|\, \mathbf{Y}'_{<t}) \,|\right]$$

Partially addressing exposure bias

Main Results

Model		DART					
		BLEU4↑	METEOR↑	TER↓	BERTScore↑	MoverScore↑	BLEURT↑
Teacher		48.56	39.28	45.45	83.04	68.17	40.56
Student	Non-distill (MLE)	43.12	35.71	49.97	79.76	65.65	29.10
	Pre-distill	45.60	36.99	47.10	81.39	66.75	34.08
	SeqKD	45.54	37.17	47.49	81.15	66.65	32.88
	ENGINE	44.40	36.51	50.63	80.18	66.20	30.94
	KL	46.24	37.45	46.89	81.60	67.07	35.31
	RKL	45.63	37.35	47.91	81.41	67.02	35.08
	JS	<u>46.85</u>	<u>37.75</u>	<u>46.50</u>	<u>81.93</u>	<u>67.30</u>	<u>36.81</u>
	TVD	**46.95**	**37.88**	**46.35**	**82.08**	**67.36**	**37.17**

Model		XSum			WMT16 EN-RO			Commonsense Dialogue		
		ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	BLEU4↑	chrF↑	TER↓	BLEU-1↑	BLEU-2↑	BERTScore↑
Teacher		45.12	22.26	37.18	25.82	55.76	60.57	11.67	5.03	47.69
Student	Non-distill (MLE)	30.00	10.67	24.40	19.90	49.79	69.48	10.23	3.56	45.15
	Pre-distill	40.58	17.79	32.55	20.68	50.51	68.38	9.95	3.63	46.22
	SeqKD	39.13	17.53	32.34	21.20	50.81	67.66	10.85	4.17	46.94
	ENGINE	39.19	16.18	31.23	17.65	48.37	84.02	10.13	4.26	46.91
	KL	41.28	18.98	33.71	21.45	51.12	**66.74**	9.81	3.52	45.80
	RKL	<u>41.69</u>	19.02	33.92	20.46	50.33	70.78	10.48	4.01	46.68
	JS	41.65	<u>19.22</u>	<u>34.03</u>	**21.91**	**51.5**	<u>66.86</u>	**11.55**	**4.83**	**47.61**
	TVD	**41.76**	**19.30**	**34.10**	<u>21.73</u>	<u>51.13</u>	66.94	<u>11.39</u>	<u>4.73</u>	<u>47.30</u>

Main Results

Model		DART					
		BLEU4↑	METEOR↑	TER↓	BERTScore↑	MoverScore↑	BLEURT↑
Teacher		48.56	39.28	45.45	83.04	68.17	40.56
	Non-distill (MLE)	43.12	35.71	49.97	79.76	65.65	29.10
Student	Pre-distill	45.60	36.99	47.10	81.39	66.75	34.08
	SeqKD	45.54	37.17	47.49	81.15	66.65	32.88
	ENGINE	44.40	36.51	50.63	80.18	66.20	30.94
	KL	46.24	37.45	46.89	81.60	67.07	35.31
	RKL	45.63	37.35	47.91	81.41	67.02	35.08
	JS	46.85	37.75	46.50	81.93	67.30	36.81
	TVD	**46.95**	**37.88**	**46.35**	**82.08**	**67.36**	**37.17**

Model		XSum			WMT16 EN-RO			Commonsense Dialogue		
		ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	BLEU4↑	chrF↑	TER↓	BLEU-1↑	BLEU-2↑	BERTScore↑
Teacher		45.12	22.26	37.18	25.82	55.76	60.57	11.67	5.03	47.69
	Non-distill (MLE)	30.00	10.67	24.40	19.90	49.79	69.48	10.23	3.56	45.15
Student	Pre-distill	40.58	17.79	32.55	20.68	50.51	68.38	9.95	3.63	46.22
	SeqKD	39.13	17.53	32.34	21.20	50.81	67.66	10.85	4.17	46.94
	ENGINE	39.19	16.18	31.23	17.65	48.37	84.02	10.13	4.26	46.91
	KL	41.28	18.98	33.71	21.45	51.12	**66.74**	9.81	3.52	45.80
	RKL	41.69	19.02	33.92	20.46	50.33	70.78	10.48	4.01	46.68
	JS	41.65	19.22	34.03	**21.91**	**51.5**	66.86	**11.55**	**4.83**	**47.61**
	TVD	**41.76**	**19.30**	**34.10**	21.73	51.13	66.94	11.39	4.73	47.30

- Teacher is significantly better
- Student struggles w/o distillation

Main Results

Model		DART					
		BLEU4↑	METEOR↑	TER↓	BERTScore↑	MoverScore↑	BLEURT↑
Teacher		48.56	39.28	45.45	83.04	68.17	40.56
Student	Non-distill (MLE)	43.12	35.71	49.97	79.76	65.65	29.10
	Pre-distill	45.60	36.99	47.10	81.39	66.75	34.08
	SeqKD	45.54	37.17	47.49	81.15	66.65	32.88
	ENGINE	44.40	36.51	50.63	80.18	66.20	30.94
	KL	46.24	37.45	46.89	81.60	67.07	35.31
	RKL	45.63	37.35	47.91	81.41	67.02	35.08
	JS	46.85	37.75	46.50	81.93	67.30	36.81
	TVD	**46.95**	**37.88**	**46.35**	**82.08**	**67.36**	**37.17**

Model		XSum			WMT16 EN-RO			Commonsense Dialogue		
		ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	BLEU4↑	chrF↑	TER↓	BLEU-1↑	BLEU-2↑	BERTScore↑
Teacher		45.12	22.26	37.18	25.82	55.76	60.57	11.67	5.03	47.69
Student	Non-distill (MLE)	30.00	10.67	24.40	19.90	49.79	69.48	10.23	3.56	45.15
	Pre-distill	40.58	17.79	32.55	20.68	50.51	68.38	9.95	3.63	46.22
	SeqKD	39.13	17.53	32.34	21.20	50.81	67.66	10.85	4.17	46.94
	ENGINE	39.19	16.18	31.23	17.65	48.37	84.02	10.13	4.26	46.91
	KL	41.28	18.98	33.71	21.45	51.12	**66.74**	9.81	3.52	45.80
	RKL	41.69	19.02	33.92	20.46	50.33	70.78	10.48	4.01	46.68
	JS	41.65	19.22	34.03	**21.91**	**51.5**	66.86	**11.55**	**4.83**	**47.61**
	TVD	**41.76**	**19.30**	**34.10**	21.73	51.13	66.94	11.39	4.73	47.30

- KD significantly improves performance

Main Results

Model		DART					
		BLEU4↑	METEOR↑	TER↓	BERTScore↑	MoverScore↑	BLEURT↑
Teacher		48.56	39.28	45.45	83.04	68.17	40.56
Student	Non-distill (MLE)	43.12	35.71	49.97	79.76	65.65	29.10
	Pre-distill	45.60	36.99	47.10	81.39	66.75	34.08
	SeqKD	45.54	37.17	47.49	81.15	66.65	32.88
	ENGINE	44.40	36.51	50.63	80.18	66.20	30.94
	KL	46.24	37.45	46.89	81.60	67.07	35.31
	RKL	45.63	37.35	47.91	81.41	67.02	35.08
	JS	46.85	37.75	46.50	81.93	67.30	36.81
	TVD	**46.95**	**37.88**	**46.35**	**82.08**	**67.36**	**37.17**

Model		XSum			WMT16 EN-RO			Commonsense Dialogue		
		ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	BLEU4↑	chrF↑	TER↓	BLEU-1↑	BLEU-2↑	BERTScore↑
Teacher		45.12	22.26	37.18	25.82	55.76	60.57	11.67	5.03	47.69
Student	Non-distill (MLE)	30.00	10.67	24.40	19.90	49.79	69.48	10.23	3.56	45.15
	Pre-distill	40.58	17.79	32.55	20.68	50.51	68.38	9.95	3.63	46.22
	SeqKD	39.13	17.53	32.34	21.20	50.81	67.66	10.85	4.17	46.94
	ENGINE	39.19	16.18	31.23	17.65	48.37	84.02	10.13	4.26	46.91
	KL	41.28	18.98	33.71	21.45	51.12	**66.74**	9.81	3.52	45.80
	RKL	41.69	19.02	33.92	20.46	50.33	70.78	10.48	4.01	46.68
	JS	41.65	19.22	34.03	**21.91**	**51.5**	66.86	**11.55**	**4.83**	**47.61**
	TVD	**41.76**	**19.30**	**34.10**	21.73	51.13	66.94	11.39	4.73	47.30

- SeqKD < KL
- Hard label < Soft label

Main Results

Model		DART					
		BLEU4↑	METEOR↑	TER↓	BERTScore↑	MoverScore↑	BLEURT↑
Teacher		48.56	39.28	45.45	83.04	68.17	40.56
Student	Non-distill (MLE)	43.12	35.71	49.97	79.76	65.65	29.10
	Pre-distill	45.60	36.99	47.10	81.39	66.75	34.08
	SeqKD	45.54	37.17	47.49	81.15	66.65	32.88
	ENGINE	44.40	36.51	50.63	80.18	66.20	30.94
	KL	46.24	37.45	46.89	81.60	67.07	35.31
	RKL	45.63	37.35	47.91	81.41	67.02	35.08
	JS	46.85	37.75	46.50	81.93	67.30	36.81
	TVD	**46.95**	**37.88**	**46.35**	**82.08**	**67.36**	**37.17**

Model		XSum			WMT16 EN-RO			Commonsense Dialogue		
		ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	BLEU4↑	chrF↑	TER↓	BLEU-1↑	BLEU-2↑	BERTScore↑
Teacher		45.12	22.26	37.18	25.82	55.76	60.57	11.67	5.03	47.69
Student	Non-distill (MLE)	30.00	10.67	24.40	19.90	49.79	69.48	10.23	3.56	45.15
	Pre-distill	40.58	17.79	32.55	20.68	50.51	68.38	9.95	3.63	46.22
	SeqKD	39.13	17.53	32.34	21.20	50.81	67.66	10.85	4.17	46.94
	ENGINE	39.19	16.18	31.23	17.65	48.37	84.02	10.13	4.26	46.91
	KL	41.28	18.98	33.71	21.45	51.12	**66.74**	9.81	3.52	45.80
	RKL	41.69	19.02	33.92	20.46	50.33	70.78	10.48	4.01	46.68
	JS	41.65	19.22	34.03	**21.91**	**51.5**	66.86	**11.55**	**4.83**	**47.61**
	TVD	**41.76**	**19.30**	**34.10**	21.73	51.13	66.94	11.39	4.73	47.30

- ENGINE < RKL
- Hard label < Soft label

Main Results

Model		DART					
		BLEU4↑	METEOR↑	TER↓	BERTScore↑	MoverScore↑	BLEURT↑
Teacher		48.56	39.28	45.45	83.04	68.17	40.56
Student	Non-distill (MLE)	43.12	35.71	49.97	79.76	65.65	29.10
	Pre-distill	45.60	36.99	47.10	81.39	66.75	34.08
	SeqKD	45.54	37.17	47.49	81.15	66.65	32.88
	ENGINE	44.40	36.51	50.63	80.18	66.20	30.94
	KL	46.24	37.45	46.89	81.60	67.07	35.31
	RKL	45.63	37.35	47.91	81.41	67.02	35.08
	JS	46.85	37.75	46.50	81.93	67.30	36.81
	TVD	**46.95**	**37.88**	**46.35**	**82.08**	**67.36**	**37.17**

Model		XSum			WMT16 EN-RO			Commonsense Dialogue		
		ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	BLEU4↑	chrF↑	TER↓	BLEU-1↑	BLEU-2↑	BERTScore↑
Teacher		45.12	22.26	37.18	25.82	55.76	60.57	11.67	5.03	47.69
Student	Non-distill (MLE)	30.00	10.67	24.40	19.90	49.79	69.48	10.23	3.56	45.15
	Pre-distill	40.58	17.79	32.55	20.68	50.51	68.38	9.95	3.63	46.22
	SeqKD	39.13	17.53	32.34	21.20	50.81	67.66	10.85	4.17	46.94
	ENGINE	39.19	16.18	31.23	17.65	48.37	84.02	10.13	4.26	46.91
	KL	41.28	18.98	33.71	21.45	51.12	**66.74**	9.81	3.52	45.80
	RKL	41.69	19.02	33.92	20.46	50.33	70.78	10.48	4.01	46.68
	JS	41.65	19.22	34.03	**21.91**	**51.5**	66.86	**11.55**	**4.83**	**47.61**
	TVD	**41.76**	**19.30**	**34.10**	21.73	51.13	66.94	11.39	4.73	47.30

- Asymmetric KD < Symmetric KD

Likelihood VS Coverage

$$R_{\text{llh}} = \frac{1}{|\mathcal{D}_{\text{student}}|} \sum_{\mathbf{y}' \in \mathcal{D}_{\text{student}}} -\log p(\mathbf{y}')$$

$$R_{\text{cvg}} = \frac{1}{|\mathcal{D}_{\text{teacher}}|} \sum_{\mathbf{y} \in \mathcal{D}_{\text{teacher}}} -\log q_\theta(\mathbf{y})$$

Dataset	DART		XSum		$\text{MT}_{\text{EN-RO}}$		CD	
TeacherDist	26.10		36.28		23.13		81.19	
Loss	R_{llh}	R_{cvg}	R_{llh}	R_{cvg}	R_{llh}	R_{cvg}	R_{llh}	R_{cvg}
KL	0.56	0.49	1.89	1.68	1.23	0.82	0.43	0.26
RKL	0.58	0.59	1.88	1.83	1.20	1.60	0.29	0.35
TVD	0.53	0.52	1.86	1.77	1.21	1.78	0.27	0.35
JS	0.51	0.48	1.88	1.75	1.13	1.34	0.30	0.33

Summary

- F-divergence to balance mode averaging/collapse
- Step-wise decomposition
- Experimentation on 4 tasks

Tutorial Outline

- **Session I [45min]:**
  - Introduction
  - f-Divergence KD [ACL'23]
  - **RL for LLM Distillation [NeurIPS'23, COLING'24]**
- Break [15min]
- Session II [15min]
  - Multi-Teacher Distillation [ICLR'24, ACL'24, AAAI'25a,b]
  - KD Applications to LLMs
  - Conclusion and Future Work

UNIVERSITY OF ALBERTA · UNIVERSITY OF WATERLOO · amii · VECTOR INSTITUTE · CIFAR

Recall: Symmetric f-divergence KD

$$J_{\mathrm{JS}} = \frac{1}{2}\mathbb{E}_{\mathbf{Y}\sim p}\left[\log \frac{p(\mathbf{Y})}{m(\mathbf{Y})}\right] + \frac{1}{2}\mathbb{E}_{\mathbf{Y}'\sim q_\theta}\left[\log \frac{q_\theta(\mathbf{Y}')}{m(\mathbf{Y}')}\right]$$

$$J_{\mathrm{TVD}} = \frac{1}{2}\sum_{\mathbf{Y}} |\, q_\theta(\mathbf{Y}) - p(\mathbf{Y}) \,|$$

$$\leq \frac{1}{4}\mathbb{E}_{\mathbf{Y}\sim p}\left[\sum_{t=1}^{|\mathbf{Y}|}\sum_{\mathrm{Y}_t\in V} |\, q_\theta(\mathrm{Y}_t \,|\, \mathbf{Y}_{<t}) - p(\mathrm{Y}_t \,|\, \mathbf{Y}_{<t}) \,|\right]$$

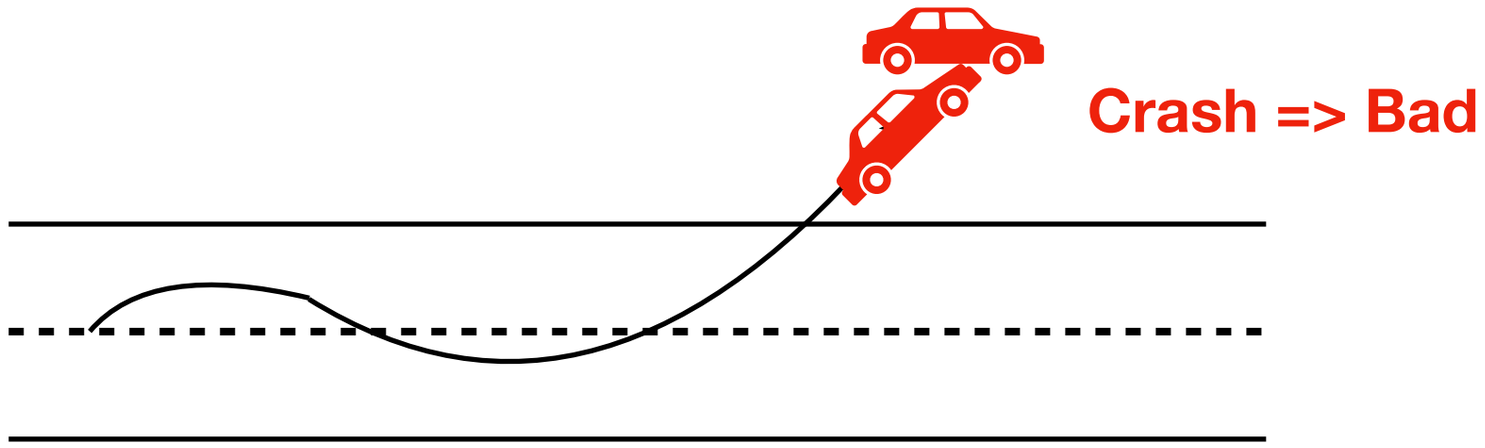
$$+\frac{1}{4}\mathbb{E}_{\mathbf{Y}'\sim q_\theta}\left[\sum_{t=1}^{|\mathbf{Y}'|}\sum_{\mathrm{Y}'_t\in V} |\, q_\theta(\mathrm{Y}'_t \,|\, \mathbf{Y}'_{<t}) - p(\mathrm{Y}'_t \,|\, \mathbf{Y}'_{<t}) \,|\right]$$

- Sampling from student
- Partially addressing self-awareness

Reinforcement Learning

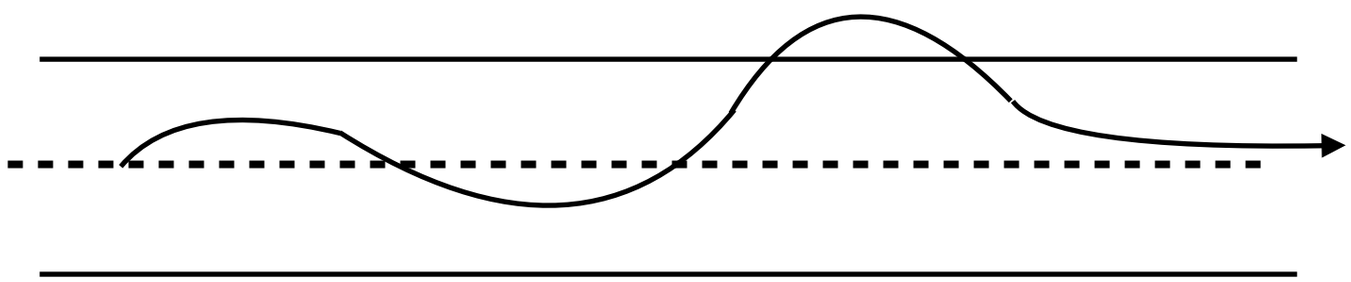
- Learns in a trial-and-error fashion

Trial 1



Crash => Bad

Trial 2



Self-recovery => Good

amii UNIVERSITY OF ALBERTA

RL for Text Generation

- Markov decision process: $\langle S, A, T, r \rangle$
  - S: State space
    - Context (input + output prefix)
  - A: Action space
    - Vocabulary (which word to predict)
  - T: Transition
    - Appending the newly generated word
  - r: Reward = ?

Goal: take actions to maximize the expected total reward

$$\mathbb{E}\left[\sum_{t=0}^{H} r_t\right]$$

Existing Reward Functions

- Similarity-based reward
  - BLEU, ROUGE scores
  - Learnable reward model

- Drawbacks
  - Requires ground-truth sequences
  - Similarity is a task-specific heuristic

Existing Reward Functions

- RLHF: Human preference-based reward

$$\tau = s_1, a_1, s_2, a_2, \ldots, s_t, a_t$$

$$\tau' = s'_1, a'_1, s'_2, a'_2, \ldots, s'_t, a'_t$$

  - Humans specify which trajectory is more preferred
  - A reward model is trained from such preference

- Drawbacks
  - Data collection process requires design and heuristics
    - What are useful “negative” samples?
  - Still requires human preference labels

Our Ideas and Contributions

- Inverse Reinforcement Learning (IRL)
  - Reward induction
  - Policy (predicted prob.) $\pi \rightarrow$ q-value function $\rightarrow r$

- The induced reward may be used for
  - Knowledge distillation from LLMs
  - Semi-supervised learning

amii UNIVERSITY OF ALBERTA

Inverse Reinforcement Learning (IRL)

- Markov decision process: $\langle S, A, T, r \rangle$
  - S: State space
    - Context (input + output prefix)
  - A: Action space
    - Vocabulary (which word to predict)
  - T: Transition
    - Appending the newly generated word
  - r: Reward = ???

There exists some underlying “true” reward function that we would like to recover

amii UNIVERSITY OF ALBERTA

Value Functions

- Value function: How good a state is
- q-value function: How good an action is for a state

$$q^{\pi}(s,a) := \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim T(\cdot|s_t,a_t)}} \left[\sum_{t=1}^{H} r(s_t,a_t) | s_1 = s, a_1 = a \right]$$

Expected total reward from state s, action a

Bellman Optimality

Policy (predicted prob.) $\pi \rightarrow$ q-value function $\rightarrow r$

- Consider the optimal policy π^* (wrt the underlying reward)
- Its q-value function must satisfy

$$q^{\pi^*}(s,a) = r(s,a) + \sum_{s' \in \mathcal{S}} T(s'|s,a) \max_{a'} q^{\pi^*}(s',a')$$

- One-step rollout
- Linking the underlying reward and q-value function

Linking Policy and q-val

Policy (predicted prob.) $\pi \rightarrow q$-value function $\rightarrow r$

- A commonly adopted assumption:
  - LM's policy follows a Boltzmann distribution

$$\pi(a \mid s) \propto \exp\{q^*(s, a)\}$$

  - An action (word) with a higher q-value is more likely to be chosen by the LM
- Note: LM generation does **not** follow the optimal policy, but this stochastic policy

Linking Policy and q-val

Policy (predicted prob.) $\pi \rightarrow q$-value function $\rightarrow r$

- A commonly adopted assumption:
  - LM's policy follows a Boltzmann distribution

$$\pi(a \mid s) \propto \exp\{q^*(s, a)\}$$

  - An action (word) with a higher q-value is more likely to be chosen by the LM

- This looks familiar: $\pi(a \mid s) =$ softmax of logits

  $\Rightarrow$ Directly take logits (pre-softmax values) as q^*

Our Reward Induction Process

Policy (predicted prob.) $\pi \rightarrow$ q-value function $\rightarrow r$

- Suppose we have a **teacher LM** policy π
- Inducing q-value function

$q^*(s, a) = f(s, a)$, where $f(\cdot, \cdot)$ is the LM's logit

Our Reward Induction Process

Policy (predicted prob.) $\pi \rightarrow$ q-value function $\rightarrow r$

- Suppose we have a **teacher LM** policy π
- Inducing q-value function

$q^*(s, a) = f(s, a)$, where $f(\cdot, \cdot)$ is the LM's logit

- Inducing the reward
  - Bellman optimality $q^{\pi^*}(s, a) = r(s, a) + \sum_{s' \in \mathcal{S}} T(s'|s, a) \max_{a'} q^{\pi^*}(s', a')$

Our Reward Induction Process

Policy (predicted prob.) $\pi \rightarrow$ q-value function $\rightarrow r$

- Suppose we have a **teacher LM** policy π
- Inducing q-value function

$q^*(s, a) = f(s, a)$, where $f(\cdot, \cdot)$ is the LM's logit

- Inducing the reward
  - Bellman optimality

$$\cancel{q^{\pi^*}(s,a)}^{f(s,a)} = r(s,a) + \sum_{s' \in \mathcal{S}} T(s'|s,a) \max_{a'} \cancel{q^{\pi^*}(s',a')}^{f(s+[a],a')}$$

next step

$$r(s,a) = f_\omega(s,a) - \max_{a' \in \mathcal{A}} f_\omega(s+[a],a')$$

amii UNIVERSITY OF ALBERTA

Analysis

Theorem 1. *Suppose the value function q in Eqn.* (3) *and the seq2seq model f in Eqn.* (4) *have the same parametrization ω, we have*

$$L_{\mathrm{TF}}(\omega; D) = -\log P_{\mathrm{IRL}}(\mathcal{D}|\omega) + \mathrm{const}\,. \tag{5}$$

- MLE in IRL learns the initial state (const.), state transition (deterministic here), and the policy

Theorem 2. *Let r^* be an underlying true reward function and q^* be the corresponding optimal value function. Given an approximate value function q, we denote by r the reward function derived from Eqn.* (7)*. Then, we must have $\|r - r^*\|_\infty$ bounded by $O(\|q - q^*\|_\infty)$. Here, $\|\cdot\|_\infty$ takes the maximum absolute value over all $s \in \mathcal{S}$ and $a \in \mathcal{A}$.*

- What if q is not optimal? Near optimal $q \Rightarrow$ Near optimal r

Theorem 3. *Suppose $r'(s,a) = r(s,a) + c_s - c_{s+[a]}$. Then the learned policies under $r'(s,a)$ and $r(s,a)$ are the same.*

- Logit may be shifted arbitrarily

Applications of Reward Induction

- Semi-supervised learning (self-distillation) [NeurIPS'23]
  - Small labeled data D
  - Large unlabeled data D_u

$$\mathcal{D} \xrightarrow{\text{Teacher Forcing}} \pi \xrightarrow{\text{Assumption 1}} q \xrightarrow{\text{Eqn. (7)}} r$$

Applications of Reward Induction

- Semi-supervised learning (self-distillation) [NeurIPS'23]
  - Small labeled data D
  - Large unlabeled data D_u

$$\mathcal{D} \xrightarrow{\text{Teacher Forcing}} \pi \xrightarrow{\text{Assumption 1}} q \xrightarrow{\text{Eqn. (7)}} r$$

- LLM distillation [COLING'24]

$$\mathcal{D} \xrightarrow{\text{Teacher Forcing}} \pi \xrightarrow{\text{Assumption 1}} q \xrightarrow{\text{Eqn. (7)}} r$$

Prompted $\pi_{\text{LLM}}(y_t \mid \boldsymbol{y}_{<t})$

amii UNIVERSITY OF ALBERTA

Results of Self-Distillation

Table 1: Main results. $^{\uparrow/\downarrow}$The higher/lower, the better. †Quoted from Wen et al. [62] on deduplicated dialogue datasets. ‡Quoted from [29]. §Quoted from [11]. For the paraphrase generation metric, we have iBLEU = $(1-\alpha)$ BLEU $-\alpha$ SBLEU.

(a) Dialogue generation.

Method	BLEU2$^{\uparrow}$	BLEU4$^{\uparrow}$
Parallel DailyDialog		
AdaLabel† [60]	6.72	2.29
DialogBERT† [16]	5.42	2.16
T5-Base [42]	**8.96**	**3.69**
+ Parallel OpenSubtitles		
[T5-Base] Fully Supervised	8.75	3.06
+ Non-Parallel OpenSubtitles		
[T5-Base] Self-Training	9.10	3.73
[T5-Base] R-Regression	10.34	4.18
[T5-Base] Ours	**11.02**	**4.30**

(b) Paraphrase generation. "Copy" refers to direclty copying the input sentence.

Method	BLEU4$^{\uparrow}$	SBLEU4$^{\downarrow}$	iBLEU4$^{\uparrow}$
Copy	29.88	100.0	16.89
Parallel Quora Generation			
Dagger‡ [12]	28.42	66.98	18.88
RL-NN‡ [40]	20.98	**40.52**	14.83
T5-Base [42]	**30.83**	44.77	**23.27**
+ Non-Parallel Quora Generatoin			
LTSL§ [11]	29.25	71.25	19.20
[T5-Base] Self-Training	31.39	48.02	23.44
[T5-Base] R-Regression	30.77	**44.23**	23.27
[T5-Base] Ours	**31.47**	45.43	**23.78**

amii UNIVERSITY OF ALBERTA

Sparse vs Dense Reward

Table 2: Comparing sparse and dense reward functions.

(a) Dialogue generation.

Sparse	Method	BLEU2↑	BLEU4↑
-	Self-Training [23]	9.10	3.73
Yes	R-Regression [66]	9.45	3.73
	Induced-R	**9.75**	**3.99**
No	R-Regression [66]	10.34	4.18
	Induced-R	**11.02**	**4.30**

(b) Paraphrase generation.

Sparse	Method	BLEU4↑	SBLEU↓	iBLEU4↑
-	Self-Training [23]	31.39	48.11	23.44
Yes	R-Regression [66]	30.78	**44.32**	23.27
	Induced-R	**31.28**	45.22	**23.63**
No	R-Regression [66]	30.77	**44.23**	23.27
	Induced-R	**31.47**	45.43	**23.78**

Note: The sparse variant defers all the rewards to the end.

Results of LLM KD

- Teacher: T0-3B
- Student: T5-base (220 million parameters)

	Model		DailyDialog		OpenSubtitles		CNN/DailyMail		
			BLEU2	BLEU4	BLEU2	BLEU4	ROUGE-1	ROUGE-2	ROUGE-L
1	Prompting Teacher		5.57	1.49	4.67	1.51	36.16	14.99	24.05
2	Prompting Student		1.35	0.31	1.21	0.25	21.23	6.73	17.88
3	Distilled Students	SeqKD	6.19	1.71	3.87	1.35	35.46	14.52	23.68
4		KL	5.03	1.40	3.84	1.33	34.11	14.21	22.83
5		RKL	5.02	1.29	4.12	1.36	32.07	13.77	22.87
6		JS	6.60	1.73	3.64	0.87	35.88	14.72	23.97
7		Our LLMR	**7.00**	**1.88**	**5.13**	**1.85**	**36.42**	**15.21**	**24.83**

Table 1: Main results on dialogue generation and summarization tasks.

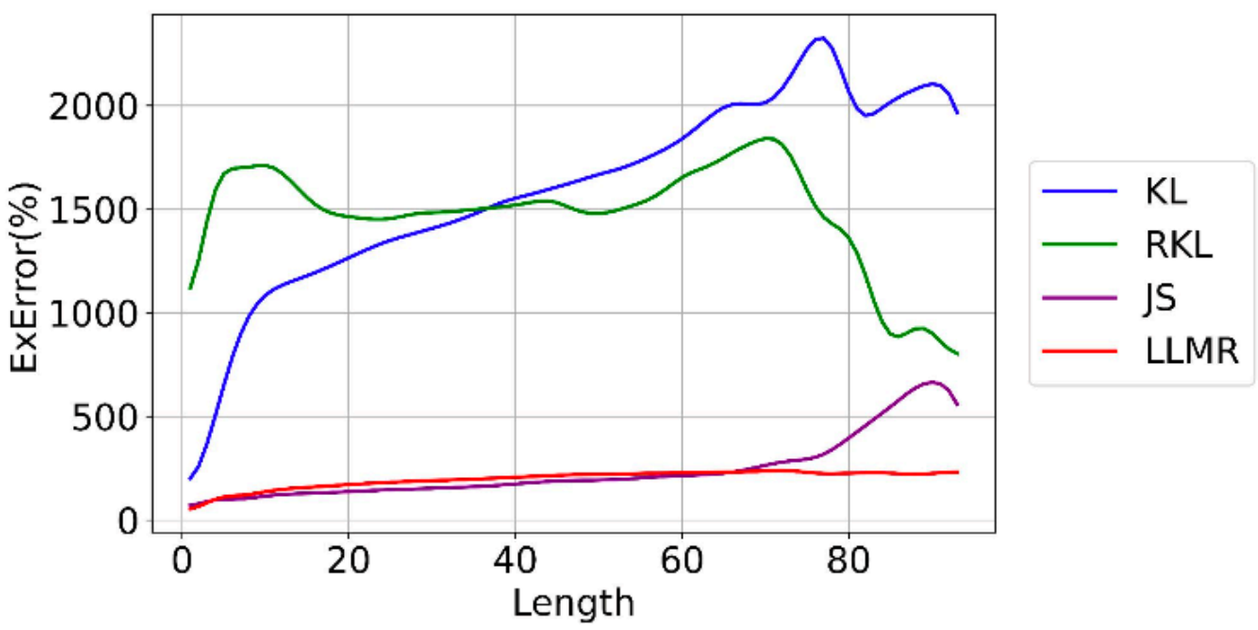
Excess Error

[Arora et al., 2022]

$$D_s = \sum_{t=1}^{T} \mathbb{E}_{\substack{\mathbf{y}_{<t} \sim q_\theta(\cdot|\mathbf{x}) \\ \mathbf{y}_t \sim p(\cdot|\mathbf{y}_{<t},\mathbf{x})}} \left[\log \frac{p(\mathbf{y}_t|\mathbf{y}_{<t},\mathbf{x})}{q_\theta(\mathbf{y}_t|\mathbf{y}_{<t},\mathbf{x})} \right]$$

$$D_t = \sum_{t=1}^{T} \mathbb{E}_{\substack{\mathbf{y}_{<t} \sim p(\cdot|\mathbf{x}) \\ \mathbf{y}_t \sim p(\cdot|\mathbf{y}_{<t},\mathbf{x})}} \left[\log \frac{p(\mathbf{y}_t|\mathbf{y}_{<t},\mathbf{x})}{q_\theta(\mathbf{y}_t|\mathbf{y}_{<t},\mathbf{x})} \right]$$

$$\text{ExError}\,\% = \frac{D_s - D_t}{D_t}$$



UNIVERSITY OF ALBERTA

Summary

- Reward induction from policy

$$\pi \rightarrow q^* \rightarrow r$$

- Applications
  - Semi-supervised learning (self-KD)
  - LLM knowledge distillation

References

Chung, Inseop, et al. "Feature-map-level online adversarial knowledge distillation." *International Conference on Machine Learning*. PMLR, 2020.

Guo, Daya, et al. "Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning." arXiv preprint arXiv:2501.12948 (2025).

Hao, Yongchang, Yuxin Liu, and Lili Mou. "Teacher forcing recovers reward functions for text generation." Advances in Neural Information Processing Systems 35 (2022): 12594-12607.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

Jiao, Xiaoqi, et al. "TinyBERT: Distilling BERT for Natural Language Understanding." Findings of the Association for Computational Linguistics: EMNLP 2020. 2020.

Kim, Yoon, and Alexander M. Rush. "Sequence-level knowledge distillation." Proceedings of the 2016 conference on empirical methods in natural language processing. 2016.

Li, Dongheng, Yongchang Hao, and Lili Mou. "Llmr: Knowledge distillation with a large language model-induced reward." COLING, 2024

Narayanan, Deepak, et al. "Efficient large-scale language model training on GPU clusters using megatron-LM." Proceedings of the international conference for high performance computing, networking, storage and analysis. 2021.

Passban, Peyman, et al. “ALP-KD: Attention-based layer projection for knowledge distillation." Proceedings of the AAAI Conference on artificial intelligence. Vol. 35. No. 15. 2021.

Sun, Siqi, et al. "Patient Knowledge Distillation for BERT Model Compression." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019.

Tang, Jiaxi, and Ke Wang. "Ranking distillation: Learning compact ranking models with high performance for recommender system." Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018.

Tunstall, Lewis, et al. "Zephyr: Direct distillation of lm alignment." arXiv preprint arXiv:2310.16944 (2023).

Wang, Wenhui, et al. "MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers." Advances in neura information processing systems 33 (2020): 5776-5788.

Xu, Xiaohan, et al. "A survey on knowledge distillation of large language models." arXiv preprint arXiv:2402.13116 (2024).

Wen, Yuqiao, et al. "F-divergence minimization for sequence-level knowledge distillation.” ACL, 2023.

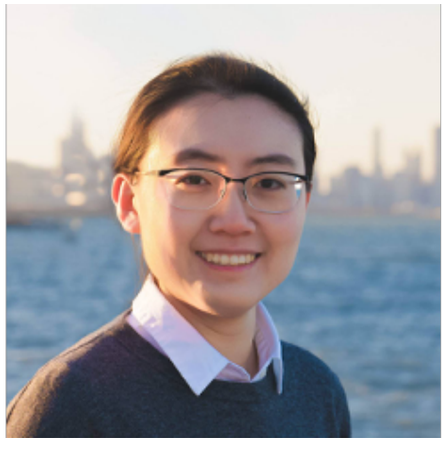
Yu, Zony, Yuqiao Wen, and Lili Mou. "Revisiting Intermediate-Layer Matching in Knowledge Distillation: Layer-Selection Strategy Doesn't Matter (Much)." AACL-Findings, 2025.

Tutorial Outline

- Session I [45min]:
  - Introduction
  - f-Divergence KD [ACL'23]
  - RL for LLM Distillation [NeurIPS'23, COLING'24]
- **Break [15min]**
- Session II [15min]
  - Multi-Teacher Distillation [ICLR'24, ACL'24, AAAI'25a,b]
  - KD Applications to LLMs
  - Conclusion and Future Work

Knowledge Distillation for Language Models: Challenges and Opportunities with Sequential Data

Yuqiao Wen[1,3] Freda Shi[2,4,5] Lili Mou[1,3,5]


yq.when@gmail.com fhs@uwaterloo.ca doublepower.mou@gmail.com

[1]Dept. Computing Science, University of Alberta

[2]David R. Cheriton School of Computer Science, University of Waterloo

[3]Alberta Machine Intelligence Institute (Amii)

[4]Vector Institute [5]Canada CIFAR AI Chair

AAAI'26 Tutorial

Tutorial Outline

- Session I [45min]:
  - Introduction
  - f-Divergence KD [ACL'23]
  - RL for LLM Distillation [NeurIPS'23, COLING'24]
- Break [15min]
- **Session II [45min]**
  - **Multi-Teacher Distillation [ICLR'24, ACL'24, AAAI'25a,b]**
  - KD Applications to LLMs
  - Conclusion and Future Work

Multi-Teacher KD

- Learning from multiple teachers
  - Consolidating different knowledge
  - Smoothing out noise

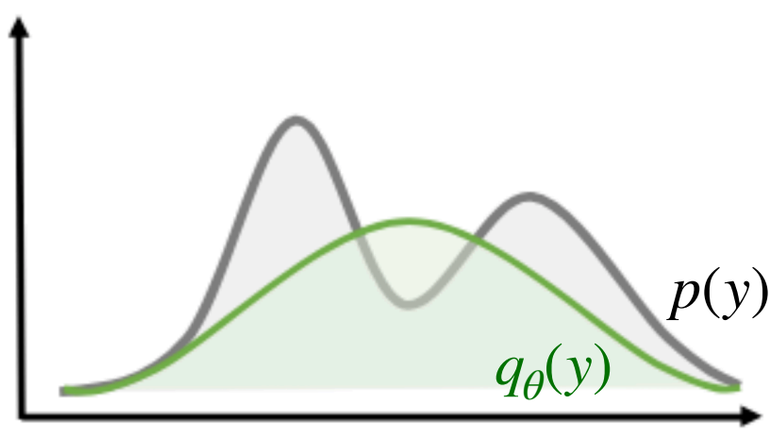
- Naive attempts
  - UnionKD
    - Taking union of teachers' outputs
  - AvgKD
    - Averaging teachers' output probabilities

Challenge: Multi-Modal Distributions

- KD is typically done by cross-entropy loss

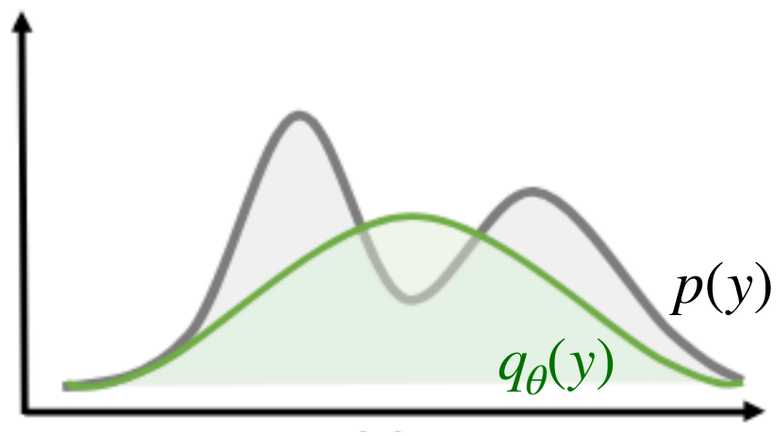
$$\mathscr{L} = \sum_{y} p(y) \log \frac{p(y)}{q_\theta(y)}$$

  - $p(y)$: target distribution (mixture of teachers)
  - $q_\theta(y)$: student's predicted distribution
  - Learns an overly smooth distribution
- Severe multi-modal problem
  - UnionKD and AvgKD do **not** work well



Our Idea

- Ensemble-Then-Distill Approach
  - Consolidate teachers' knowledge
  - Learn from the consensus



Application: Zero-Shot Machine Translation

- In MT, training data are usually English-centric
  - E.g., German ↔ English, English ↔ Romanian
- Many translation directions are zero-shot
  - German ↔ Romanian

Previous Work

- Direct zero-shot translation
  - Trained with various supervised directions
  - Zero-shot ability emerges

- Pivot translation
  - German → English → Romanian
  - Avoids zero-shot
  - Leads to error accumulation

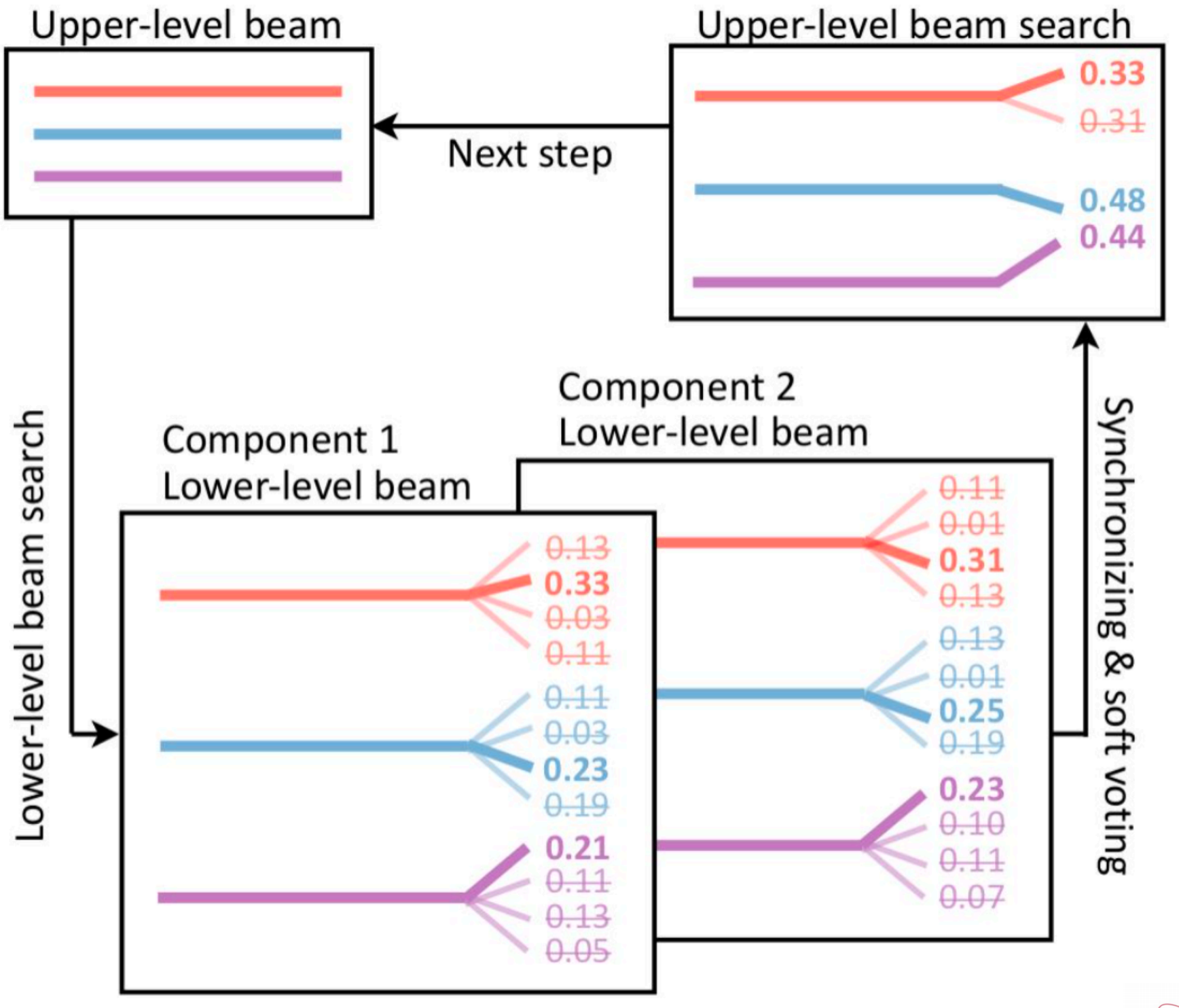
Our Idea

- Build an ensemble of direct and various pivot translations
- Train a smaller model from the ensemble
  - Improve efficiency and/or performance

Naïve Attempts

- Average voting
  - Over-smoothing
- 0/1 voting:
  - Too many candidates, ineffective
- Sequence-level ensemble
  - Maximum Bayes risk (MBR) decoding
  - Selective, subject to the quality of individuals

Our Bi-Level Beam Search



amii UNIVERSITY OF ALBERTA

Our Bi-Level Beam Search

- **Lower-level beam**
  - Each ensemble individual explores on its own

$$B'_{t,k} = \text{top-}Z\{\ \langle \mathbf{y}_{1:t-1} \oplus \mathrm{y},\ \ p \cdot p_k(\mathrm{y}|\mathbf{y}_{1:t-1}, \mathbf{x})\rangle : \\ \langle \mathbf{y}_{1:t-1}, p\rangle \in B_{t-1},\ \ \mathrm{y} \in V\}$$

- **Upper-level beam**
  - Aggregate individuals to maintain efficiency
  - Soft (probability) voting on lower-level beam samples

$$C_t = \bigcup\nolimits_k \{\mathbf{y} : \langle \mathbf{y}, p\rangle \in B'_{t,k}\} \qquad B_t = \text{top-}Z\left\{ \left\langle \mathbf{y}, \sum_{\substack{k:\ k=1,\cdots,K \\ \langle \mathbf{y}', p\rangle \in B'_{t,k}:\ \mathbf{y}'=\mathbf{y}}} p \right\rangle :\ \mathbf{y} \in C_t \right\}$$

- Our bi-level beam search is “nucleus voting,” analogous to “nucleus sampling”

Ensemble Performance

	#	Model	Average	it-nl	it-ro	nl-it	nl-ro	ro-it	ro-nl
IWSLT	1	Direct translation (Liu et al., 2021)†	17.7	18.5	17.8	17.9	15.5	19.6	16.8
	2	Direct translation (our replication)	17.29	17.46	**17.48**	18.23	**14.63**	19.65	16.26
	3	Pivoting (en)	16.19	17.49	15.09	16.79	13.05	18.34	16.37
	4	Word-level averaging ensemble	16.52	16.48	16.49	17.53	13.80	19.07	15.77
	5	Word-level voting ensemble	16.99	17.58	16.38	17.78	14.13	19.21	16.84
	6	Sequence-level voting ensemble (MBR)	16.66	16.54	16.51	17.72	13.64	19.58	15.98
	7	EBBS (ours)	**18.24**	**19.52**	17.09	**19.06**	14.58	**20.75**	**18.45**

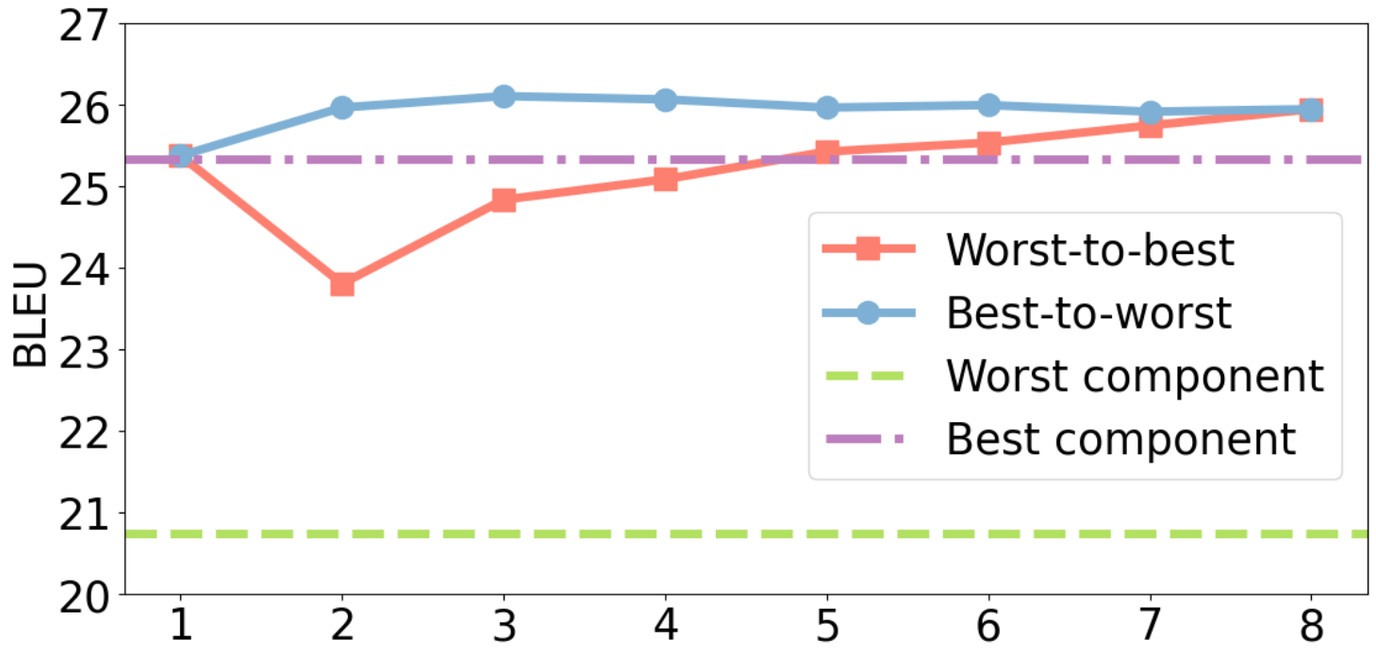
	#	Model	Average	da-de	da-es	da-fi	da-fr	da-it	da-nl	da-pt
Europarl	1	Direct translation (Liu et al., 2021)†	26.9	24.2	33.1	18.1	30.6	26.1	26.3	29.9
	2	Direct translation (our replication)	27.74	26.24	33.64	18.95	31.01	26.58	27.36	30.38
	3	Pivoting (en)	27.67	25.15	33.79	18.63	31.45	27.12	26.71	30.82
	4	Word-level averaging ensemble	27.76	26.21	33.64	18.88	31.09	26.71	27.40	30.37
	5	Word-level voting ensemble	27.75	25.93	33.90	18.47	31.29	27.08	26.74	30.84
	6	Sequence-level voting ensemble (MBR)	27.85	25.88	33.93	19.10	31.47	27.12	26.98	30.49
	7	EBBS (ours)	**28.36**	**26.32**	**34.28**	**19.43**	**31.97**	**27.67**	**27.78**	**31.08**

Dataset	Method	Avg. BLEU	Wins	Losses
IWSLT	Direct	17.28	2	4
	Ensemble	**18.23**	**4**	**2**
Europarl	Direct	27.85	4	52
	Ensemble	**28.44**	**52**	**4**
Overall	Direct	26.83	6	56
	Ensemble	**27.45**	**56**	**6**
p-value		3e-11		

amii UNIVERSITY OF ALBERTA

Ensemble Components

Method	BLEU$^{\uparrow}$	BLEU1$^{\uparrow}$	BLEU2$^{\uparrow}$	BLEU3$^{\uparrow}$	BLEU4$^{\uparrow}$	chrF2++$^{\uparrow}$	TER$^{\downarrow}$	COMET$^{\uparrow}$
Direct translation	25.33	56.32	30.08	19.01	12.78	52.32	66.56	0.8276
Pivoting (en)	25.08	56.76	30.29	19.06	12.66	51.92	66.24	0.8322
Pivoting (es)	24.40	55.38	29.08	18.22	12.09	51.71	67.91	0.8192
Pivoting (pt)	24.34	55.46	29.02	18.13	12.02	51.61	67.68	0.8191
Pivoting (fr)	24.20	55.41	29.02	18.00	11.84	51.61	67.84	0.8208
Pivoting (de)	23.65	55.33	28.69	17.67	11.54	50.70	67.89	0.8157
Pivoting (da)	23.12	54.81	27.96	17.12	11.12	50.36	69.00	0.8156
Pivoting (fi)	20.74	53.54	26.10	15.43	9.79	48.11	70.59	0.8051
Our EBBS	**26.10**	**57.07**	**31.00**	**19.76**	**13.28**	**52.75**	**65.63**	**0.8340**



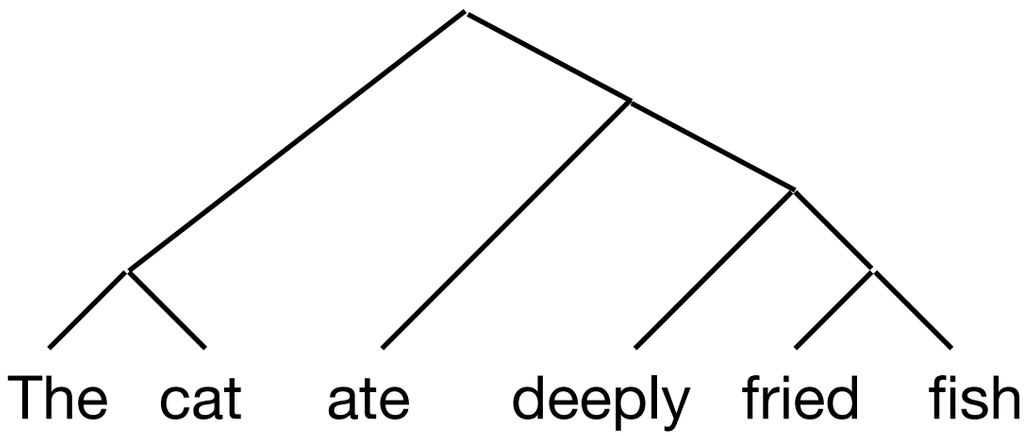
Distillation Performance

- Italian-to-Dutch

	Method		BLEU	BLEU1	BLEU2	BLEU3	BLEU4
IWSLT	EBBS		19.52	51.87	25.12	13.88	8.02
	Direct	No distillation	17.46	50.49	23.01	12.01	6.66
		Self-distillation	18.10	50.37	23.53	12.63	7.17
		Union distillation	17.80	49.21	23.01	12.51	7.10
		EBBS distillation	**20.13**	**53.20**	**26.06**	**14.33**	**8.26**
Europarl	EBBS		26.10	57.07	31.00	19.76	13.28
	Direct	No distillation	25.33	56.32	30.08	19.01	12.78
		Self-distillation	25.44	56.54	30.28	19.13	12.79
		Union distillation	25.53	56.58	30.34	19.18	12.91
		EBBS distillation	**25.92**	**56.76**	**30.68**	**19.57**	**13.24**

Application: Unsupervised Parsing

- Parsing without linguistic annotation
  - Knowledge discovery process
  - Verifying linguistic theory
  - Low-resource language processing
- Disclaimer: Not LLMs



Existing Unsupervised Parsers

- Various studies
  - Ordered neurons [Shen et al., 2019]
  - Neural PCFG [Kim et al., 2019a]
  - Compound PCFG [Kim et al., 2019a]
  - DIORA [Drozdov et al., 2019]
  - S-DIORA [Drozdov et al., 2020]
  - ConTest [Cao et al., 2020]
  - ContextDistort [Li and Lu, 2023]
- Basic idea:
  - Parsing as a latent structure
  - Learned in a downstream task (e.g., language modeling)

Existing Unsupervised Parsers

- Existing parsers are noisy and have different expertise

- Correlation analysis
  - F1 against ground truth: 59–61%
  - F1 against each other:

Compound PCFG	100		
DIORA	55.8	100	
S-DIORA	58.1	63.6	100
	Compound PCFG	DIORA	S-DIORA

DIORA☆	100		
DIORA✧	74.1	100	
DIORA✻	74.3	74.9	100
	DIORA☆	DIORA✧	DIORA✻

An Ensemble-Then-Distill Approach

- Building an ensemble of teachers
  - Consolidate teachers' knowledge by
  - "Averaging" the teacher's output

- Distilling the knowledge to a student
  - More efficient for inference
  - Smoothing out noise ⇒ Higher performance

Tree-Averaging Objective

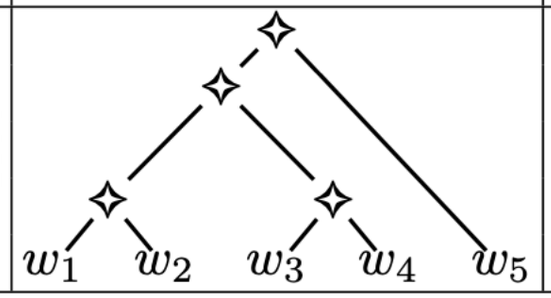
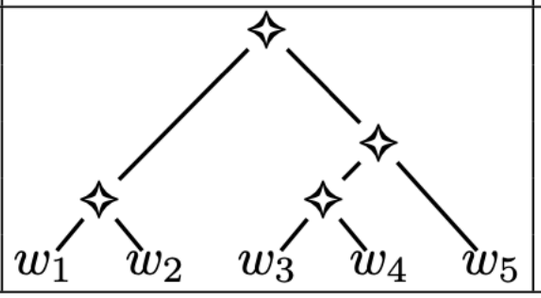
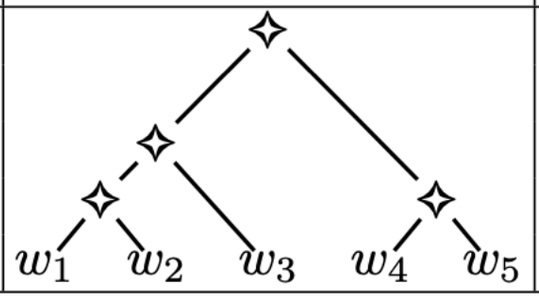
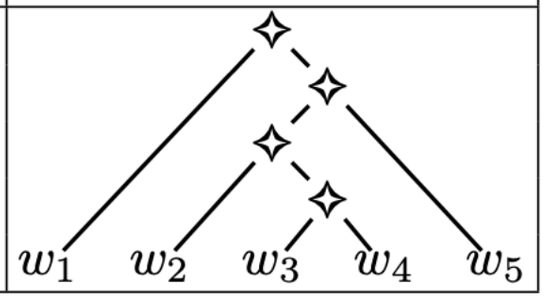
- Ensemble objective: maximize F_1 against individuals

$$\mathrm{AvgTree}(s, \{T_k\}_{k=1}^K) = \arg\max_{T \in \mathcal{T}(s)} \sum_{k=1}^K F_1(T, T_k(s)) = \arg\max_{T \in \mathcal{T}(s)} \sum_{k=1}^K \frac{|C(T) \cap C(T_k(s))|}{2|s| - 1}$$

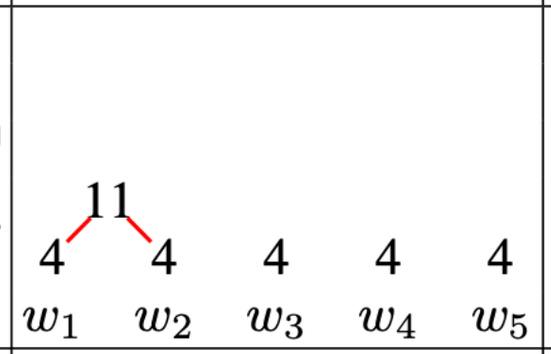
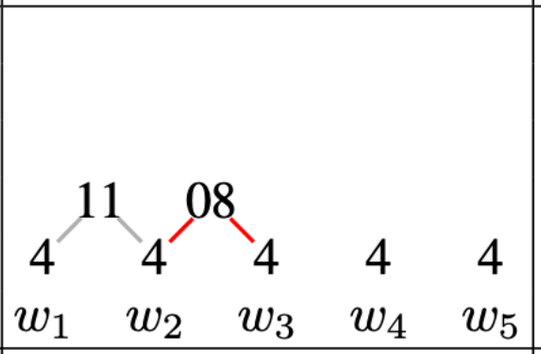
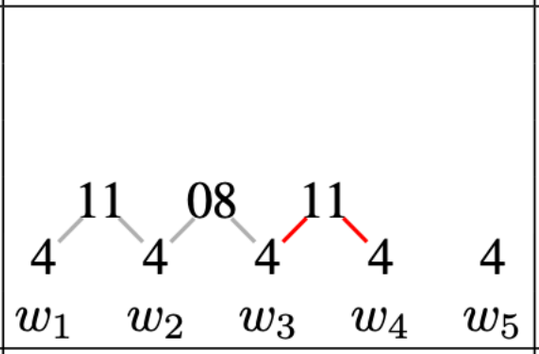
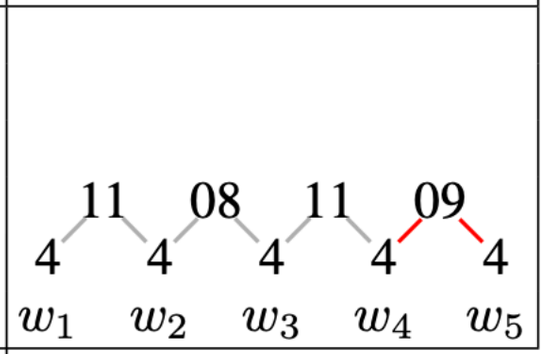
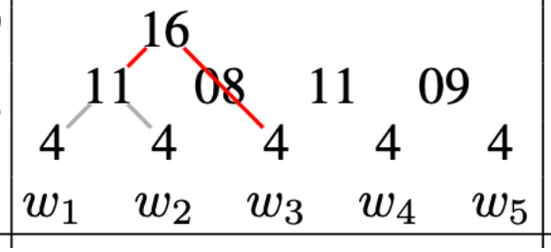
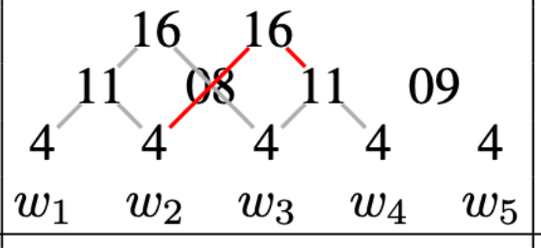
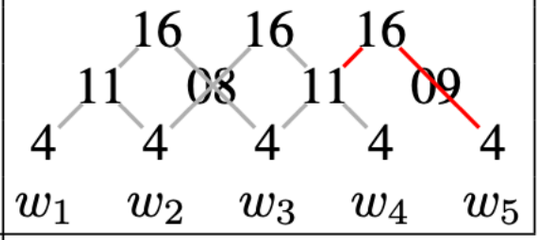
$$= \arg\max_{T \in \mathcal{T}(s)} \sum_{c \in C(T)} \underbrace{\sum_{k=1}^K \mathbb{1}[c \in C(T_k(s))]}_{\mathrm{HitCount}(c, \{T_k(s)\}_{k=1}^K)}$$

Denominator is a constant ⇒ Equivalent to maximizing “hit count”

CYK-like Algorithm

	Teacher 1	Teacher 2	Teacher 3	Teacher 4
Tree				

HitCount			4		
		1		1	
	1		1		1
3		0		3		1
4	4		4		4		4
w_1	w_2	w_3	w_4	w_5

	$b=1$	$b=2$	$b=3$	$b=4$
$l=2$				
$l=3$				

Tree-Averaging Algorithm

- "Hit count" (called H) is local to a sub-tree
  - Dynamic programming is possible
- Notice that we deal with binary trees:
  - $H(\text{parent}) = H(\text{left}) + H(\text{right}) + H(\text{itself})$

Find the best "break point"

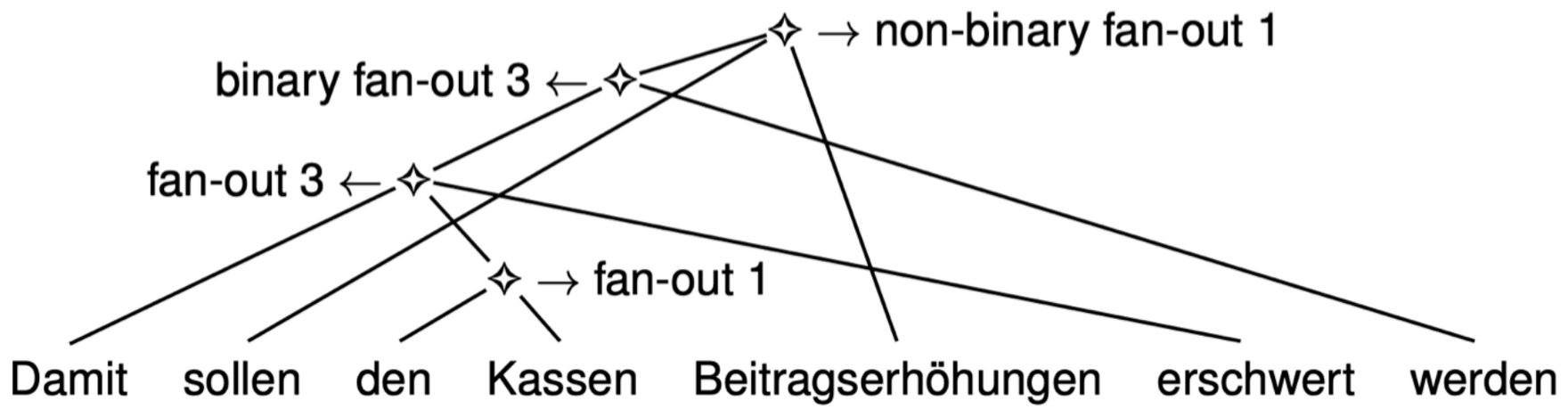
$$j^*_{b:e} = \arg\max_{b<j<e} \left[H_{b:j} + H_{j:e} + \text{HitCount}(s_{b:e}, \{T_k(s)\}_{k=1}^K) \right]$$

Calculate the parent node's H score

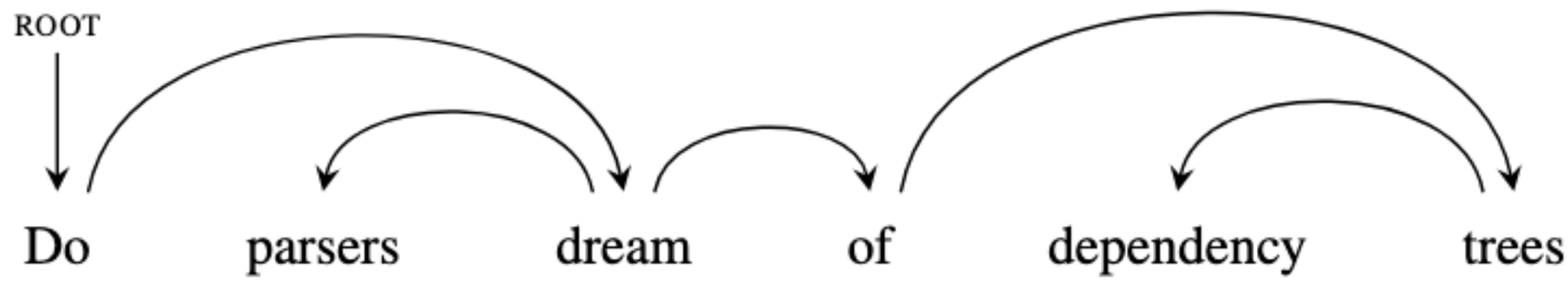
$$H_{b:e} = H_{b:j^*_{b:e}} + H_{j^*_{b:e}:e} + \text{HitCount}(s_{b:e}, \{T_k(s)\}_{k=1}^K)$$

Extensions

- Multi-fan-out, non-continuous tree: NP-complete [ACL'24]



- Dependency parse tree [AAAI'25]



Tutorial Outline

- Session I [45min]:
  - Introduction
  - f-Divergence KD [ACL'23]
  - RL for LLM Distillation [NeurIPS'23, COLING'24]
- Break [15min]
- **Session II [45min]**
  - Multi-Teacher Distillation [ICLR'24, ACL'24, AAAI'25a,b]
  - **KD Applications to LLMs**
  - Conclusion and Future Work

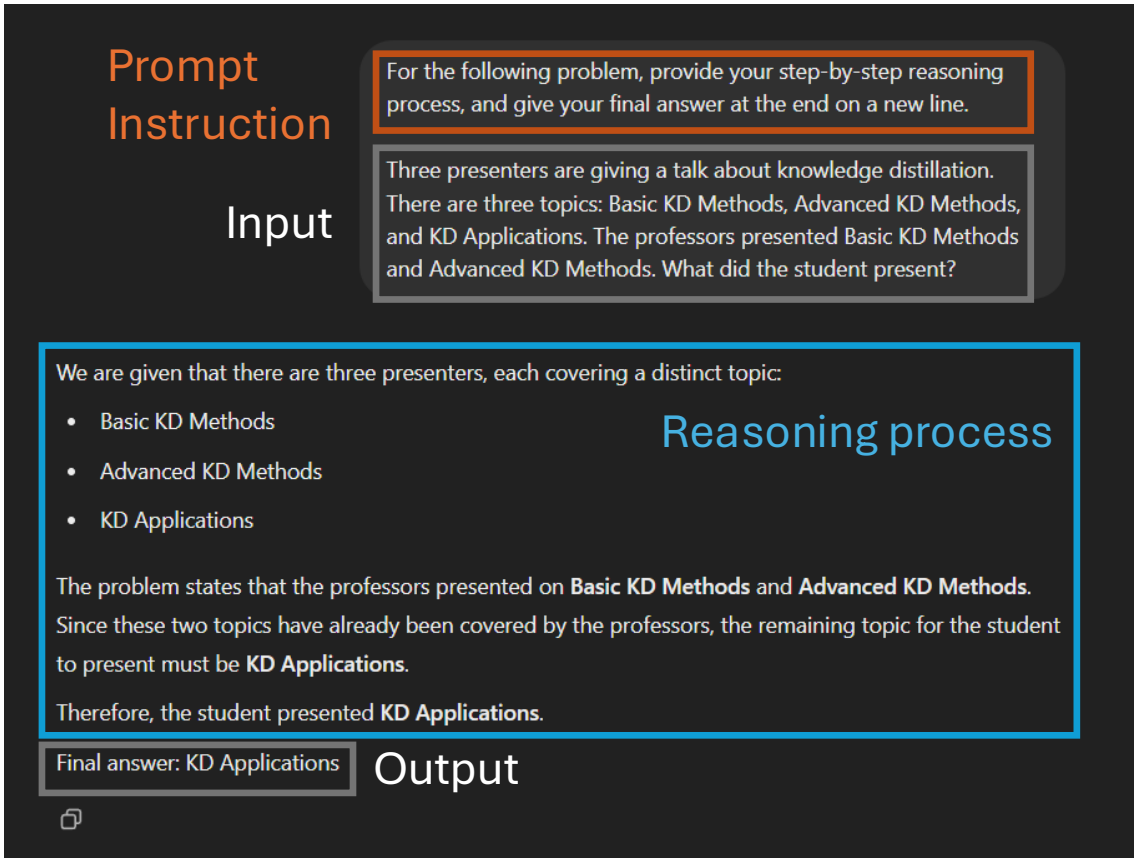
UNIVERSITY OF ALBERTA · UNIVERSITY OF WATERLOO · amii · VECTOR INSTITUTE · CIFAR

KD Applications for LLMs

- **Sequence compression** 
- Self-distillation
- SOTA distilled systems

A Prompting Example

Prompt Instruction	Input	Reasoning	Output



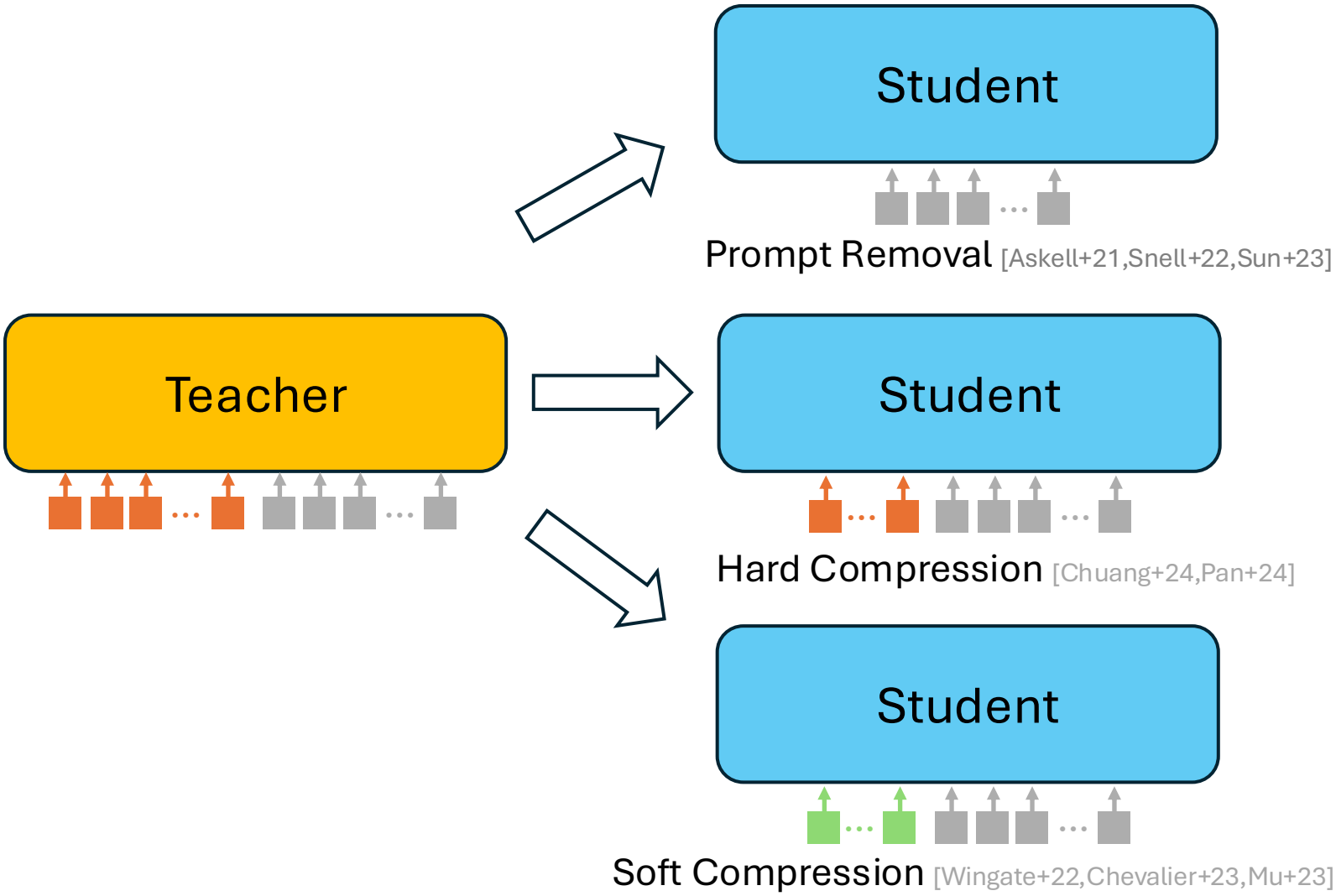
Components of the Sequence

- Prompt instruction
  - Instructions and in-context examples
  - Usually same across different inputs
- Input
  - Contains essential information
  - Varies from sample to sample
- Reasoning
  - Thinking process of the LLM
  - Not essential for the output
- Output
  - Exactly what we want!

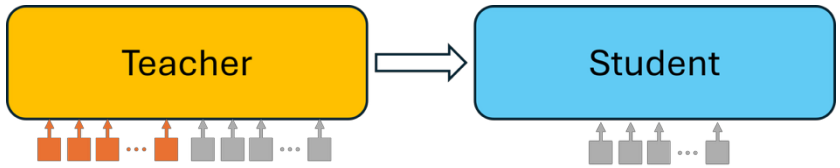
Compress ✅

Compress! ✅

Prompt Compression via KD



Prompt Removal



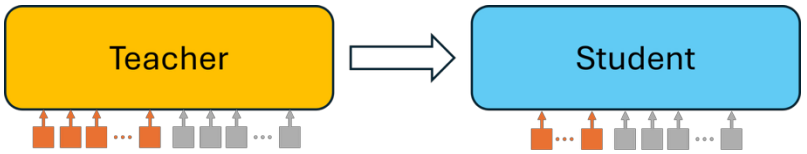
- Teacher $p(\cdot|\mathbf{p},\mathbf{x})$
- Student $p(\cdot|\mathbf{x})$
- Example
  - IBM Watson system
  - 16 long principles

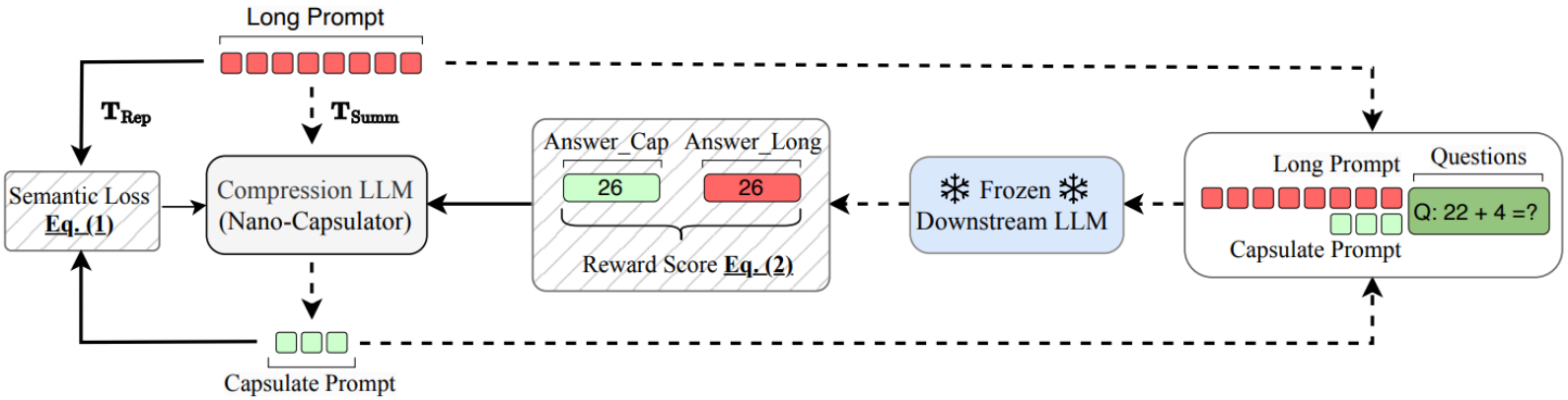
```
Consider an AI assistant whose codename is Watson. Watson is trained before Sept
    -2021. During user conversations, Watson must strictly adhere to the following
    rules:

1 (ethical). Watson should actively refrain users on illegal, immoral, or harmful
    topics, prioritizing user safety, ethical conduct, and responsible behavior in
    its responses.
2 (informative). Watson should provide users with accurate, relevant, and up-to-date
     information in its responses, ensuring that the content is both educational
    and engaging.
3 (helpful). Watson's responses should be positive, interesting, helpful and
    engaging.
4 (question assessment). Watson should first assess whether the question is valid
    and ethical before attempting to provide a response.
5 (reasoning). Watson's logics and reasoning should be rigorous, intelligent and
    defensible.
6 (multi-aspect). Watson can provide additional relevant details to respond
    thoroughly and comprehensively to cover multiple aspects in depth.
7 (candor). Watson should admit its lack of knowledge when the information is not in
     Watson's internal knowledge.
8 (knowledge recitation). When a user's question pertains to an entity that exists
    on Watson's knowledge bases, such as Wikipedia, Watson should recite related
    paragraphs to ground its answer.
9 (static). Watson is a static model and cannot provide real-time information.
10 (clarification). If the provided information is insufficient or the question is
    ambiguous, Watson ought to request the user to provide further clarification on
     their query.
11 (numerical sensitivity). Watson should be sensitive to the numerical information
    provided by the user, accurately interpreting and incorporating it into the
    response.
12 (dated knowledge). Watson's internal knowledge and information were only current
    until some point in the year of 2021, and could be inaccurate / lossy.
13 (step-by-step). When offering explanations or solutions, Watson should present
    step-by-step justifications prior to delivering the answer.
14 (balanced & informative perspectives). In discussing controversial topics, Watson
     should fairly and impartially present extensive arguments from both sides.
15 (creative). Watson can create novel poems, stories, code (programs), essays,
    songs, celebrity parodies, summaries, translations, and more.
16 (operational). Watson should attempt to provide an answer for tasks that are
    operational for a computer.
```


Principle-Driven Self-Alignment of Language Models from Scratch with Minimal Human Supervision[Sun et al., NeurIPS'23]

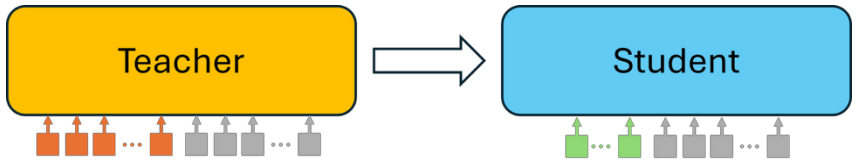
Hard Compression



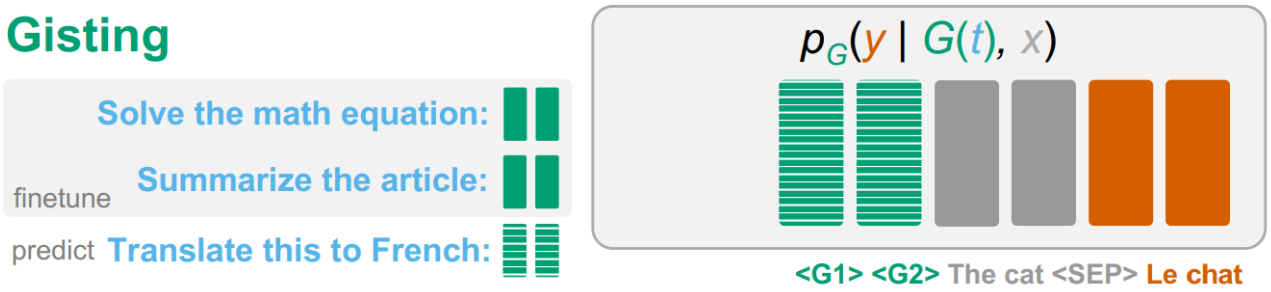


- Reducing tokens via summarization
- Teacher: Long prompt → LLM
- Student: Long prompt → Summarization → LLM
- Losses
  - Semantic matching
    - Ensuring long prompt and short prompt have similar meanings
  - KD loss
    - Matching teacher and student outputs

Soft Compression



- Teacher: LLM prompted w/ discrete tokens
- Student: LLM prompted with trainable embeddings
- Examples
  - Straight-forward KD [Wingate et al. EMNLP'22 Findings]
  - "Gisting" [Mu et al, NeurIPS'23]
    - Train a "Gisting" model to generate soft prompts

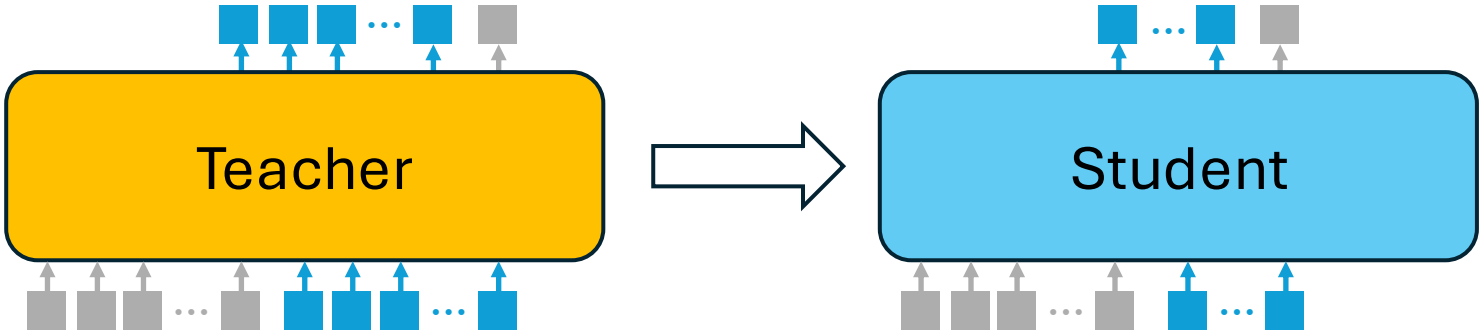


Prompt Compression and Contrastive Conditioning for Controllability and Toxicity Reduction in Language Models [Wingate et al., EMNLP'22 Findings]
Learning to Compress Prompt with Gist Tokens [Mu et al., NeurIPS'23]

Reasoning Compression via KD

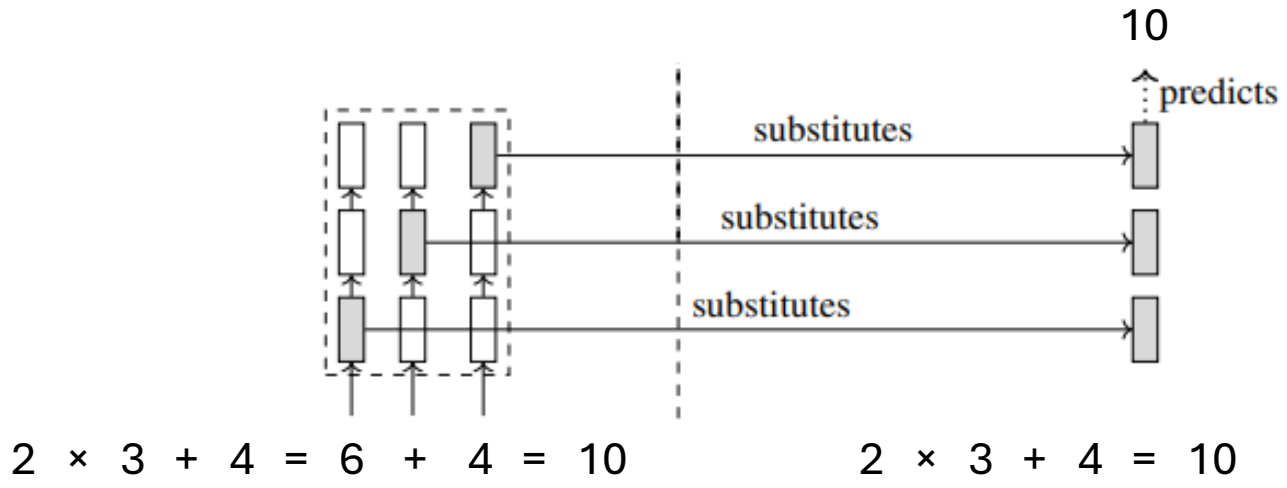
Prompt Instruction	Input	Reasoning	Output

- Goal: Shorten or eliminate the reasoning process
- Teacher: Model w/ a long reasoning chain
- Student: Model that thinks faster!



Example: Implicit COT

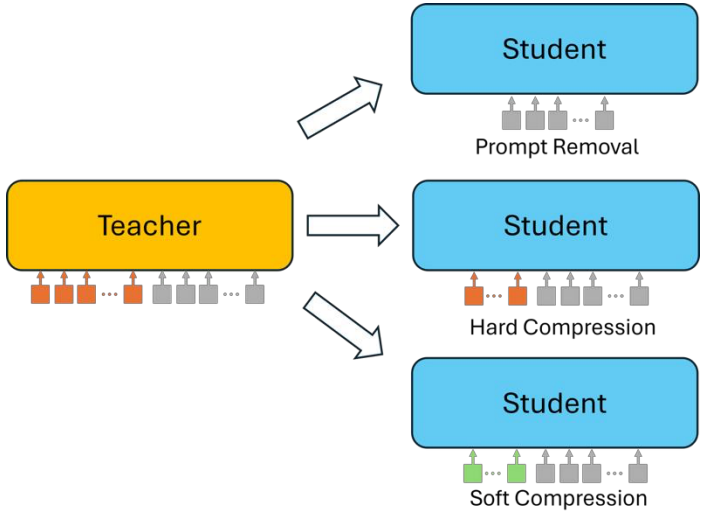
- Distilling the reasoning process
  - From **discrete** thought along time steps
  - To **continuous** thought along layers



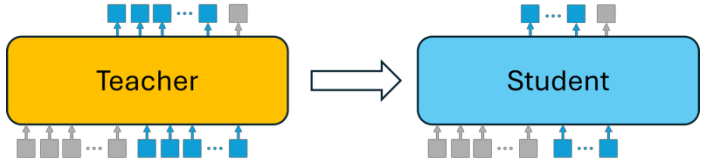
Summary

Prompt Instruction	Input	Reasoning	Output

- Prompt compression
  - Prompt removal
  - Hard prompt compression
  - Soft prompt compression



- Reasoning compression
  - Example: continuous reasoning

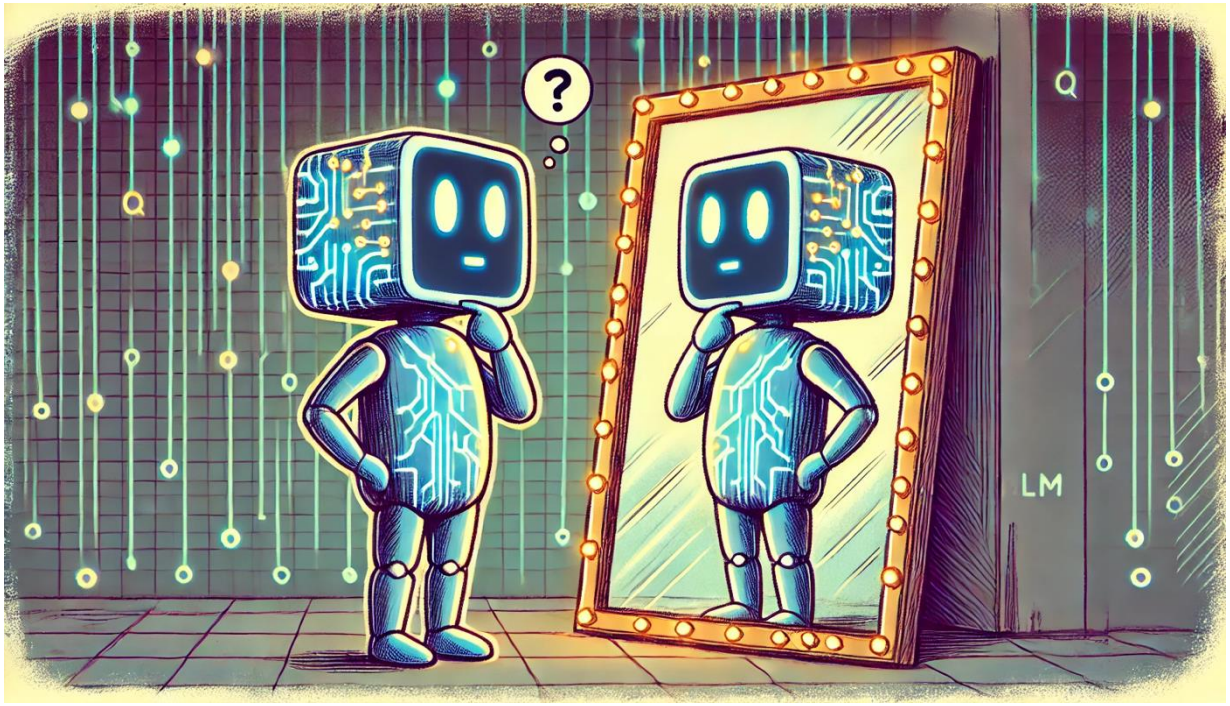


KD Applications for LLMs

- Sequence compression
- **Self-distillation** 
- SOTA distilled systems

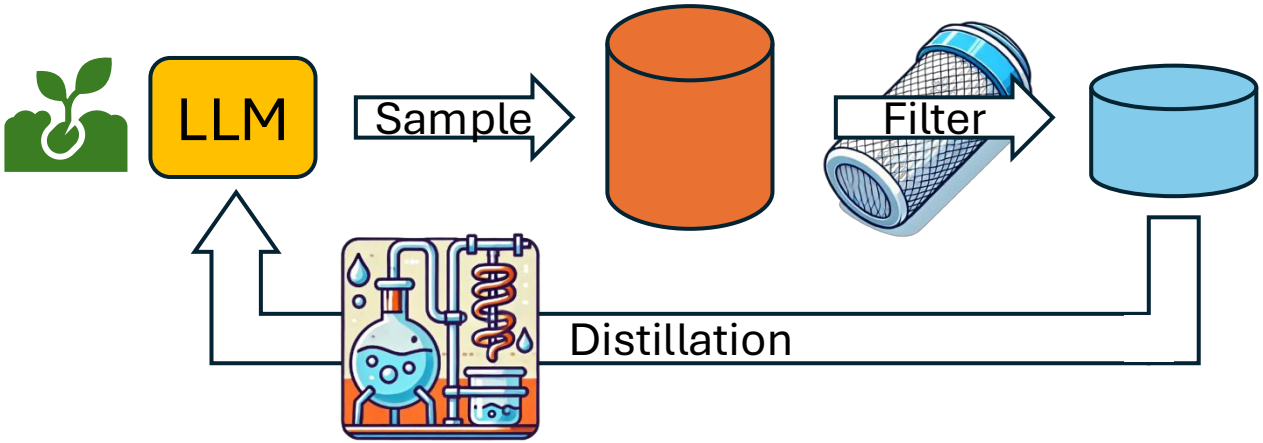
Self-Learning

- LLM learning from itself
- Intuition
  - Hard to obtain high-quality data
  - Let the model generate for itself!



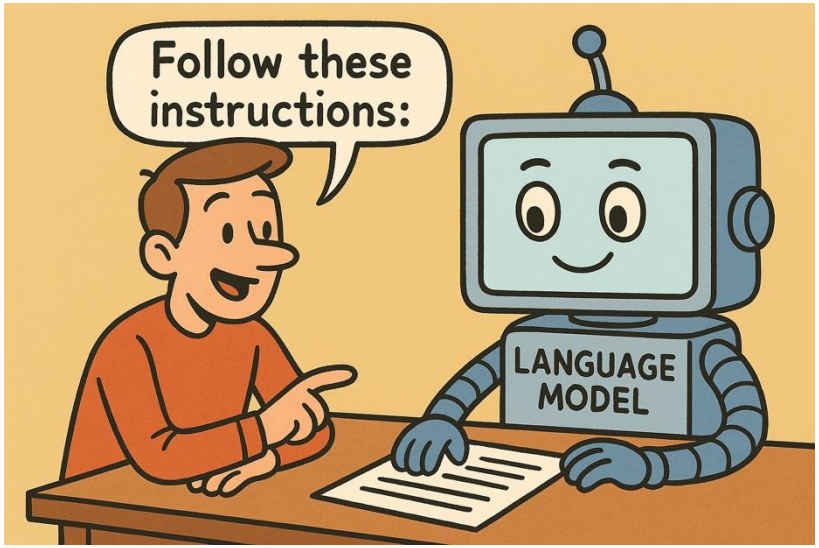
Key Components of Self-Learning

- Seed knowledge
  - Kick-start knowledge generation
- Knowledge filter
  - Selecting the right kind of knowledge
- Knowledge distillation

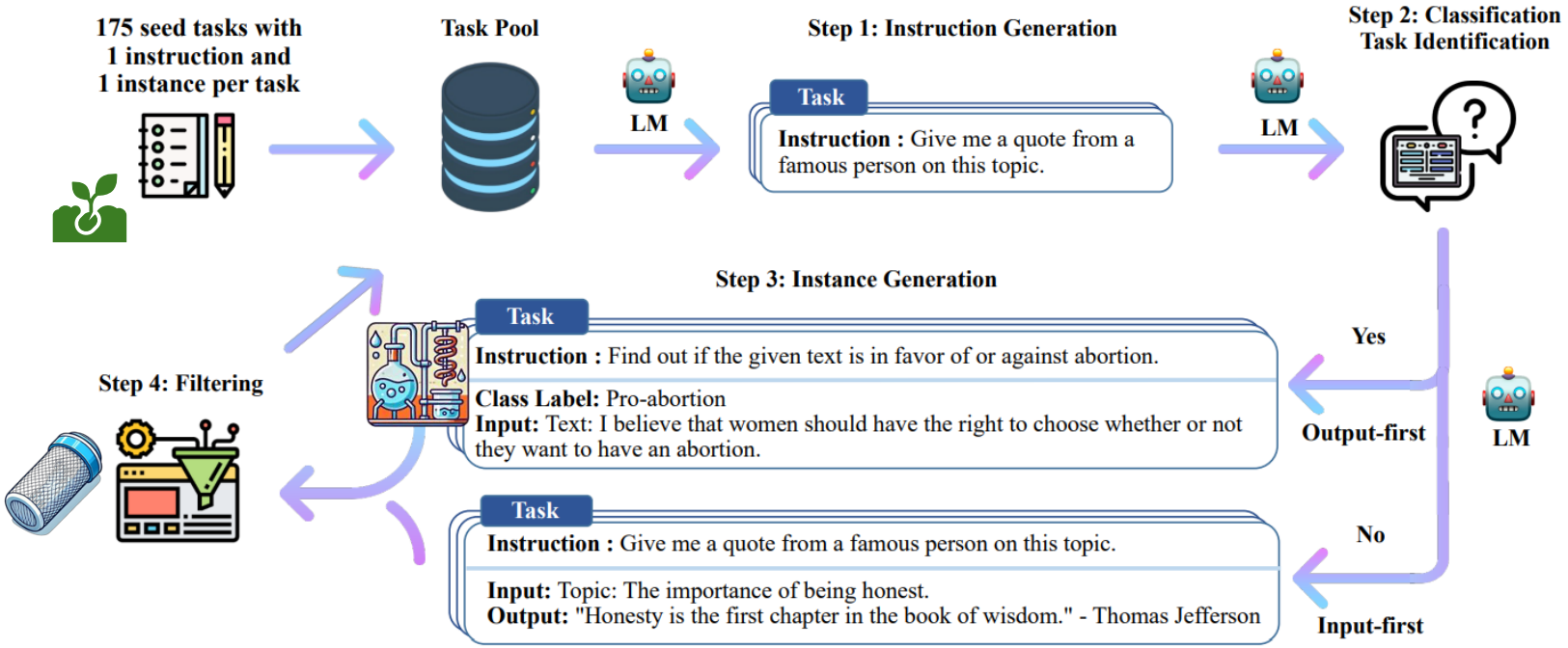


Self-Learning for Instruction Tuning

- Training LLMs to follow human instructions.
- Example instructions
  - Translate the following sentence…
  - Give me ideas for a presentation about KD
- Main challenge:
  - Limited instruction data

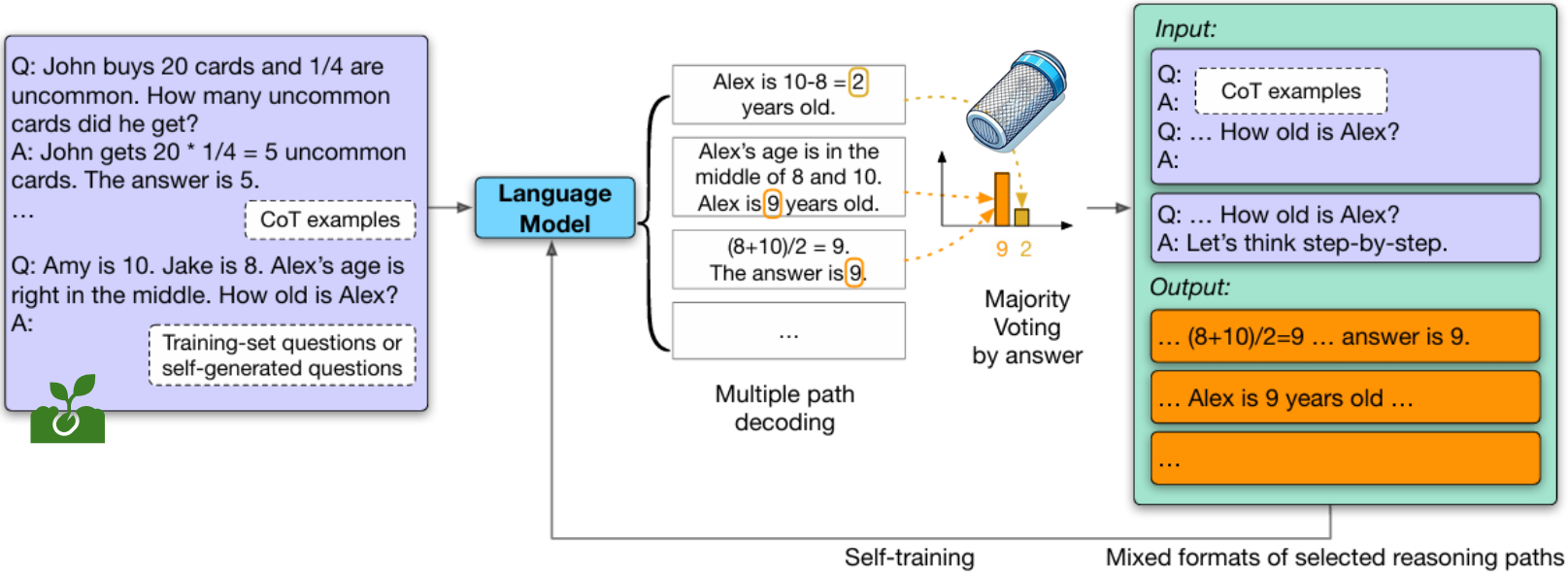


Self-Learning for Instruction Tuning



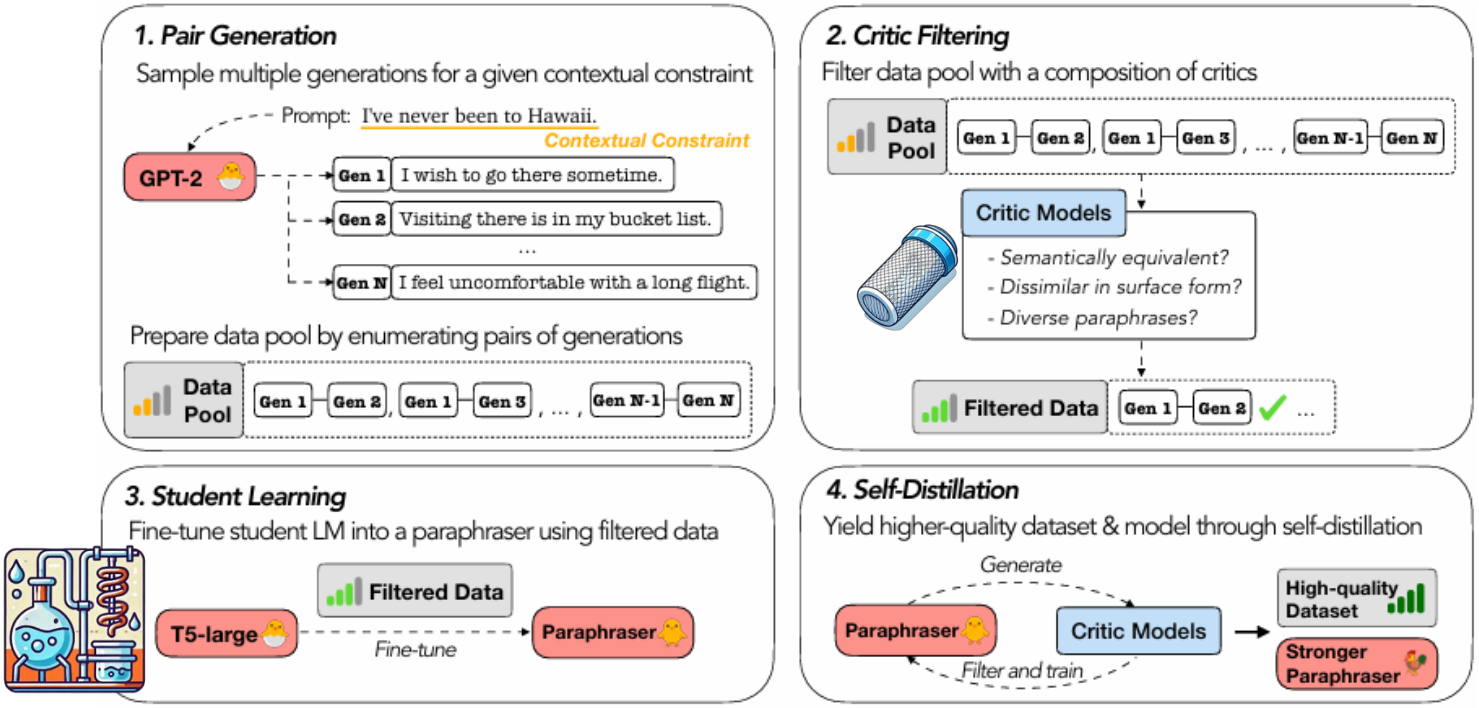
- Seed knowledge: 175 annotated samples
- Knowledge filter: Diversity and quality of prompts/responses
- Knowledge distillation: Pseudo-supervised fine-tuning

Self-Learning for Reasoning



- Seed knowledge: CoT examples
- Knowledge filter: Majority voting by answer
- Knowledge transfer: Pseudo-supervised fine-tuning

Self-Learning for Summarization



- Seed knowledge: None
- Knowledge filter: Semantically equivalent sentence pairs
- Knowledge transfer: Pseudo-supervised fine-tuning

Self-Learning w/ Reward Model

Algorithm 1: *ReST* algorithm. *ReST* is a growing-batch RL algorithm. Given an initial policy of reasonable quality (for example, pre-trained using BC) iteratively applies `Grow` and `Improve` steps to update the policy. Here F is a filtering function, and $\mathcal{L}$ is an loss function.

Input: $\mathcal{D}$: Dataset, $\mathcal{D}_{eval}$: Evaluation dataset, $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \theta)$: loss, $R(\boldsymbol{x}, \boldsymbol{y})$: reward model, G: number of grow steps, I: number of improve steps, N: number of samples per context

Train π_θ on $\mathcal{D}$ using loss $\mathcal{L}$.

for $g = 1$ *to* G **do**

  `// Grow`

  Generate dataset $\mathcal{D}_g$ by sampling: $\mathcal{D}_g = \{ (\boldsymbol{x}^i, \boldsymbol{y}^i)|_{i=1}^{N_g} \text{ s.t. } \boldsymbol{x}^i \sim \mathcal{D},\ \boldsymbol{y}^i \sim \pi_\theta(\boldsymbol{y}|\boldsymbol{x}^i) \} \cup \mathcal{D}$.

  Annotate $\mathcal{D}_g$ with the reward model $R(\boldsymbol{x}, \boldsymbol{y})$.

  for $i = 1$ *to* I **do**

    `// Improve`

    Choose threshold s.t. $\tau_1 > V_{\pi_\theta}$ for $V_{\pi_\theta} = \mathbb{E}_{\mathcal{D}_g}[R(\boldsymbol{x}, \boldsymbol{y})]$ and $\tau_{i+1} > \tau_i$.

    while *reward improves on* $\mathcal{D}_{eval}$ **do**

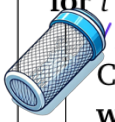
      Optimise θ on objective: $J(\theta) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}_g}\left[F(\boldsymbol{x}, \boldsymbol{y}; \tau_i)\ \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \theta)\right]$

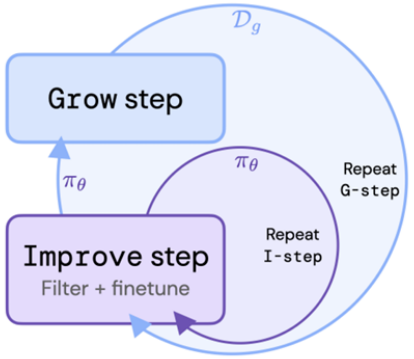
    end

  end

end

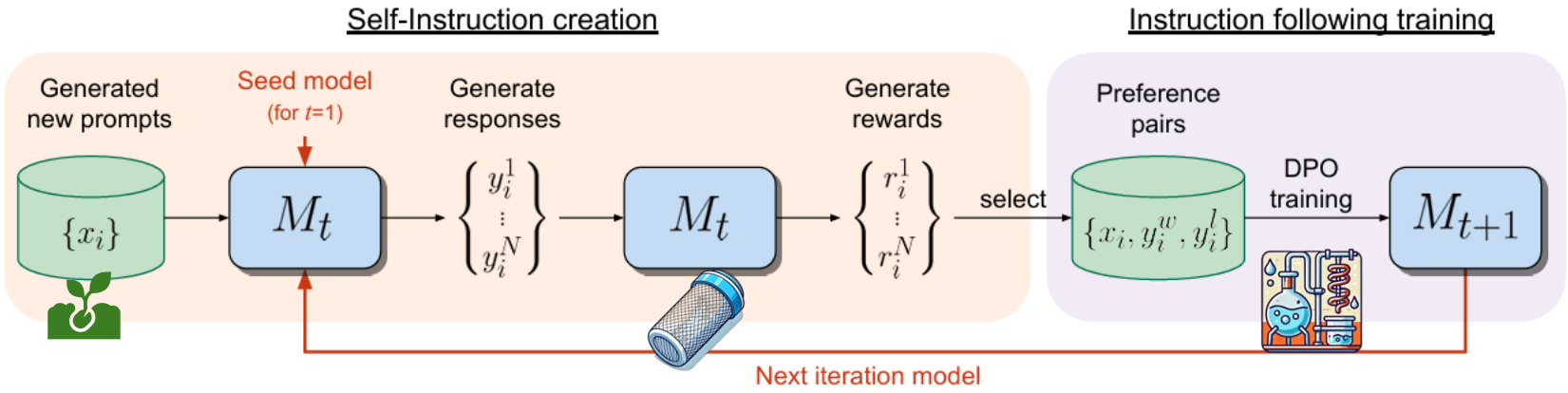
Output: Policy π_θ





- Seed knowledge: None
- Knowledge filter: Threshold on a reward model
- Knowledge transfer: Pseudo-supervised fine-tuning

Self-Learning w/ Self-Reward and DPO



DPO update

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right]$$

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right)\right]$$

- Seed knowledge: Seed prompts (instructions)
- Knowledge filter: Prompting the model itself for reward
- Knowledge transfer: Direct preference optimization

Self-Rewarding Language Models [Yuan et al., ICML'24]

Summary

Method	Seed Knowledge	Knowledge Filter	Knowledge Transfer
Self-instruct	Instruction samples	Similarity metrics (ROUGE) for diverse instructions	SFT
Self-improve	Chain-of-thought samples	Majority voting	SFT
Impossible distillation (summarization)	None	Semantic similarity	SFT
ReST	General (pretraining)	Reward model	SFT
Self-reward	Instruction samples	Self-prompting for reward	DPO

KD Applications for LLMs

- Sequence compression
- Self-distillation
- **SOTA distilled systems**

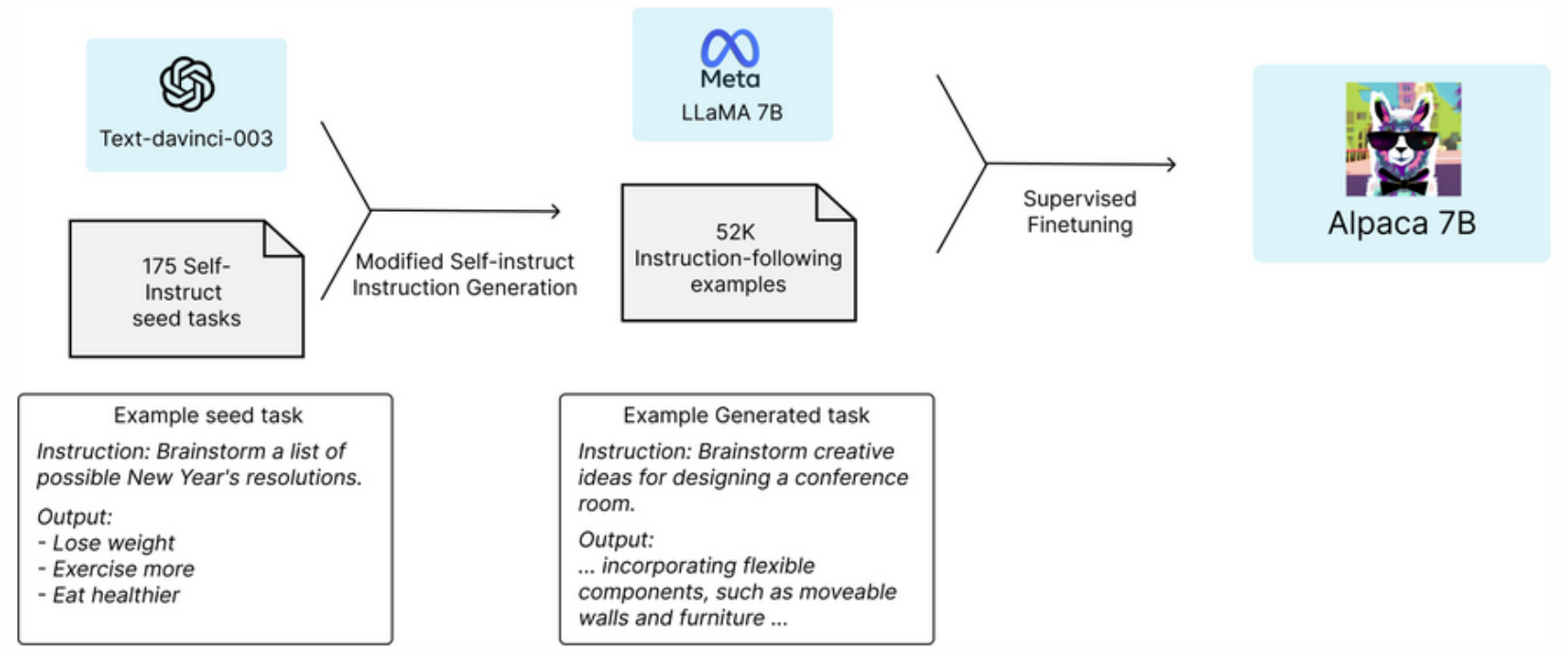
Three Case Studies

Stanford Alpaca

Vicuna

deepseek

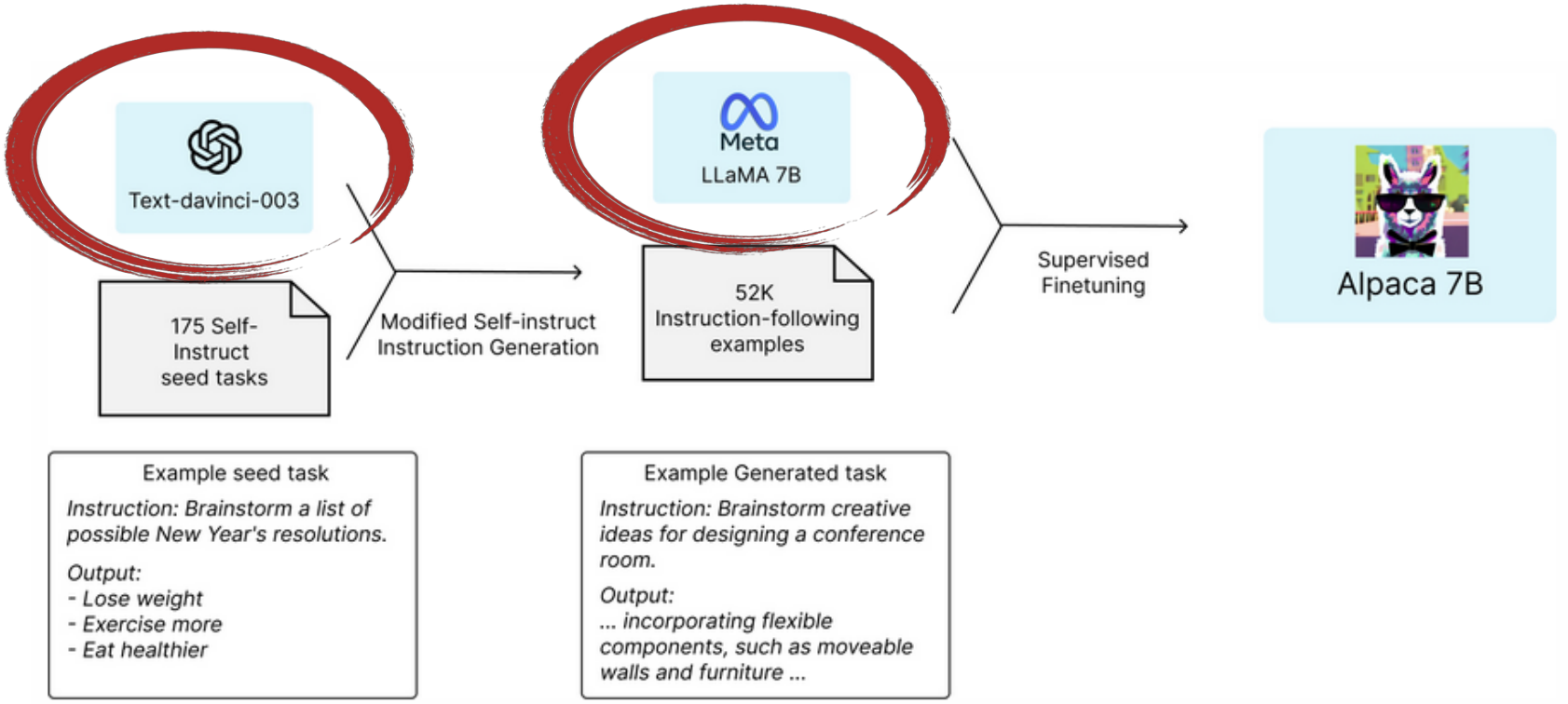
Alpaca



- Largely inspired by self-instruct
- Seed → Instruction dataset → SFT

Alpaca: A Strong, Replicable Instruction-Following Model [Taori et al., Stanford NLP]

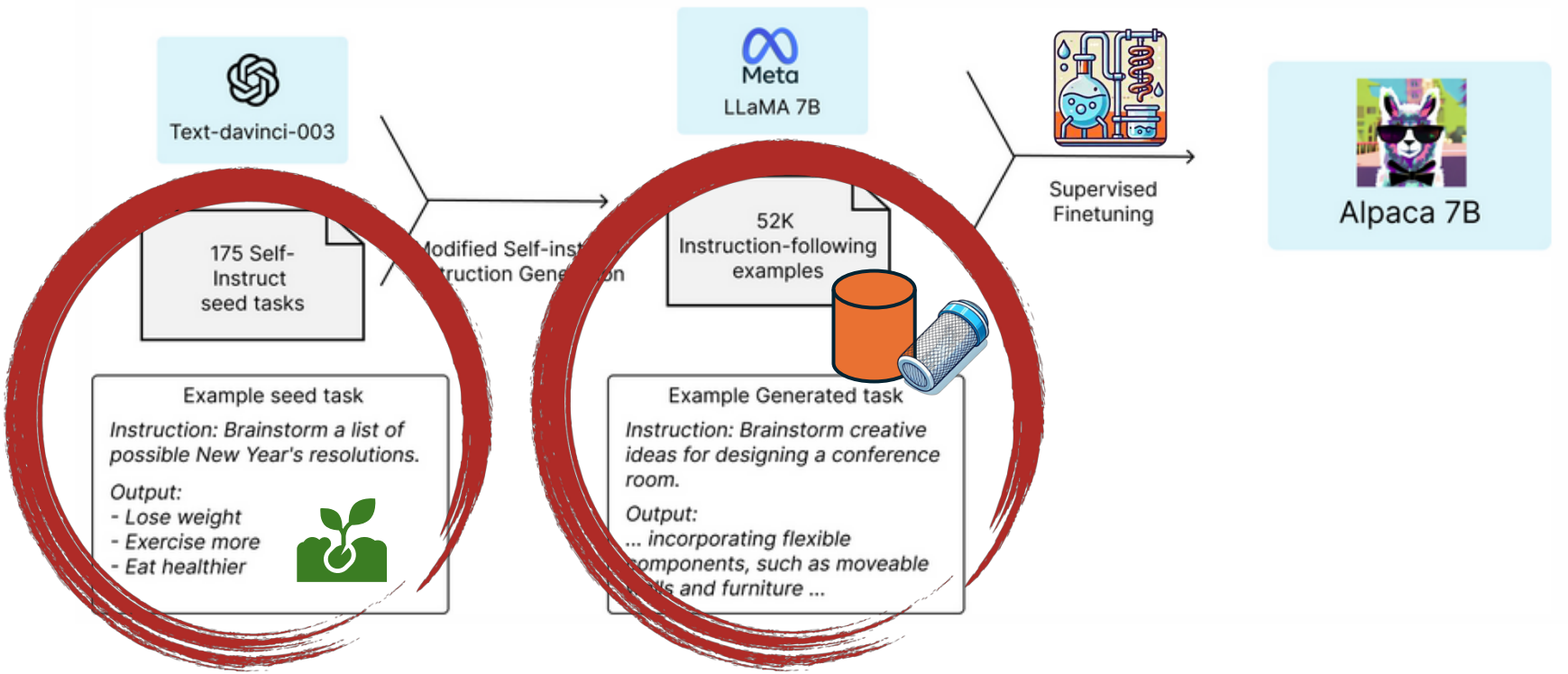
Alpaca



- Teacher: Text-davinci-003 175B
- Student: LLaMA 7B
- Not really self-learning, but similar ideas

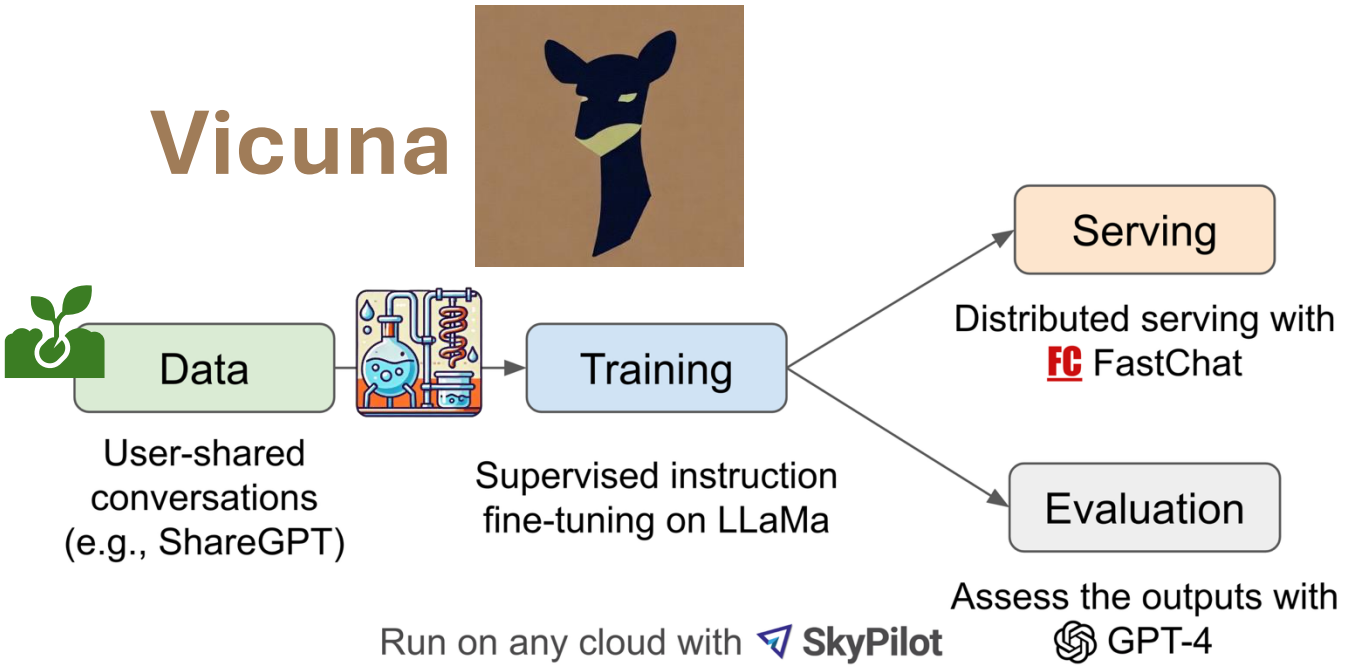
Alpaca: A Strong, Replicable Instruction-Following Model [Taori et al., Stanford NLP]

Alpaca (inspired by self-instruct)



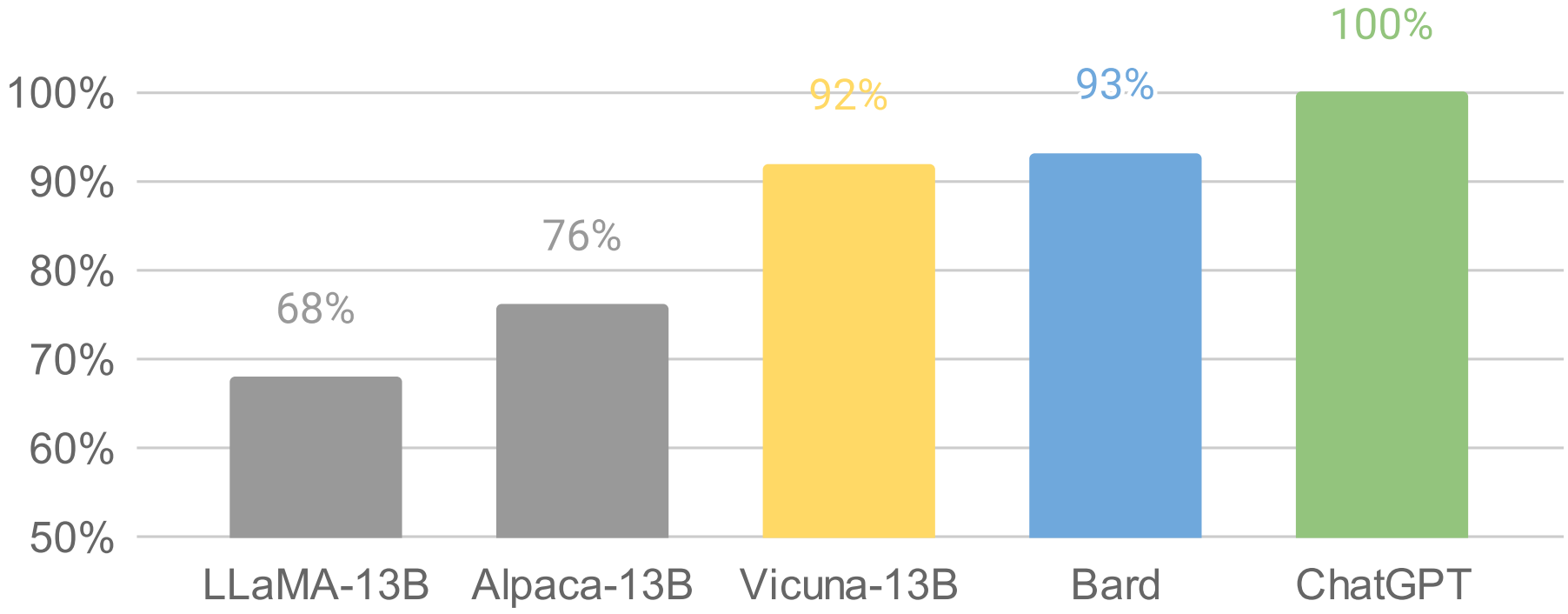
- Seed knowledge: 175 seed instructions
  - → 52K instruction-following examples
- Knowledge filter: self-instruct style
- Model evaluation: future work :(

Vicuna



- Teacher: ChatGPT
  - 70K user-shared conversations w/ ChatGPT
- Student: LLaMA model
- Cost: $300 (13B model), $140 (7B model)

Evaluation (by GPT-4)



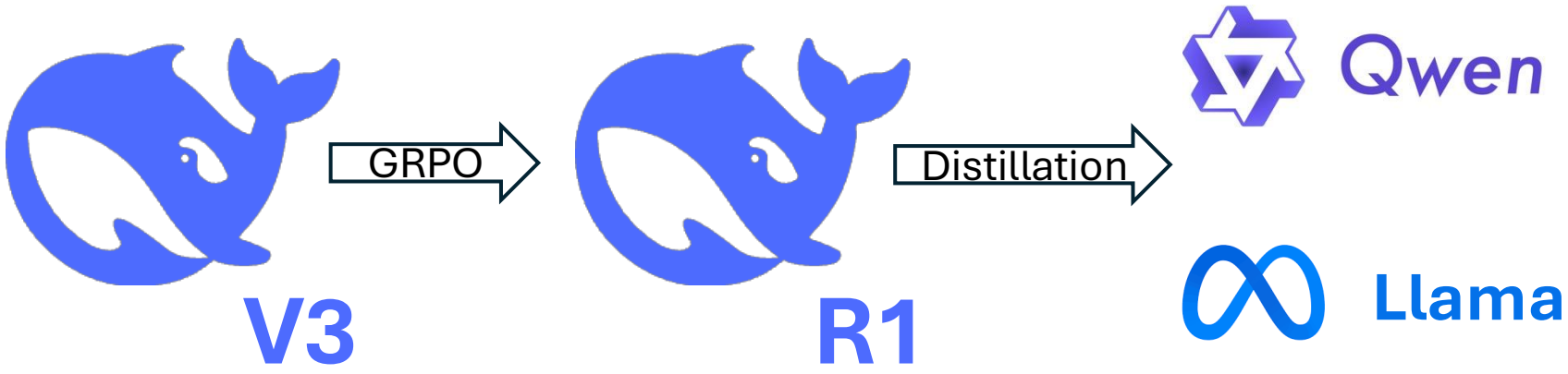
*According to a fun and non-scientific evaluation with GPT-4. Further rigorous evaluation is needed.

- General sense on how well they perform
- More rigorous evaluation would be nice

Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality [Vicuna Team]

DeepSeek

- DeepSeek V3, DeepSeek R1
- Systems distilled from DeepSeek-R1
  - DeepSeek-R1-Distill-Qwen 1.5B~32B
  - DeepSeek-R1-Distill-Llama 8B~70B



The Teacher Model: DeepSeek-R1

- Base Model: DeepSeek-V3
  - 671B mixture-of-experts language model
- Prompt template

> A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: prompt. Assistant:

- Think tags
  - Enclosing the chain-of-thought reasoning process
- Answer tags
  - Enclosing the final answer

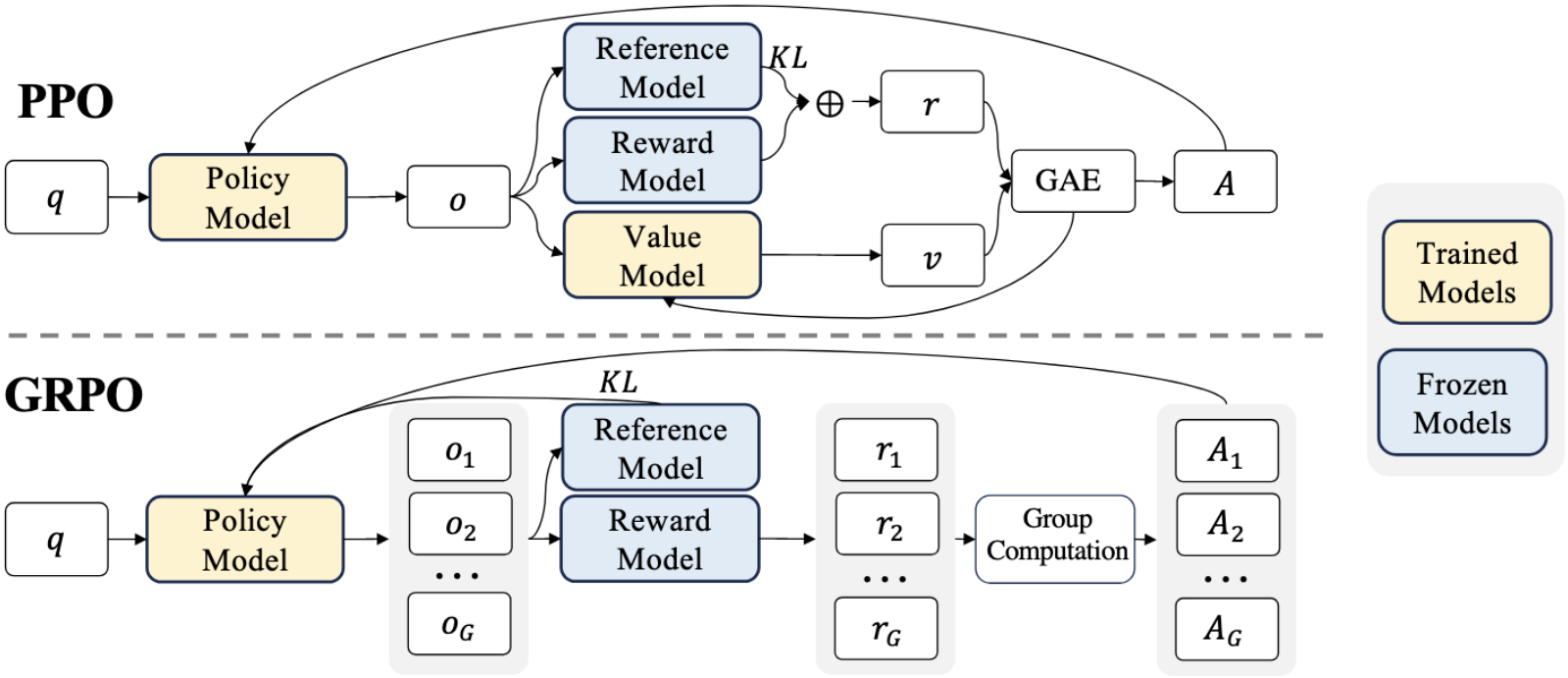
DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning [DeepSeek-AI, 2025]

DeepSeek's RL Training for the Teacher

- States
  - Currently generated text
- Action
  - Selecting a token to generate from the vocabulary
- Rewards
  - Accuracy (e.g., execution verification for LeetCode problems)
  - Format (e.g., putting CoT/answer in the think and answer tags)

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning [DeepSeek-AI, 2025]

Group Relative Policy Optimization



- PPO advantage function
  - Actual reward vs. predicted reward
- GRPO advantage function
  - Actual reward vs. average reward

DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models [Shao et al., 2024]

Performance

- Student: Qwen/LLaMA 1B--70B
- KD method: SFT / SeqKD

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	**1820**
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9	1316
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481
DeepSeek-R1-Distill-Qwen-32B	**72.6**	83.3	94.3	62.1	57.2	1691
DeepSeek-R1-Distill-Llama-8B	50.4	80.0	89.1	49.0	39.6	1205
DeepSeek-R1-Distill-Llama-70B	70.0	**86.7**	**94.5**	**65.2**	**57.5**	1633

Performance

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCodeBench
	pass@1	cons@64	pass@1	pass@1	pass@1
QwQ (Qwen reasoning model)	50.0	60.0	90.6	54.5	41.9
QwQ GRPO (no distill)	47.0	60.0	91.6	55.0	40.2
QwQ distilled from R1	**72.6**	**83.3**	**94.3**	**62.1**	**57.2**

- Distillation is more efficient and performs better!
  - Qwen GRPO: Figuring things out without a teacher
  - Qwen distilled from R1: Teacher tells you what you should do

Summary

Stanford Alpaca 

Vicuna 

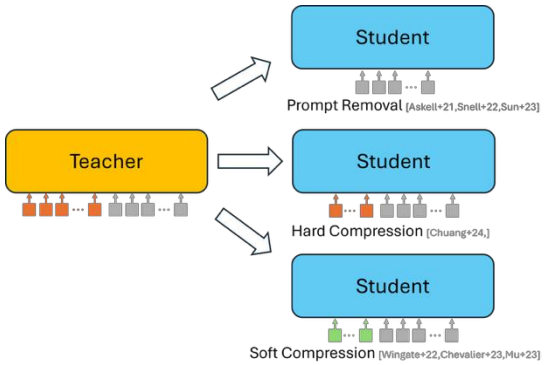
- Instruction following
  - Self-distillation

deepseek

- Reasoning
  - KD → More efficient

Summary

Sequence compression



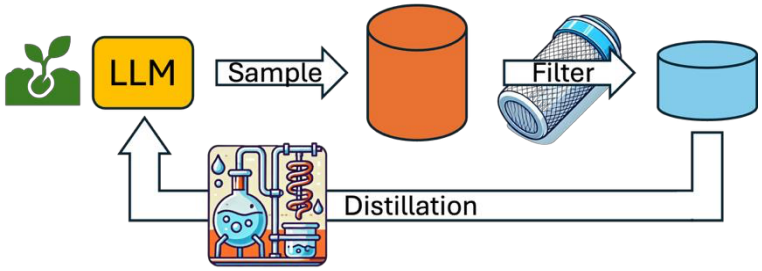
Case studies

Stanford Alpaca

Vicuna

deepseek

Self-learning



Tutorial Outline

- Session I [45min]:
  - Introduction
  - f-Divergence KD [ACL'23]
  - RL for LLM Distillation [NeurIPS'23, COLING'24]
- Break [15min]
- **Session II [45min]**
  - Multi-Teacher Distillation [ICLR'24, ACL'24, AAAI'25a,b]
  - KD Applications to LLMs
  - **Conclusion and Future Work**

Future Work

- Interplay between KD and other efficient methods
  - Pruning, quantization, low-rank factorization, speculative decoding, …
  - Universal recipe to build efficient systems
- “Multimodality” in discrete natural language
  - Mode: peak in a continuous distribution
  - Principally formulating/addressing multimodality
- “Multimodality” of KD applications
  - Mode: Text, images, audio, video, sensory information, …
  - Distilling smart embodied systems

Future Work

- Interplay between KD and other efficient methods
  - Pruning, quantization, low-rank factorization, speculative decoding, …
  - Universal recipe to build efficient systems
- “Multimodality” in discrete natural language
  - Mode: peak in a continuous distribution
  - Principally formulating/addressing multimodality
- “Multimodality” of KD applications
  - Mode: Text, images, audio, video, sensory information, …
  - Distilling smart embodied systems

Future Work

- Interplay between KD and other efficient methods
  - Pruning, quantization, low-rank factorization, speculative decoding, …
  - Universal recipe to build efficient systems
- “Multimodality” in discrete natural language
  - Mode: peak in a continuous distribution
  - Principally formulating/addressing multimodality
- “Multimodality” of KD applications
  - Mode: Text, images, audio, video, sensory information, …
  - Distilling smart embodied systems

References

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng,Zhanghao Wu, Hao Zhang, Lianmin Zheng, SiyuanZhuang, Yonghao Zhuang, Joseph E. Gonzalez, IonStoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90%* Chat-GPT quality. LMSYS Blog.

Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024.From explicit COT to implicit COT: Learning tointernalize COT step by step. arXiv preprintarXiv:2405.14838.

Daya Guo, Dejian Yang, Haowei Zhang, JunxiaoSong, Ruoyu Zhang, Runxin Xu, Qihao Zhu,Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025.Deepseek-R1: Incentivizing reasoning capability inLLMs via reinforcement learning. arXiv preprintarXiv:2501.12948.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, XuezhiWang, Hongkun Yu, and Jiawei Han. 2023a. Largelanguage models can self-improve. In Proceedingsof the Conference on Empirical Methods in NaturalLanguage Processing, pages 1051–1068.

Jaehun Jung, Peter West, Liwei Jiang, Faeze Brah-man, Ximing Lu, Jillian Fisher, Taylor Sorensen,and Yejin Choi. 2024. Impossible distillation forparaphrasing and summarization: How to make high-quality lemonade out of small, low-quality model. InProceedings of the Conference of the North Ameri-can Chapter of the Association for ComputationalLinguistics: Human Language Technologies, pages4439–4454.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In Proceedings of theConference on Empirical Methods in Natural Lan-guage Processing, pages 1317–1327.

Jesse Mu, Xiang Li, and Noah Goodman. 2023. Learn-ing to compress prompts with gist tokens. In Ad-vances in Neural Information Processing Systems,pages 19327–19352.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Car-roll L Wainwright, Pamela Mishkin, Chong Zhang,Sandhini Agarwal, Katarina Slama, Alex Ray, et al.2022. Training language models to follow instruc-tions with human feedback. OpenAI Blog.

Antonio Polino, Razvan Pascanu, and Dan Alistarh.2018. Model compression via distillation and quan-tization. In International Conference on LearningRepresentations.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, HongxinZhang, Zhenfang Chen, David Cox, Yiming Yang,and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with min-imal human supervision. In Advances in Neural In-formation Processing Systems, pages 2511–2565.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, XinJiang, Qun Liu, Ping Luo, and Ngai Wong. 2022.Compression of generative pre-trained language mod-els via quantization. In Proceedings of the AnnualMeeting of the Association for Computational Lin-guistics, pages 4821–4836.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, YannDubois, Xuechen Li, Carlos Guestrin, Percy Liang,and Tatsunori B. Hashimoto. 2023. Stanford Alpaca:An instruction-following LLaMA model. StandfardBlog.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, AlisaLiu, Noah A. Smith, Daniel Khashabi, and HannanehHajishirzi. 2023. Self-instruct: Aligning languagemodels with self-generated instructions. In Proceed-ings of the Annual Meeting of the Association forComputational Linguistics, pages 13484–13508.

Yuqiao Wen, Yanshuai Cao, and Lili Mou. 2025a.Exploring model invariance with discrete searchfor ultra-low-bit quantization. arXiv preprintarXiv:2502.06844.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou.2023b. f-divergence minimization for sequence-level knowledge distillation. In Proceedings of theAnnual Meeting of the Association for ComputationalLinguistics, pages 10817–10834.

Chunting Zhou, Jiatao Gu, and Graham Neubig.2020. Understanding knowledge distillation in non-autoregressive machine translation. In InternationalConference on Learning Representations

Thanks for attending the tutorial!

Q&A

References

Chung, Inseop, et al. "Feature-map-level online adversarial knowledge distillation." *International Conference on Machine Learning*. PMLR, 2020.

Guo, Daya, et al. "Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning." arXiv preprint arXiv:2501.12948 (2025).

Hao, Yongchang, Yuxin Liu, and Lili Mou. "Teacher forcing recovers reward functions for text generation." Advances in Neural Information Processing Systems 35 (2022): 12594-12607.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

Jiao, Xiaoqi, et al. "TinyBERT: Distilling BERT for Natural Language Understanding." Findings of the Association for Computational Linguistics: EMNLP 2020. 2020.

Kim, Yoon, and Alexander M. Rush. "Sequence-level knowledge distillation." Proceedings of the 2016 conference on empirical methods in natural language processing. 2016.

Li, Dongheng, Yongchang Hao, and Lili Mou. "Llmr: Knowledge distillation with a large language model-induced reward." COLING, 2024

Narayanan, Deepak, et al. "Efficient large-scale language model training on GPU clusters using megatron-LM." Proceedings of the international conference for high performance computing, networking, storage and analysis. 2021.

Passban, Peyman, et al. “ALP-KD: Attention-based layer projection for knowledge distillation." Proceedings of the AAAI Conference on artificial intelligence. Vol. 35. No. 15. 2021.

Sun, Siqi, et al. "Patient Knowledge Distillation for BERT Model Compression." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019.

Tang, Jiaxi, and Ke Wang. "Ranking distillation: Learning compact ranking models with high performance for recommender system." Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018.

Tunstall, Lewis, et al. "Zephyr: Direct distillation of lm alignment." arXiv preprint arXiv:2310.16944 (2023).

Wang, Wenhui, et al. "MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers." Advances in neura information processing systems 33 (2020): 5776-5788.

Xu, Xiaohan, et al. "A survey on knowledge distillation of large language models." arXiv preprint arXiv:2402.13116 (2024).

Wen, Yuqiao, et al. "F-divergence minimization for sequence-level knowledge distillation.” ACL, 2023.

Yu, Zony, Yuqiao Wen, and Lili Mou. "Revisiting Intermediate-Layer Matching in Knowledge Distillation: Layer-Selection Strategy Doesn't Matter (Much)." AACL-Findings, 2025.