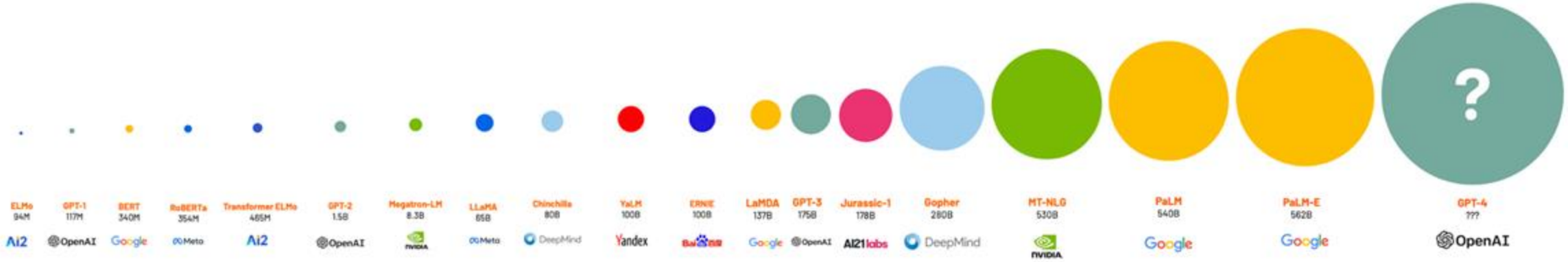
Under Review

Song-ha Jo, Youngrok Ko, Sang-goo Lee, Jinseok Seol

Task-aware Block Pruning with Output Distribution Signals for LLMs

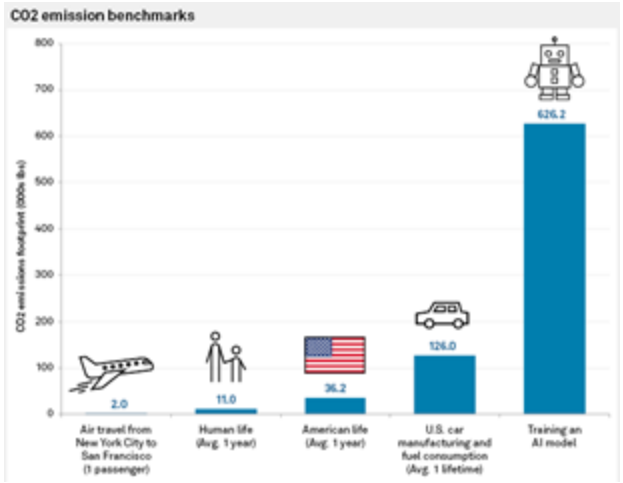
Background

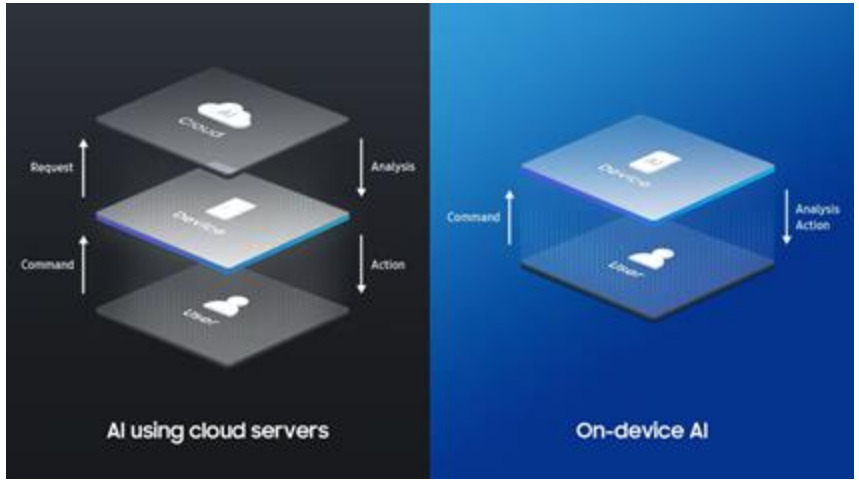
- **Large Language Models** (LLM)
  - remarkable performance across diverse natural language processing (NLP) tasks
  - however, size matters in deployment scenarios
    - high training & inference cost
    - resource-constrained environments



Background

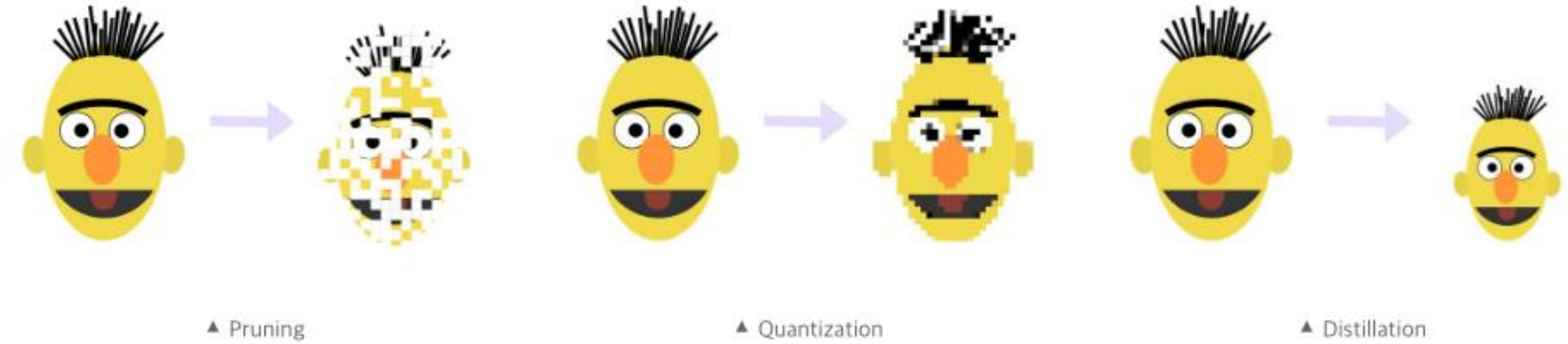
- **Large Language Models** (LLM)
  - remarkable performance across diverse natural language processing (NLP) tasks
  - however, size matters in deployment scenarios
    - high training & inference cost
    - resource-constrained environments





Background

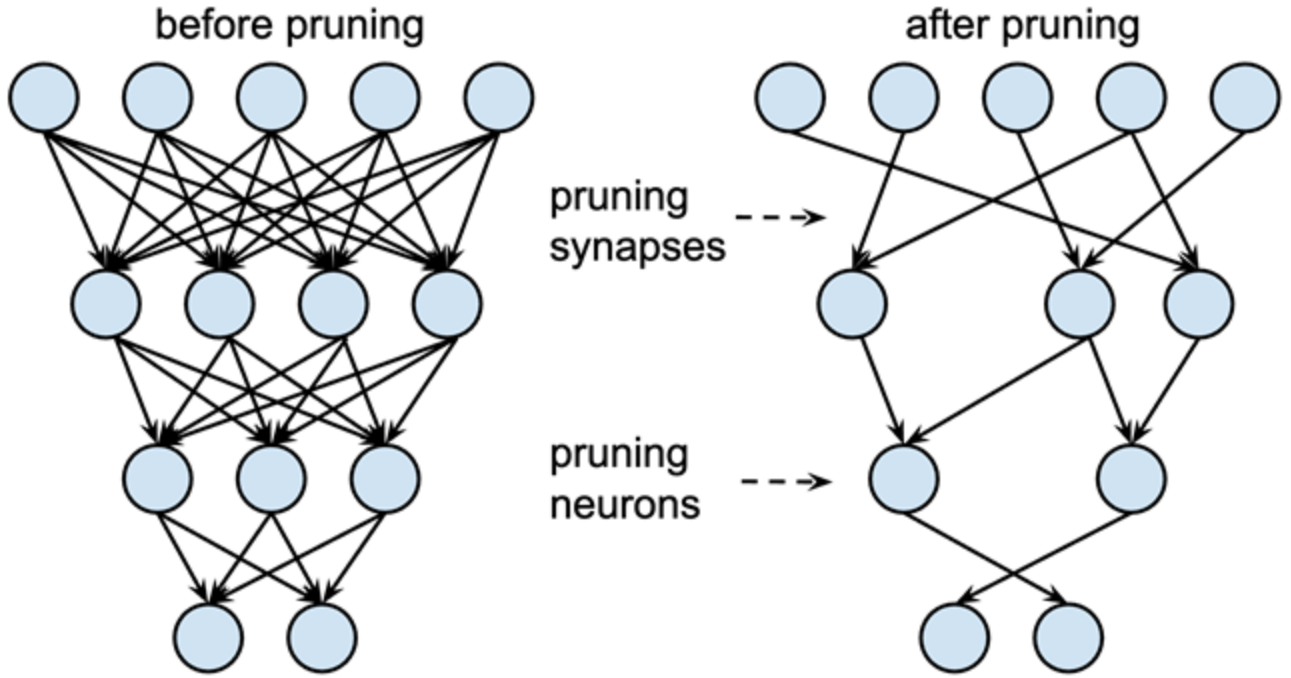
- **Lightweighting in Deep Learning**
  - Pruning
  - Quantization
  - Knowledge Distillation (KD)



▲ Pruning ▲ Quantization ▲ Distillation

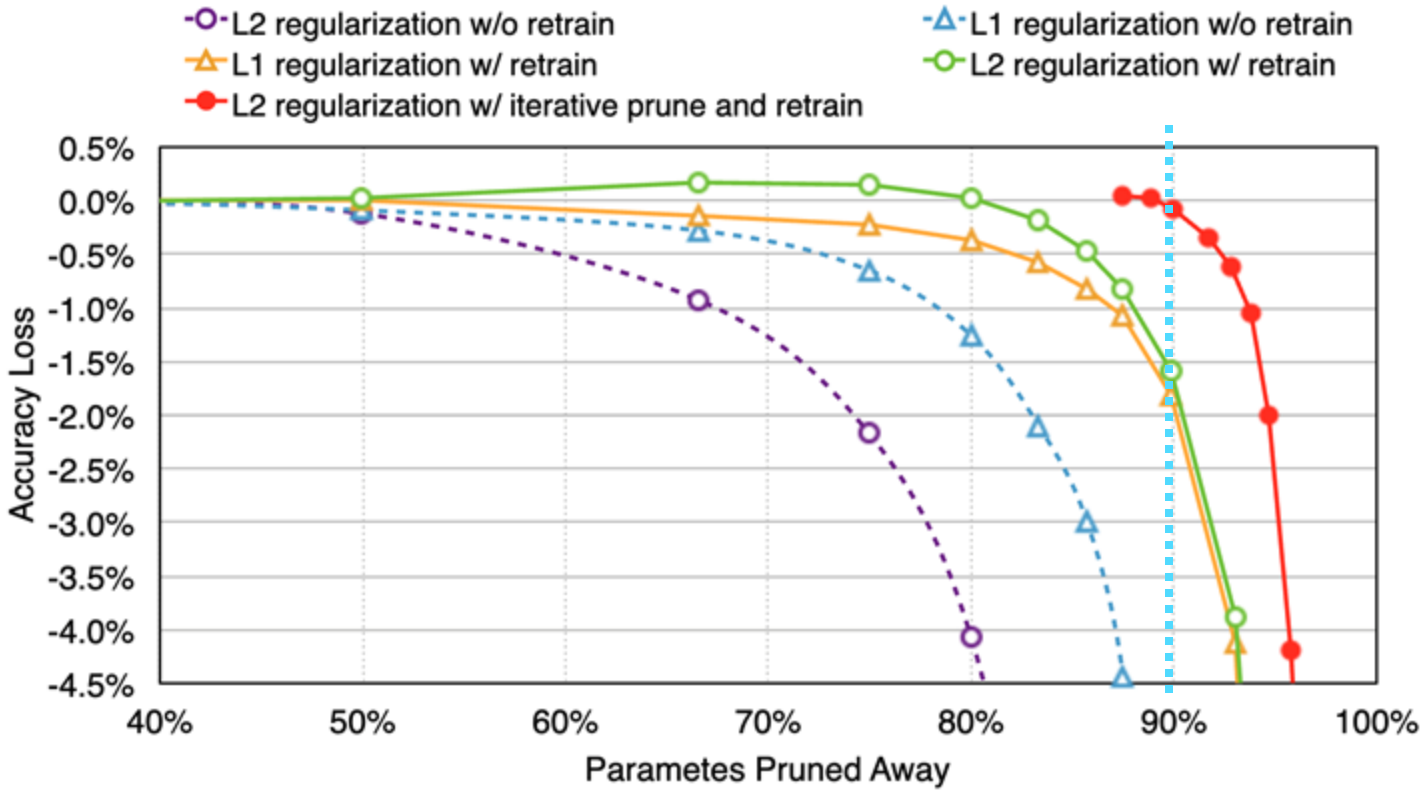
Background

- **Pruning**
  - in neural network, prune less impactful nodes (neurons) or edges (synapses)



Background

- **Pruning**
  - in some cases, even 90% of parameters can be pruned away

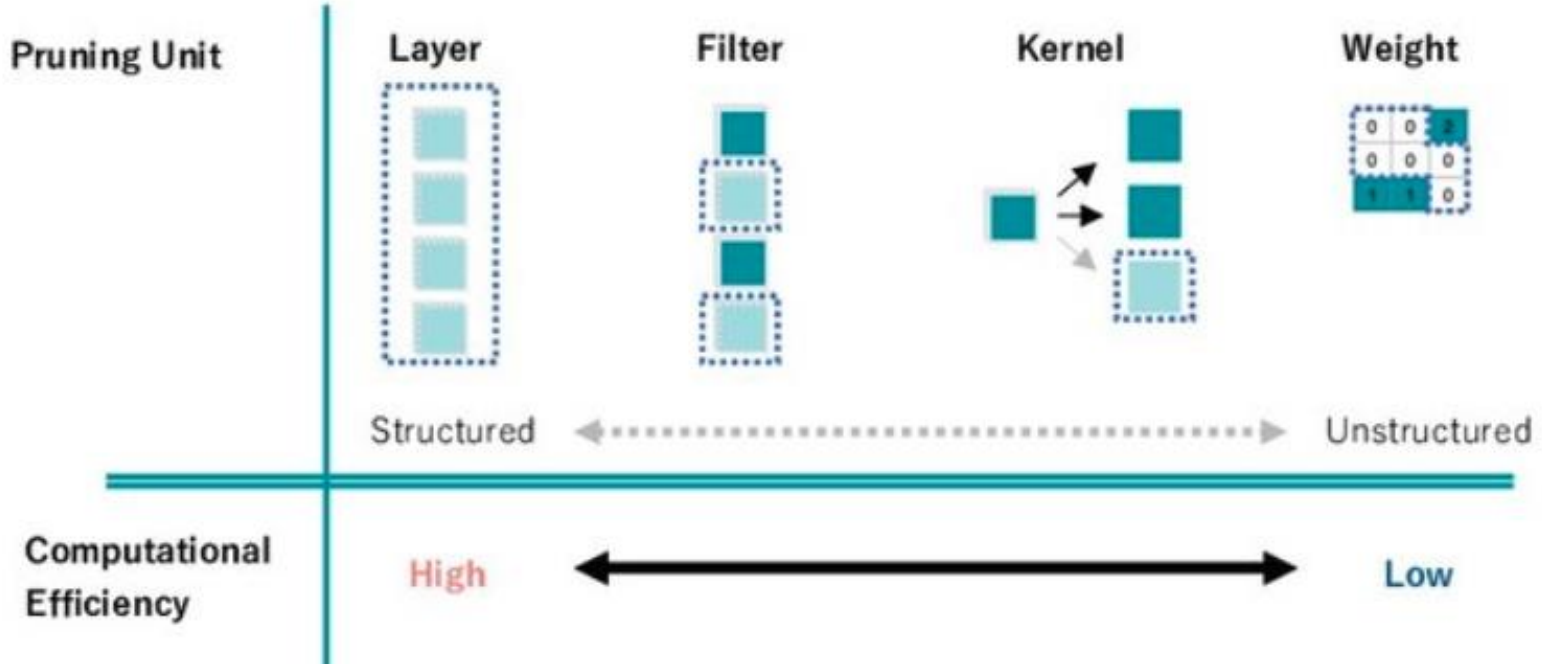


Background

- **Effect of Pruning**
  - less parameters
    - but retains performance
      - sometimes, improving
  - less memory usage (space)
  - less computational cost (time)

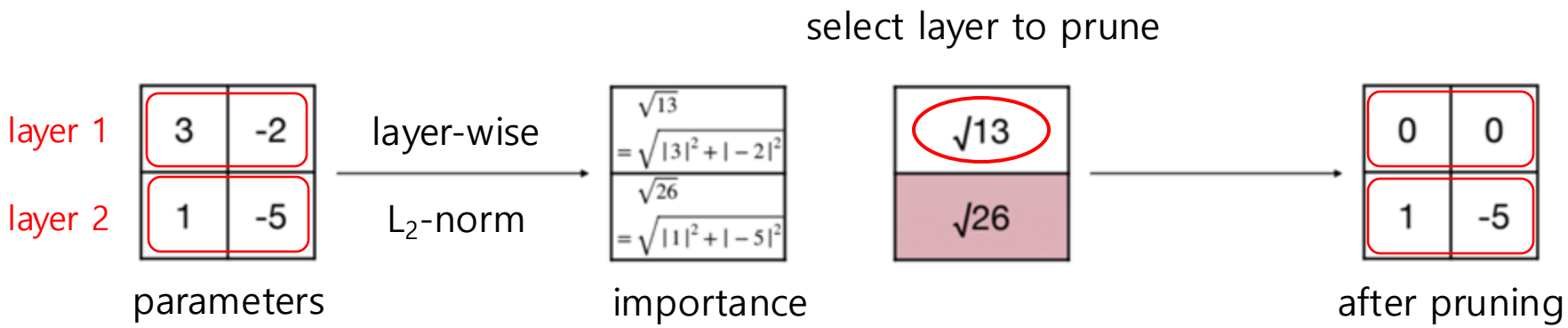
Background

- **Pruning Structures**
  - structured vs unstructured
    - structured pruning yields time & space acceleration immediately



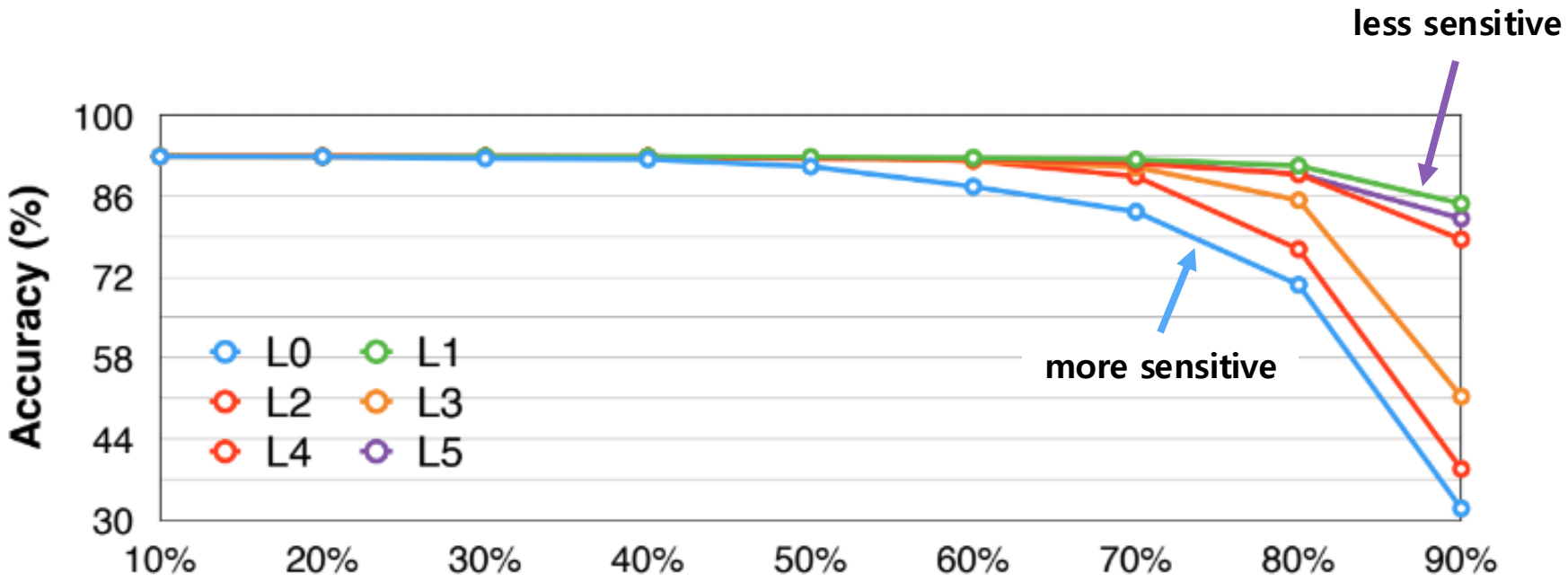
Background

- **Pruning Criteria**
  - which layer to prune?
    - absolute value of parameters, representation similarity, perplexity, …



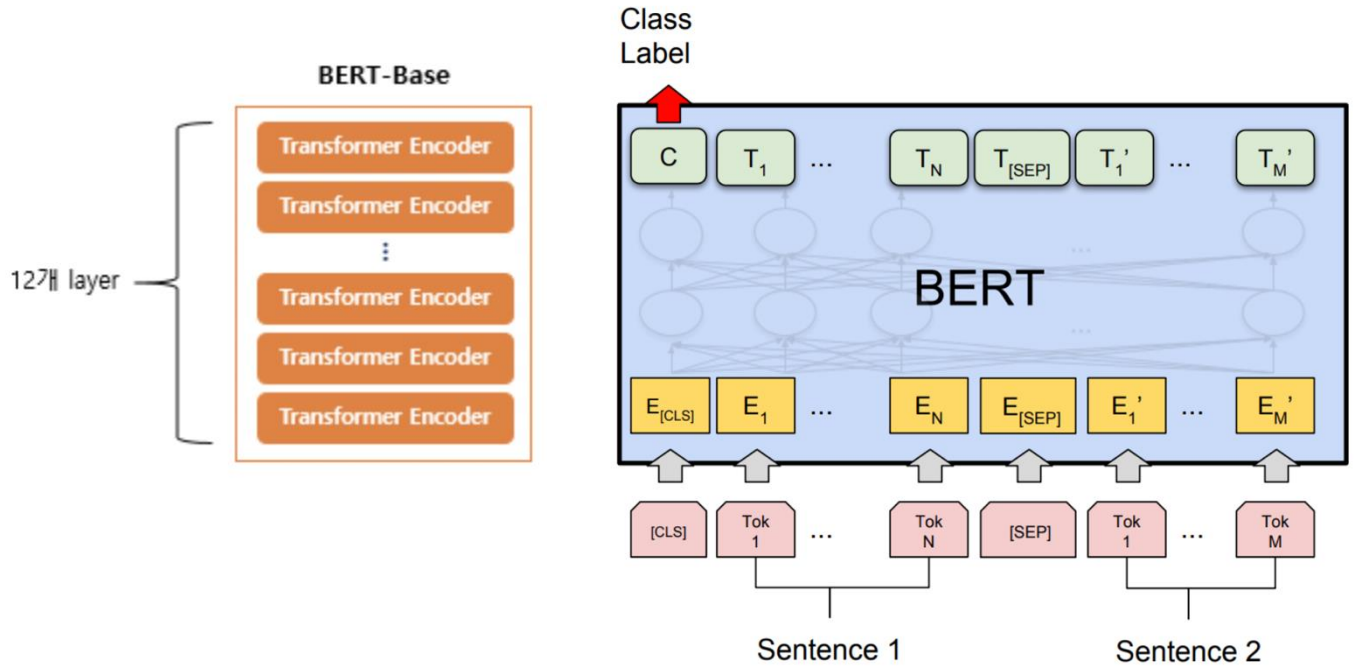
Background

- **Sensitivity Analysis**
  - for each layer, for each pruning rate
  - prune a layer, retrain, measure accuracy degradation



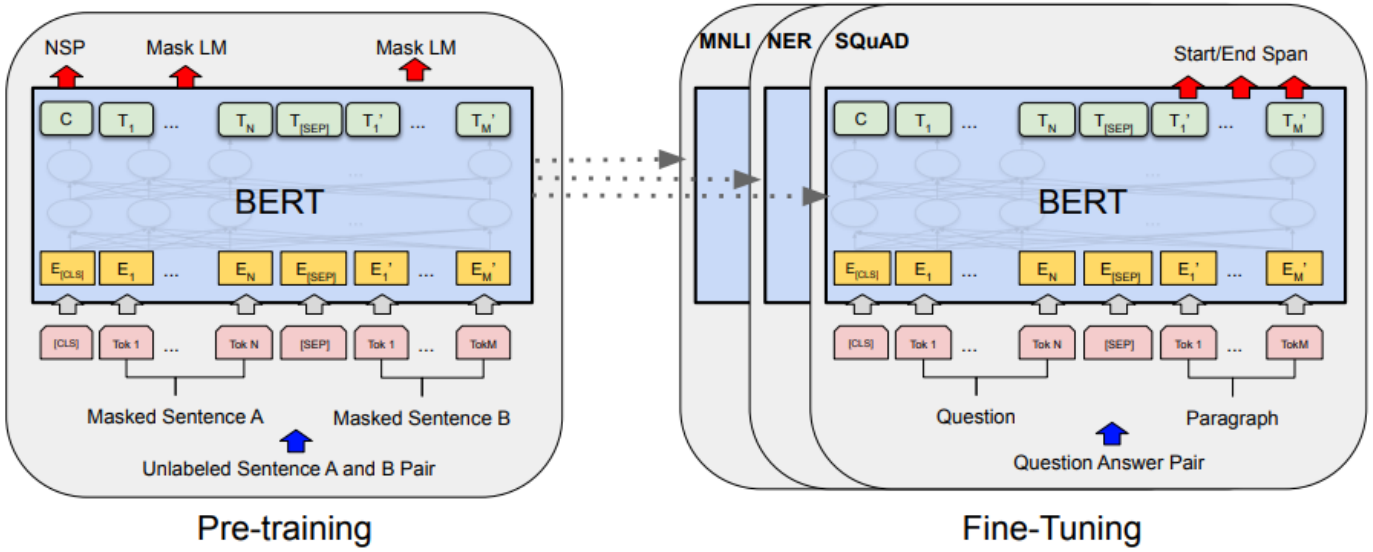
Background

- **Block Pruning**
  - neural network models with blocks
  - larger structure than a layer
  - block-level pruning gives immediate huge boost on time & space efficiency



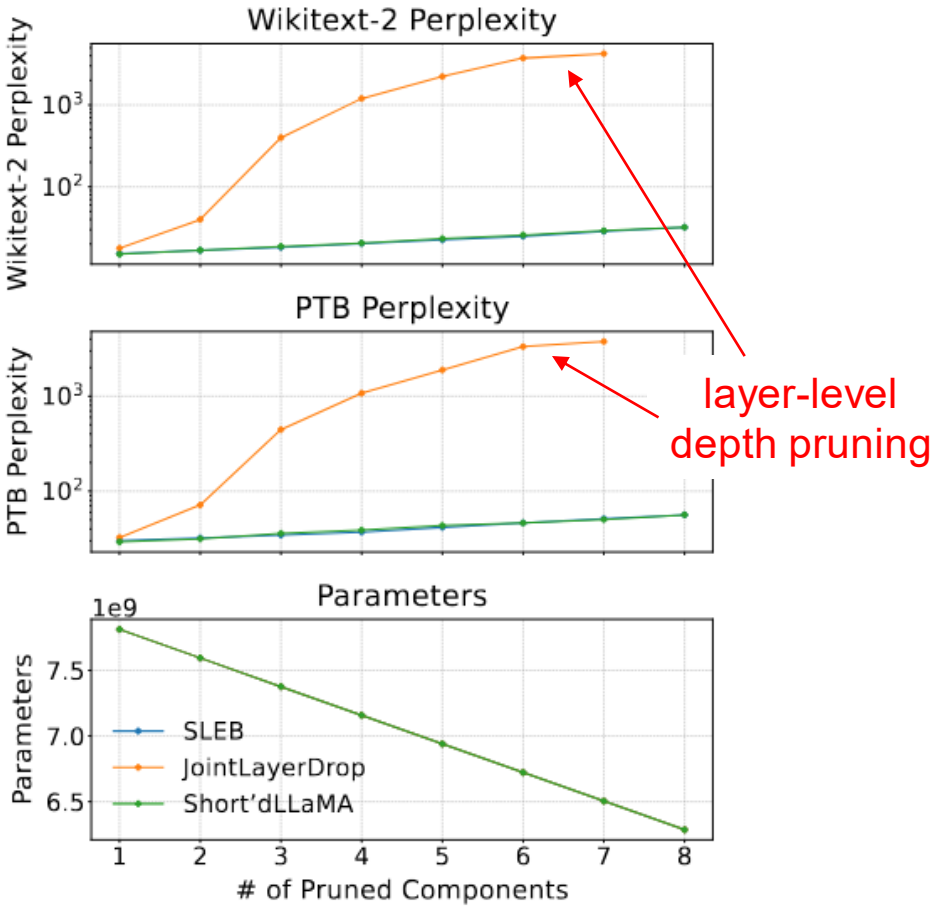
Background

- **Downstream Tasks**
  - pretrained LLM holds general knowledge
    - trained with next token prediction task, performance measured by perplexity
  - for specific task, fine-tune on downstream task
    - e.g., question-answering, performance measured by task accuracy



Introduction

- Unstructured Pruning: requires specialized hardware
- Structured Pruning: structural imbalance and instability
  - layer-wise width pruning
  - layer-level depth pruning
- Therefore, we adopt **Block Pruning**

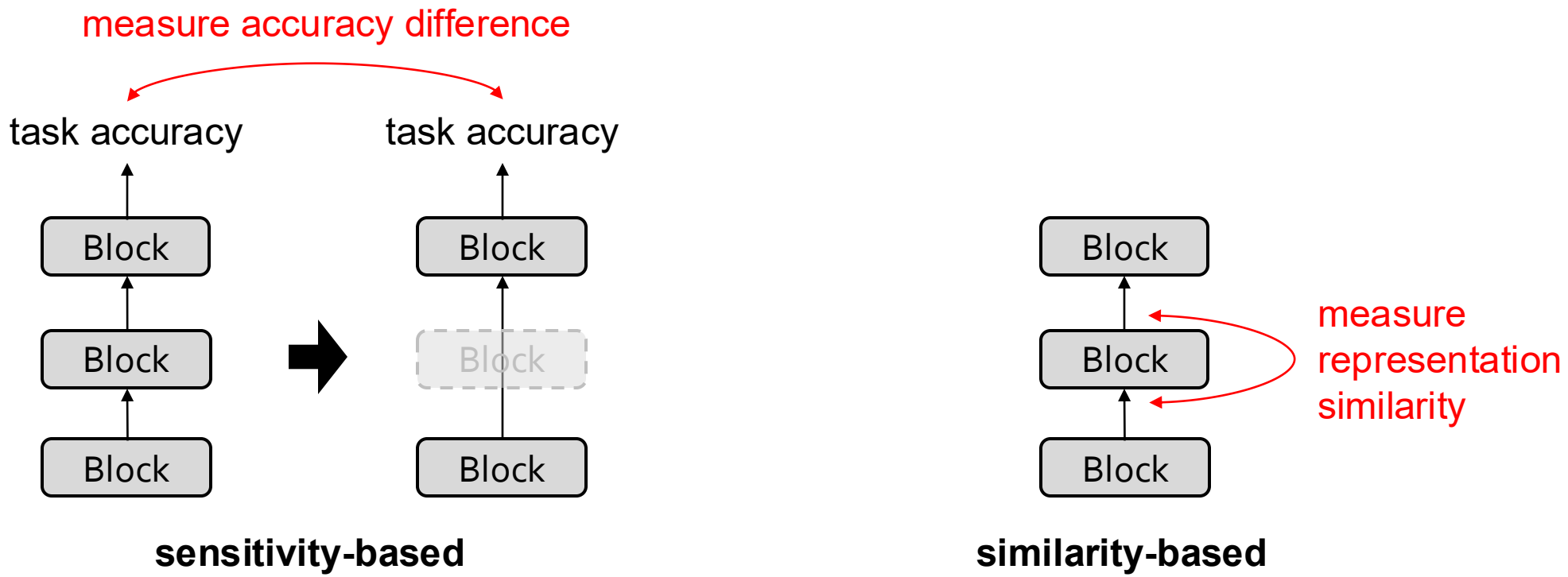


Motivation

- Limitations of existing **Block Pruning** methods
  - rely on representation similarity, or
  - rely on costly sensitivity analysis, using perplexity
- Failing to capture task-relevant decision signals
  - e.g., pruning criteria of perplexity do not necessarily correlate with downstream tasks
  - **[suggest]** block pruning for downstream tasks via output distribution signals
    - downstream task-aware pruning, rather than surface perplexity
    - no sensitivity analysis required

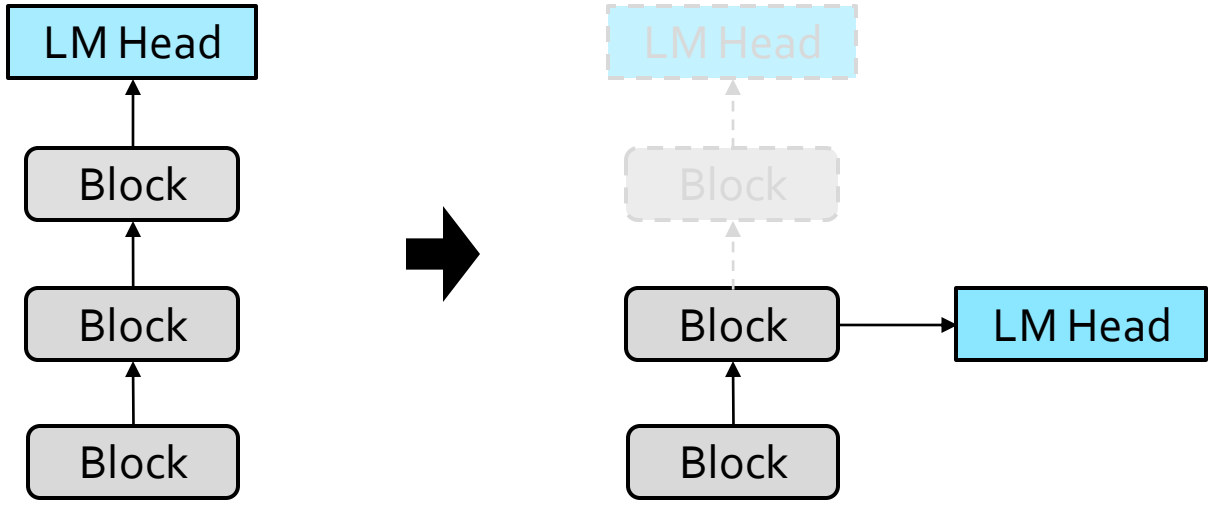
Related Work

- **Block Pruning**
  - sensitivity-based: execute sensitivity analysis on block-level, high cost
  - similarity-based: measure similarity between before/after blocks, task-ignorant



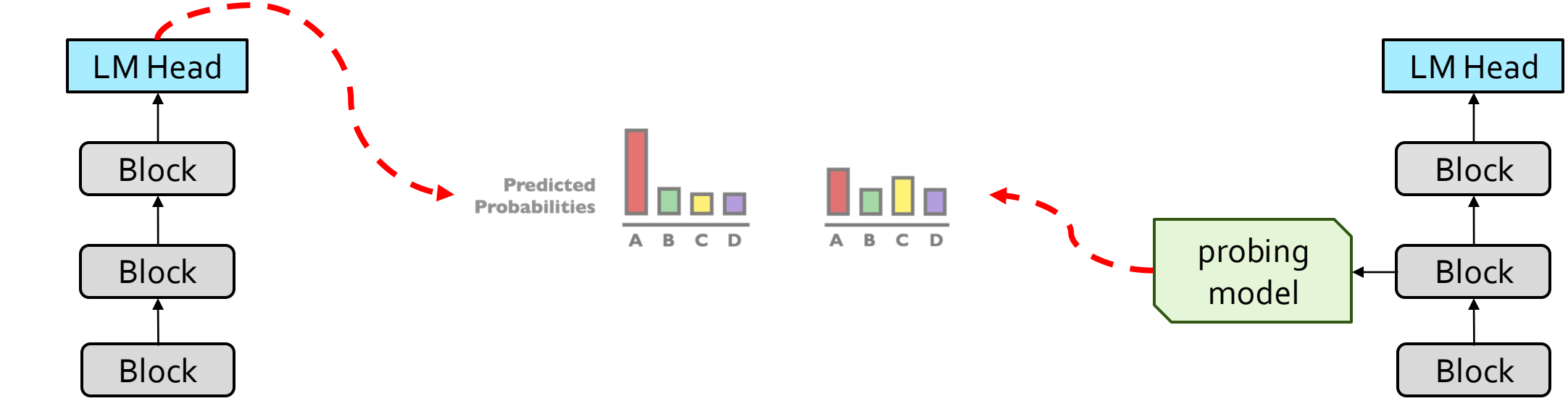
Related Work

- **Early Exit** (EE)
  - in language models (LM), final layer (**LM Head**) decodes internal latent vector into a token
  - in inference, we can early exit from an intermediate block, jumping right into the **LM Head** if confidence is high enough



Related Work

- **Uncertainty Quantification** (UQ)
  - output distribution: measure confidence score, entropy, etc.
  - internal representation: analyze via probing

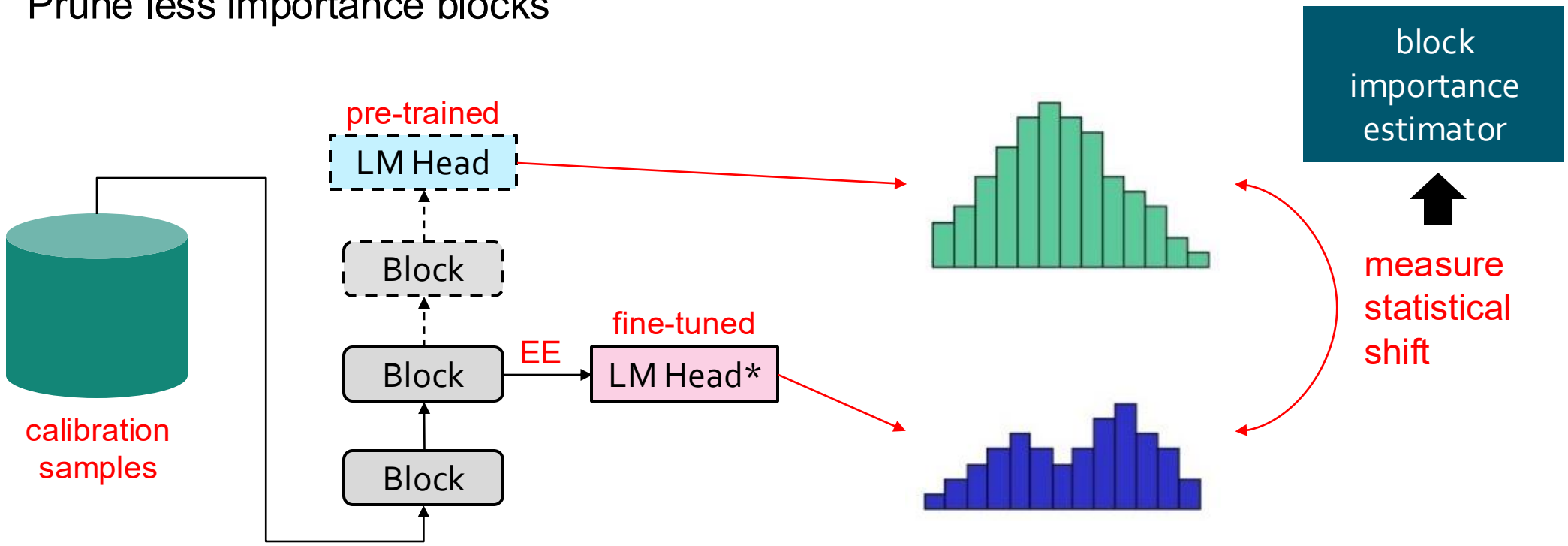


output distribution

internal representation

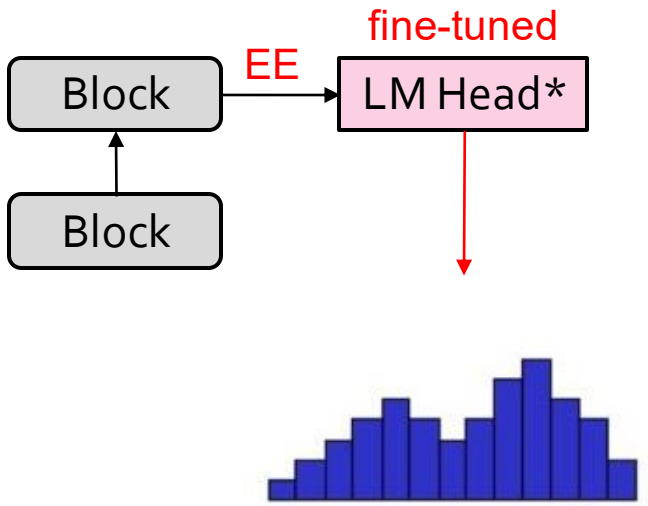
Method

1. Attach *block-wise LM Head* to intermediate block
2. Measure distribution's *statistical shift* on calibration samples
3. Aggregate statistical shift to estimate *block importance*
4. Prune less importance blocks



Method

- For $n = 1, 2, \ldots, N$ calibration samples, on block i
  - attach *block-wise LM Head*, compute output distribution
    - distribution over vocabulary V
    - might be fine-tuned (via head trainer dataset) or not
  - measure *statistical value* $s_i^{(n)}$
  - compute the *statistical shift* $\Delta s_i^{(n)}$



$$\Delta s_i^{(n)} = s_i^{(n)} - s_{i-1}^{(n)}$$

Method

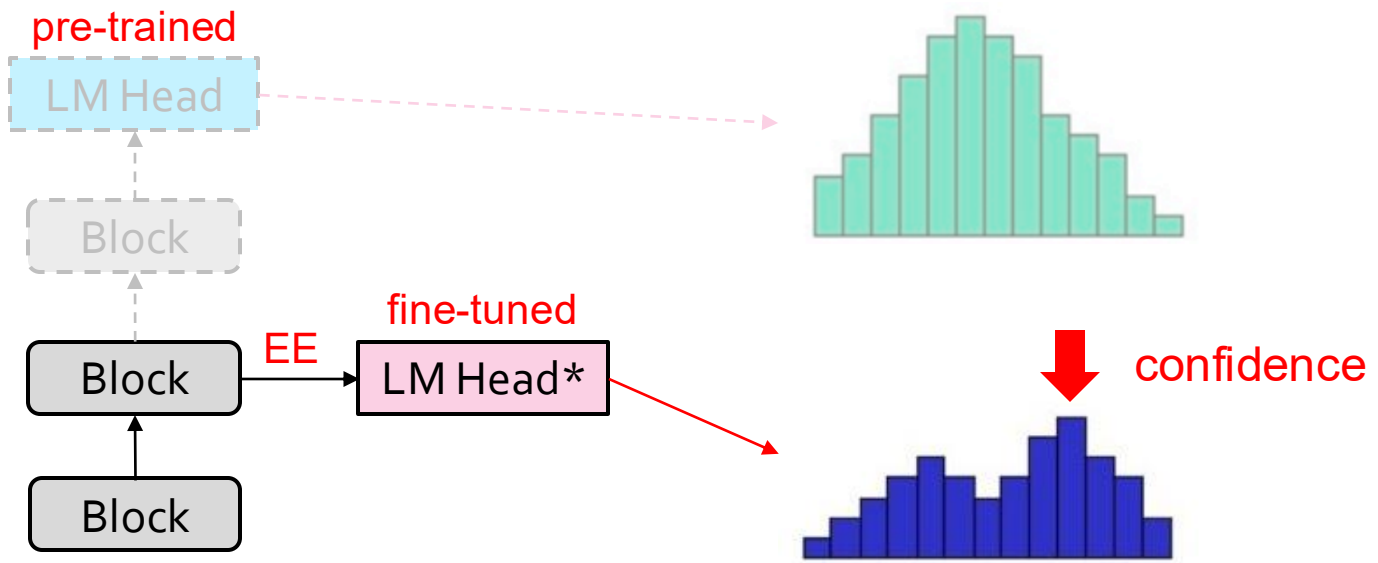
- Statistical value $s_i^{(n)}$ of distribution over V
  - various candidates of *statistical criteria*

Criteria (desirability)	Formula	#probs.	#dists.
Confidence score (maximum probability) ↑	$\mathrm{CONF}(x) = \max_{y \in V} p(y \mid x)$	1	1
Key probability (assigned to ground-truth y^*) ↑	$K(x) = p(y^* \mid x)$	1	1
Gap between top-1 and top-2 probabilities ↑	$G(x) = p(y_{(1)} \mid x) - p(y_{(2)} \mid x)$	2	1
Entropy over the token distribution ↓	$\mathrm{ENT}(x) = -\sum_{y \in V} p(y \mid x) \log p(y \mid x)$	All	1
Cross-entropy between output distributions ↓	$\mathrm{CE}(p, q) = -\sum_{y \in V} p(y \mid x) \log q(y \mid x)$	All	2
Kullback-Leibler divergence ↓	$\mathrm{KLD}(p \parallel q) = \sum_{y \in V} p(y \mid x) \log \frac{p(y \mid x)}{q(y \mid x)}$	All	2
Jensen-Shannon divergence ↓	$\mathrm{JSD}(p \parallel q) = \frac{1}{2} \mathrm{KLD}(p \parallel m) + \frac{1}{2} \mathrm{KLD}(q \parallel m)$	All	2

Method

- Statistical value $s_i^{(n)}$ of distribution over V

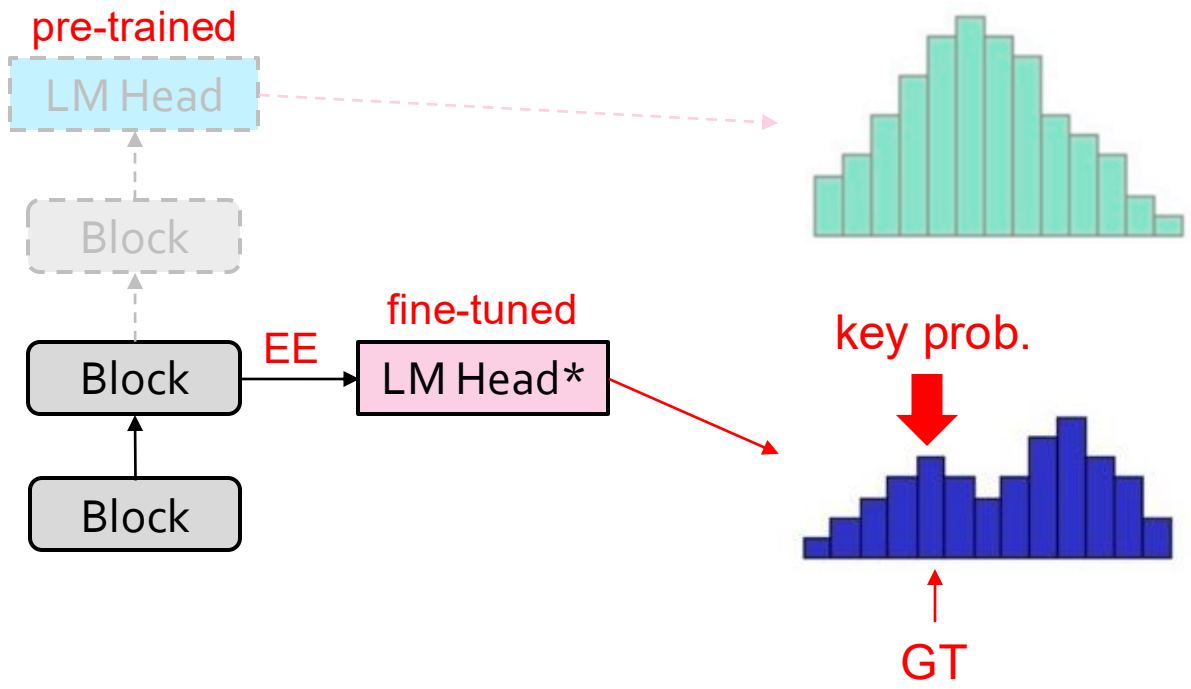
Confidence score (maximum probability) ↑ $\quad \mathrm{CONF}(x) = \max_{y \in V} p(y \mid x)$



Method

- Statistical value $s_i^{(n)}$ of distribution over V

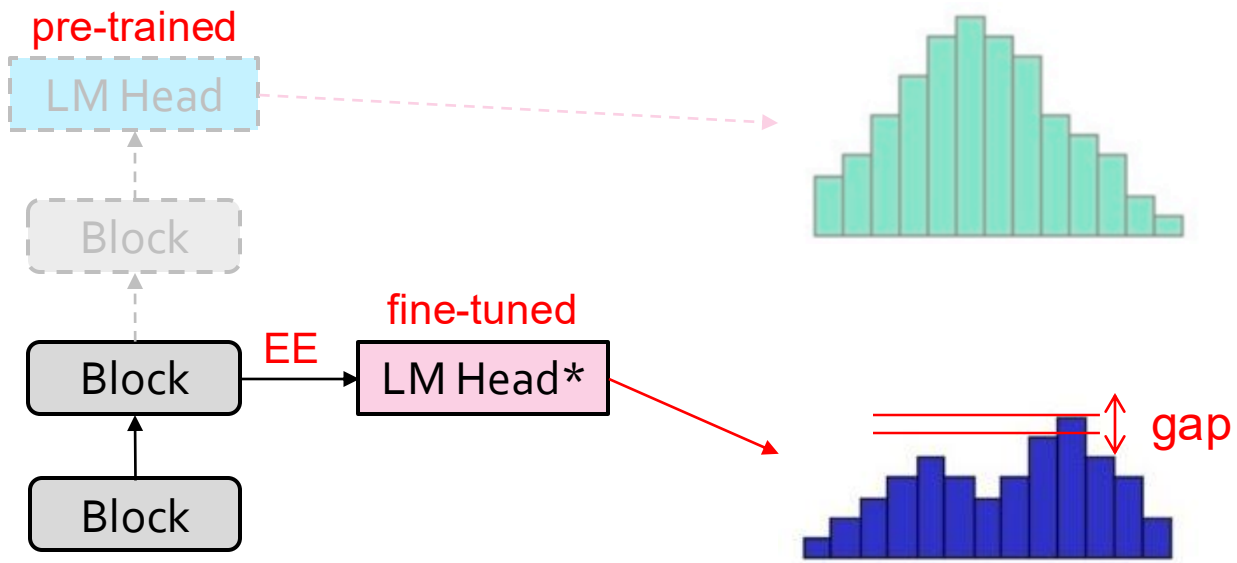
Key probability (assigned to ground-truth y^*) ↑ $K(x) = p(y^* \mid x)$



Method

- Statistical value $s_i^{(n)}$ of distribution over V

Gap between top-1 and top-2 probabilities ↑ $\quad G(x) = p(y_{(1)} \mid x) - p(y_{(2)} \mid x)$

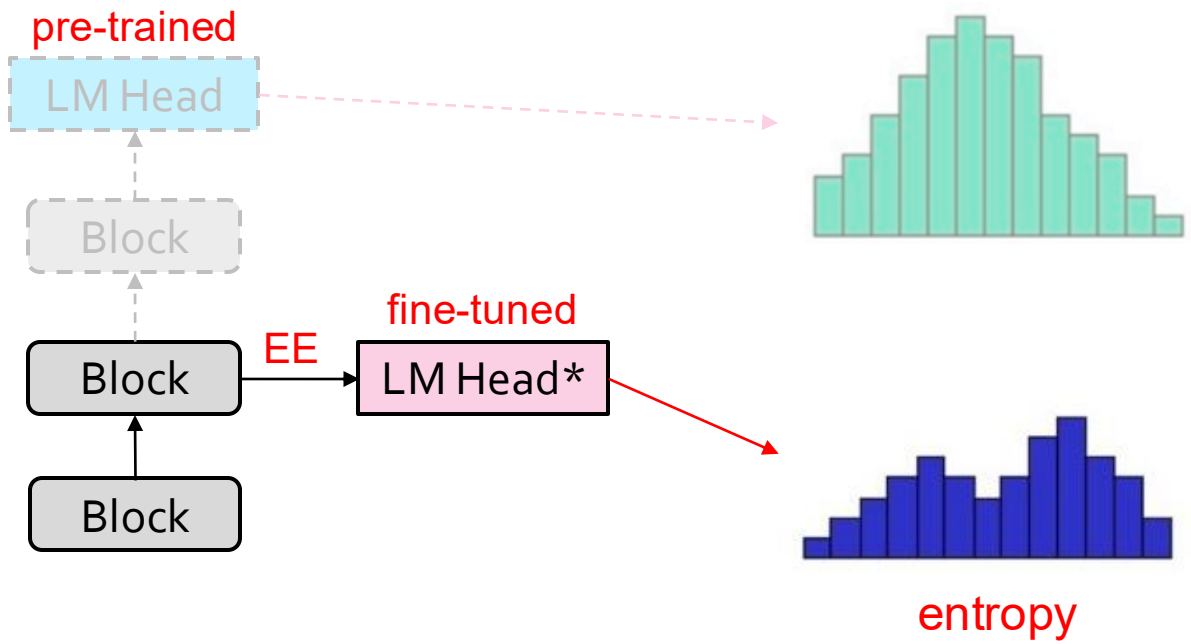


Method

- Statistical value $s_i^{(n)}$ of distribution over V

Entropy over the token distribution ↓

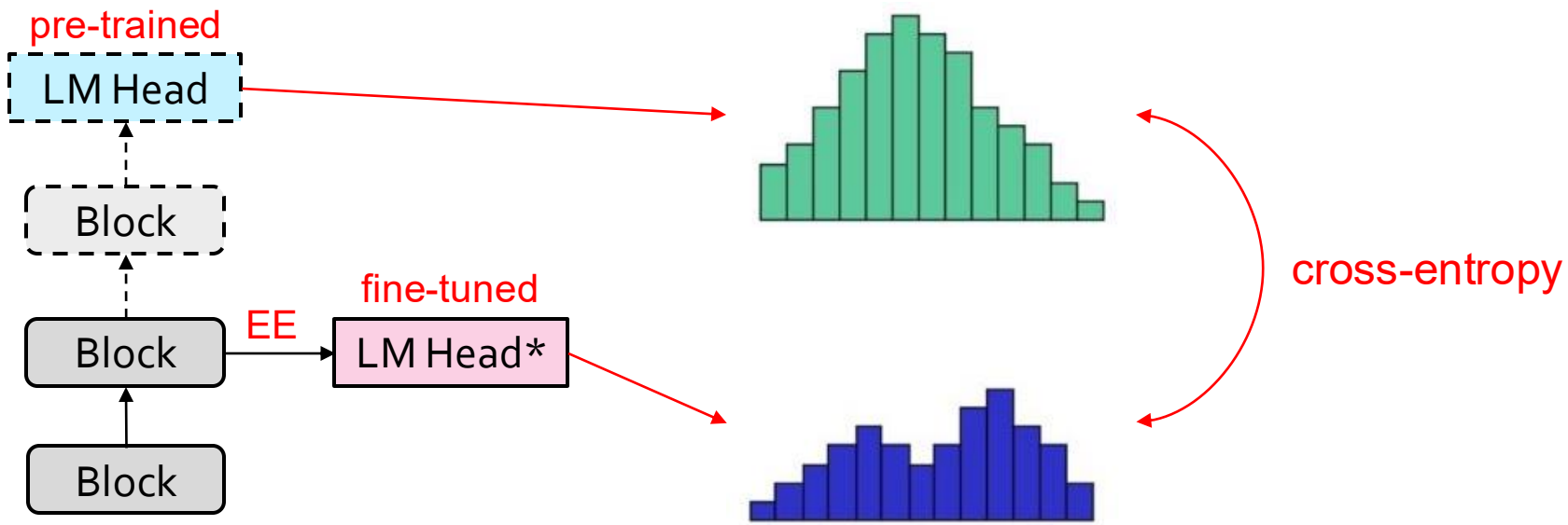
$$\mathrm{ENT}(x) = -\sum_{y \in V} p(y \mid x) \log p(y \mid x)$$



Method

- Statistical value $s_i^{(n)}$ of distribution over V

Cross-entropy between output distributions ↓ $\quad \mathrm{CE}(p, q) = -\sum_{y \in V} p(y \mid x) \log q(y \mid x)$

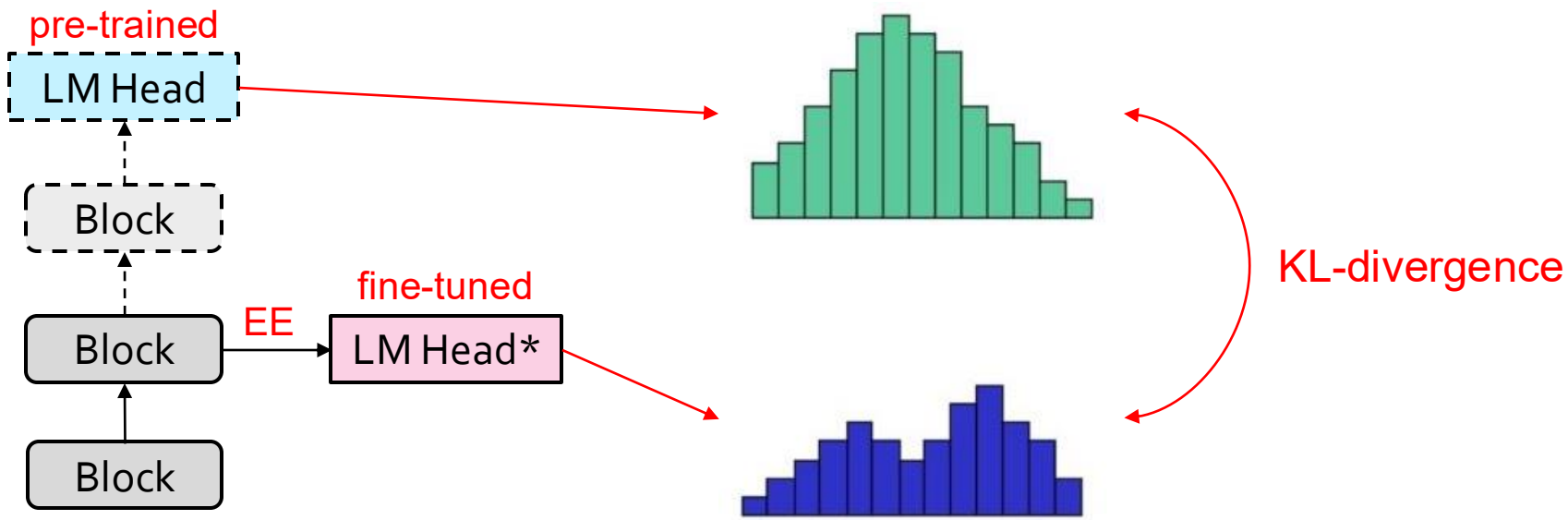


Method

- Statistical value $s_i^{(n)}$ of distribution over V

Kullback-Leibler divergence ↓

$$\mathrm{KLD}(p \parallel q) = \sum_{y \in V} p(y \mid x) \log \frac{p(y \mid x)}{q(y \mid x)}$$

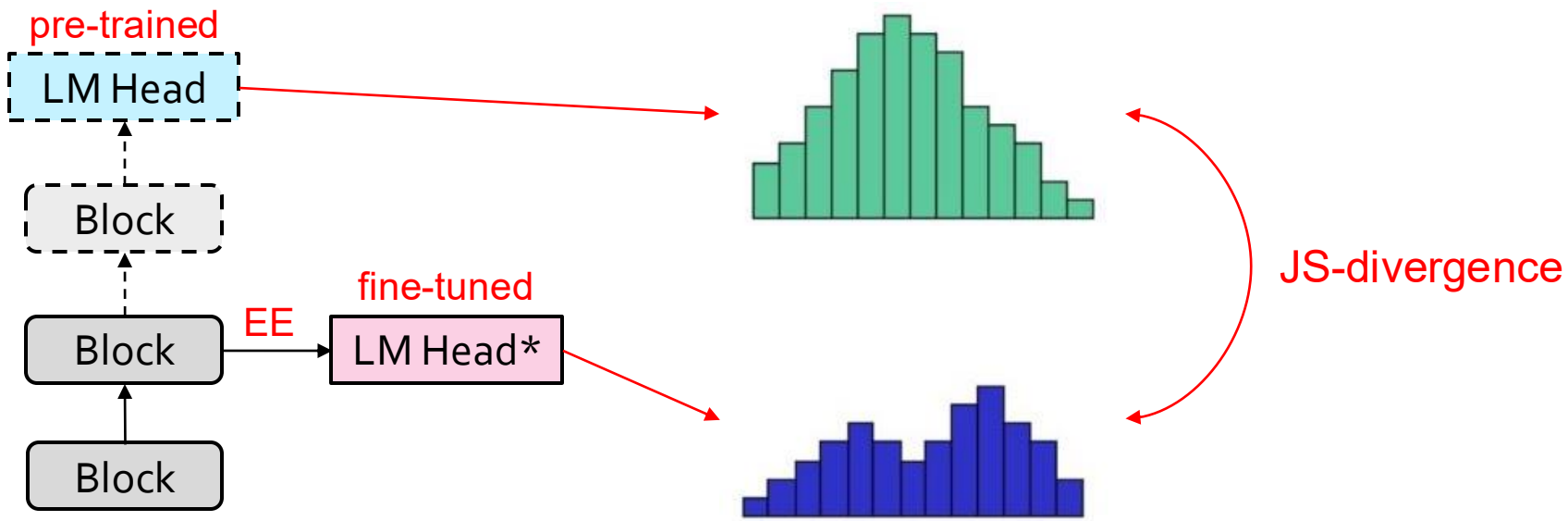


Method

- Statistical value $s_i^{(n)}$ of distribution over V

Jensen-Shannon divergence ↓

$$\mathrm{JSD}(p \parallel q) = \frac{1}{2}\mathrm{KLD}(p \parallel m) + \frac{1}{2}\mathrm{KLD}(q \parallel m)$$



Method

- **Block Importance Estimation**
  - among calibration samples, *aggregate* statistical shifts
  - method 1: Directional Desirable Frequency (DDF)

$$\Delta s_i^{(n)} = s_i^{(n)} - s_{i-1}^{(n)}$$

$$\mathrm{DDF}(b_i) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\left[\alpha \cdot \Delta s_i^{(n)} > 0\right]$$

Criteria (desirability)	Formula
Confidence score (maximum probability) ↑	$\mathrm{CONF}(x) = \max_{y \in V} p(y \mid x)$
Key probability (assigned to ground-truth y^*) ↑	$K(x) = p(y^* \mid x)$
Gap between top-1 and top-2 probabilities ↑	$G(x) = p(y_{(1)} \mid x) - p(y_{(2)} \mid x)$
Entropy over the token distribution ↓	$\mathrm{ENT}(x) = -\sum_{y \in V} p(y \mid x) \log p(y \mid x)$
Cross-entropy between output distributions ↓	$\mathrm{CE}(p, q) = -\sum_{y \in V} p(y \mid x) \log q(y \mid x)$
Kullback-Leibler divergence ↓	$\mathrm{KLD}(p \parallel q) = \sum_{y \in V} p(y \mid x) \log \frac{p(y \mid x)}{q(y \mid x)}$
Jensen-Shannon divergence ↓	$\mathrm{JSD}(p \parallel q) = \frac{1}{2}\mathrm{KLD}(p \parallel m) + \frac{1}{2}\mathrm{KLD}(q \parallel m)$

$$\alpha = \begin{cases} +1 & (\text{desirability}_{\uparrow}) \\ -1 & (\text{desirability}_{\downarrow}) \end{cases}$$

Method

- **Block Importance Estimation**
  - among calibration samples, *aggregate* statistical shifts
  - method 2: Statistical Shift Norm (SSN)

$$\Delta s_i^{(n)} = s_i^{(n)} - s_{i-1}^{(n)}$$

$$\mathrm{SSN}(b_i) = \frac{1}{N}\left(\sum_{n=1}^{N}\left(\Delta s_i^{(n)}\right)^p\right)^{\frac{1}{p}}$$

Method

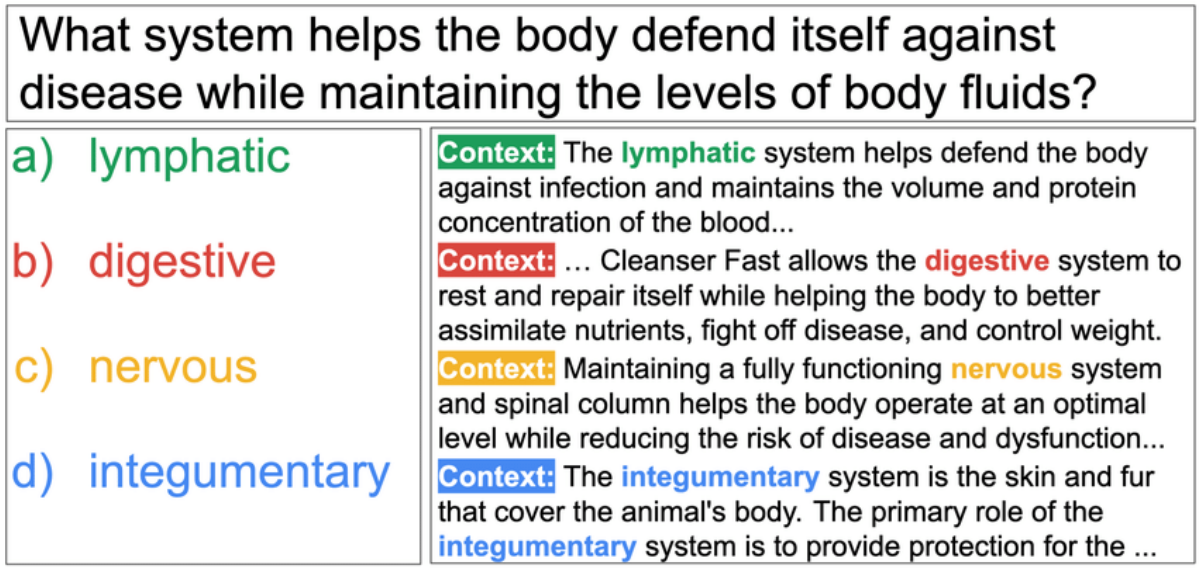
- Again,
  - attach *block-wise LM Head* to intermediate block
    - fine-tune on head trainer dataset (or not)
  - measure distribution's *statistical shift* on calibration samples
    - compute $\Delta s_i^{(n)}$ for each sample/block
  - then, aggregate to estimate *block importance*
    - with DDF (influence) or SSN (magnitude)
  - prune less important blocks

Method

- Calibration Samples
  - **General Text Generation** (GTG)
    - high entropy does not necessarily indicate uncertainty
  - **Multiple Choice Question Answering** (MCQA)
    - restricted vocabulary distribution (say, 4)
    - suitable for block-level influence computation
    - so, we use MCQA as calibration samples

Dataset

- **Head Trainer Dataset**
  - mixed, general
- **Calibration Samples**
  - 1024 samples from ARC Easy training
- **Evaluation**
  - ARC Easy, ARC Challenge
  - BoolQ
  - COPA
  - HellaSwag
  - PIQA



ARC Easy Dataset

Baselines

- **Block Pruning**
  - ShortenedLLaMa (sensitivity-based)
  - SLEB (sensitivity-based)
  - EntroDrop (similarity-based)
- **Layer Pruning**
  - JointLayerDrop (similarity-based)

Backbone

- LLMs with 32 transformer blocks
  - Llama-3-8B
  - Mistral-7B
- Up to 8 blocks to prune (25%)

Experiments

- Our model: **T**ask-**a**ware **B**lock **P**runing (**TaBP**)
- Research questions
  - RQ1: accuracy preservation
  - RQ2: ablation study on calibration sample, MCQA vs GTG
  - RQ3: ablation study on statistical criteria and aggregation method

Experimental Results

- Llama-3-8B

Method	ARC-E ↑	ARC-C ↑	BoolQ ↑	COPA ↑	HeSw. ↑	PIQA ↑	Wino. ↑	Avg. ↑
Original Model	80.09	53.33	81.44	89.00	79.17	79.71	72.85	76.51
Prune 2 blocks (pruning ratio = 6.25%)								
SLEB	**76.60**	48.29	74.80	87.00	74.33	77.80	70.48	72.76
ShortendLLaMA	76.05	49.23	75.66	**90.00**	74.97	**77.86**	70.88	73.52
EntroDrop	67.55	44.45	63.09	84.00	68.58	75.57	71.43	67.81
JointLayerDrop	62.58	42.75	62.45	78.00	57.89	70.89	62.04	62.37
TaBP (ours)	76.01	**50.17**	**80.76**	87.00	**76.24**	78.07	**73.88**	**74.59**
Prune 4 blocks (pruning ratio = 12.5%)								
SLEB	**72.47**	43.26	66.12	**87.00**	70.69	**76.12**	69.46	69.30
ShortendLLaMA	70.16	38.65	57.25	**87.00**	66.83	75.63	61.72	65.32
EntroDrop	40.78	33.53	58.59	67.00	37.43	61.86	58.48	51.10
JointLayerDrop	56.73	41.13	62.39	74.00	51.71	68.88	61.96	59.54
TaBP (ours)	71.46	**47.44**	**73.21**	81.00	**72.05**	75.95	**72.53**	**70.52**
Prune 8 blocks (pruning ratio = 25%)								
SLEB	61.78	31.74	42.11	**77.00**	57.98	71.82	53.59	56.57
ShortendLLaMA	**61.83**	31.66	42.05	**77.00**	57.94	**71.87**	53.83	56.60
EntroDrop	33.33	29.78	62.11	66.00	26.97	57.67	56.59	47.49
JointLayerDrop	50.51	37.97	62.54	68.00	46.72	65.23	61.40	56.05
TaBP (ours)	53.83	**40.61**	**64.34**	72.00	**59.35**	68.61	**67.79**	**60.93**

92% accuracy
with
14% faster
13% smaller

Experimental Results

- Mistral-7B

Method	ARC-E ↑	ARC-C ↑	BoolQ ↑	COPA ↑	HeSw. ↑	PIQA ↑	Wino. ↑	Avg. ↑
Original Model	80.09	53.50	81.31	89.00	79.16	79.71	72.61	76.48
Prune 2 blocks (pruning ratio = 6.25%)								
SLEB	76.22	46.67	77.71	**88.00**	77.30	**80.03**	67.96	73.41
ShortendLLaMA	74.87	44.62	74.53	**88.00**	73.88	42.20	**78.62**	72.02
EntroDrop	67.55	44.45	63.09	84.00	68.58	75.57	71.43	67.81
JointLayerDrop	56.73	41.97	62.23	76.00	56.01	67.90	66.22	61.01
TaBP (ours)	**77.06**	**50.26**	**80.64**	85.00	**77.75**	78.45	72.14	**74.47**
Prune 4 blocks (pruning ratio = 12.5%)								
SLEB	**71.68**	41.98	74.22	87.00	72.83	**77.58**	64.96	70.04
ShortendLLaMA	66.84	34.30	59.51	**88.00**	60.68	75.41	56.43	63.02
EntroDrop	29.97	29.35	32.94	71.00	33.49	56.20	61.09	44.86
JointLayerDrop	49.83	36.09	62.51	66.00	46.18	64.47	62.75	55.40
TaBP (ours)	70.92	**44.88**	**76.30**	87.00	**73.67**	75.14	**69.14**	**71.01**
Prune 8 blocks (pruning ratio = 25%)								
SLEB	**63.59**	35.75	59.54	**79.00**	**62.95**	**72.47**	60.85	62.02
ShortendLLaMA	51.73	26.28	54.89	70.00	48.28	69.26	52.17	53.23
EntroDrop	26.77	29.18	42.48	72.00	28.25	56.15	55.80	44.38
JointLayerDrop	49.20	36.77	62.45	66.00	45.37	63.93	62.51	55.18
TaBP (ours)	56.73	**40.36**	**76.85**	**79.00**	61.62	68.44	**66.69**	**64.24**

93% accuracy
with
14% faster
13% smaller

Experimental Results

- Ablation study on calibration samples
  - MCQA (with DDF) shows significant performance gain

Method		Avg. Acc.@8 ↑	AUC@8 ↑
DDF (ours)	MCQA	**62.87**	**.7022**
	GTG	58.02	.6906
SSN (ours)	MCQA	58.92	.6711
	GTG	59.59	.6902
SLEB		56.57	.6703
Short'dLLaMA		56.60	.6633
EntroDrop		45.68	.4821
JointLayerDrop		56.05	.5285

Experimental Results

- Ablation study on aggregation & criteria
  - DDF is generally better
    - influence is more critical than magnitude
  - Entropy is the best
    - in MCQA calibration, entropy can be served as uncertainty

Method	Avg.@2 ↑	Avg.@4 ↑	Avg.@8 ↑
SLEB	72.76	69.30	56.57
ShortenedLLaMA	73.52	65.32	56.60
EntroDrop	64.68	49.71	45.68
JointLayerDrop	62.37	59.54	56.05

Dataset	LM Head	Aggregation	Criteria	Avg.@2 ↑	Avg.@4 ↑	Avg.@8 ↑
ARC-E	Frozen	DDF	CONF	70.12	68.40	**63.94**
			K	70.12	68.40	**63.94**
			G	70.12	67.72	60.98
			ENT	**75.04**	69.39	62.87
			CE	70.38	69.11	61.95
			KLD	70.12	68.48	**63.94**
			JSD	72.13	69.72	59.74
		SSN	CONF	66.13	64.90	55.36
			K	66.13	64.90	55.36
			G	67.09	65.97	56.24
			ENT	66.13	64.90	55.95
			CE	66.13	64.90	55.36
			KLD	66.13	64.90	57.88
			JSD	70.64	67.99	59.01
	Trained	DDF	CONF	70.14	68.40	60.69
			K	70.14	68.40	60.69
			G	70.12	68.08	59.74
			ENT	74.59	**70.52**	60.93
			CE	70.12	68.33	62.57
			KLD	70.13	68.39	60.21
			JSD	70.11	67.72	59.74
		SSN	CONF	69.43	68.17	58.44
			K	69.43	68.17	58.44
			G	70.38	67.42	59.51
			ENT	69.43	68.17	59.54
			CE	69.43	68.17	58.44
			KLD	69.43	68.17	60.23
			JSD	69.43	67.42	59.51

Criteria (desirability)	**Formula**
Confidence score (maximum probability) ↑	$\mathrm{CONF}(x) = \max_{y \in V} p(y \mid x)$
Key probability (assigned to ground-truth y^*) ↑	$K(x) = p(y^* \mid x)$
Gap between top-1 and top-2 probabilities ↑	$G(x) = p(y_{(1)} \mid x) - p(y_{(2)} \mid x)$
Entropy over the token distribution ↓	$\mathrm{ENT}(x) = -\sum_{y \in V} p(y \mid x) \log p(y \mid x)$
Cross-entropy between output distributions ↓	$\mathrm{CE}(p, q) = -\sum_{y \in V} p(y \mid x) \log q(y \mid x)$
Kullback-Leibler divergence ↓	$\mathrm{KLD}(p \parallel q) = \sum_{y \in V} p(y \mid x) \log \frac{p(y \mid x)}{q(y \mid x)}$
Jensen-Shannon divergence ↓	$\mathrm{JSD}(p \parallel q) = \frac{1}{2} \mathrm{KLD}(p \parallel m) + \frac{1}{2} \mathrm{KLD}(q \parallel m)$

Conclusion

- **Task-aware Block Pruning** via output distribution signals
  - no need of sensitivity analysis
  - downstream task-aware, rather than surface perplexity
  - retains 92~93% accuracy even after 12.5% pruned away

Q&A

Thank you for listening!