VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY

Prefix-Tuning: Optimizing Continuous Prompts for Generation

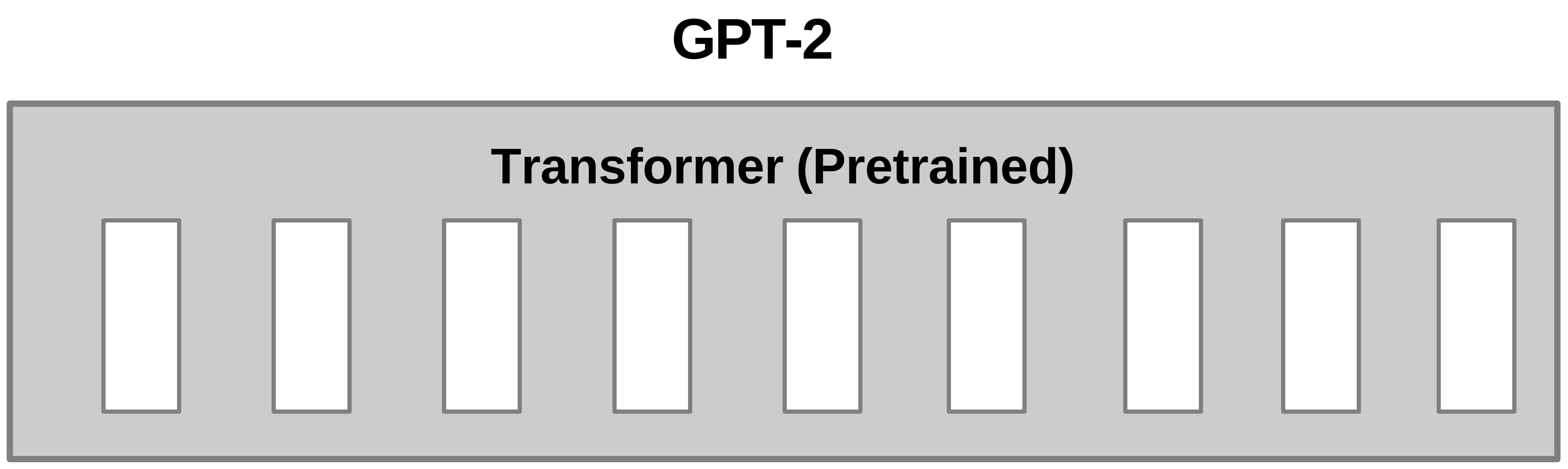
Lecturers: PhD. Luong Ngoc Hoang

Members: Tran Van Tinh
Vu Bao Quoc
Dinh Van Hoan
Than The Tung

OUTLINE

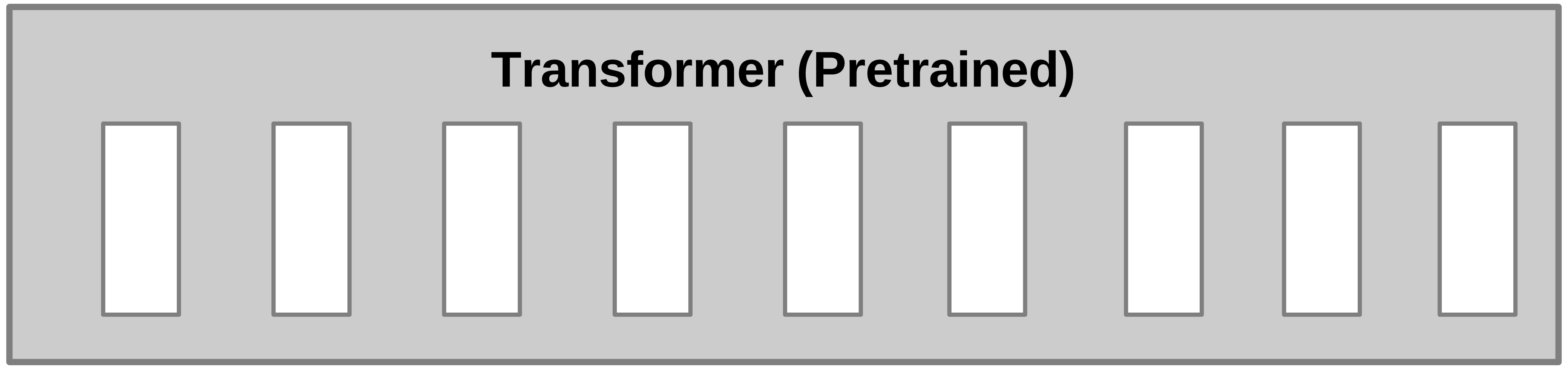
1. Introduction
2. Related Work
3. Prefix-tuning (Intuition + Method)
4. Results (Experiments + Ablation Studies)
5. Demo

Why optimizing prompts?



Why optimizing prompts?

GPT-2



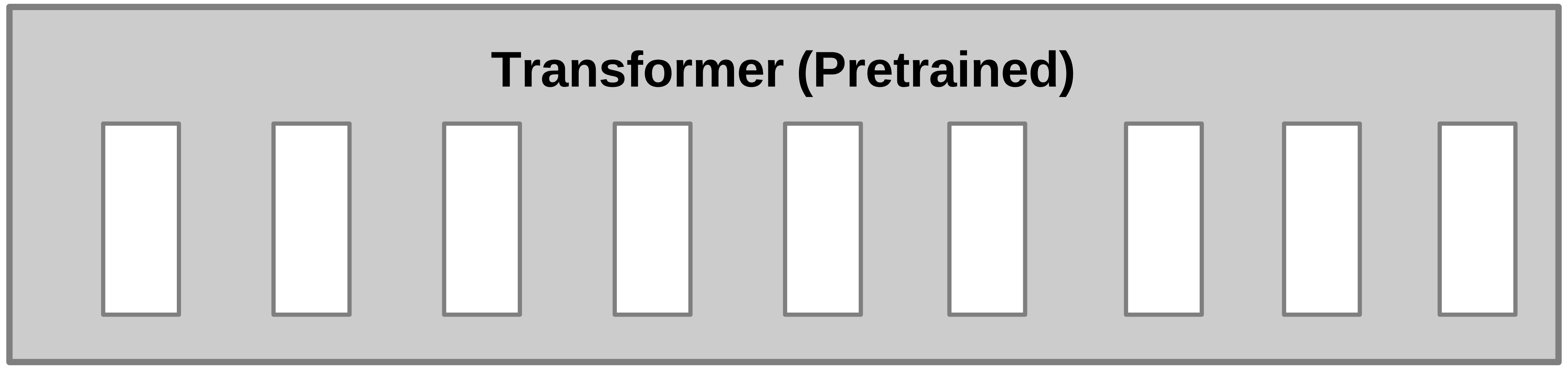
Fine-tune →

Tasks

Table-to-Text
Summarization
Translation
Dialog Generation
…

Why optimizing prompts?

GPT-2



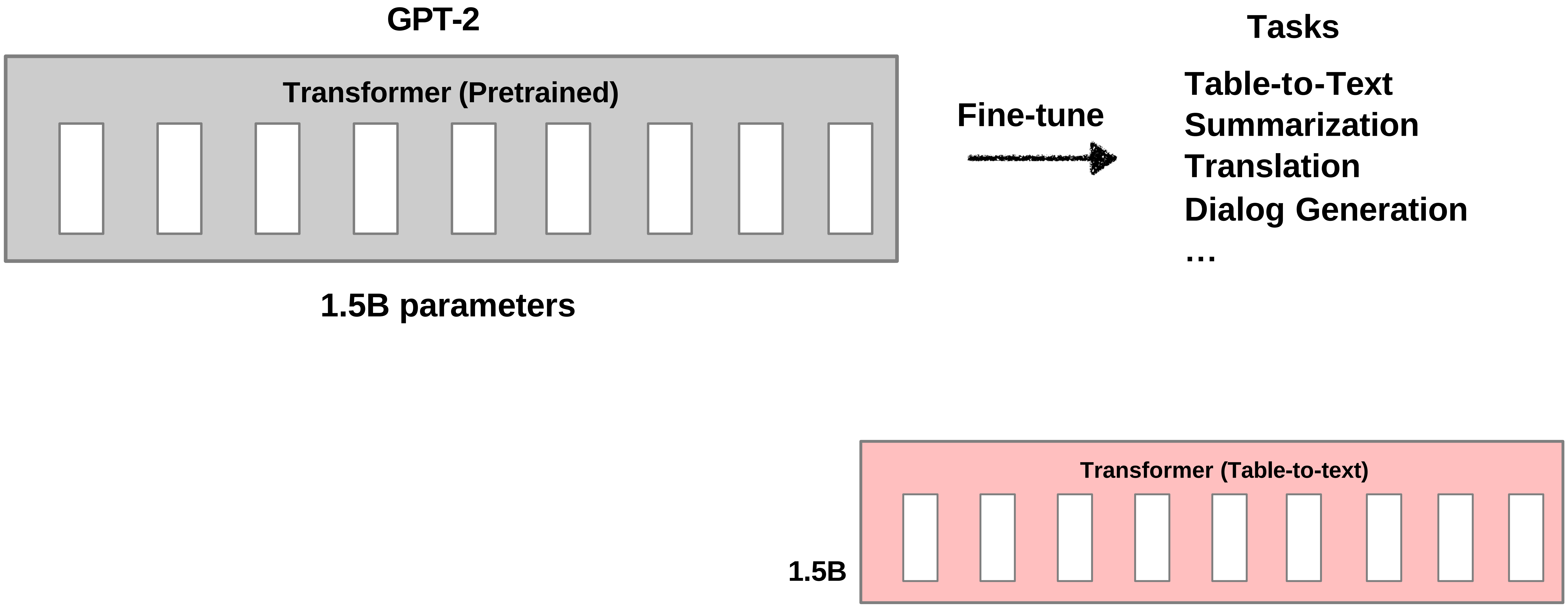
1.5B parameters

Fine-tune →

Tasks

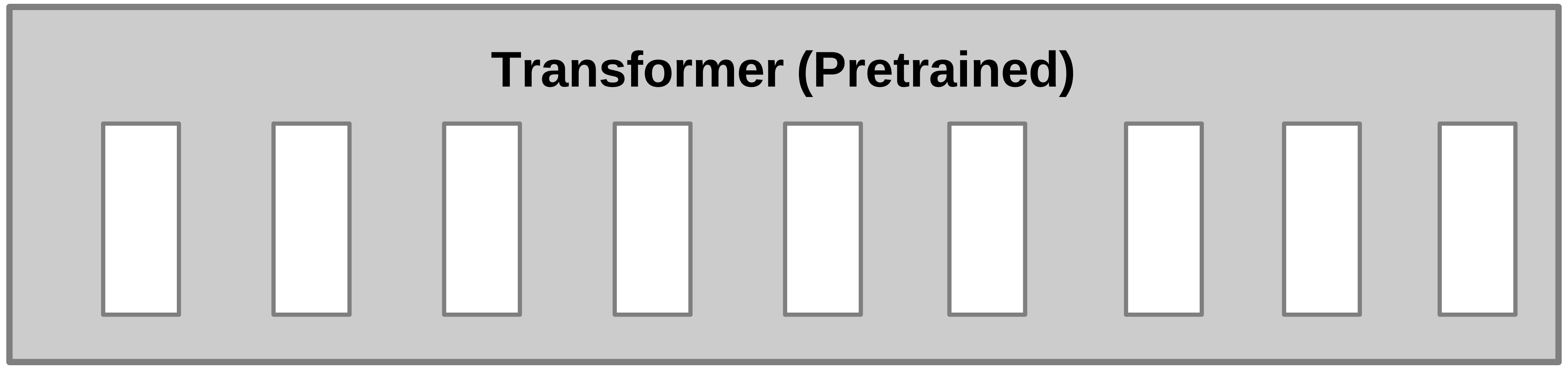
Table-to-Text
Summarization
Translation
Dialog Generation
…

Why optimizing prompts?



Why optimizing prompts?

GPT-2



1.5B parameters

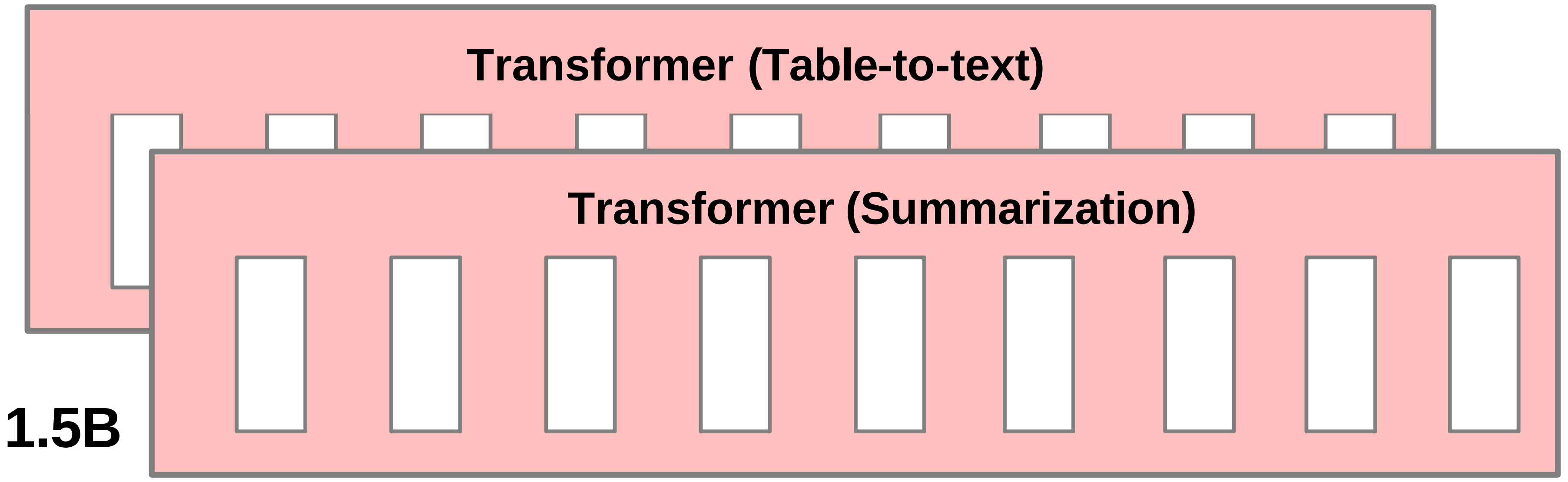
Fine-tune

Tasks

Table-to-Text
Summarization
Translation
Dialog Generation
…

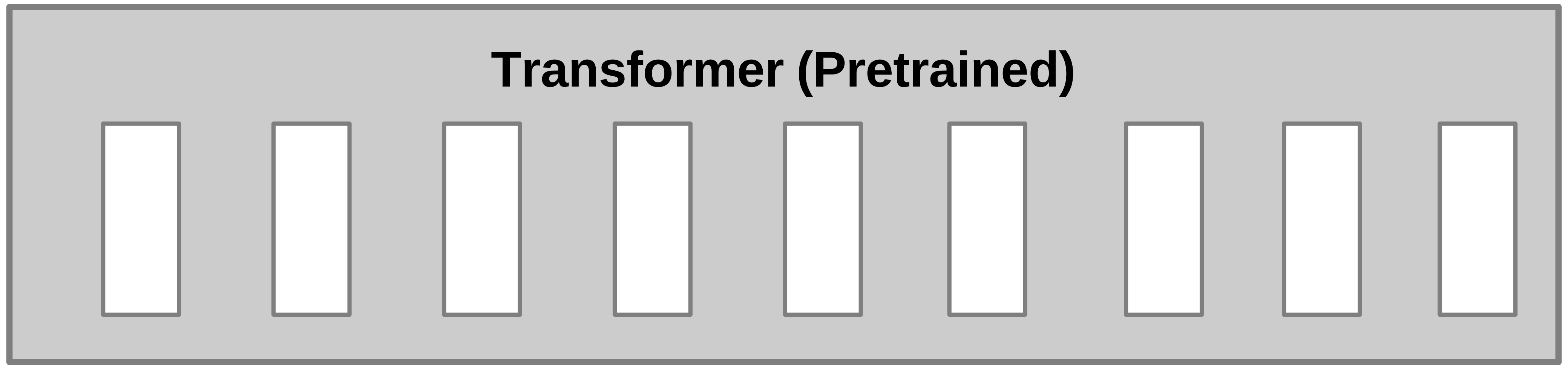
1.5B

1.5B



Why optimizing prompts?

GPT-2

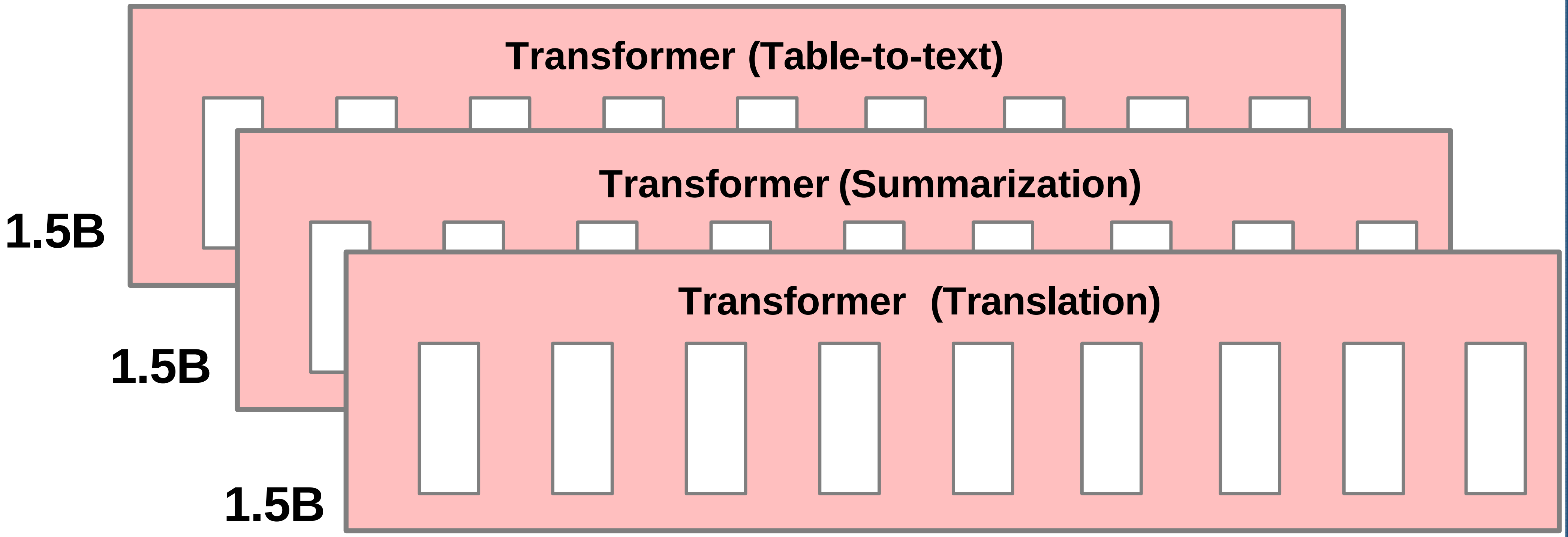


1.5B parameters

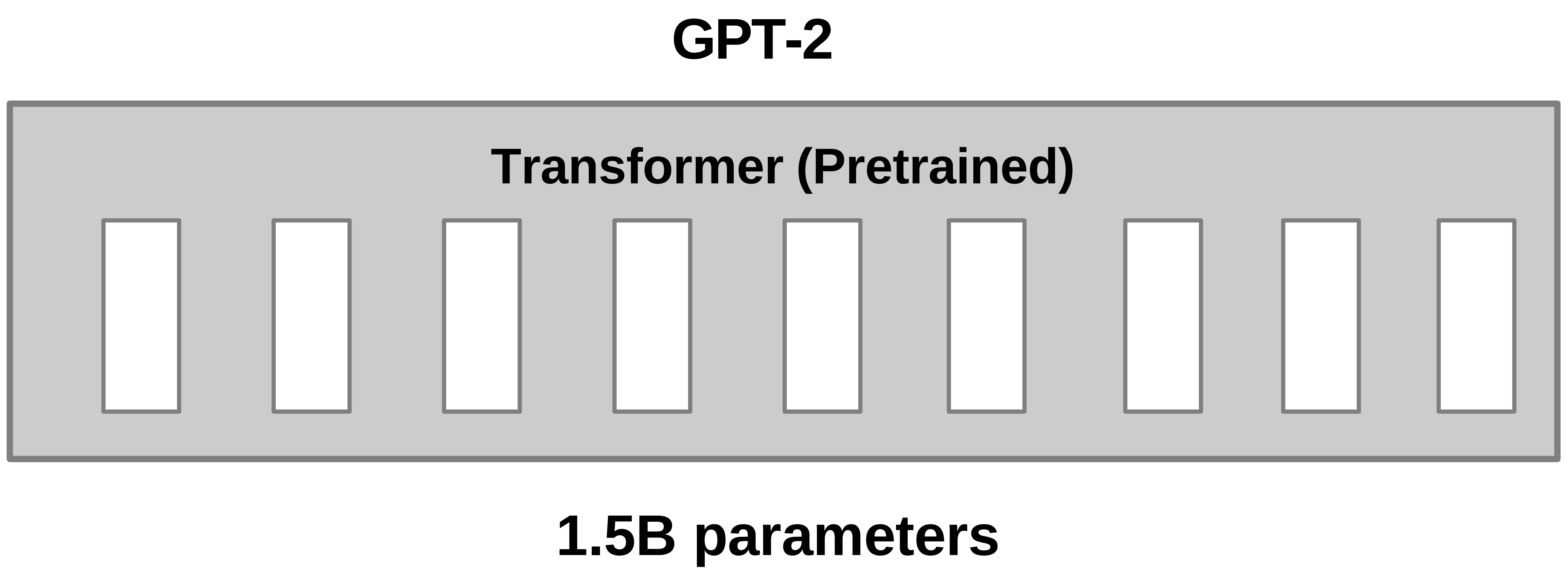
Fine-tune →

Tasks

Table-to-Text
Summarization
Translation
Dialog Generation
…



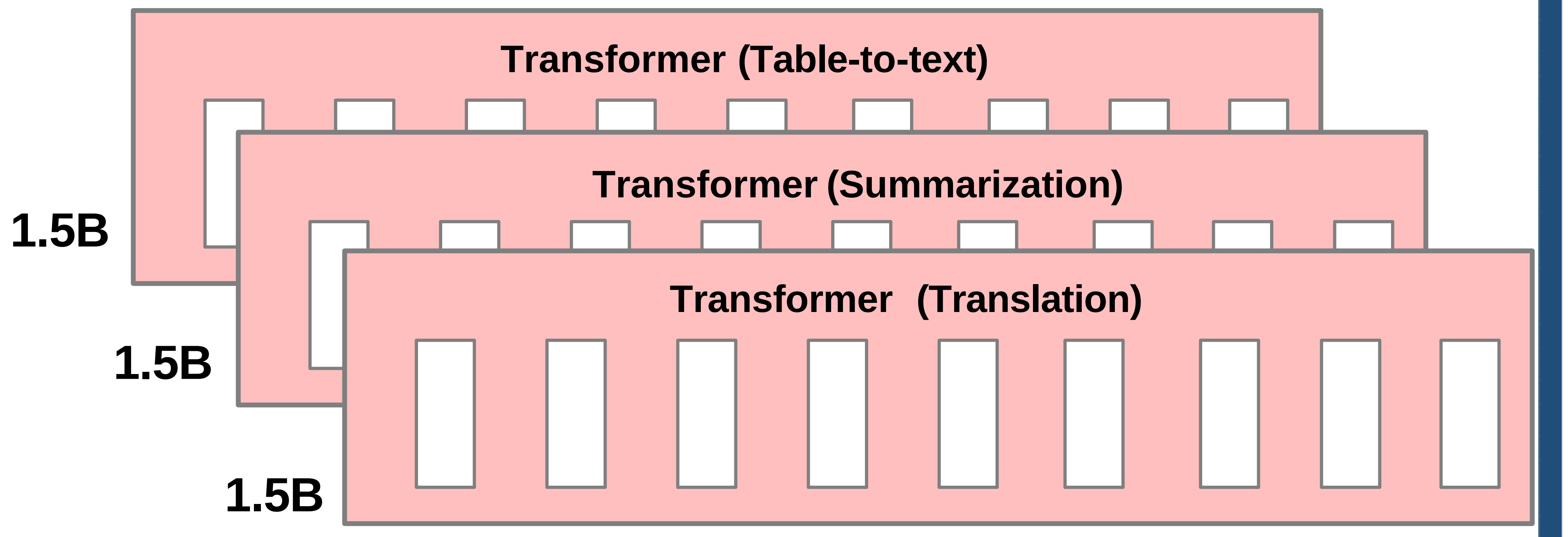
Why optimizing prompts?



Fine-tune

Tasks

Table-to-Text
Summarization
Translation
Dialog Generation
…



Expensive to store and update a full model copy for each task.

In-context Learning

Prompt

Instruction — **Summarize the following data table:**

Example — **TABLE: name: Alimentum | area: city centre | family friendly: no**
A: There is a place in the city centre, Alimentum, that is not family-friendly.

Input — **TABLE: name: Starbucks | area: riverside | customer rating: 5 star**

GPT-3

Output — **A:** There is a place in the riverside, Starbucks, that has a 5-star customer rating.

🙂 **In-context learning:**
No task-specific training

In-context Learning

Prompt

Instruction: Summarize the following data table:

Example: TABLE: name: Alimentum | area: city centre | family friendly: no
A: There is a place in the city centre, Alimentum, that is not family-friendly.

Input: TABLE: name: Starbucks | area: riverside | customer rating: 5 star

GPT-3

Output: A: There is a place in the riverside, Starbucks, that has a 5-star customer rating.

🙂 **In-context learning:**
No task-specific training

❌ **Cannot exploit large training set.**

In-context Learning

Prompt

Instruction	**Summarize the following data table:**
Example	**TABLE: name: Alimentum | area: city centre | family friendly: no** **A: There is a place in the city centre, Alimentum, that is not family-friendly.**
Input	**TABLE: name: Starbucks | area: riverside | customer rating: 5 star**

GPT-3

Output	**A:** There is a place in the riverside, Starbucks, that has a 5-star customer rating.

In-context learning:
No task-specific training

❌ **Cannot exploit large training set.**

❌ **Manually written prompts may be suboptimal.**

In-context Learning

Prompt

- **Instruction**: Summarize the following data table:
- **Example**: TABLE: name: Alimentum | area: city centre | family friendly: no
  A: There is a place in the city centre, Alimentum, that is not family-friendly.
- **Input**: TABLE: name: Starbucks | area: riverside | customer rating: 5 star

GPT-3

Output: **A:** There is a place in the riverside, Starbucks, that has a 5-star customer rating.

🙂 **In-context learning:**
No task-specific training

❌ **Cannot exploit large training set.**

❌ **Manually written prompts may be suboptimal.**

❌ **Doesn't generalize to smaller LM like GPT-2.**

Prefix-tuning

Fine-tuning
(100% parameters)

❌ **Expensive to store a copy of the full LM for each task.**

In-context learning
(0% parameters)

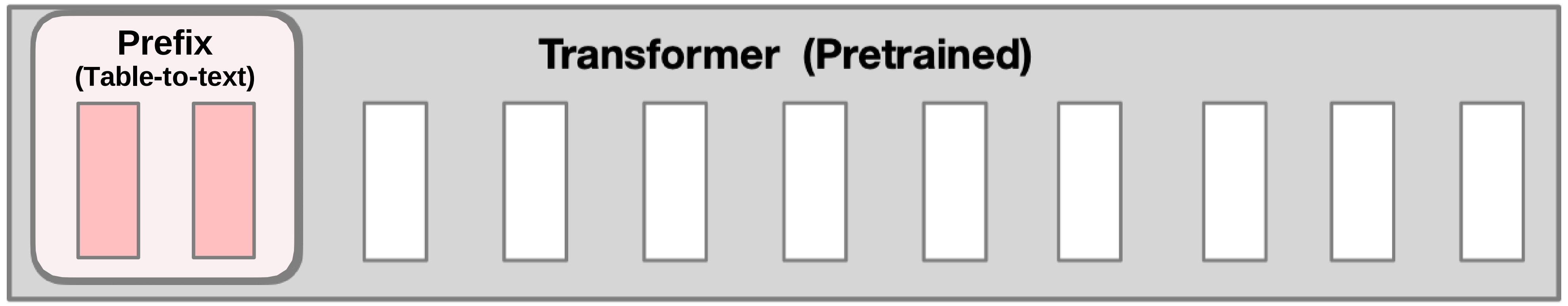
❌ **Cannot exploit large training sets.**
❌ **Manually written prompts may be suboptimal.**

Prefix-tuning

Fine-tuning
(100% parameters)

❌ **Expensive to store a copy of the full LM for each task.**

Freezing the LM parameters



250K parameters

Prefix-tuning
(0.1% parameters)

In-context learning
(0% parameters)

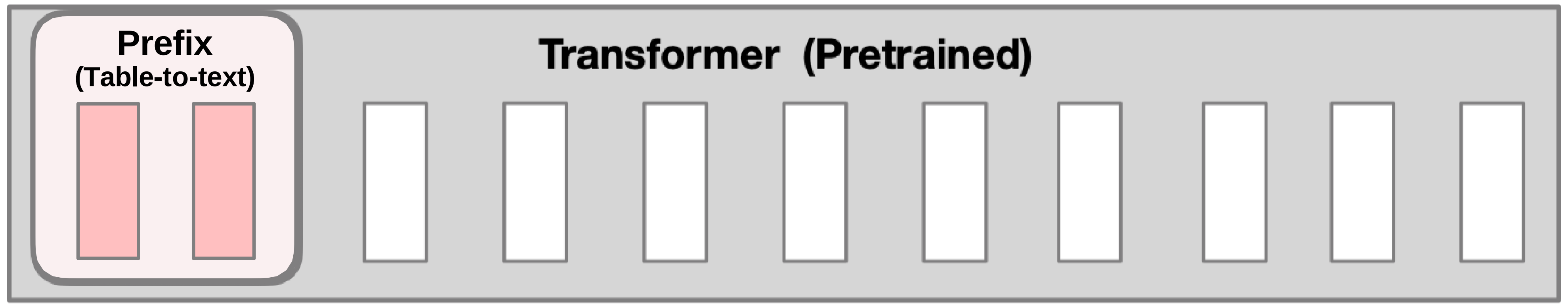
❌ **Cannot exploit large training sets.**
❌ **Manually written prompts may be suboptimal.**

Prefix-tuning v.s. Fine-tuning

Fine-tuning
(100% parameters)

❌ **Expensive to store a copy of the full LM for each task.**

Freezing the LM parameters



250K parameters

Prefix-tuning
(0.1% parameters)

✅ **Very lightweight.**

In-context learning
(0% parameters)

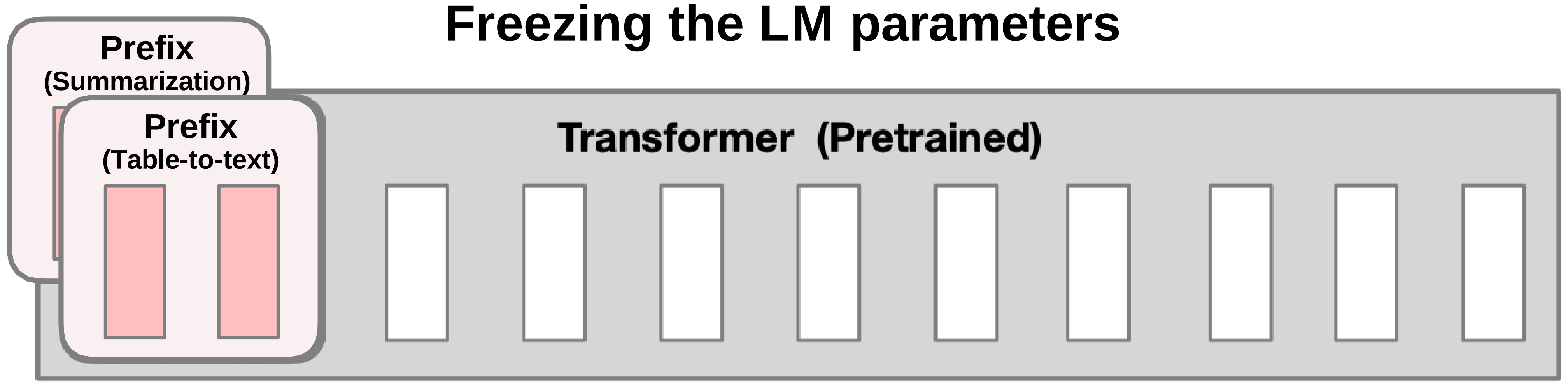
❌ **Cannot exploit large training sets.**
❌ **Manually written prompts may be suboptimal.**

Prefix-tuning v.s. Fine-tuning

Fine-tuning
(100% parameters)

❌ Expensive to store a copy of the full LM for each task.

Freezing the LM parameters



250K parameters

Prefix-tuning
(0.1% parameters)

✅ Very lightweight.

In-context learning
(0% parameters)

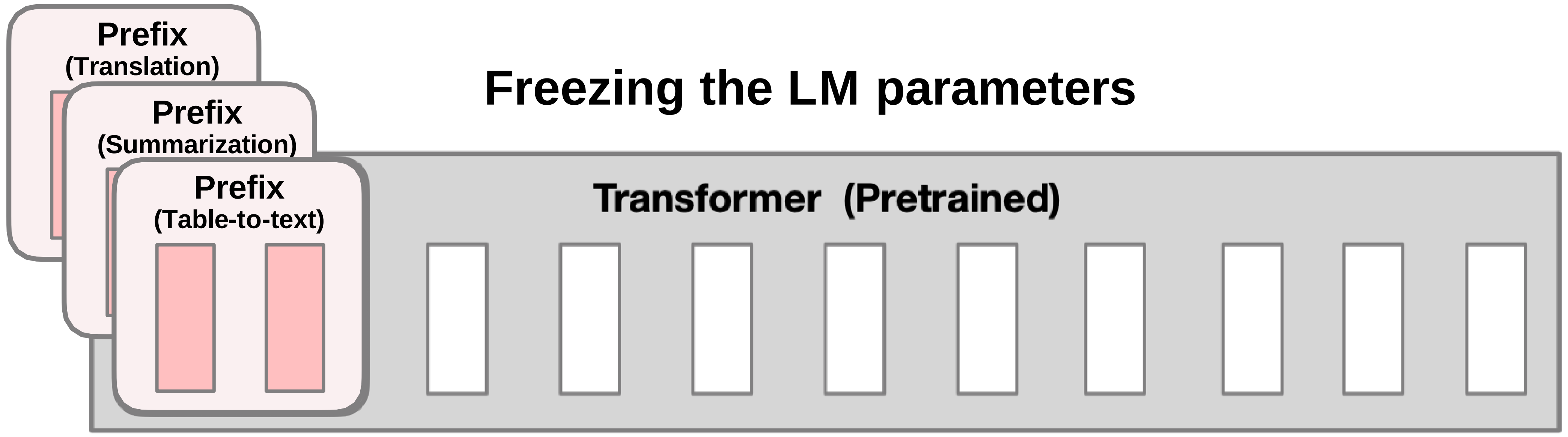
❌ Cannot exploit large training sets.
❌ Manually written prompts may be suboptimal.

Prefix-tuning v.s. Fine-tuning

Fine-tuning
(100% parameters)

❌ **Expensive to store a copy of the full LM for each task.**

Freezing the LM parameters



250K parameters

Prefix-tuning
(0.1% parameters)

✅ **Very lightweight.**

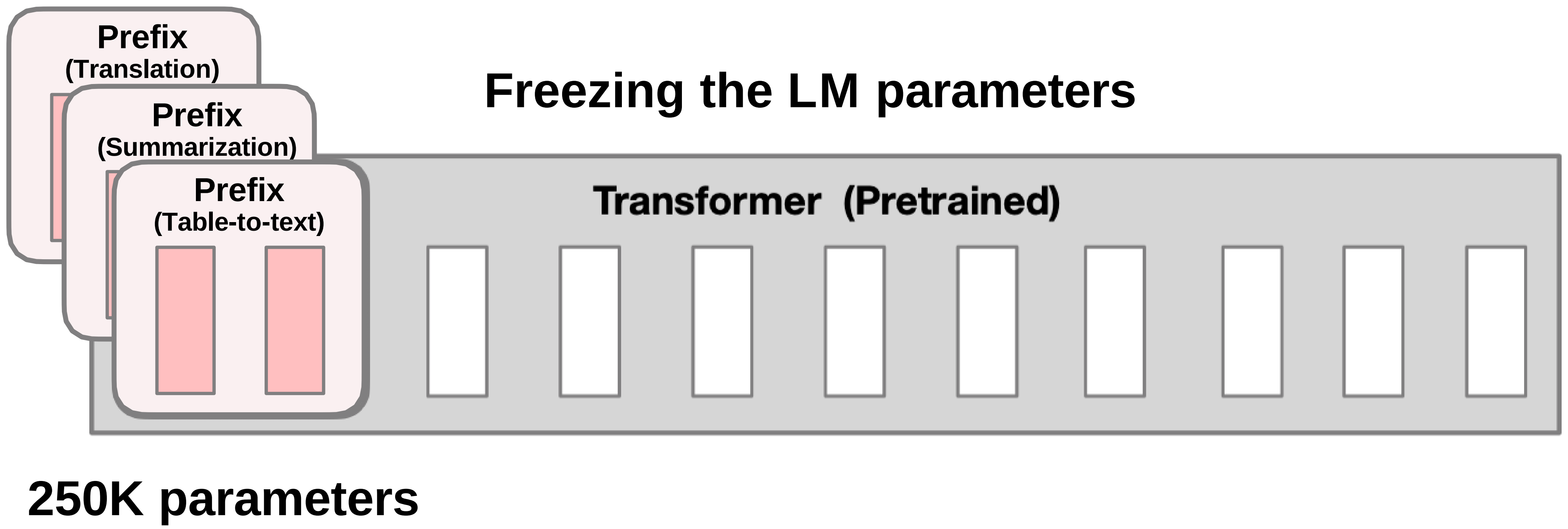
In-context learning
(0% parameters)

❌ **Cannot exploit large training sets.**
❌ **Manually written prompts may be suboptimal.**

Prefix-tuning v.s. In-context learning

Fine-tuning
(100% parameters)

❌ **Expensive to store a copy of the full LM for each task.**



Prefix-tuning
(0.1% parameters)

✅ **Very lightweight.**
✅ **Can exploit large training set via the trainable prefix.**

In-context learning
(0% parameters)

❌ **Cannot exploit large training sets.**
❌ **Manually written prompts may be suboptimal.**

RELATED WORK

[1] Parameter-efficient transfer learning for NLP —Houlsby et. al. 2019

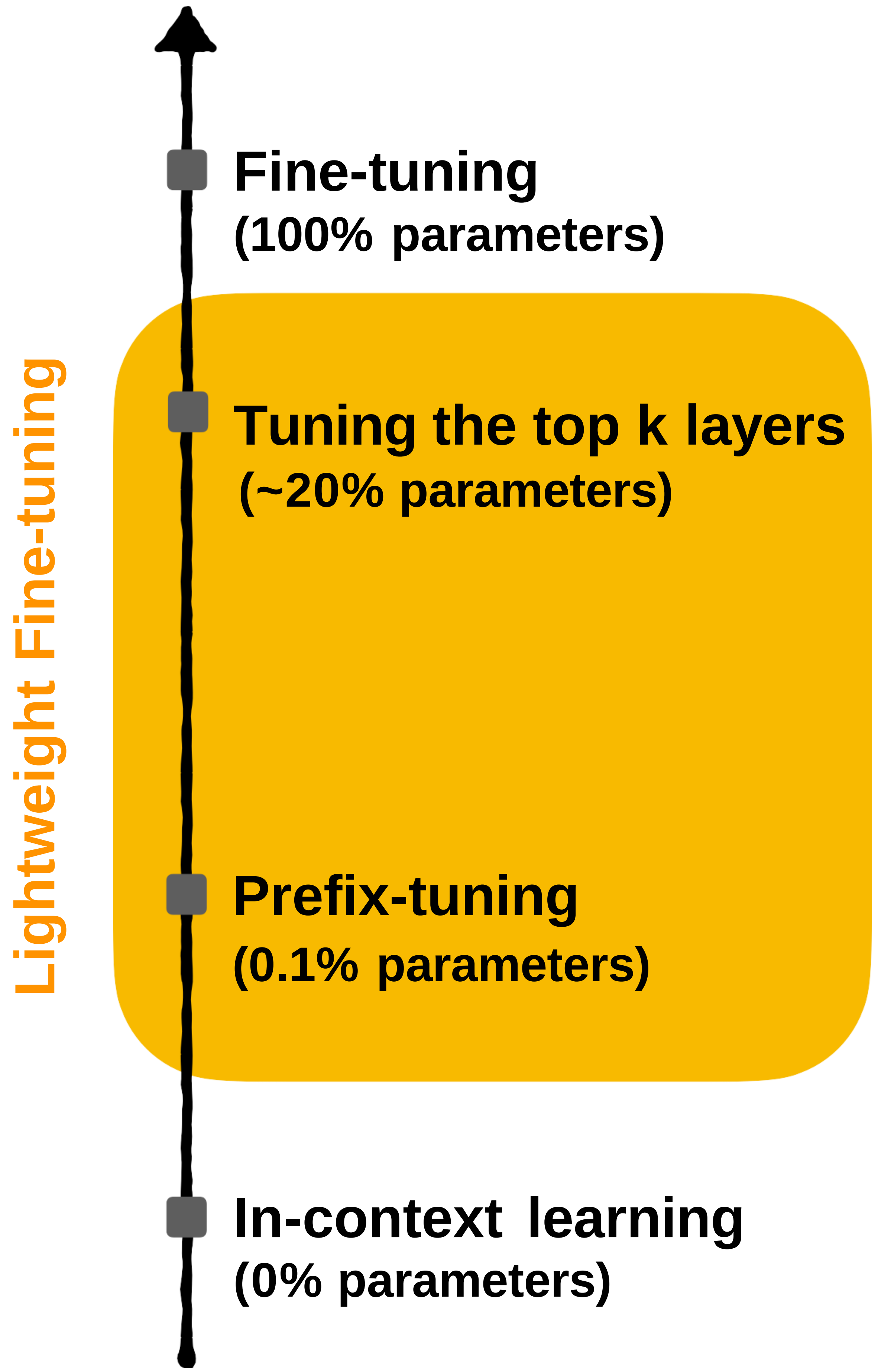
Lightweight Fine-tuning

Fine-tuning
(100% parameters)

Prefix-tuning
(0.1% parameters)

In-context learning
(0% parameters)

[1] Parameter-efficient transfer learning for NLP —Houlsby et. al. 2019



[1] Parameter-efficient transfer learning for NLP —Houlsby et. al. 2019

Lightweight Fine-tuning

- **Fine-tuning** (100% parameters)
- **Tuning the top k layers** (~20% parameters)
- **Adapter-tuning** [1] (3-4% parameters)
- **Prefix-tuning** (0.1% parameters)
- **In-context learning** (0% parameters)

[1] Parameter-efficient transfer learning for NLP —Houlsby et. al. 2019

Lightweight Fine-tuning

- **Fine-tuning**
  (100% parameters)
- **Tuning the top k layers**
  (~20% parameters)
- **Adapter-tuning** [1]
  (3-4% parameters)
- **Prefix-tuning**
  (0.1% parameters)
- **In-context learning**
  (0% parameters)

❌ **Performance drop compared with fine-tuning.**

RELATED WORK

[1] Parameter-efficient transfer learning for NLP —Houlsby et. al. 2019

Lightweight Fine-tuning

Fine-tuning
(100% parameters)

Tuning the top k layers
(~20% parameters)

❌ Performance drop compared with fine-tuning.

Adapter-tuning [1]
(3-4% parameters)

🆗 Moderately lightweight: 30x reduction compared to fine-tuning.
✅ Maintains comparable performance to fine-tuning.

Prefix-tuning
(0.1% parameters)

In-context learning
(0% parameters)

[1] Parameter-efficient transfer learning for NLP —Houlsby et. al. 2019

Lightweight Fine-tuning

Fine-tuning
(100% parameters)

Tuning the top k layers
(~20% parameters)

❌ Performance drop compared with fine-tuning.

Adapter-tuning [1]
(3-4% parameters)

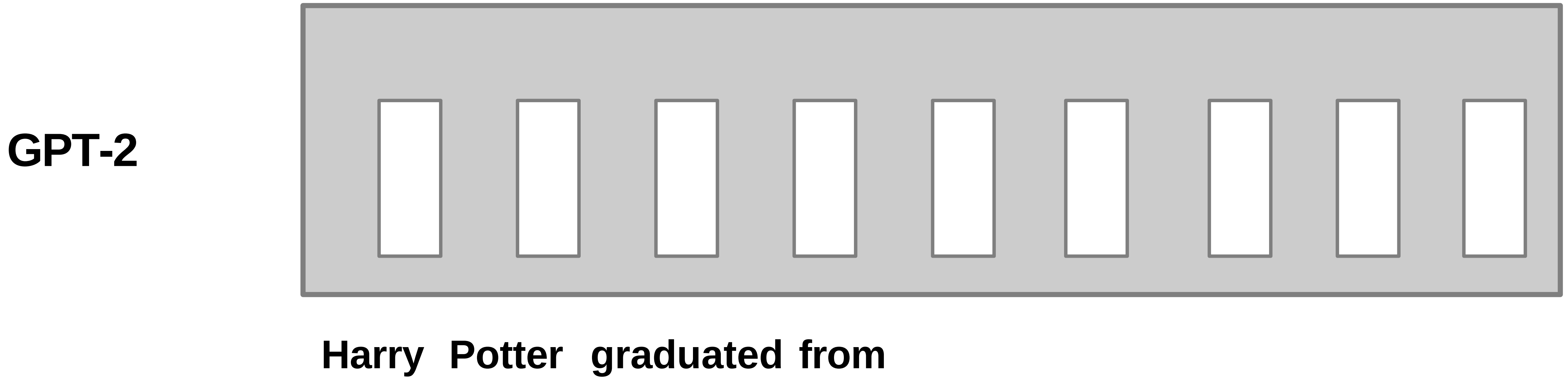
🆗 Moderately lightweight: 30x reduction compared to fine-tuning.
✅ Maintains comparable performance to fine-tuning.

Prefix-tuning
(0.1% parameters)

✅ Very lightweight: 1000x reduction compared to fine-tuning.
✅ Maintains comparable performance to fine-tuning.

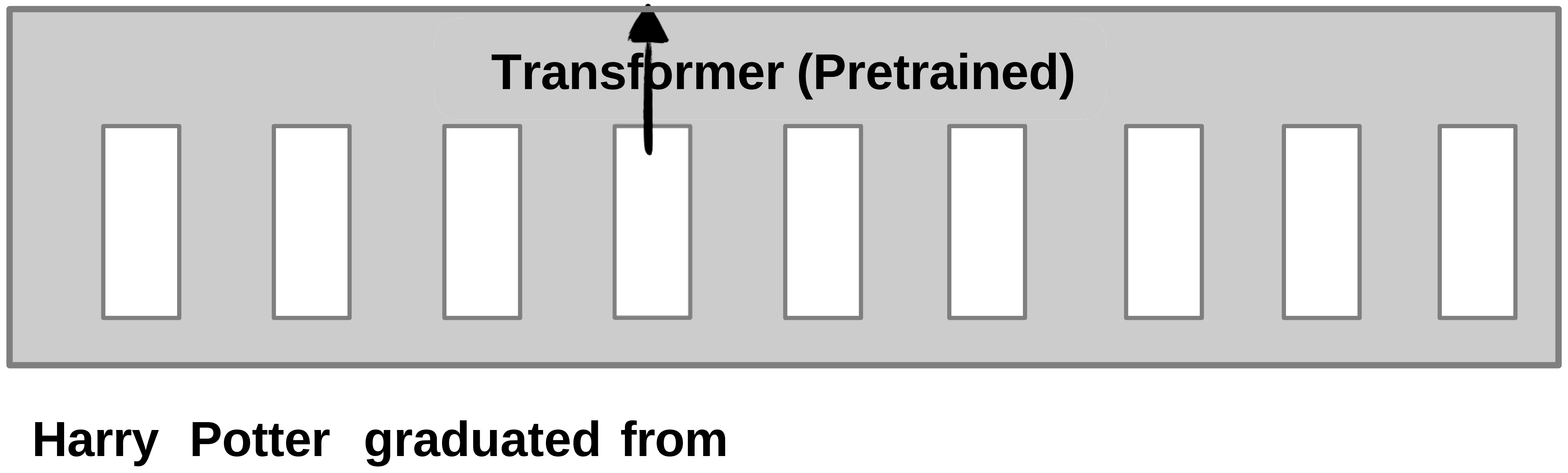
In-context learning
(0% parameters)

Prefix-tuning draws inspiration from prompting



Prefix-tuning draws inspiration from prompting

GPT-2



Prefix-tuning draws inspiration from prompting

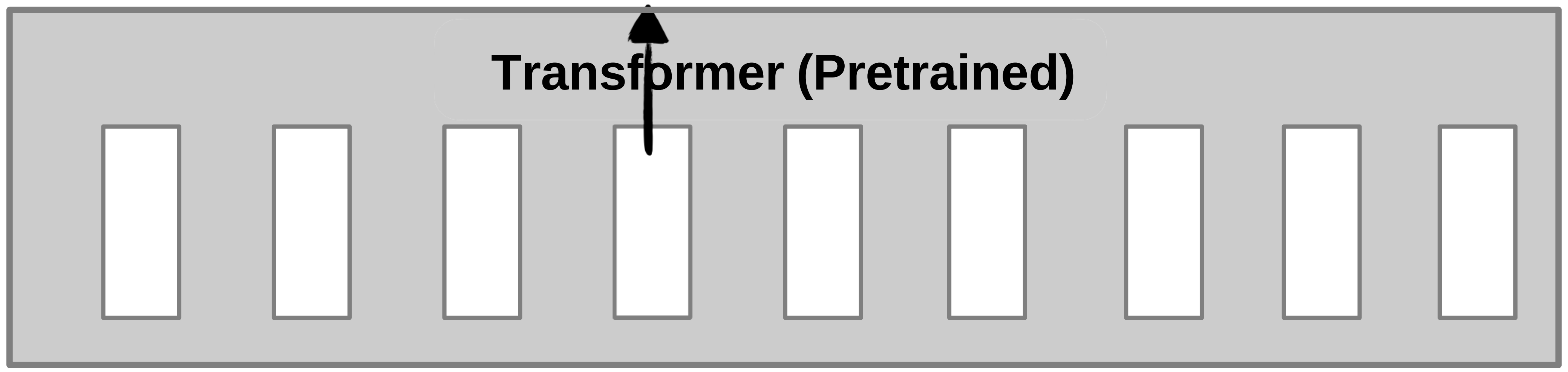
P(Hogwarts | Harry Potter graduated from) 0.8

P(Oxford | Harry Potter graduated from) 0.05

P(is | Harry Potter graduated from) 0.0001

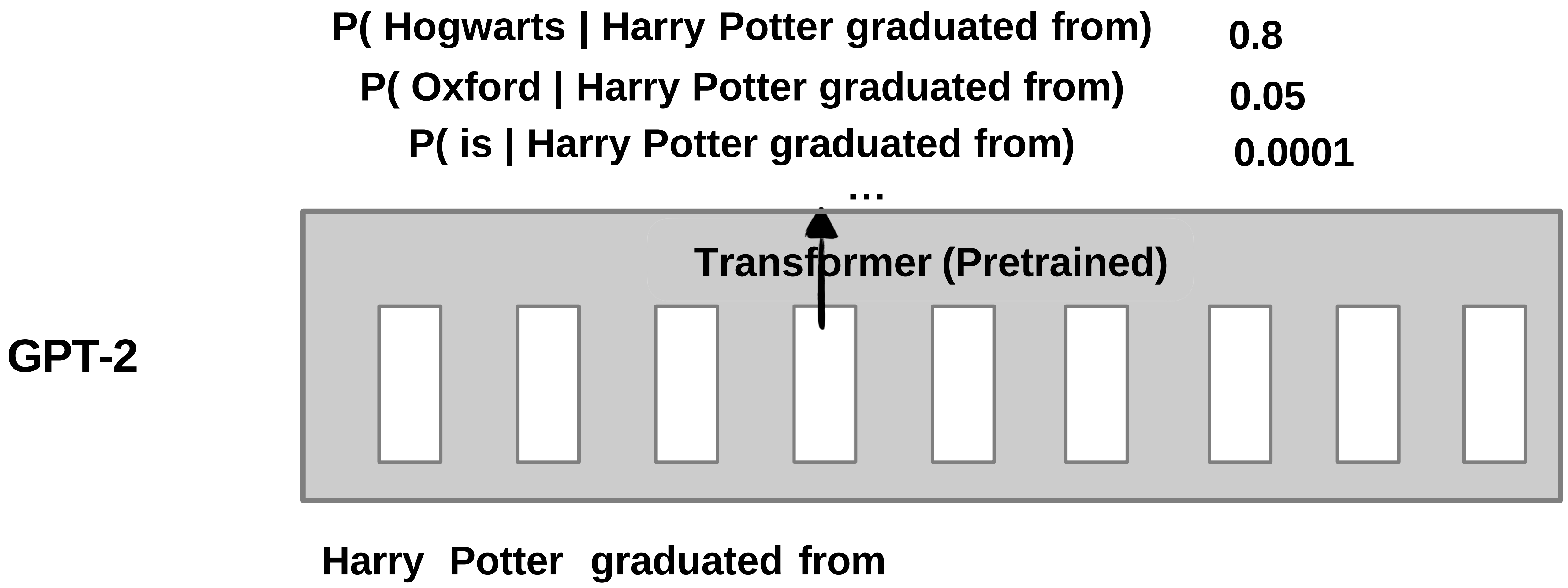
…

GPT-2



Harry Potter graduated from

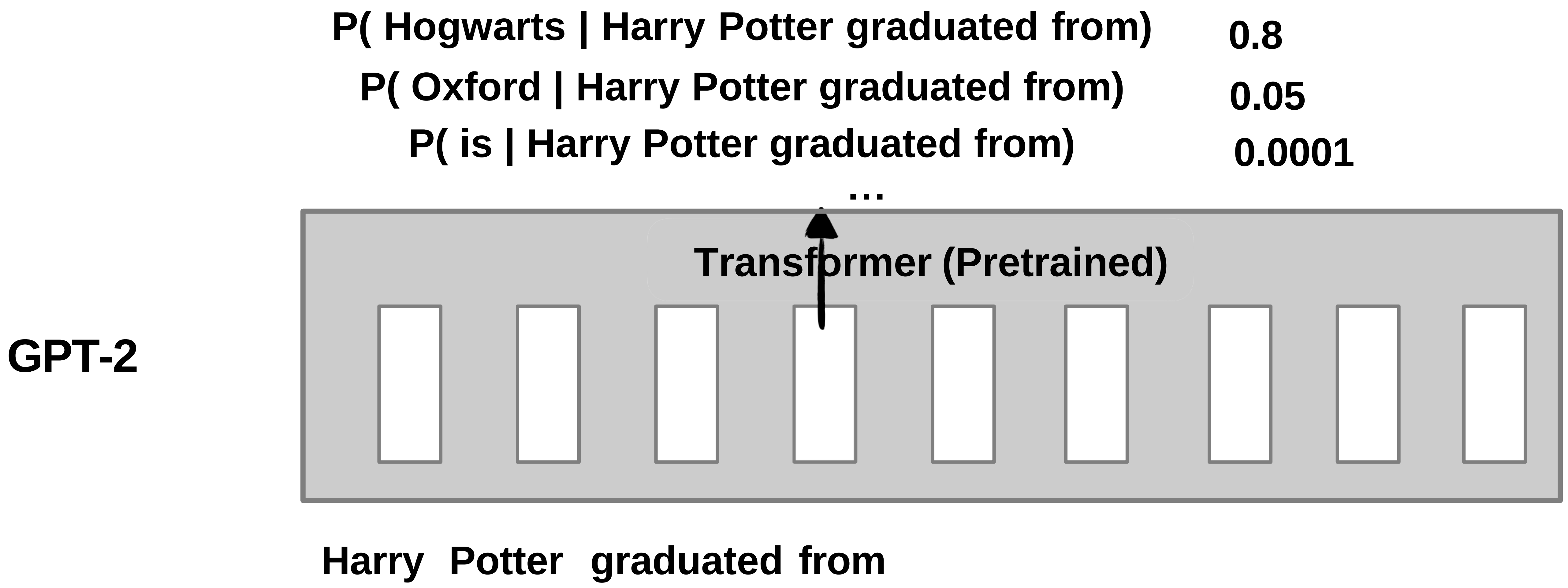
Prefix-tuning draws inspiration from prompting



Goal: how to make the LM assign higher probability to a word (e.g. "Hogwarts")?

[without parameter updates]

Prefix-tuning draws inspiration from prompting

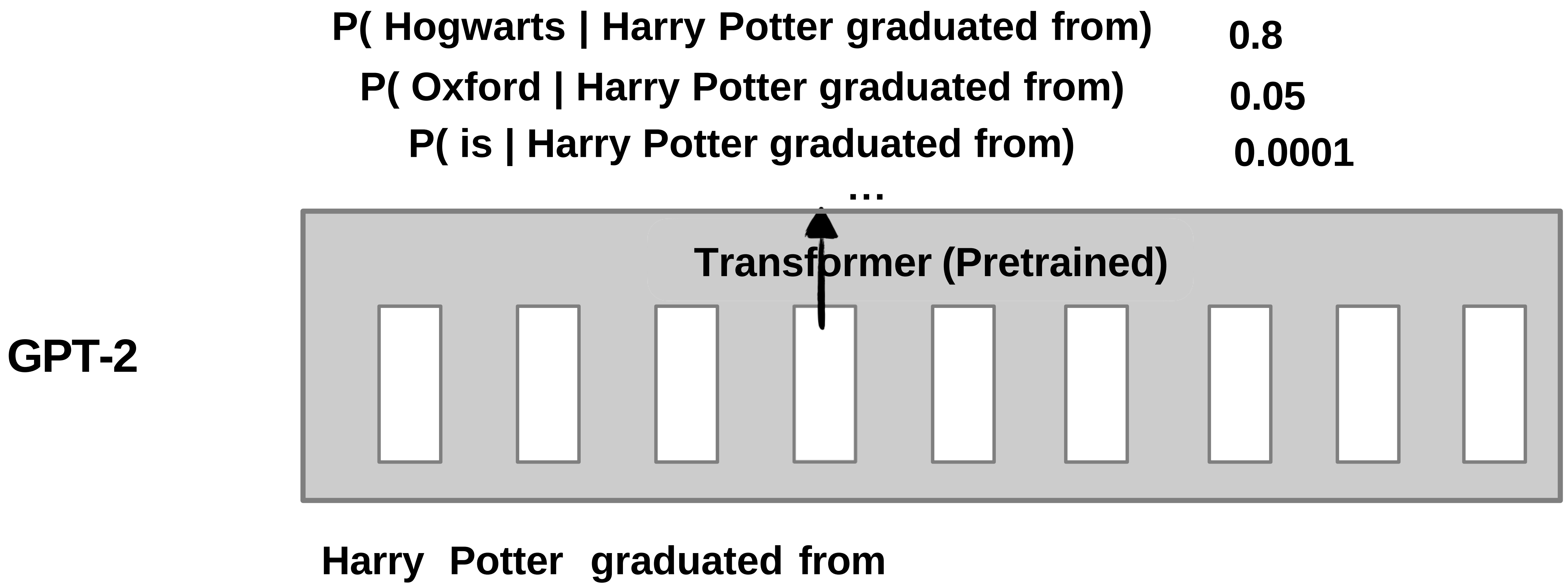


Goal: how to make the LM assign higher probability to a word (e.g. "Hogwarts")?

[without parameter updates]

Hogwarts **P(Hogwarts)**

Prefix-tuning draws inspiration from prompting

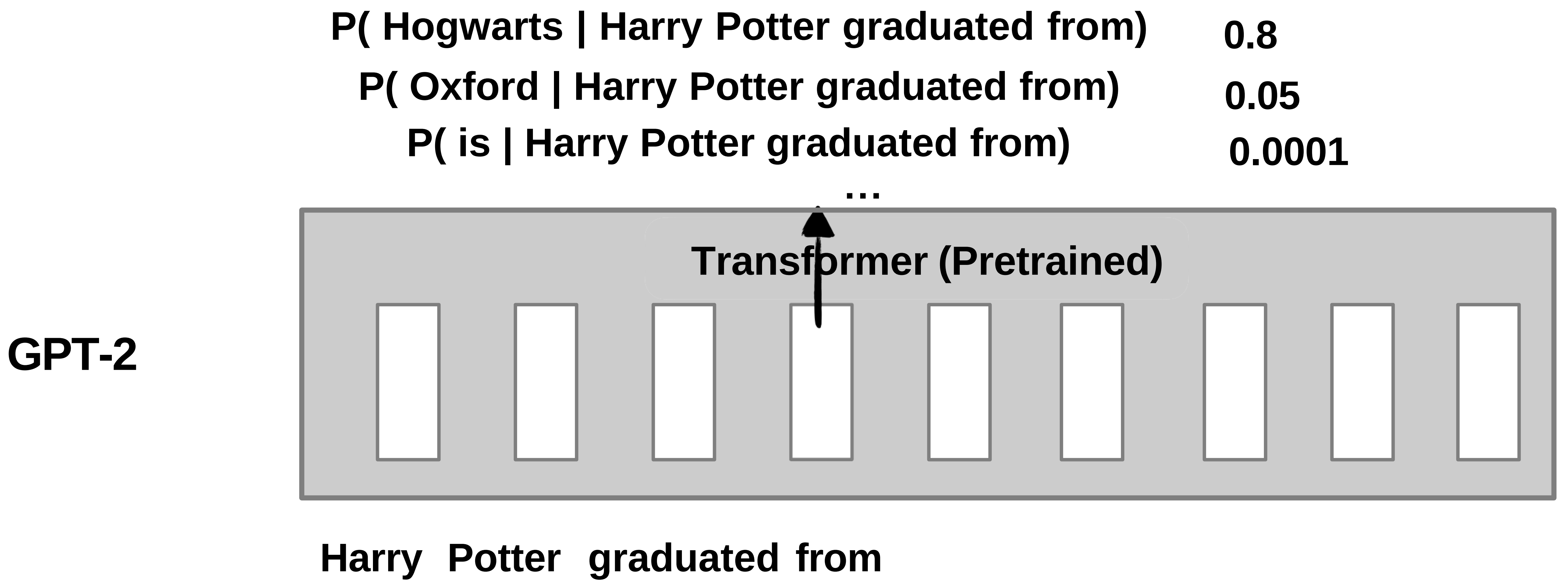


Goal: how to make the LM assign higher probability to a word (e.g. “Hogwarts”)?

[without parameter updates]

Harry Potter graduated from Hogwarts **P(Hogwarts)**

Prefix-tuning draws inspiration from prompting

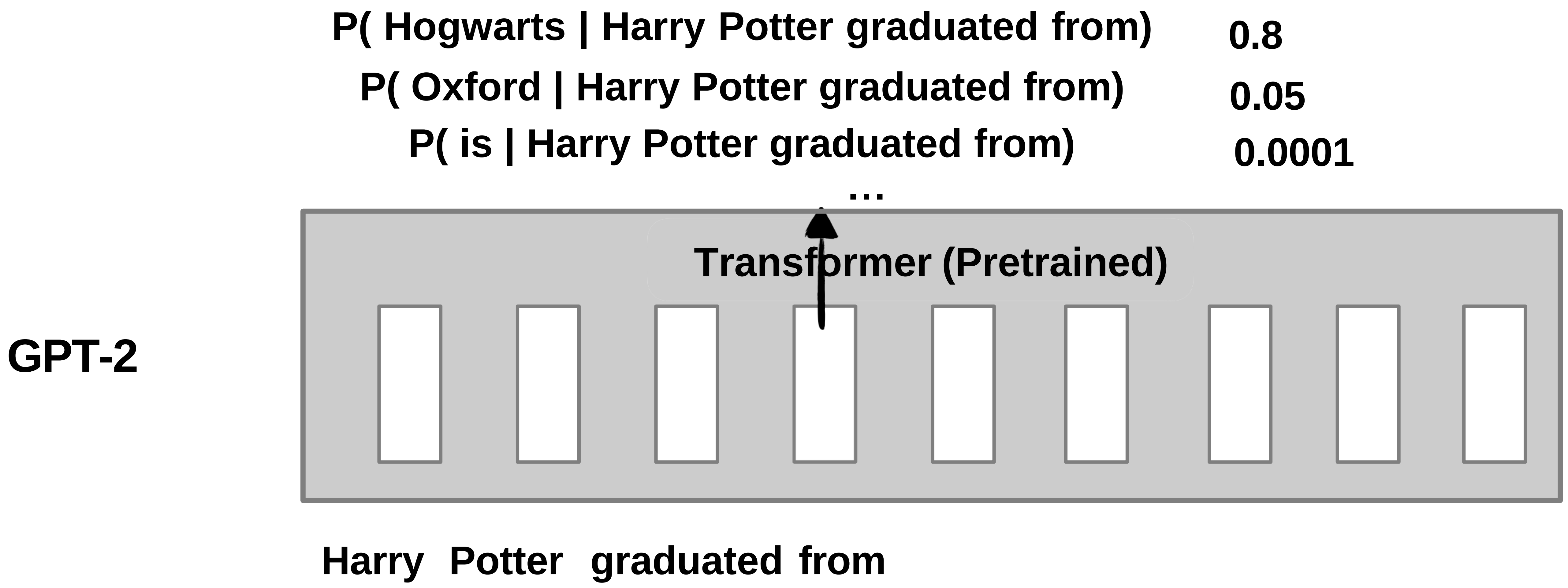


Goal: how to make the LM assign higher probability to a word (e.g. "Hogwarts")?

[without parameter updates]

Harry Potter graduated from Hogwarts **P(Hogwarts) << P(Hogwarts | Harry Potter …)**

Prefix-tuning draws inspiration from prompting



Goal: how to make the LM assign higher probability to a word (e.g. "Hogwarts")?

[without parameter updates]

Harry Potter graduated from Hogwarts **P(Hogwarts) << P(Hogwarts | Harry Potter …)**

Takeaway: prepending a proper context is enough to steer the LM to generate a word/phrase/sentence.

Intuition

Takeaway: prepending a proper context is enough to steer the LM to generate a word/phrase/sentence.

Can we find a context that steers the LM to solve an NLG task?

Intuition

Takeaway: prepending a proper context is enough to steer the LM to generate a word/phrase/sentence.

Can we find a context that steers the LM to solve an NLG task?

$$\max\ P(y \mid x)$$

Input Table (x): name[Clowns] customer-rating[1 out of 5] eatType[coffee shop] food[Chinese] area[riverside] near[Clare Hall]

Textual Description (y): Clowns is a coffee shop inthe riverside area near Clare Hall that has a rating 1 out of 5 . They serve Chinese food .

Intuition

Takeaway: prepending a proper context is enough to steer the LM to generate a word/phrase/sentence.

Can we find a context that steers the LM to solve an NLG task?

$\max\ P(y \mid x)$

$P(y \mid x) \ll P(y \mid t\ \ x)$

Task Instruction (t): Summarize the following table:

Input Table (x): name[Clowns] customer-rating[1 out of 5] eatType[coffee shop] food[Chinese] area[riverside] near[Clare Hall]

Textual Description (y): Clowns is a coffee shop inthe riverside area near Clare Hall that has a rating 1 out of 5 . They serve Chinese food .

Intuition

Takeaway: prepending a proper context is enough to steer the LM to generate a word/phrase/sentence.

Can we find a context that steers the LM to solve an NLG task?

max P(y | x)

P(y | x) << P(y | t x)

Task Instruction (t): Summarize the following table:

Input Table (x): name[Clowns] customer-rating[1 out of 5] eatType[coffee shop] food[Chinese] area[riverside] near[Clare Hall]

Textual Description (y): Clowns is a coffee shop inthe riverside area near Clare Hall that has a rating 1 out of 5 . They serve Chinese food .

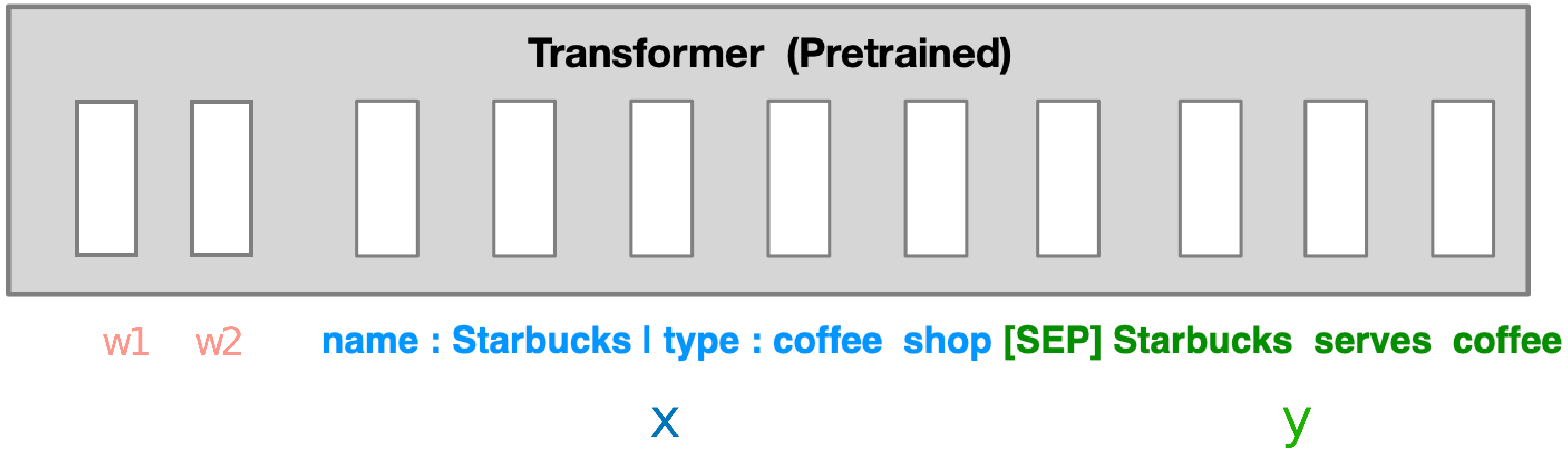
Might guide a human, but fails for moderately sized LM like GPT-2.

Intuition

Solution: Optimize the instruction!
Learn a good instruction that can steer the LM for an NLG task.

1.Optimize the discrete instruction via discrete optimization.

- ❌ **Discrete optimization is challenging.**
- ❌ **Not expressive.**



$$w_1, w_2 = \underset{w'_1, w'_2 \in \text{Vocab}}{\text{argmax}} \; \mathbb{E}_{x,y}[\log P_{\text{GPT2}}(y \mid w'_1, w'_2, x)]$$

Intuition

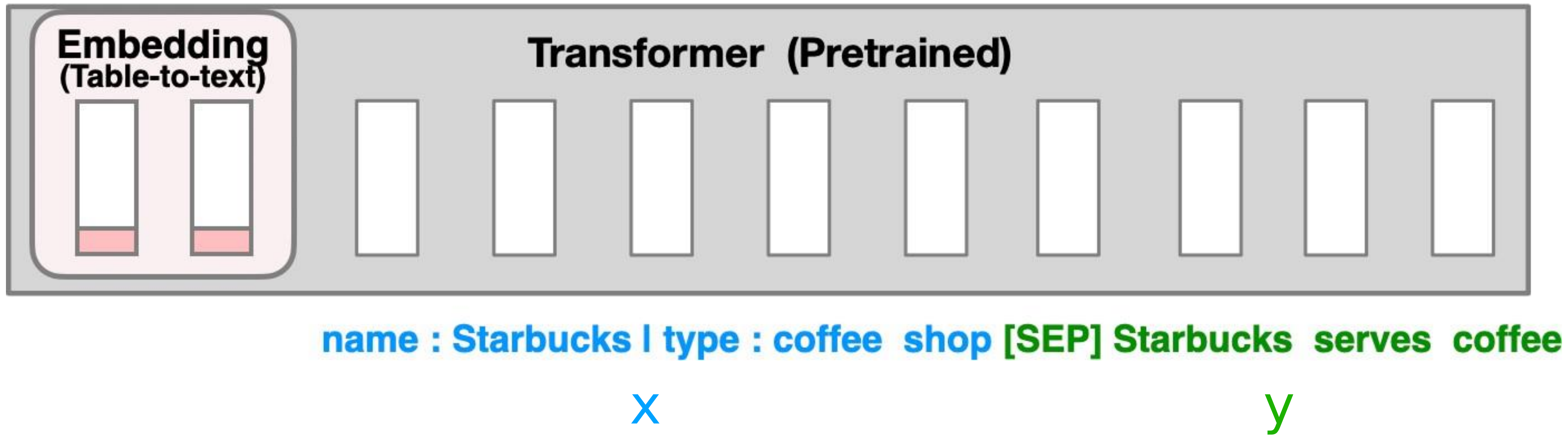
Solution: Optimize the instruction!
Learn a good instruction that can steer the LM for an NLG task.

1.Optimize the discrete instruction via discrete optimization.

- ❌ ❌ **Discrete optimization is challenging. Not expressive.**

2. Optimize the instruction as continuous word embeddings.

- ❌ **Moderately expressive.**



$$e_1, e_2 = \underset{e'_1, e'_2 \in \mathbb{R}^d}{\operatorname{argmax}} \; \mathbb{E}_{x,y}[\log P_{\mathrm{GPT2}}(y \mid e'_1, e'_2, \mathrm{emb}(x))]$$

Intuition

Solution: Optimize the instruction!
Learn a good instruction that can steer the LM for an NLG task.

1.Optimize the discrete instruction via discrete optimization.

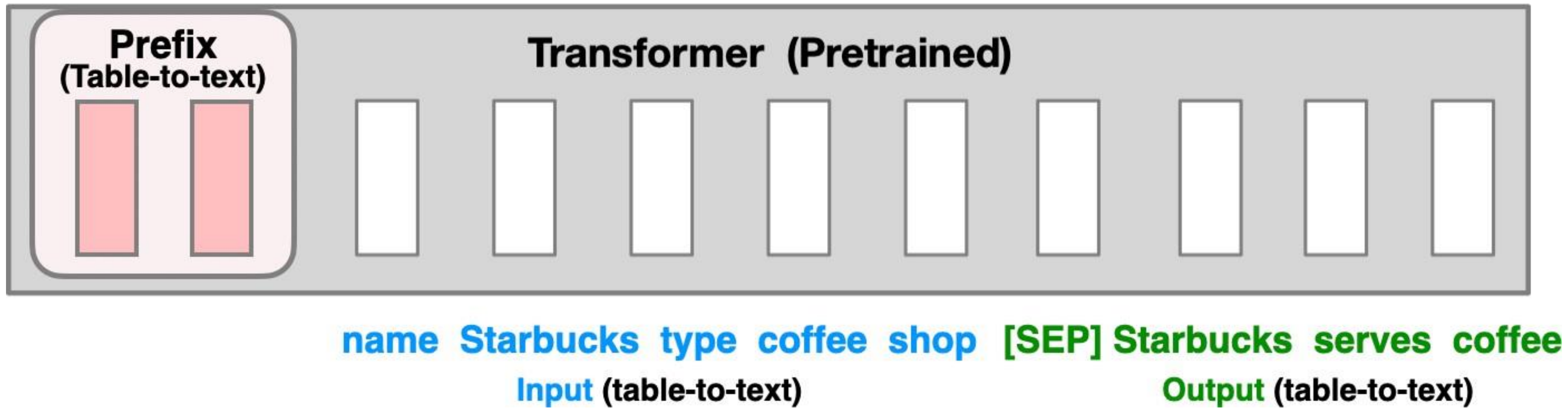
❌ Discrete optimization is challenging. Not
❌ expressive.

2. Optimize the instruction as continuous word embeddings.

❌ Moderately expressive.

3. Optimize the instruction as prefix activations of all layers.

✅ Very expressive.



$$p_1, p_2 = \underset{p'_1, p'_2 \in \mathbb{R}^{l \times d}}{\operatorname{argmax}} \mathbb{E}_{x,y}[\log P_{\mathrm{GPT2}}(y \mid p'_1, p'_2, x)]$$

Intuition

Solution: Optimize the instruction!
Learn a good instruction that can steer the LM for an NLG task.

1.Optimize the discrete instruction via discrete optimization.

❌ **Discrete optimization is challenging. Not**
❌ **expressive.**

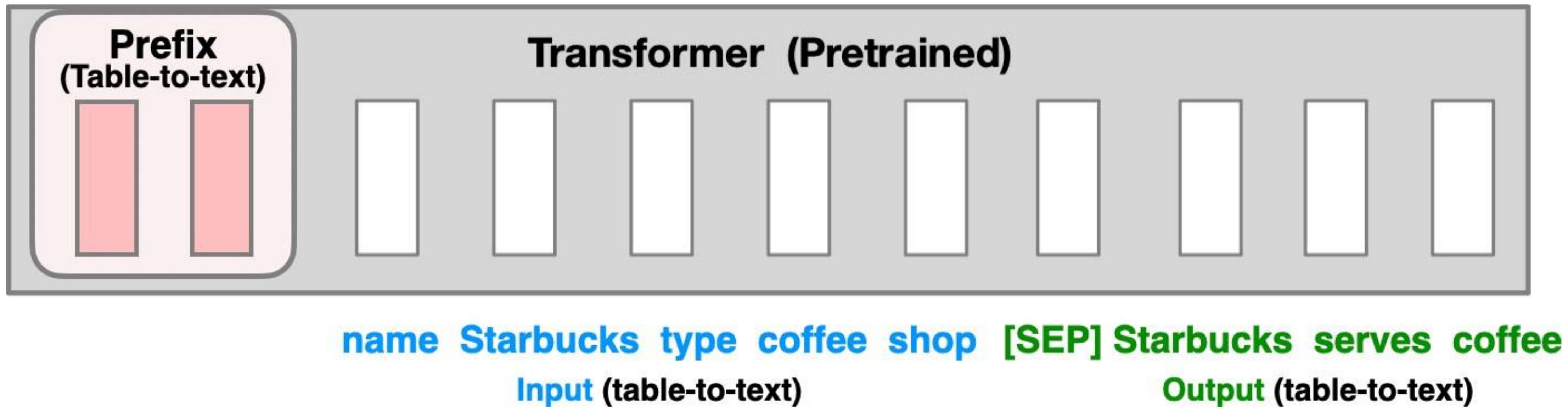
2. Optimize the instruction as continuous word embeddings.

❌ **Moderately expressive.**

3. Optimize the instruction as prefix activations of all layers.

✅ **Very expressive.**

Prefix-tuning



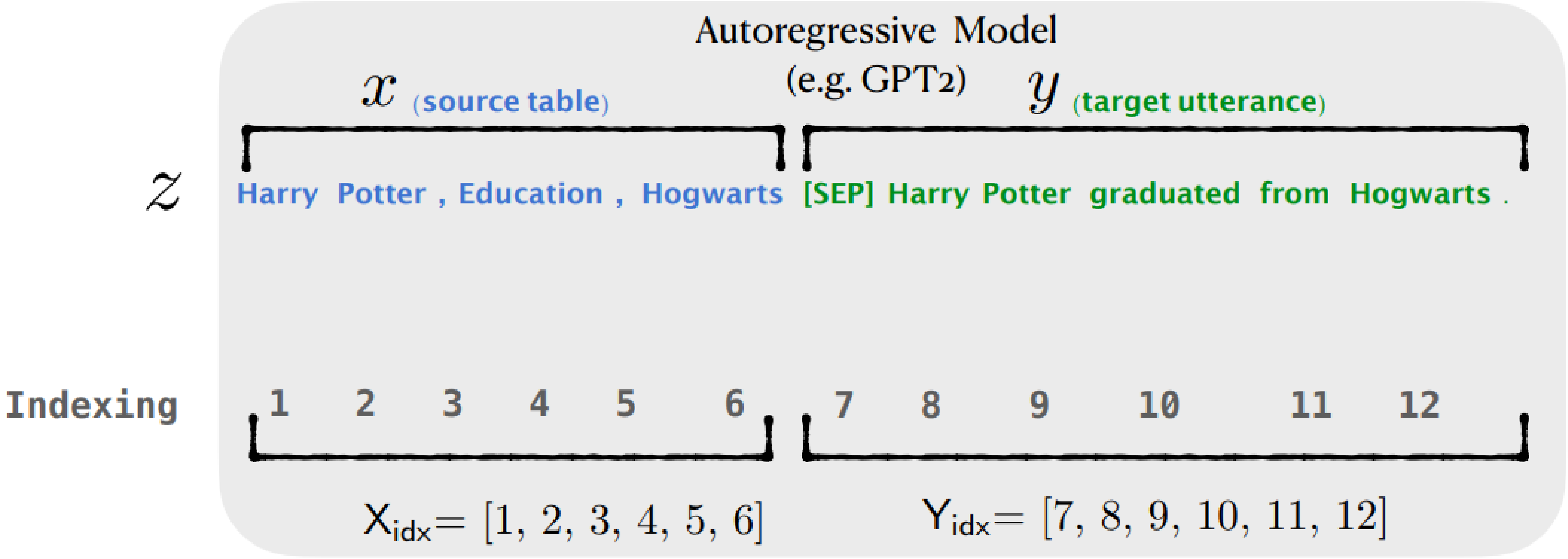
$$p_1, p_2 = \underset{p'_1, p'_2 \in \mathbb{R}^{l \times d}}{\operatorname{argmax}} \; \mathbb{E}_{x,y}[\log P_{\mathrm{GPT2}}(y \mid p'_1, p'_2, x)]$$

Table-to-text

Example:

Input Table (x):	name[Clowns] customer-rating[1 out of 5] eatType[coffee shop] food[Chinese] area[riverside] near[Clare Hall]
Textual Description (y):	Clowns is a coffee shop inthe riverside area near Clare Hall that has a rating 1 out of 5 . They serve Chinese food .

Fine-tuning

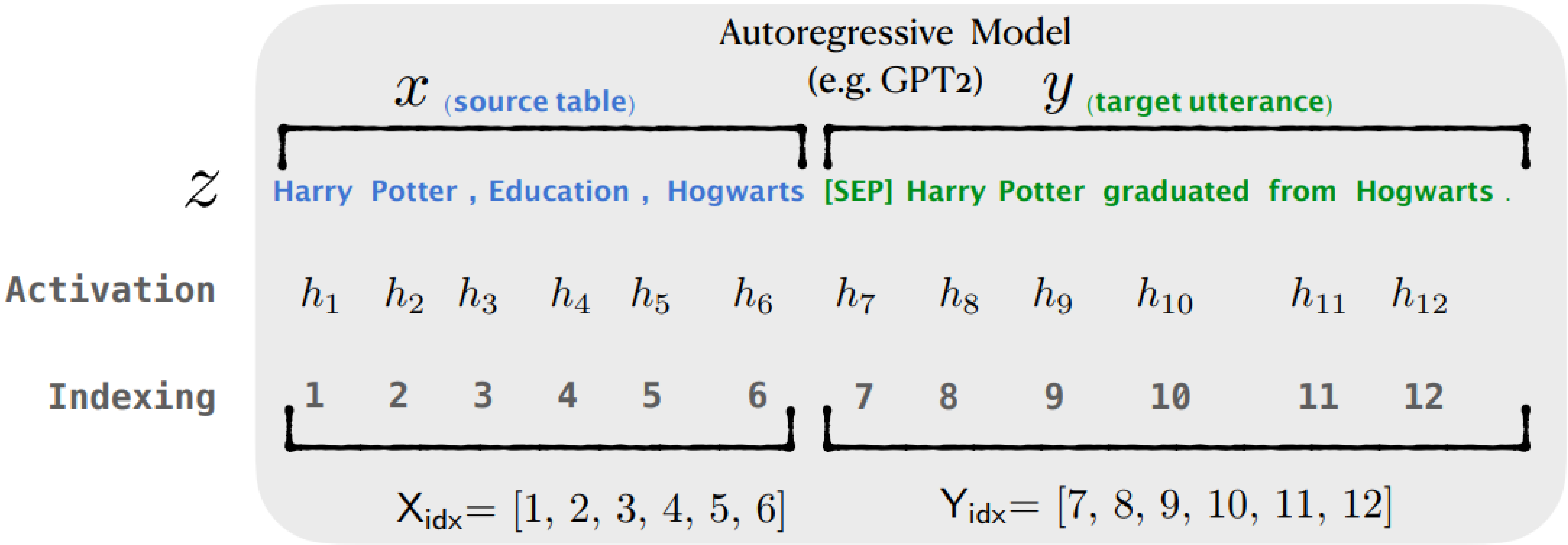


Autoregressive Model (e.g. GPT2)
x (source table)
y (target utterance)
z
Harry Potter , Education , Hogwarts [SEP] Harry Potter graduated from Hogwarts .
Indexing
1 2 3 4 5 6 7 8 9 10 11 12
X_{idx} = [1, 2, 3, 4, 5, 6]
Y_{idx} = [7, 8, 9, 10, 11, 12]

Fine-tuning

Autoregresive LM:

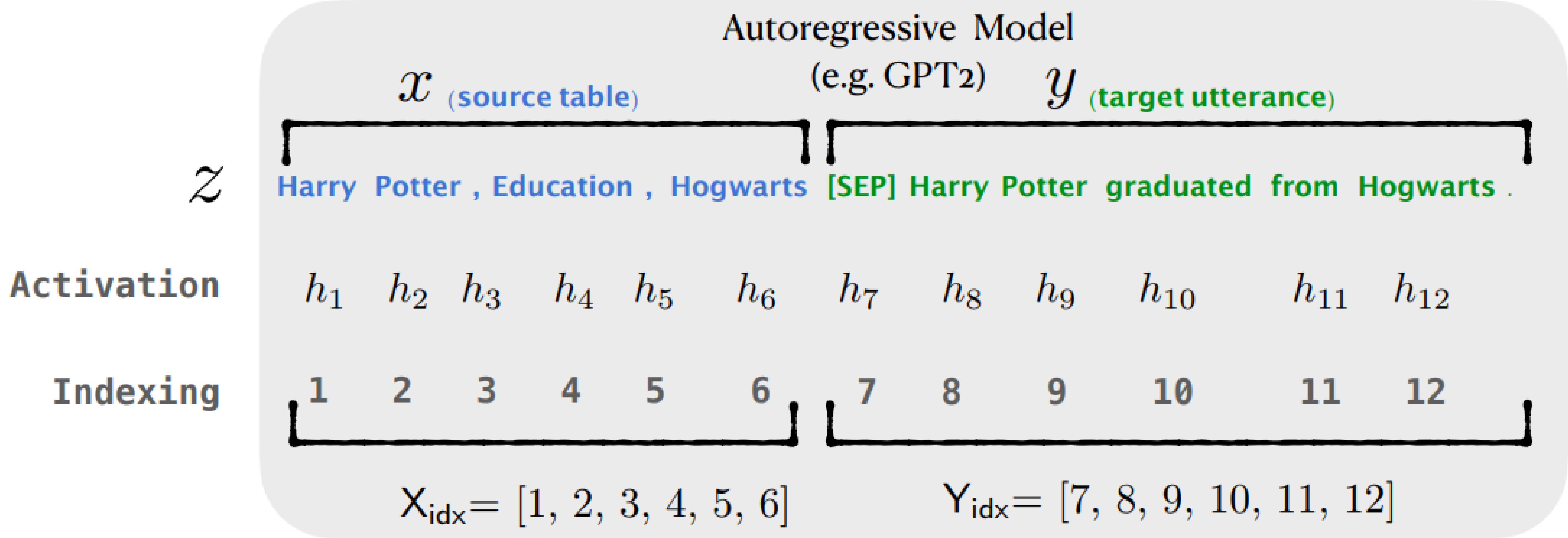
$$h_i = \mathrm{LM}(z_i, h_{<i})$$



Fine-tuning

Autoregresive LM:

$$h_i = \mathrm{LM}(z_i, h_{<i})$$

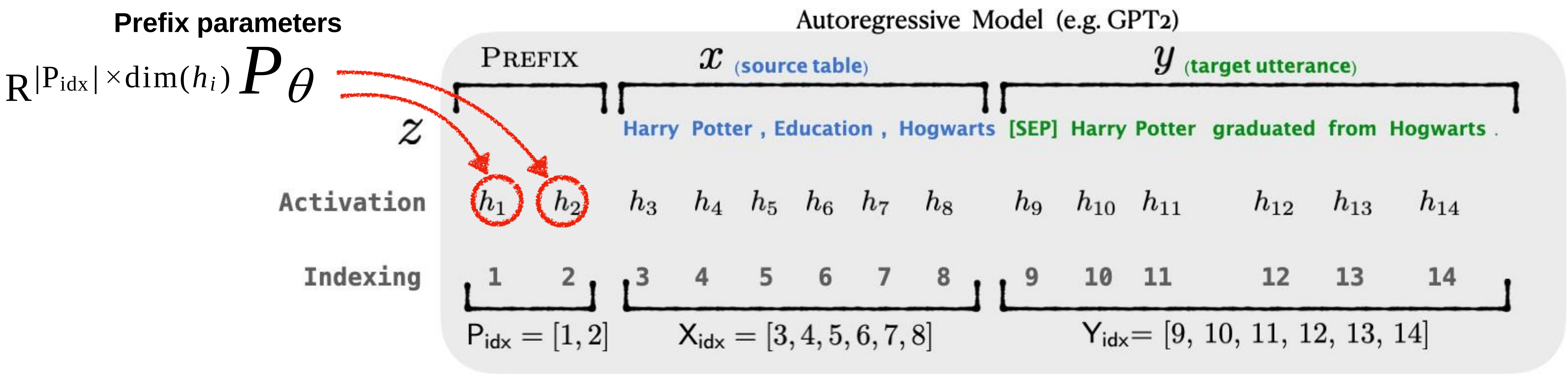


Objective:

$$\max_{\phi} \log p_{\phi}(y \mid x) = \sum_{i \in Y_{idx}} \log p_{\phi}(z_i \mid h_{<i})$$

PREFIX-TUNING

$$h_i = \begin{cases} P_\theta[i,:], & \text{if } i \in \mathsf{P}_{\mathsf{idx}}, \\ \mathrm{LM}_\phi(z_i, h_{<i}), & \text{otherwise.} \end{cases}$$



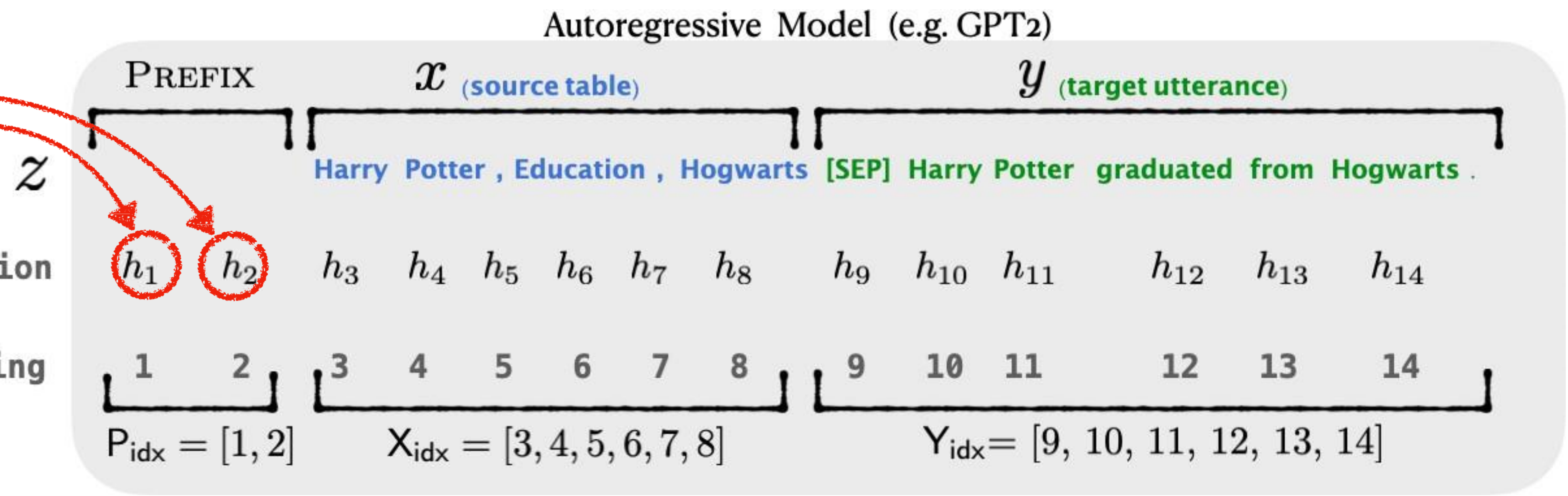
$$h_i = \begin{cases} P_\theta[i, :], & \text{if } i \in \mathsf{P}_{\mathsf{idx}}, \\ \mathrm{LM}_\phi(z_i, h_{<i}), & \text{otherwise.} \end{cases}$$

$$\max_\theta \log p_{\phi,\theta}(y \mid x) = \sum_{i \in \mathsf{Y}_{\mathsf{idx}}} \log p_{\phi,\theta}(z_i \mid h_{<i})$$

freeze LM parameters ϕ
update prefix parameters θ

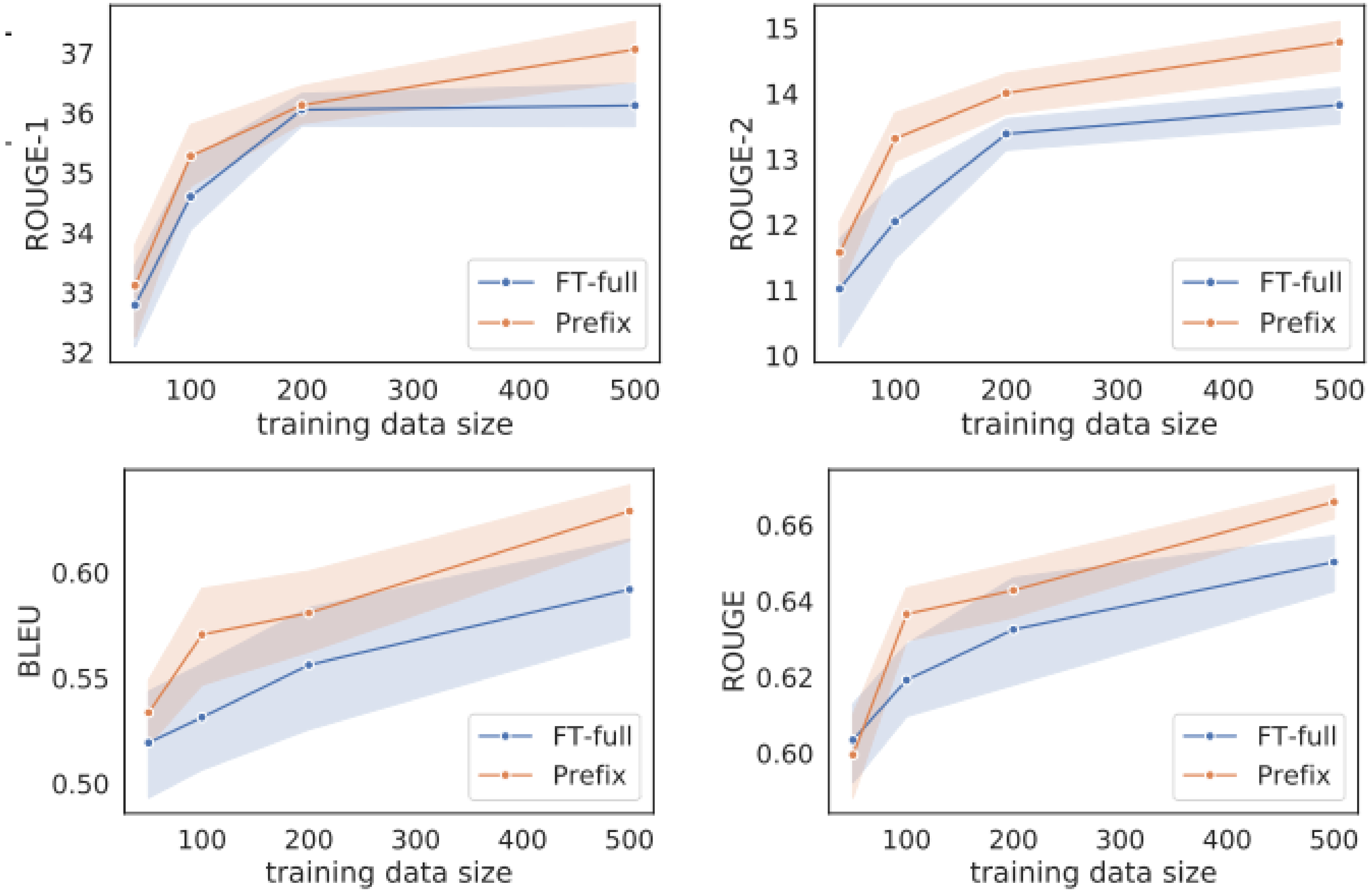
Prefix parameters

$\mathrm{R}^{|\mathrm{P}_{\mathrm{idx}}| \times \dim(h_i)}$ P_θ



	E2E					WebNLG									DART					
	BLEU	NIST	MET	R-L	CIDEr	BLEU			MET			TER ↓			BLEU	MET	TER ↓	Mover	BERT	BLEURT
						S	U	A	S	U	A	S	U	A						
GPT-2$_{\text{MEDIUM}}$																				
FT-FULL	68.8	8.71	46.1	71.1	2.43	**64.7**	26.7	45.7	**0.46**	0.30	0.38	**0.33**	0.78	0.54	46.2	**0.39**	**0.46**	**0.50**	**0.94**	**0.39**
FT-TOP2	68.1	8.59	46.0	70.8	2.41	53.6	18.9	36.0	0.38	0.23	0.31	0.49	0.99	0.72	41.0	0.34	0.56	0.43	0.93	0.21
ADAPTER(3%)	68.9	8.71	46.1	71.3	**2.47**	60.5	**47.9**	54.8	0.43	**0.38**	**0.41**	0.35	**0.46**	**0.39**	45.2	0.38	**0.46**	**0.50**	**0.94**	**0.39**
ADAPTER(0.1%)	66.3	8.41	45.0	69.8	2.40	54.5	45.1	50.2	0.39	0.36	0.38	0.40	0.46	0.43	42.4	0.36	0.48	0.47	**0.94**	0.33
PREFIX(0.1%)	**70.3**	**8.82**	**46.3**	**72.1**	2.46	62.9	45.3	**55.0**	0.44	0.37	**0.41**	0.35	0.51	0.42	**46.4**	0.38	**0.46**	**0.50**	**0.94**	**0.39**
GPT-2$_{\text{LARGE}}$																				
FT-FULL	68.5	8.78	46.0	69.9	2.45	**65.3**	43.1	55.5	**0.46**	0.38	**0.42**	**0.33**	0.53	0.42	**47.0**	**0.39**	0.46	**0.51**	**0.94**	**0.40**
Prefix	**70.3**	**8.85**	**46.2**	**71.7**	**2.47**	63.4	**47.7**	**56.3**	0.45	**0.39**	**0.42**	0.34	**0.48**	**0.40**	46.7	**0.39**	**0.45**	**0.51**	**0.94**	**0.40**
SOTA	68.6	8.70	45.3	70.8	2.37	63.9	52.8	57.1	0.46	0.41	0.44	-	-	-	-	-	-	-	-	-

Table 2: Metrics (higher is better, except for TER) for table-to-text generation on E2E (left), WebNLG (middle) and DART (right). With only 0.1% parameters, Prefix-tuning outperforms other lightweight baselines and achieves a comparable performance with fine-tuning. The best score is boldfaced for both GPT-2$_{\text{MEDIUM}}$ and GPT-2$_{\text{LARGE}}$.

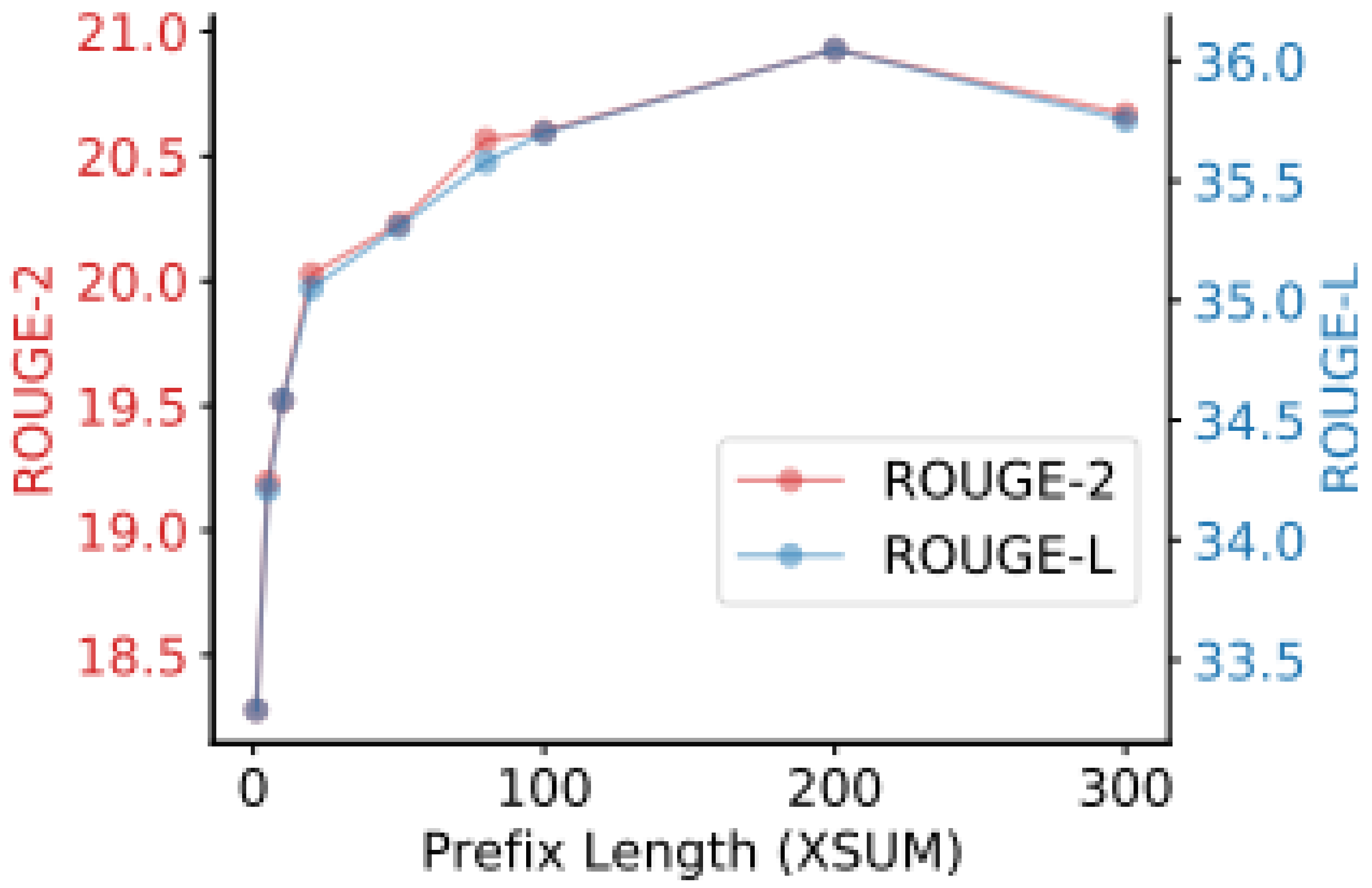


	R-1 ↑	R-2 ↑	R-L ↑
FT-FULL(Lewis et al., 2020)	45.14	22.27	37.25
PREFIX(2%)	43.80	20.93	36.05
PREFIX(0.1%)	42.92	20.03	35.05

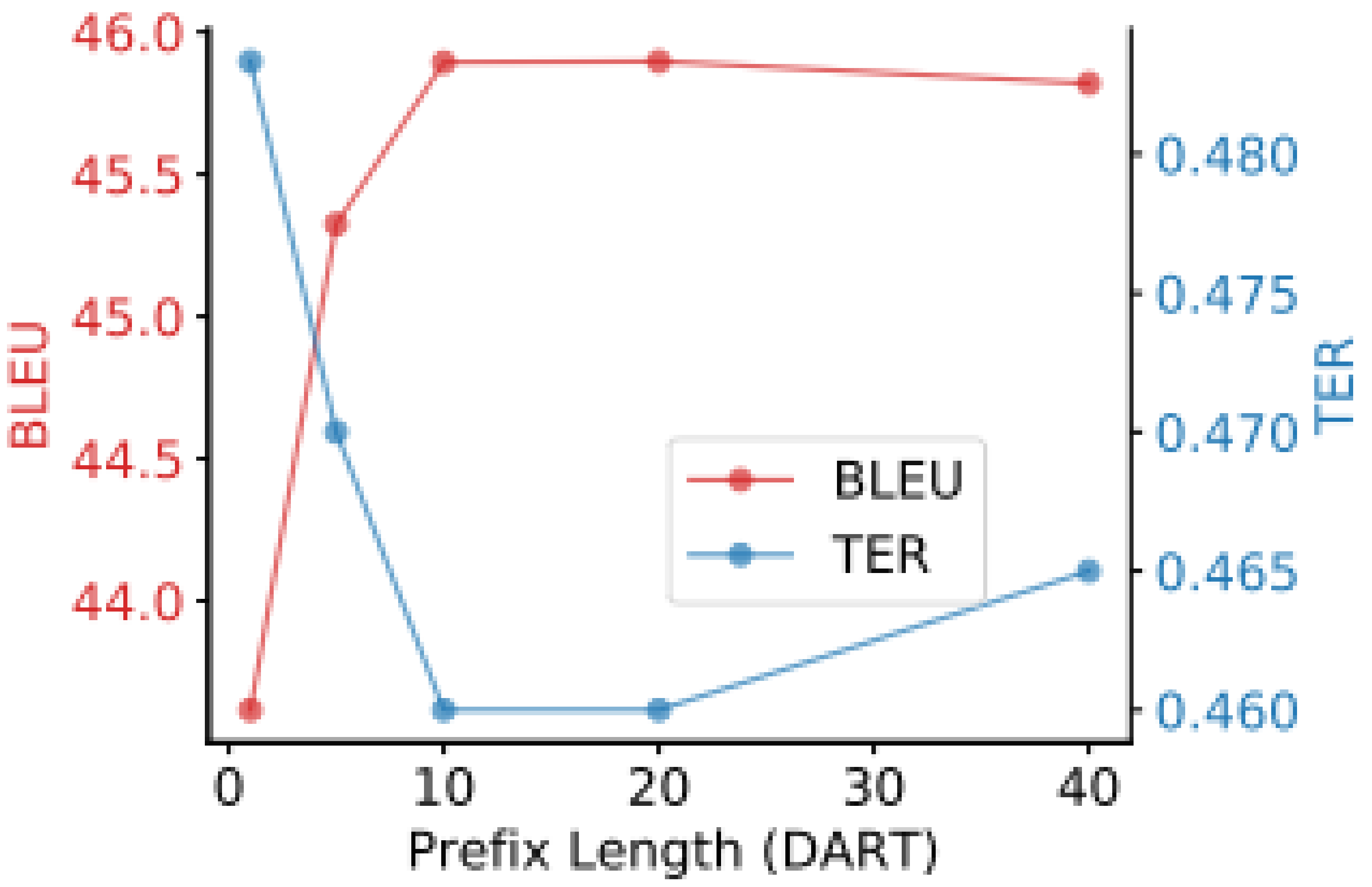
Table 3: Performance of methods on the XSUM summarization dataset. Prefix-tuning slightly underperforms fine-tuning in the full-data regime.

	news-to-sports			within-news		
	R-1 ↑	R-2 ↑	R-L ↑	R-1 ↑	R-2 ↑	R-L ↑
FT-FULL	38.15	15.51	30.26	39.20	16.35	31.15
PREFIX	39.23	16.74	31.51	39.41	16.87	31.47

Table 4: Extrapolation performance on XSUM. Prefix-tuning outperforms fine-tuning on both news-to-sports and within-news splits.



ROUGE-2
ROUGE-L
21.0
20.5
20.0
19.5
19.0
18.5
36.0
35.5
35.0
34.5
34.0
33.5
0
100
200
300
Prefix Length (XSUM)

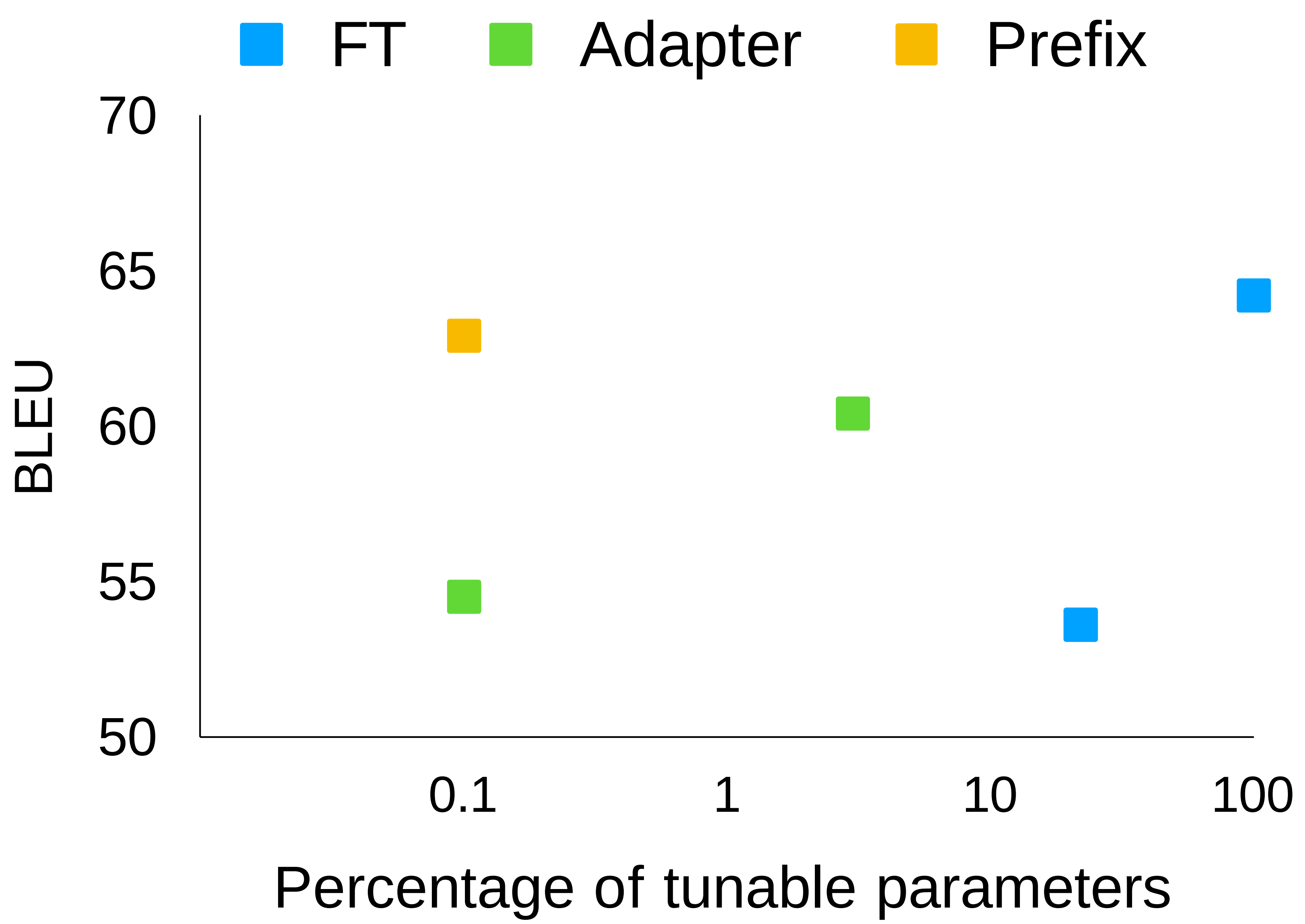


BLEU
TER
46.0
45.5
45.0
44.5
44.0
0.480
0.475
0.470
0.465
0.460
0
10
20
30
40
Prefix Length (DART)

[3] The WebNLG challenge: Generating text from RDF data —Gardent et. al. 2017

Table-to-text

dataset	WebNLG [3]
domain	train: 9 categories test: 9+5 categories
size	22K



Example

x : [Alan Tudyk, starring, Big Hero 6], [Steven T Segle, creator, Baymax], [Big Hero 6, series, Baymax]

y : Baymax is a character who appeared in Big Hero 6 starring Alan Tudyk. It was created by Steven T Seagle.

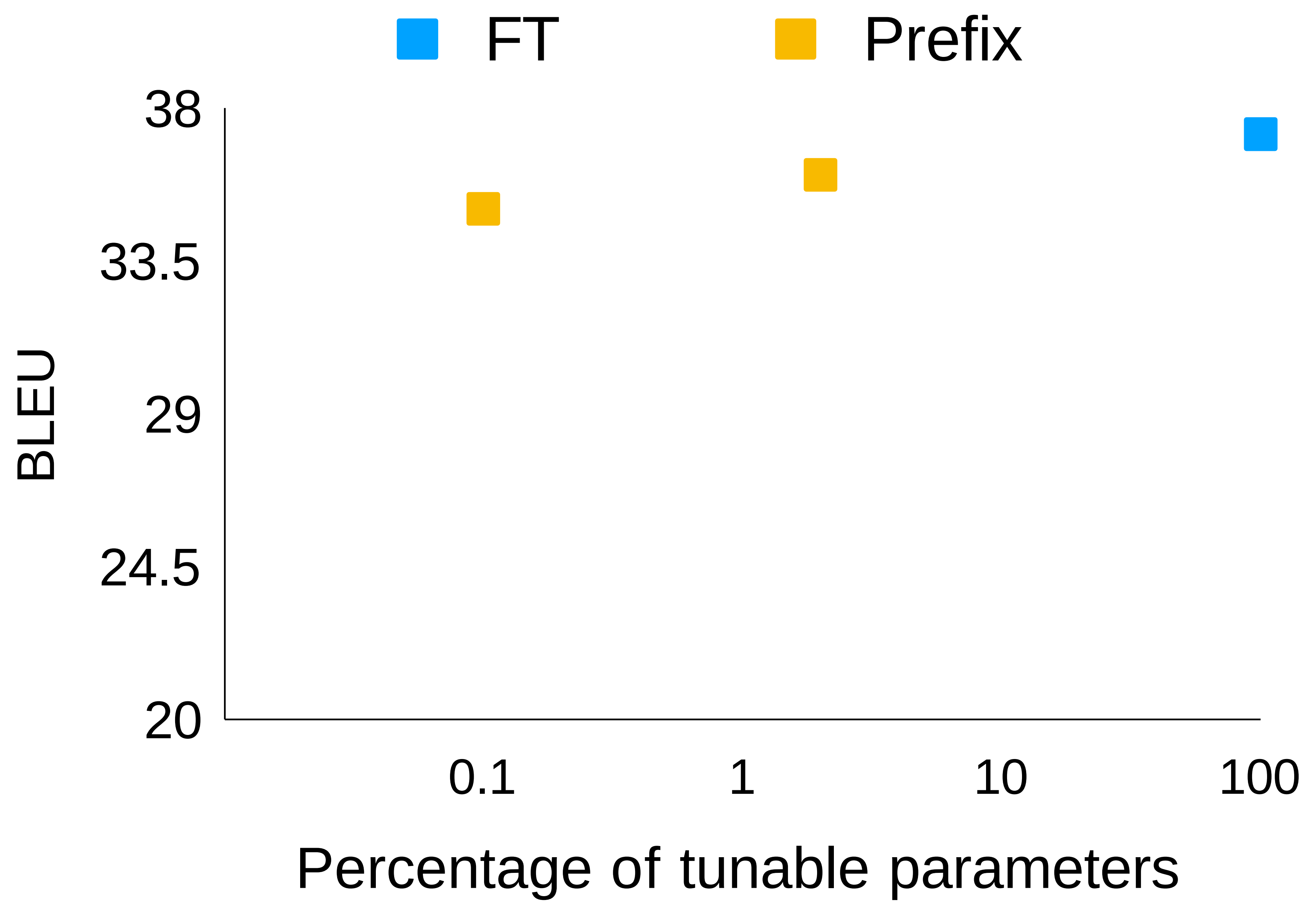
FT: Full fine-tuning with 100% tunable parameters
FT (22%): Fine-tune the top two layers, around 22%
Adapter (3%): Adapter-tuning with 3% tunable parameters
Adapter (0.1%): Adapter-tuning with 0.1% tunable parameters
Prefix (0.1%): Prefix-tuning with 0.1% tunable parameters

Takeaways:
1. Prefix-tuning is an effective and space-efficient method to adapt GPT-2 to table-to-text generation.
2. More parameter-efficient than adapter-tuning, significantly reducing parameters while improving generation quality.

[5] Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization —Narayan et. al. 2018

Summarization

Dataset	XSUM [5]
Domain	news
size	225K



x: Article: Scientists at University College London discovered people tend to think that their hands are wider and their fingers are shorter than they truly are.They say the confusion may lie in the way the brain receives information from different parts of the body. Distorted perception may dominate in some people, leading to body image problems ... [ignoring 308 words] could be very motivating for people with eating disorders to know that there was a biological explanation for their experiences, rather than feeling it was their fault."

y: Summary: The brain naturally distorts body image - a finding which could explain eating disorders like anorexia, say experts.

FT: Full fine-tuning with 100% tunable parameters
Prefix (2%): Prefix-tuning with 2% tunable parameters
Prefix (0.1%): Prefix-tuning with 0.1% tunable parameters

Takeaway:
With 2% parameters, prefix-tuning obtains slightly lower performance than fine-tuning.

Extrapolation

Example:

Trained on 9 categories

Astronaut, University, Monument, Building, ComicsCharacter, Food, Airport, SportsTeam, City, and WrittenWork

extrapolates

Test on 5 unseen categories

Athlete, Artist, MeanOfTransportation, CelestialBody, Politician

Extrapolation

Trained on 9 categories

Astronaut, University, Monument, Building, ComicsCharacter, Food, Airport, SportsTeam, City, and WrittenWork

extrapolates

Test on 5 unseen categories

Athlete, Artist, MeanOfTransportation, CelestialBody, Politician

Example:

x : [103_Colmore_Row | architect | John_Madin]
[John_Madin | birthPlace | Birmingham]
[Birmingham | leaderName | Andrew_Mitchell]

y : John Madin was born in Birmingham (with Andrew Mitchell as a key leader) and became an architect, designing 103 Colmore Row.

Extrapolation

Trained on 9 categories

Astronaut, University, Monument, Building, ComicsCharacter, Food, Airport, SportsTeam, City, and WrittenWork

extrapolates

Test on 5 unseen categories

Athlete, Artist, MeanOfTransportation, CelestialBody, Politician

Example:

x : [103_Colmore_Row | architect | John_Madin]
[John_Madin | birthPlace | Birmingham]
[Birmingham | leaderName | Andrew_Mitchell]

y : John Madin was born in Birmingham (with Andrew Mitchell as a key leader) and became an architect, designing 103 Colmore Row.

x : [Albennie_Jones | genre | Rhythm_and_blues]
[Albennie_Jones | birthPlace | Errata,_Mississippi]
[Rhythm_and_blues | derivative | Disco]

y : Albennie Jones, born in Errata, Mississippi, is a performer of rhythm and blues, of which disco is a derivative.

Extrapolation

Trained on 9 categories

Astronaut, University, Monument, Building, ComicsCharacter, Food, Airport, SportsTeam, City, and WrittenWork

extrapolates

Test on 5 unseen categories

Athlete, Artist, MeanOfTransportation, CelestialBody, Politician

Example:

x: [103_Colmore_Row | architect | John_Madin]
[John_Madin | birthPlace | Birmingham]
[Birmingham | leaderName | Andrew_Mitchell]

y: John Madin was born in Birmingham (with Andrew Mitchell as a key leader) and became an architect, designing 103 Colmore Row.

x: [Albennie_Jones | genre | Rhythm_and_blues]
[Albennie_Jones | birthPlace | Errata,_Mississippi]
[Rhythm_and_blues | derivative | Disco]

y: Albennie Jones, born in Errata, Mississippi, is a performer of rhythm and blues, of which disco is a derivative.

Extrapolation

Trained on 9 categories

Astronaut, University, Monument, Building, ComicsCharacter, Food, Airport, SportsTeam, City, and WrittenWork

extrapolates

Test on 5 unseen categories

Athlete, Artist, MeanOfTransportation, CelestialBody, Politician

Example:

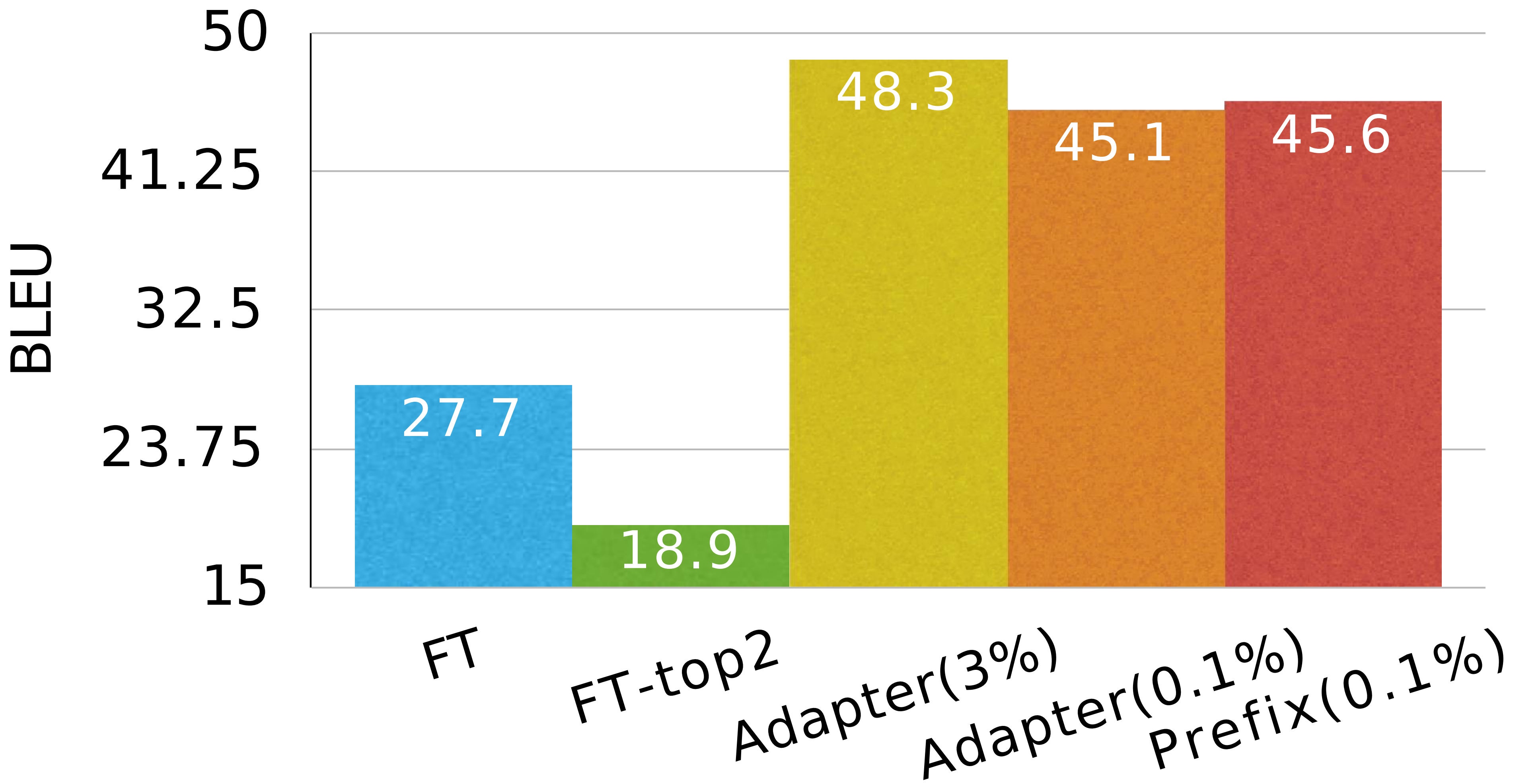
x: [103_Colmore_Row | architect | John_Madin]
[John_Madin | birthPlace | Birmingham]
[Birmingham | leaderName | Andrew_Mitchell]

y: John Madin was born in Birmingham (with Andrew Mitchell as a key leader) and became an architect, designing 103 Colmore Row.

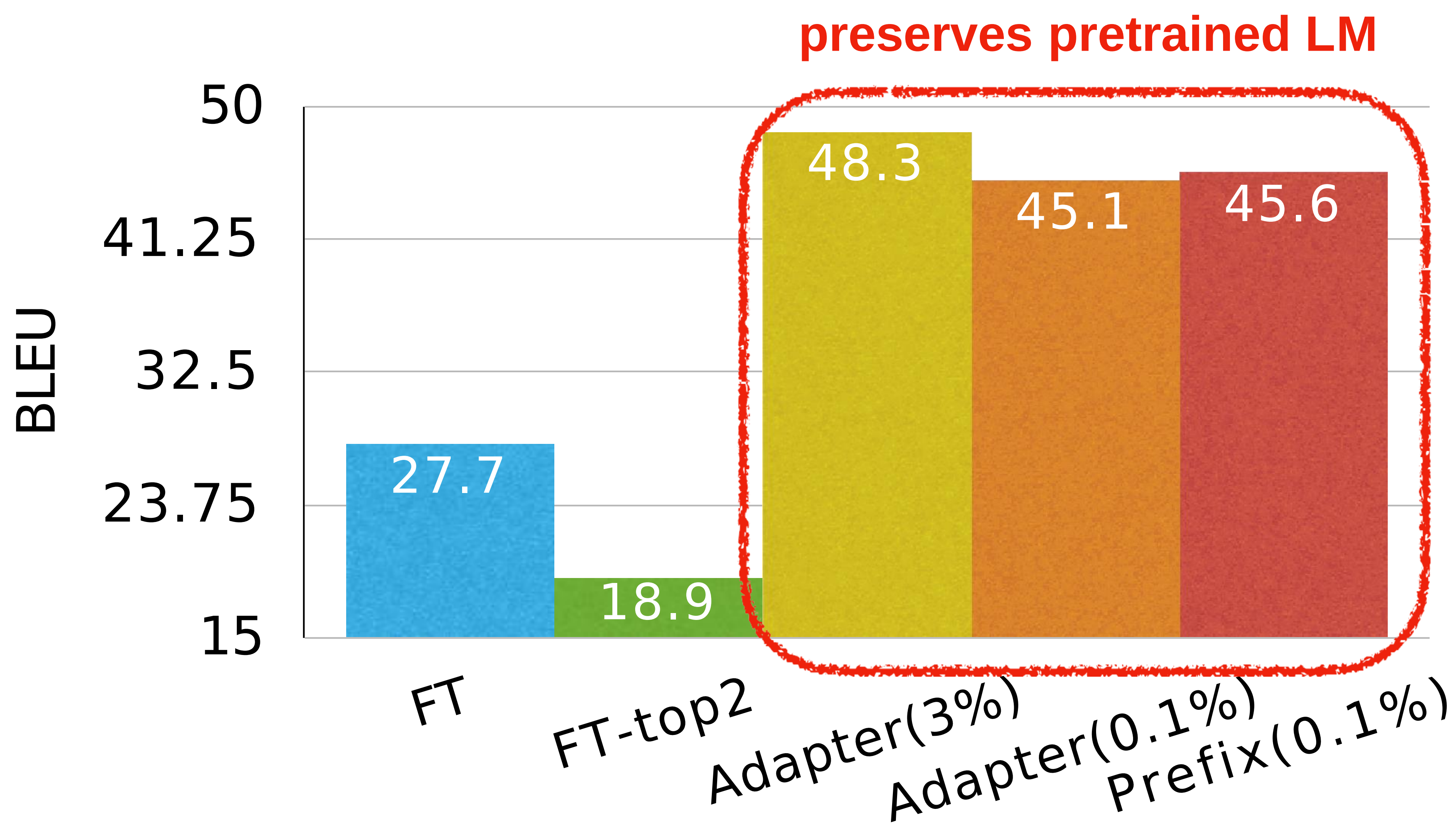
x: [Albennie_Jones | genre | Rhythm_and_blues]
[Albennie_Jones | birthPlace | Errata,_Mississippi]
[Rhythm_and_blues | derivative | Disco]

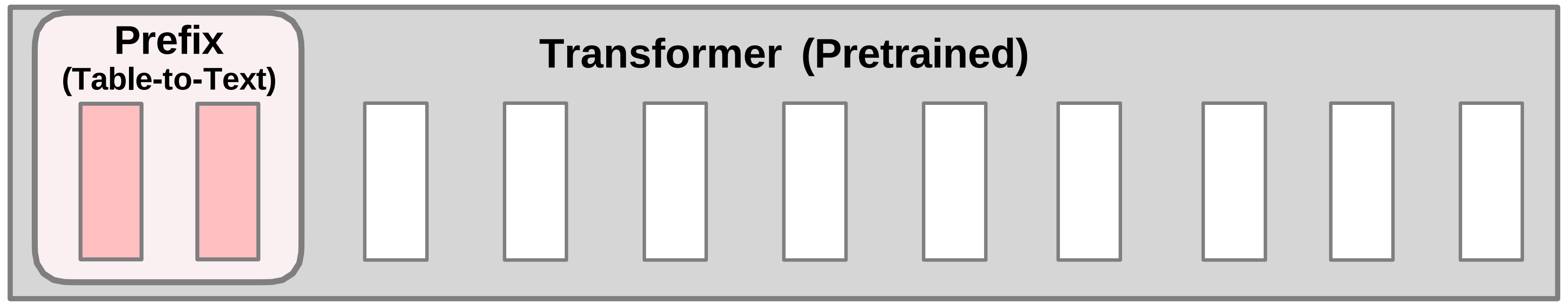
y: Albennie Jones, born in Errata, Mississippi, is a performer of rhythm and blues, of which disco is a derivative.

Extrapolation

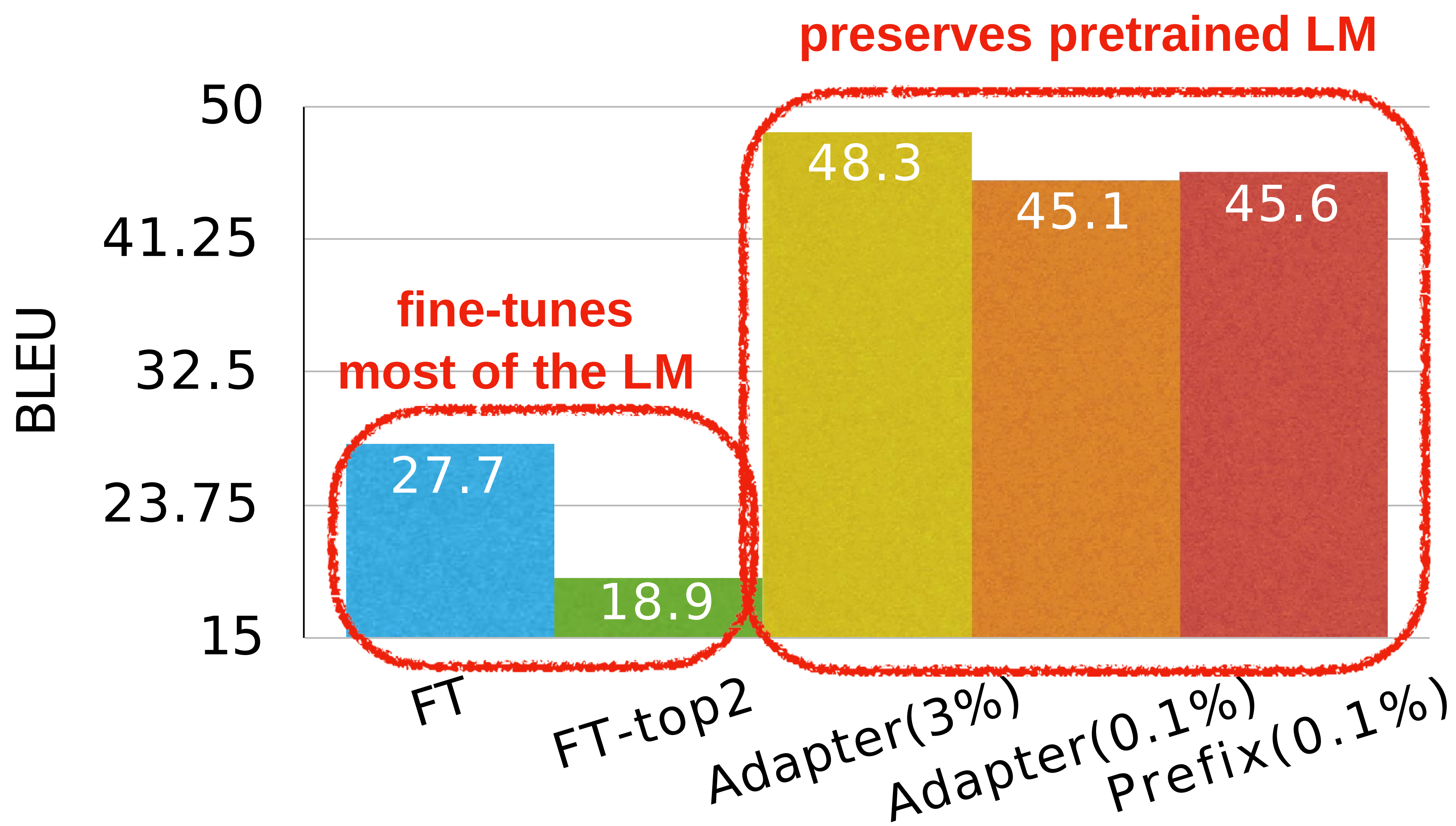


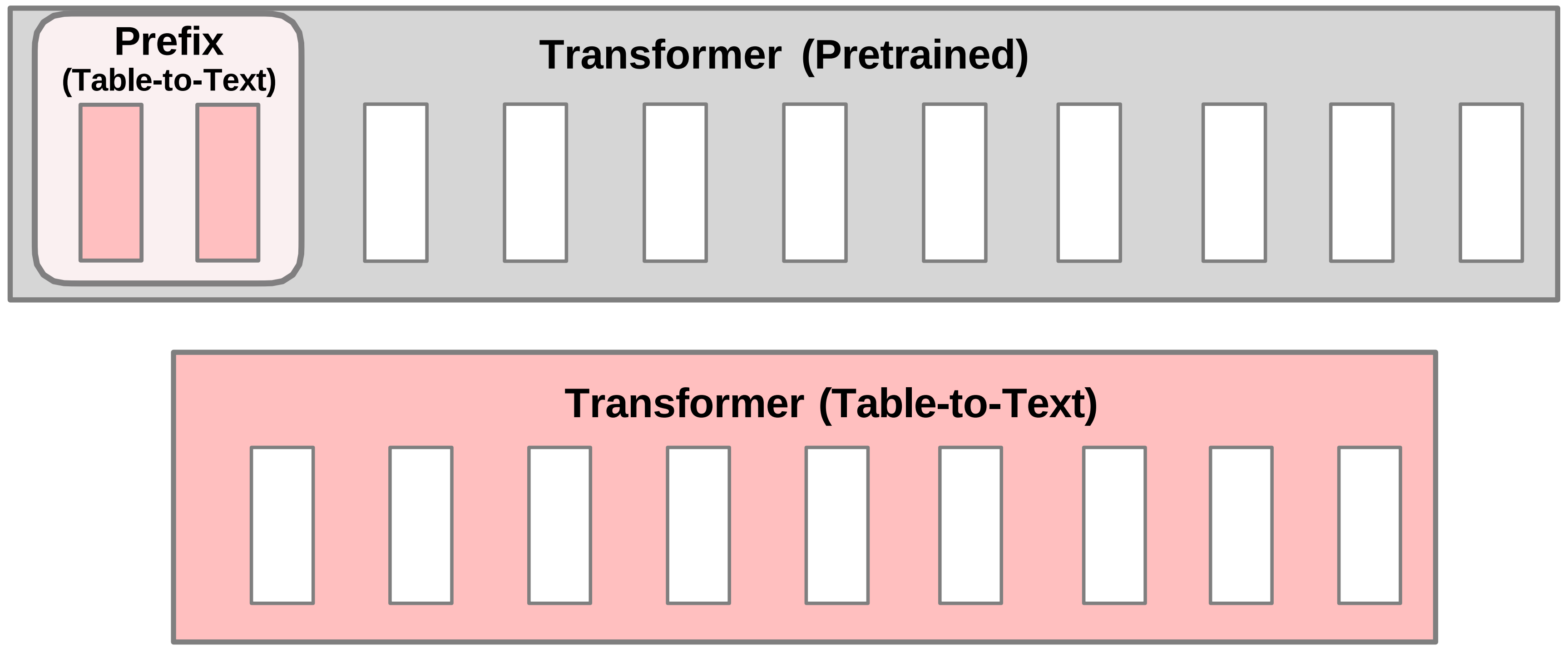
Extrapolation



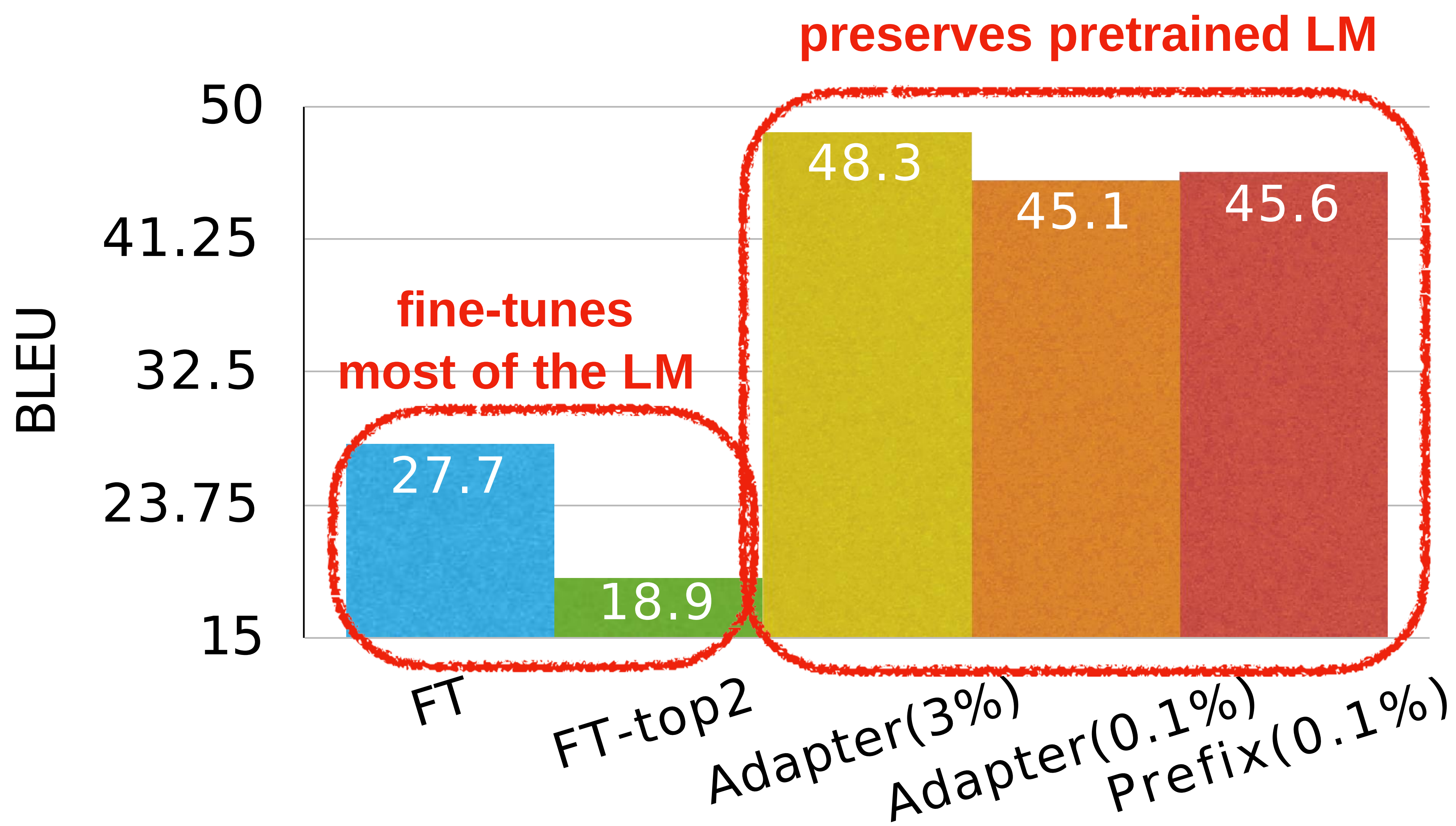


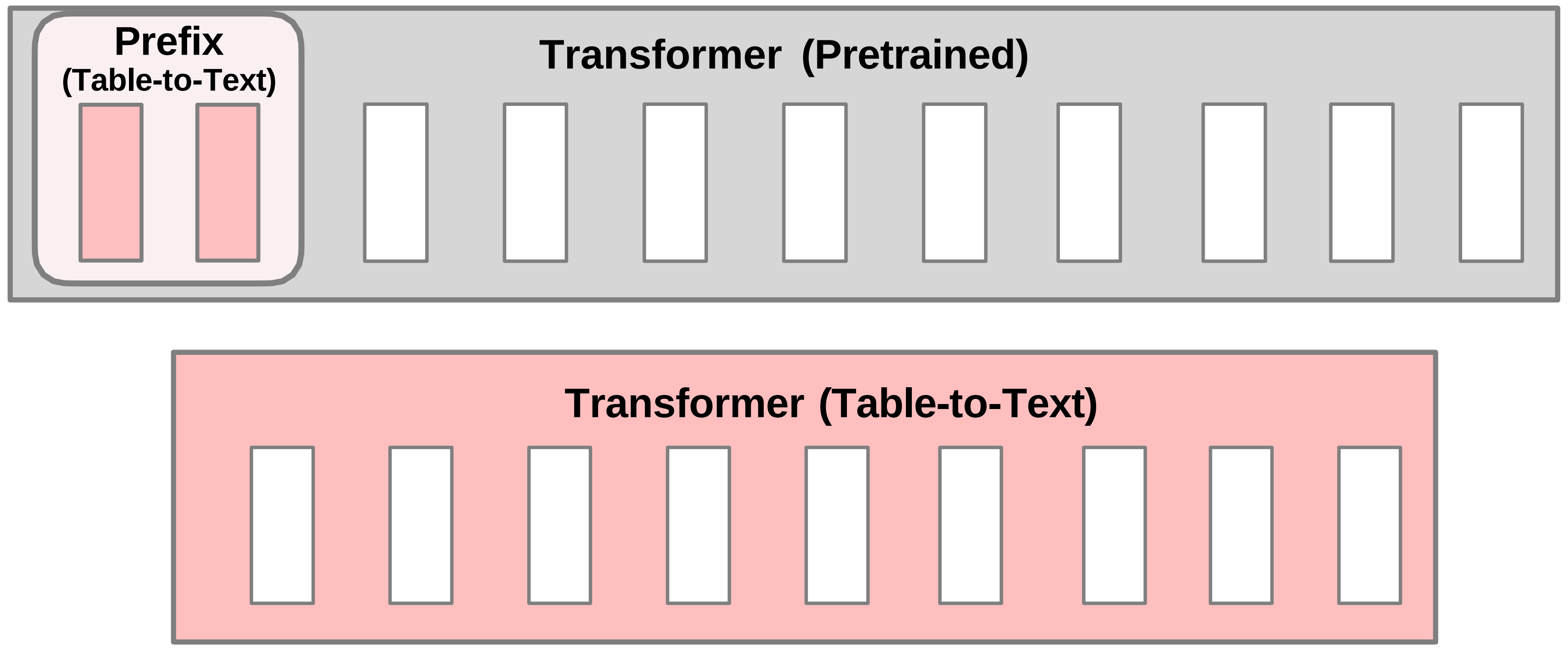
Extrapolation





Extrapolation





Takeaway:
Methods that preserve the pretrained LM achieves better extrapolation than those that fine-tunes most of the LM.

Demo available at here

	WebNLG								
	BLEU			MET			TER ↓		
	S	U	A	S	U	A	S	U	A
GPT2 Medium									
No Finetune	0.00	0.00	0.00	0.03	0.03	0.03	1.28	1.48	1.37
Prefix	62.77	44.95	54.73	0.45	0.37	0.41	0.34	0.50	0.42