

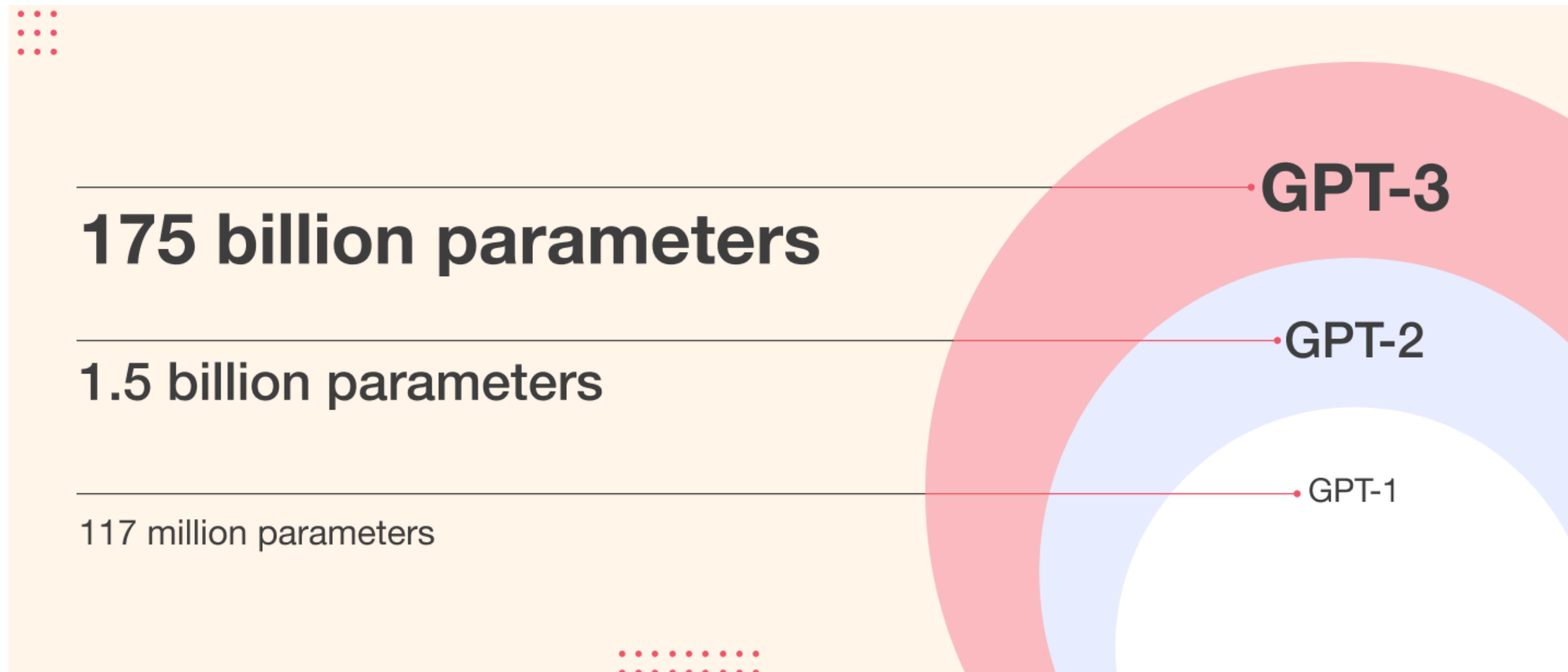
Small Language Models

Mengzhou Xia

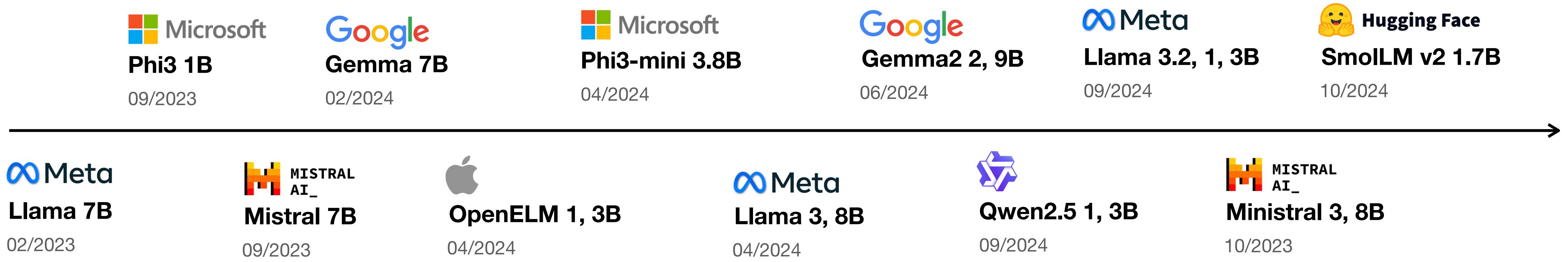
Princeton Language and Intelligence (PLI)

Princeton University

Large language models get larger



Small language models



SLM: language models <10B

Fierce competition in small model regime

Mostly open-weight

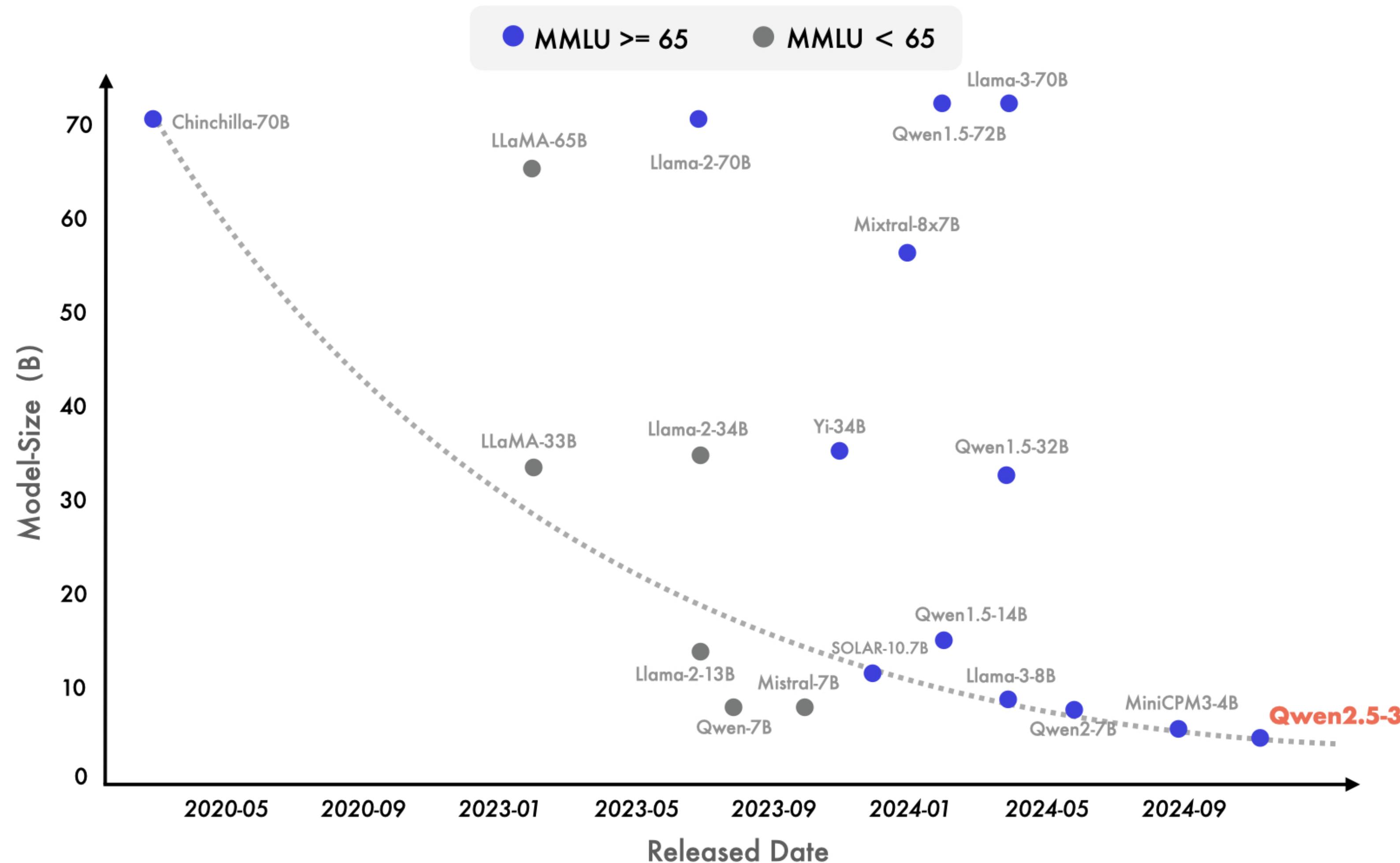
Small language models are popular

meta-llama/Llama-3.2-1B-Instruct	488	Downloads last month 1,297,753
meta-llama/Llama-3.2-3B-Instruct	498	Downloads last month 992,933
Qwen/Qwen2.5-1.5B-Instruct	115	Downloads last month 40,938,448
google/gemma-2-2b	398	Downloads last month 13,561,972

Small language models have been downloaded extensively.

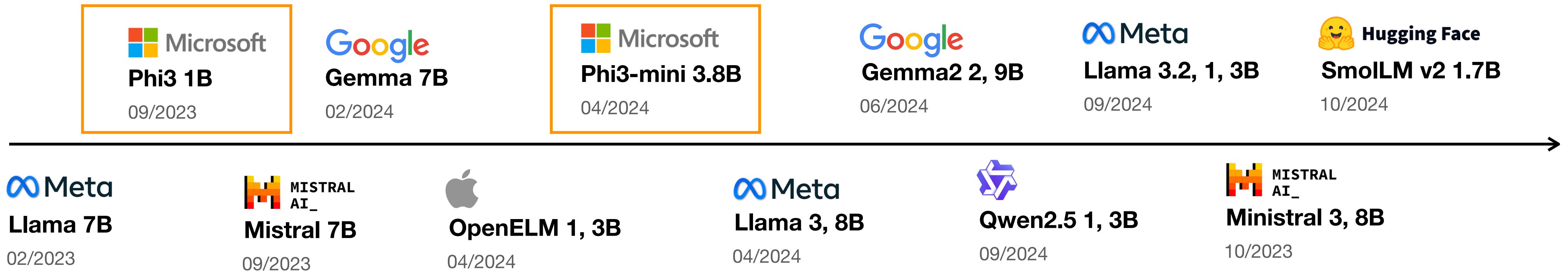
- Efficient to run and fine-tune
- Fit for on-device use
- Suited for being adaptation into specialized models

Small Language models are getting stronger



The model size that can achieve MMLU ≥ 65 gradually decreases.

Key technique: High quality data

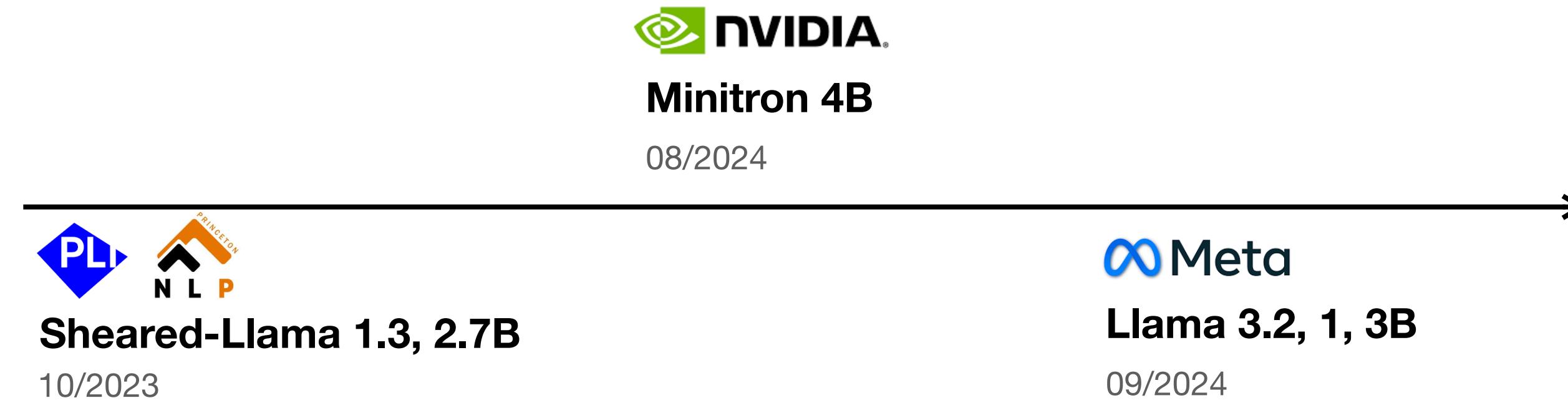


- **High quality data** (Phi series)

we have a small dataset is focused on “text-book quality educational content”, we can learn the task better, even with a smaller model

Usually have no details :(
Key assets of model developers

Key technique: Pruning



- **Pruning (Sheared-Llama, Llama 3.2, Minitron)**

Structured pruning removes groups of model weights from large models, producing smaller ones that require significantly less compute than training from scratch.

Key technique: Distillation



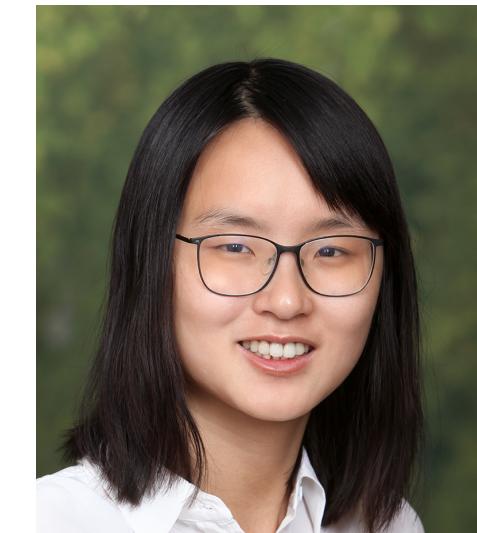
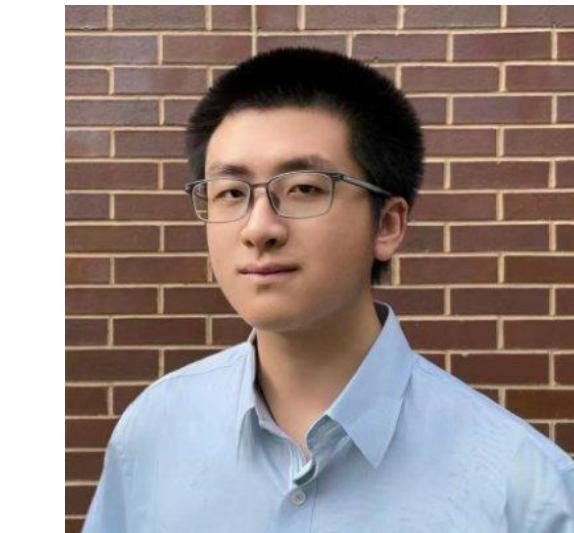
- **Distillation (Gemma2, Llama 3.2, Minitron)**

Train small models using signals from the larger models.

This Lecture

- Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning
Structured pruning to produce small and competitive LMs effectively.
- Gemma 2: Improving Open Language Models at a Practical Size
SoTA <10B model.
- MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies
Many training techniques and ablations.

Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning (ICLR'24)



Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, Danqi Chen.
arXiv:2310.06694

How to train a smaller LLM?

- **Background:** Moderate-sized language models are booming

[2023/2/7]	LLaMA-1	(7-70B parameters, 1T tokens)
[2023/5/5]	INCITE	(3B-70B parameters, 800B tokens)
[2023/6/7]	OpenLLaMA-v1	(3B-13B parameters, 1T tokens)
[2023/6/7]	OpenLLaMA-v2	(3B-13B parameters, 1T tokens)
[2023/7/18]	LLaMA-2	(7B-70B parameters, 2T tokens)
[2023/9/27]	Mistral-7B	(7B parameters, ?? tokens)

Recent efforts: TinyLlama, Phi-1, Phi-2, OLMo, MiniCPM, Gemma ..

Yet the creation of such models at any scale is still expensive

How to train a smaller LLM?

- **Background:** Moderate-sized language models are booming

 princeton-nlp/Sheared-LLaMA-1.3B

 Text Generation • Updated Jan 23 • ↓ 61.4k • ❤ 80

 princeton-nlp/Sheared-LLaMA-2.7B

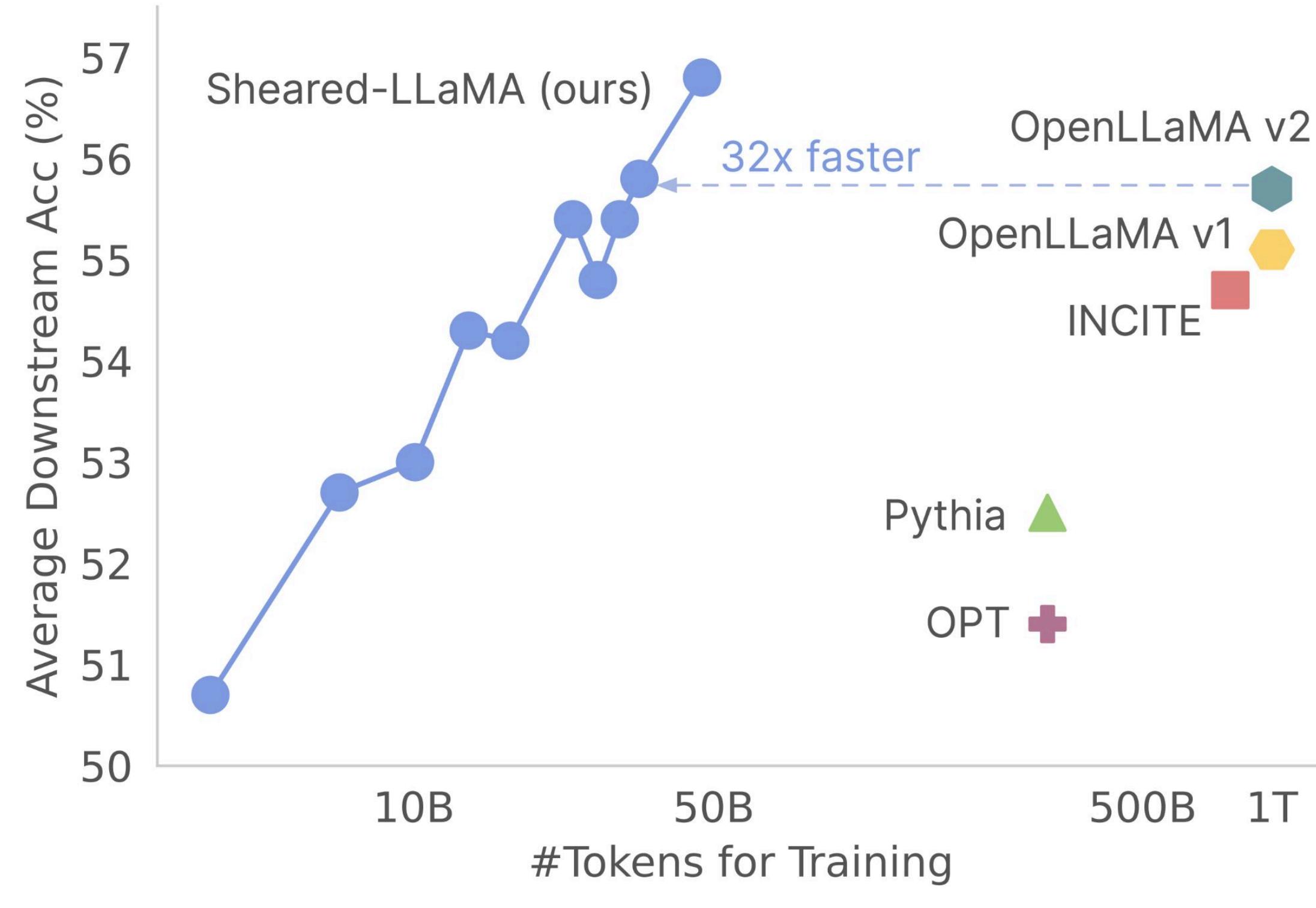
 Text Generation • Updated Jan 23 • ↓ 6.68k • ❤ 53

- **Key message:** Instead of training one from scratch, we can structurally prune a model from existing, larger pre-trained LLMs!



ShearedLLaMA

Sheared LLaMA



Model	Pre-training Data	#Tokens
LLaMA1	LLaMA data	1T
LLaMA2	<i>Unknown</i>	2T
OPT	OPT data ⁵	300B
Pythia	The Pile	300B
INCITE-Base	RedPajama	800B
OpenLLaMA v1	RedPajama	1T
OpenLLaMA v2	OpenLLaMA data ⁶	1T
TinyLlama	TinyLlama data ⁷	3T
Sheared-LLaMA	RedPajama	50B

- Structured pruning is always a more cost-effective way to build smaller models.

How do we get there? LLM Shearing

Stage 1: Pruning \implies **Stage 2:** Continued pre-training

- **Stage 1:** What final structure do we want?

Targeted structured pruning

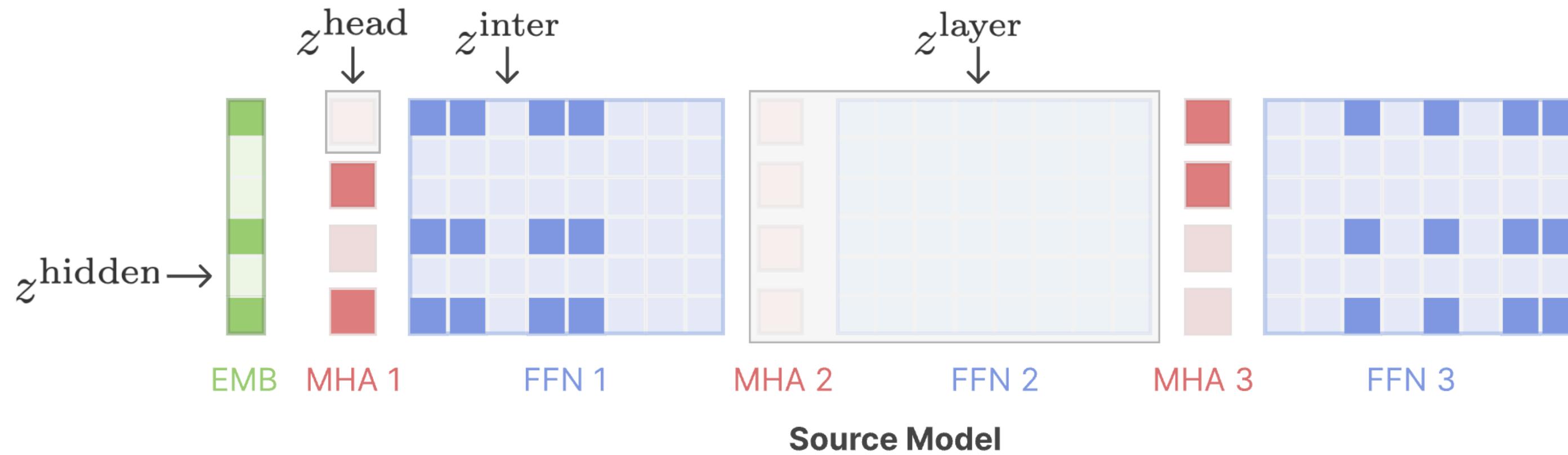
- **Pre-specify** model architecture following existing pre-trained models
- **Search** such a substructure end-to-end to maximize performance
- The output is exactly like a smaller, dense model, while previous structured pruning approaches produce non-uniform layer configurations

- **Stage 2:** How to use pre-training data more efficiently?

Dynamic batch loading

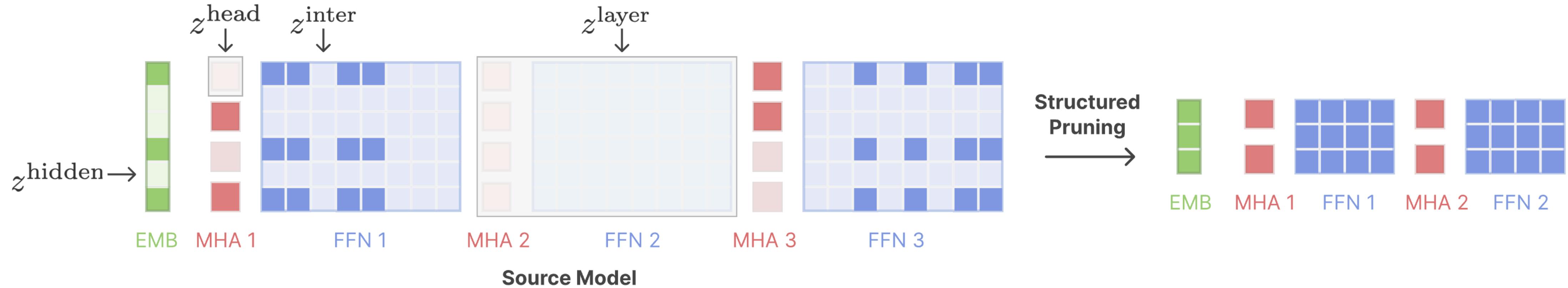
- Some domains recover faster (StackExchange), and some domains recover more slowly (C4)
- Use models to provide signal for adjusting **domain weights of data mixture**

Targeted structured pruning

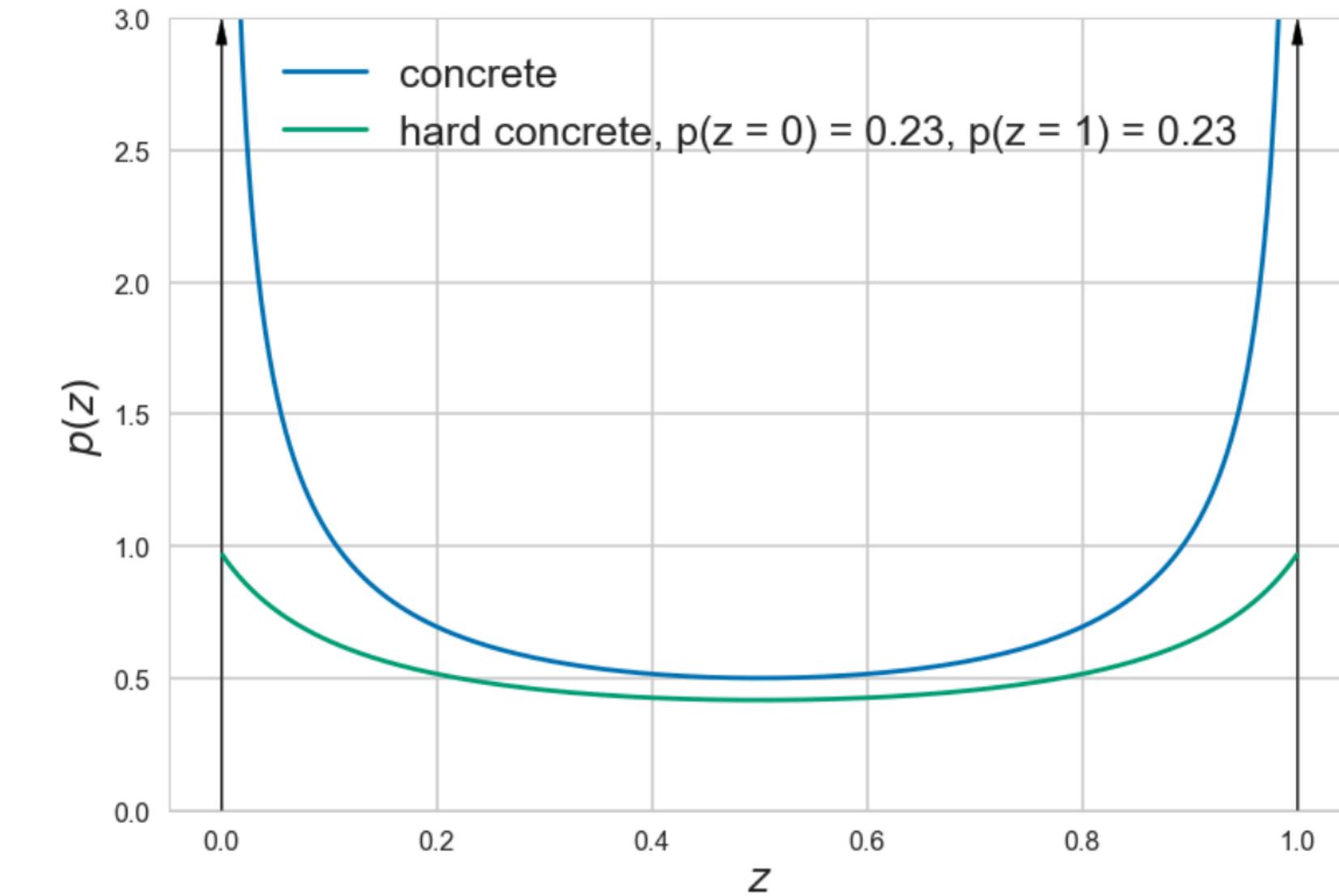


- All the pruning decisions are modeled as **masking variables** (both global and layer-wise);
- We use l_0 regularization (Louizos et al., 2017) and parameterize the masks as hard-concrete distributions

Targeted structured pruning

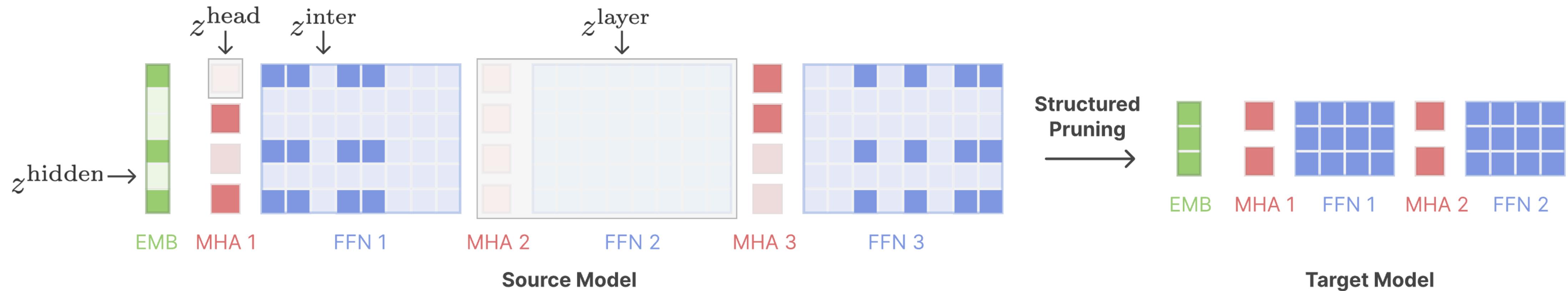


- All the pruning decisions are modeled as **masking variables** (both global and layer-wise);
- We use l_0 regularization (Louizos et al., 2017) and parameterize the masks as hard-concrete distributions



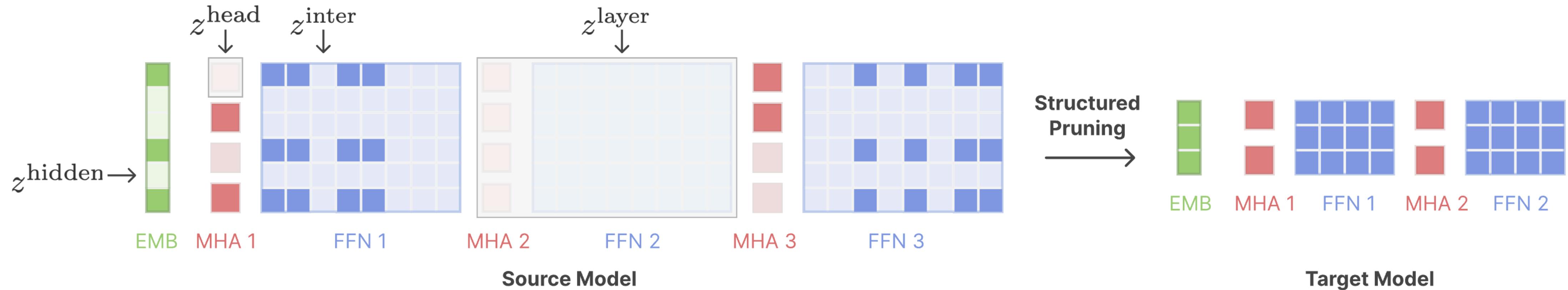
Allow z to be exactly 0.

Targeted structured pruning



- All the pruning decisions are modeled as **masking variables** (both global and layer-wise); We use l_0 regularization (Louizos et al., 2017) and parameterize the masks as hard-concrete distributions
- We use **Lagrangian multipliers** to impose constraints on the final structure of pruned models (Wang et al., 2020; Xia et al., 2022)

Targeted structured pruning



$$\tilde{\mathcal{L}}^{\text{head}}(\lambda, \phi, z) = \lambda^{\text{head}} \cdot \left(\sum z^{\text{head}} - H_{\mathcal{T}} \right) + \phi^{\text{head}} \cdot \left(\sum z^{\text{head}} - H_{\mathcal{T}} \right)^2$$

Target number of heads

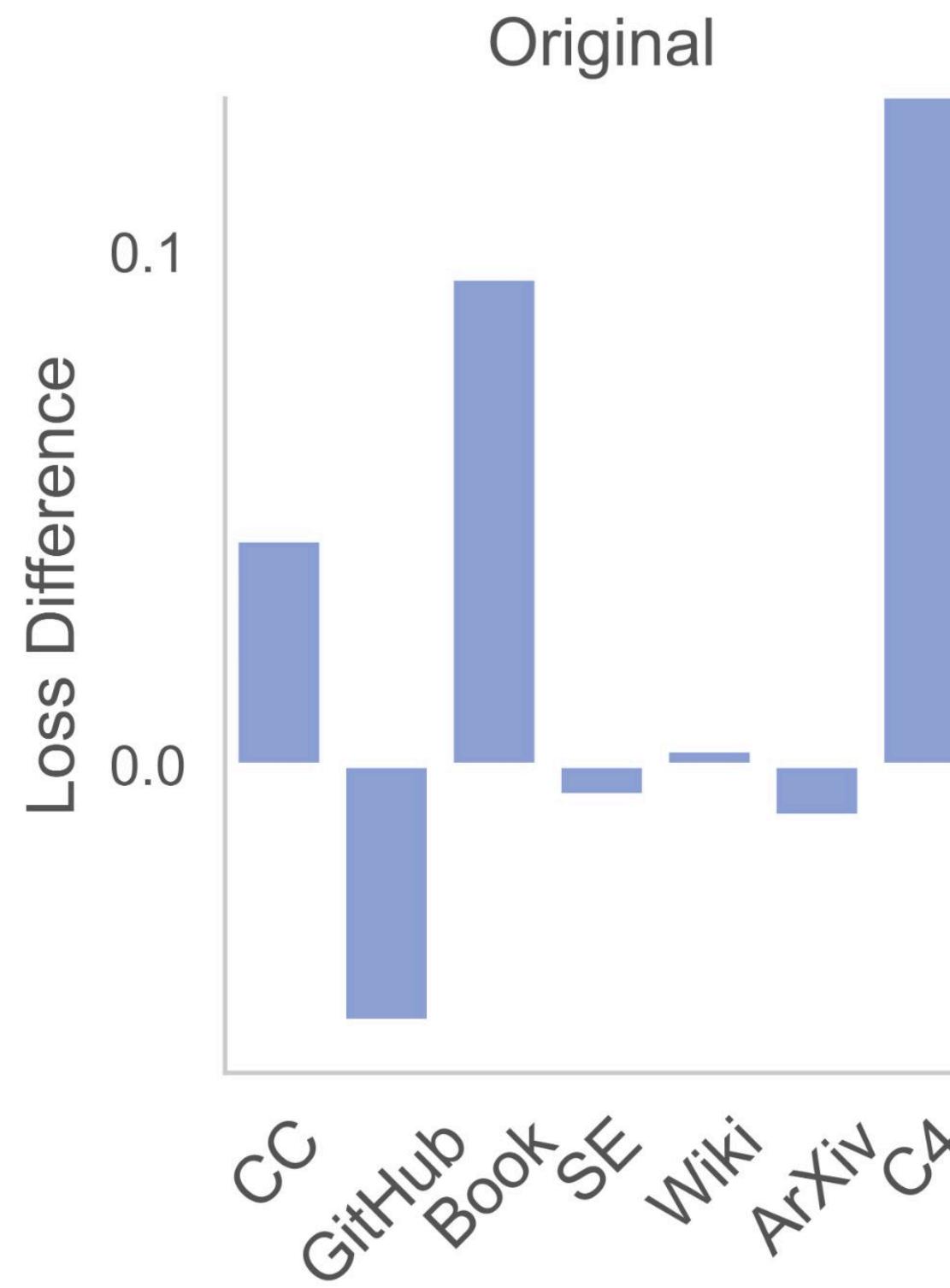
$$\mathcal{L}_{\text{prune}}(\theta, z, \lambda, \phi) = \mathcal{L}(\theta, z) + \sum_{j=1}^{L_S} \tilde{\mathcal{L}}_j^{\text{head}} + \sum_{j=1}^{L_S} \tilde{\mathcal{L}}_j^{\text{int}} + \tilde{\mathcal{L}}^{\text{layer}} + \tilde{\mathcal{L}}^{\text{hidden}}$$

The objective can be optimized end-to-end..

Dynamic batch loading

Problem: If we just continue pre-training the pruned model with an off-the-shelf pre-training dataset e.g., RedPajama...

- The loss difference is extremely imbalanced across different domains



Loss difference: compared to pre-training a model from scratch - I will discuss it soon!

Inefficient use of pre-training data

Dynamic batch loading

- **Key idea:** Load more data for domains where the loss reduction is slow

$$\Delta_t[i] \leftarrow \max \{\ell_t[i] - \ell_{\text{ref}}[i], 0\} \quad i: \text{domain index}$$

$$\alpha_t = \log(w_{t-m}) + \Delta_t \quad \text{adjust domain weights after } m \text{ steps}$$

$$w_t = \frac{\exp(\alpha_t)}{\sum_i \exp(\alpha_t[i])}.$$

- Inspired by (Xie et al., 2023) but no proxy model required. Almost no overhead!
- Where does **reference loss** come from?
 - We either estimate the reference loss by scaling law (Hoffman et al., 2022) on LLaMA-2 series, or just use the source model to calculate validation loss

Downstream task performance

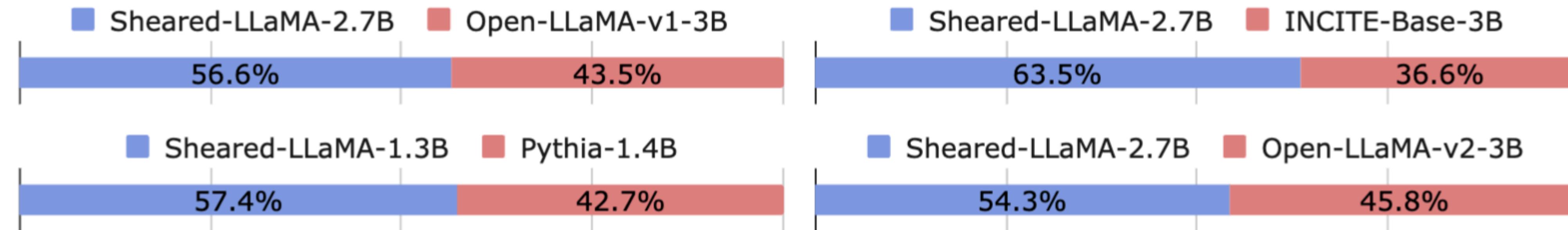
Model (#tokens for training)	Performance
LLaMA2-7B (2T) [†]	64.6
OPT-1.3B (300B) [†]	48.2
Pythia-1.4B (300B) [†]	48.9
Sheared-LLaMA-1.3B (50B)	51.0
OPT-2.7B (300B) [†]	51.4
Pythia-2.8B (300B) [†]	52.5
INCITE-3B (800B)	54.7
Open-LLaMA-3B-v1 (1T)	55.1
Open-LLaMA-3B-v2 (1T) [†]	55.7
Sheared-LLaMA-2.7B (50B)	56.7

- 11 downstream tasks across reasoning, reading comprehension, knowledge intensive tasks
- Sheared-LLaMA outperforms existing LMs with much less compute

*: TinyLLaMA-1.1B (3T): 50.0

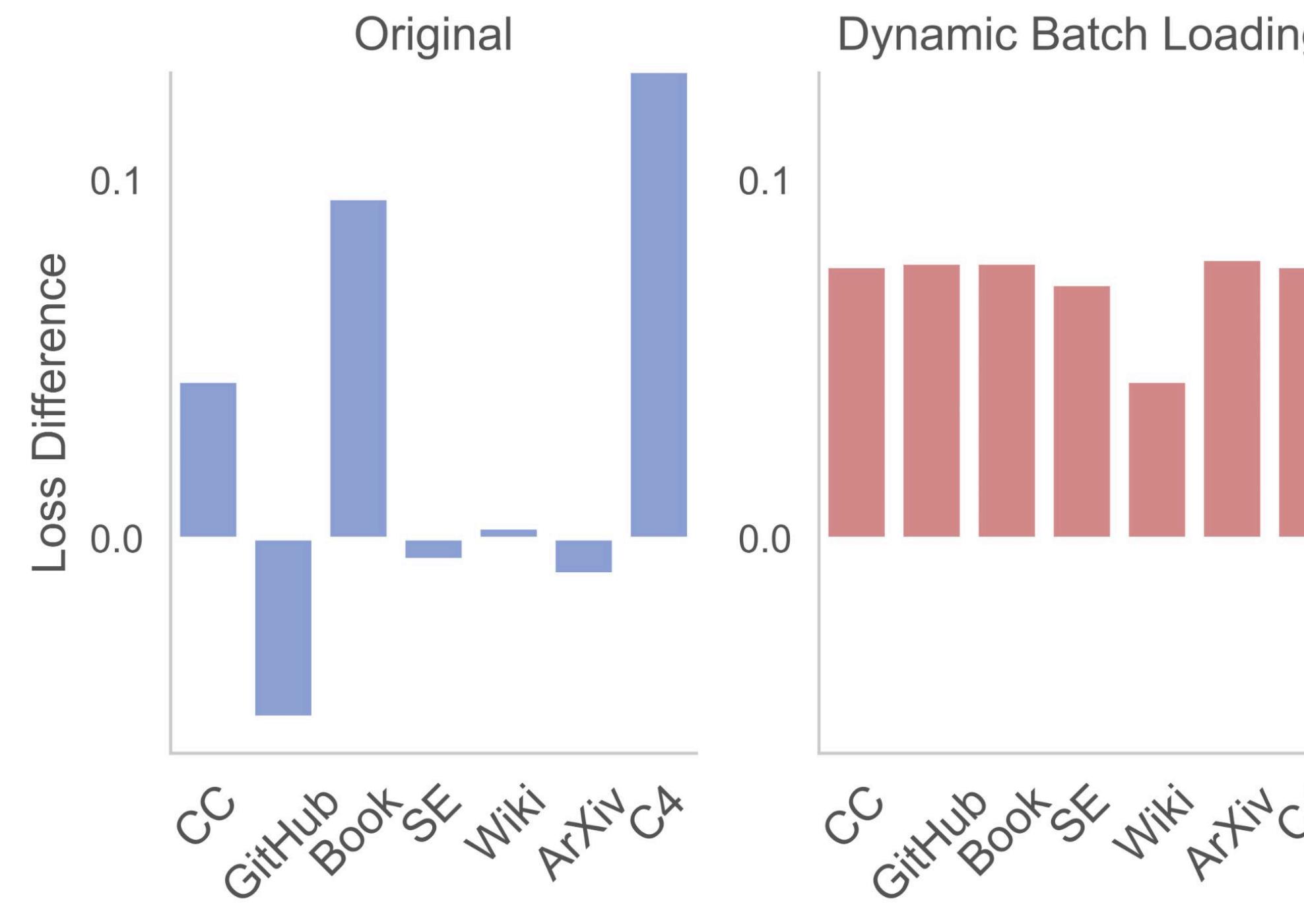
Sheared LLaMA “chat” models

- Fine-tuning and evaluation on ShareGPT, with GPT-4 as the model evaluator
- Sheared-LLaMA has the ability to generate long-form, coherent responses to human instructions

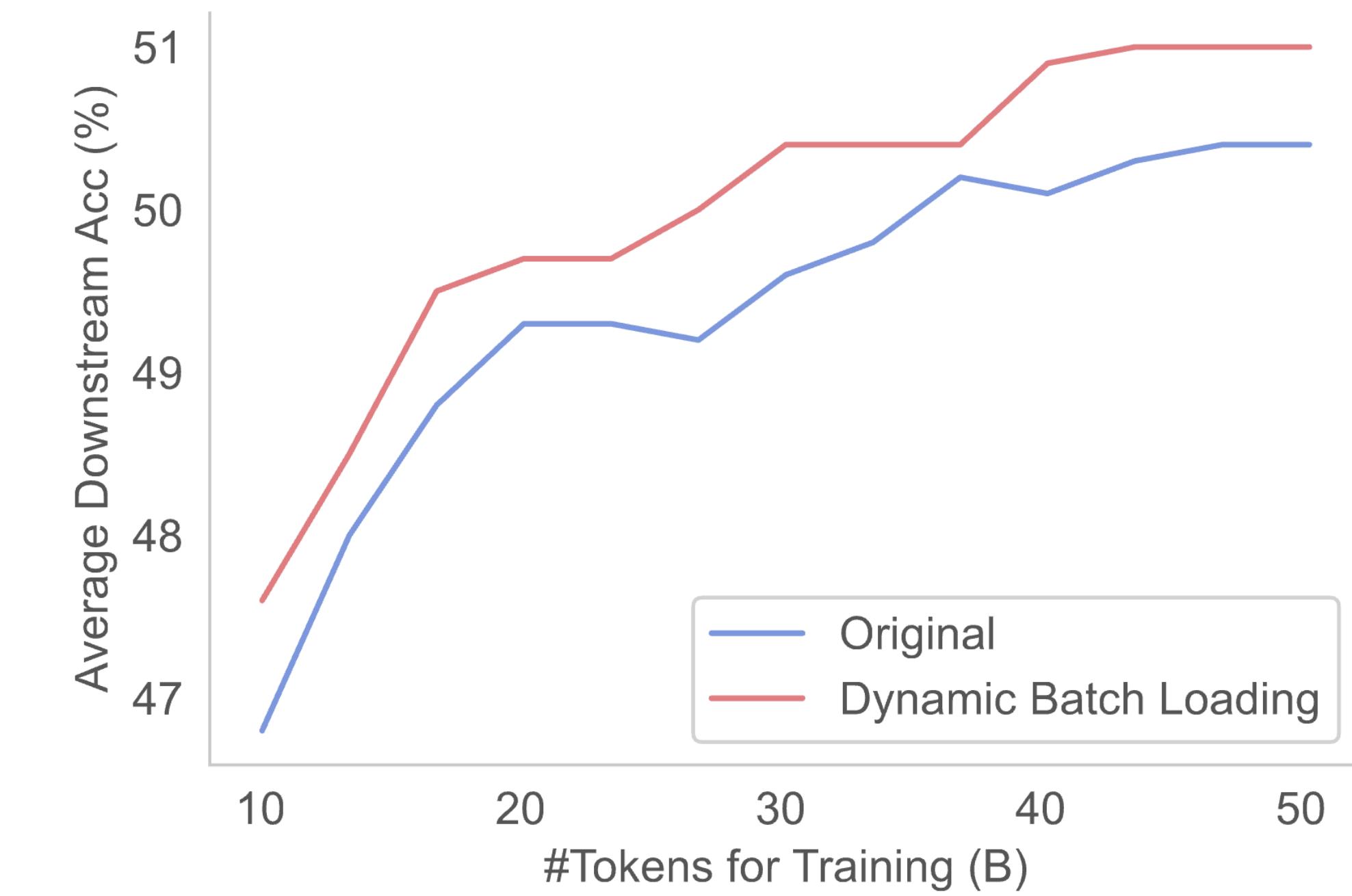


Try out these models on HuggingFace!

Impact of dynamic batch loading

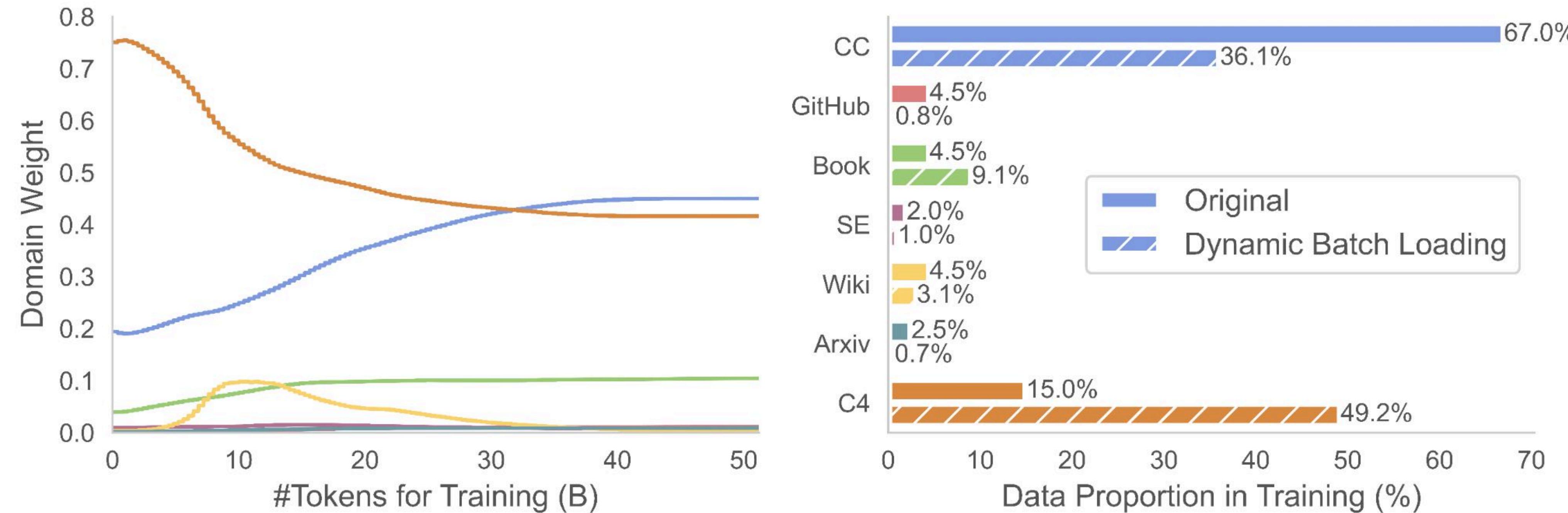


Consistent loss differences across domains



Better downstream performance

The data use across different domains



We used more data in C4 and Book, and less in any other domains!

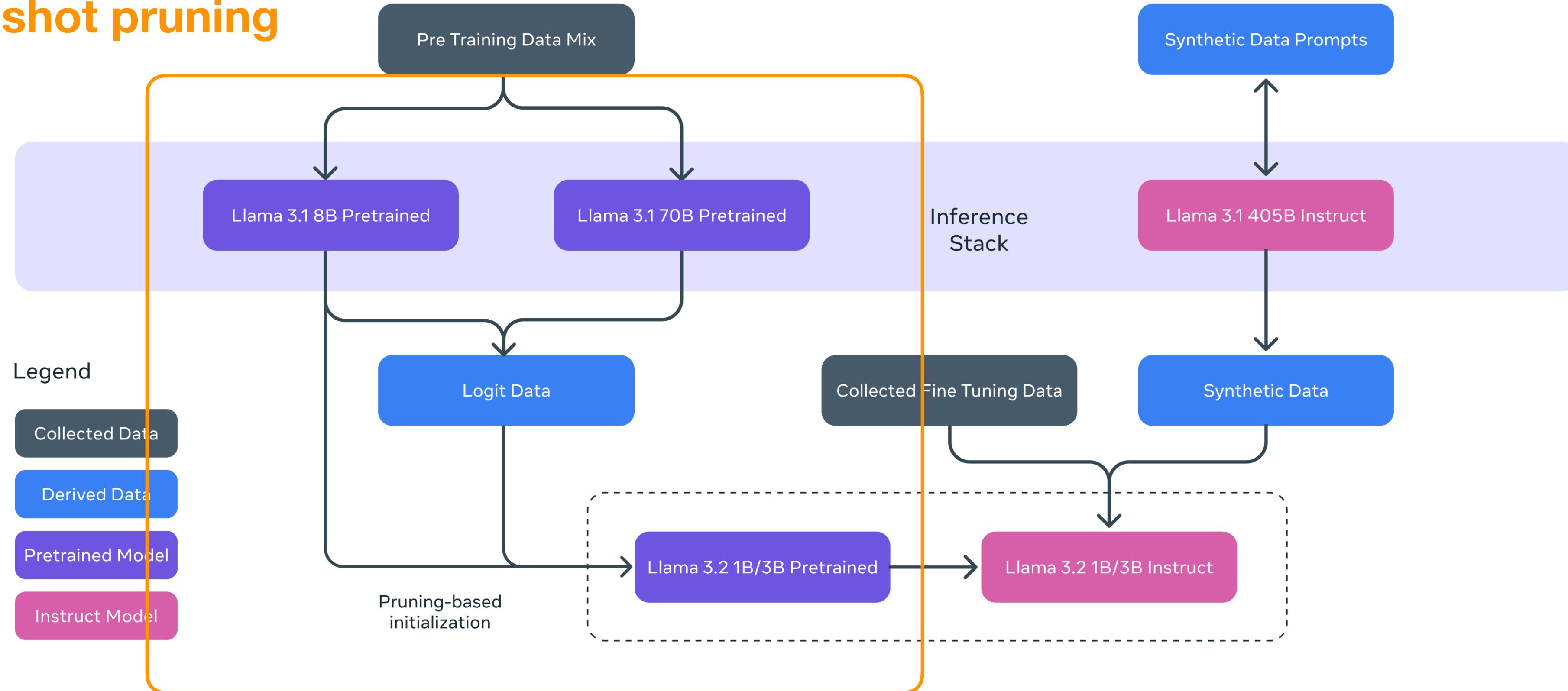
Summary

- Pruning and continued pre-training existing models is far more **cost-effective** than training models from scratch
- It highlights the importance of **domain mixture for pruning LLMs**; dynamic batch loading is a general approach!
- Limitations: Complex to implement, restricted by source data quality
- **Community efforts:**
 - People have been using ShearedLLaMA models in many use cases: data selection, draft model for speculative decoding, contrastive decoding, video captioning..
 - More pruned models e.g., VinaLLaMA, Llama3.2, Minitron

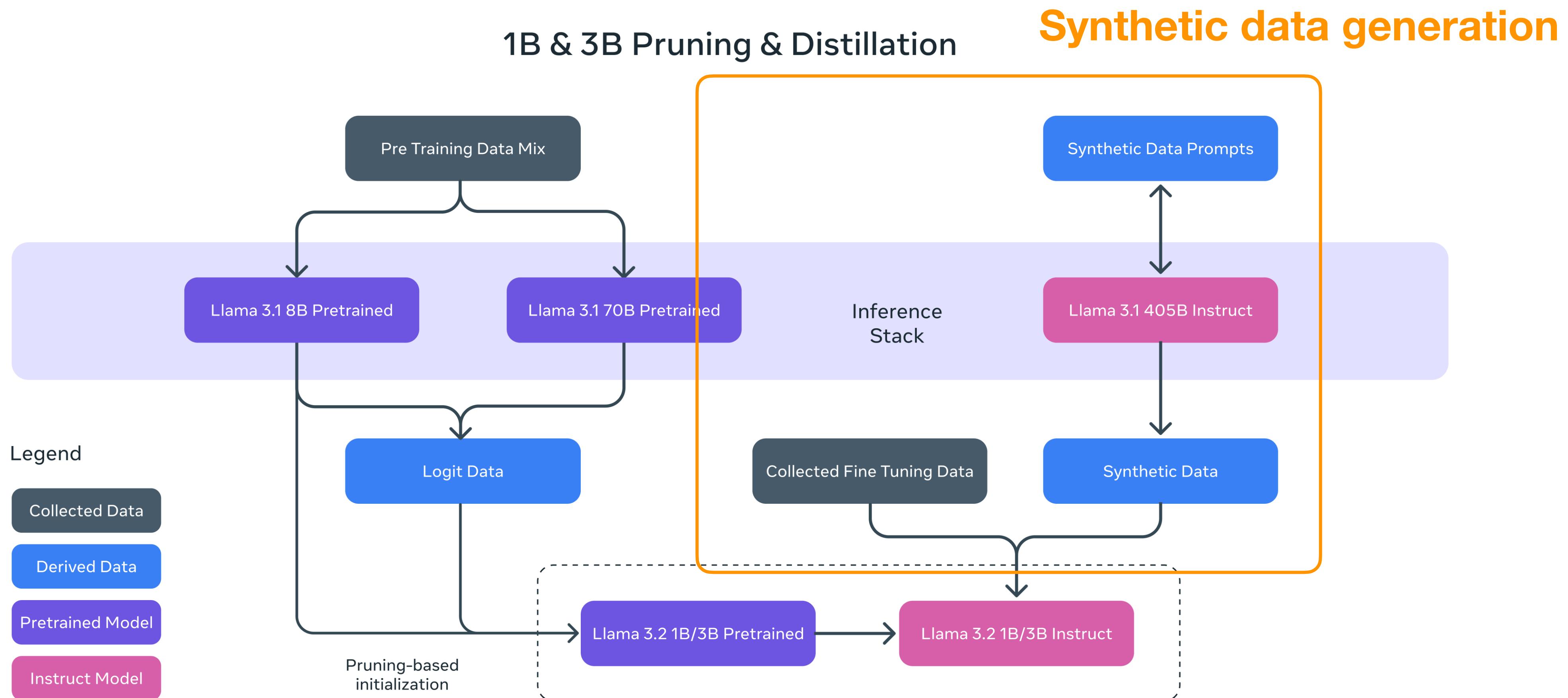
Llama 3.2 — Pruning and Distillation

1B & 3B Pruning & Distillation

One-shot pruning



Llama 3.2 — Pruning and Distillation



Gemma 2: Improving Open Language Models at a Practical Size

Gemma Team, Google DeepMind

Training Details

- Train Gemma 2 27B for 13T tokens
- Distill it to 2B and 9B

From Scratch

$$L(\theta) = - \sum_{t=1}^T \log p(y_t | y_{<t}, x; \theta)$$

Distillation

$$L_{\text{distill}}(\theta_S) = - \sum_{t=1}^T p_T(y_t | y_{<t}, x; \theta_T) \log p_S(y_t | y_{<t}, x; \theta_S)$$

	LLaMA-3 70B	Qwen1.5 32B	Gemma-2 27B
MMLU	79.2	74.3	75.2
GSM8K	76.9	61.1	74.0
ARC-c	68.8	63.6	71.4
HellaSwag	88.0	85.0	86.4
Winogrande	85.3	81.5	83.7

- Gemma 27B is comparable to Llama3 70B

Evaluation

Benchmark	metric	Gemma-1 2B	Gemma-2 2B	Mistral 7B	LLaMA-3 8B	Gemma-1 7B	Gemma-2 9B	Gemma-2 27B
MMLU	5-shot	42.3	52.2	62.5	66.6	64.4	71.3	75.2
ARC-C	25-shot	48.5	55.7	60.5	59.2	61.1	68.4	71.4
GSM8K	5-shot	15.1	24.3	39.6	45.7	51.8	68.6	74.0
AGIEval	3-5-shot	24.2	31.5	44.0 [†]	45.9 [†]	44.9 [†]	52.8	55.1
DROP	3-shot, F1	48.5	51.2	63.8*	58.4	56.3	69.4	74.2
BBH	3-shot, CoT	35.2	41.9	56.0 [◦]	61.1 [◦]	59.0 [◦]	68.2	74.9
Winogrande	5-shot	66.8	71.3	78.5	76.1	79.0	80.6	83.7
HellaSwag	10-shot	71.7	72.9	83.0	82.0	82.3	81.9	86.4
MATH	4-shot	11.8	16.0	12.7	-	24.3	36.6	42.3
ARC-e	0-shot	73.2	80.6	80.5	-	81.5	88.0	88.6
PIQA	0-shot	77.3	78.4	82.2	-	81.2	81.7	83.2
SIQA	0-shot	49.7	51.9	47.0*	-	51.8	53.4	53.7
Boolq	0-shot	69.4	72.7	83.2*	-	83.2	84.2	84.8
TriviaQA	5-shot	53.2	60.4	62.5	-	63.4	76.6	83.7
NQ	5-shot	12.5	17.1	23.2	-	23.0	29.2	34.5
HumanEval	pass@1	22.0	20.1	26.2	-	32.3	40.2	51.8
MBPP	3-shot	29.2	30.2	40.2*	-	44.4	52.4	62.6
Average (8)		44.0	50.0	61.0	61.9	62.4	70.2	74.4
Average (all)		44.2	48.7	55.6	-	57.9	64.9	69.4

- Gemma2 2B >> Gemma1 2B, Gemma2 9B >> Llama3 8B

From Scratch vs. Distillation

From Scratch

$$L(\theta) = - \sum_{t=1}^T \log p(y_t | y_{<t}, x; \theta)$$

Setup:

- Student model: 2B model
- Teacher model: 7B
- 500B tokens

Distillation

$$L_{\text{distill}}(\theta_S) = - \sum_{t=1}^T p_T(y_t | y_{<t}, x; \theta_T) \log p_S(y_t | y_{<t}, x; \theta_S)$$

	from scratch	distilled
Average (3 bench.)	60.3	67.7

Distillation leads to better performance with **a fixed amount of tokens**.

a fixed amount of compute?

Impact of distillation w.r.t. model size

Setup:

- Student model: 200M, 400M, 1B
- Teacher model: 7B
- Evaluation metric: perplexity

	200M	400M	1B
from scratch	23	19	17
distilled (7B)	21	17	15

The gain from distillation remains across different student model sizes.

Other details of Gemma2

- Use a reward model that's much larger than policy in RLHF
- Use model merging after post-training
- Post-training helps coding much more than knowledge based task
- Strong ELO rating on LMSys, rated by real users

Model	2B		9B		27B	
	PT	IT	PT	IT	PT	IT
MMLU	52.2	56.1	71.3	72.3	75.2	76.2
MBPP	30.2	36.6	52.4	59.2	62.6	67.4

Model	Elo	95% CI	Open	Model	Elo	95% CI	Open
gpt-4o-2024-05-13	1286	+2 / -3	-	gemma-2-9b-it	1187	+3 / -5	+
gpt-4o-mini-2024-07-18	1279	+5 / -4	-	qwen2-72b-instruct	1187	+3 / -3	+
claude-3-5-sonnet	1271	+3 / -4	-	gpt-4-0314	1186	+2 / -3	-
gemini-advanced-0514	1266	+2 / -3	-	qwen1.5-110b-chat	1161	+3 / -3	+
llama-3.1-405b-instruct	1262	+8 / -7	+	mistral-large-2402	1157	+3 / -3	-
gemini-1.5-pro-api-0514	1261	+2 / -3	-	yi-1.5-34b-chat	1157	+4 / -3	-
gemini-1.5-pro-api-0409	1257	+3 / -3	-	reka-flash-21b-20240226	1155	+4 / -4	-
gpt-4-turbo-2024-04-09	1256	+2 / -3	-	llama-3-8b-instruct	1151	+2 / -3	+
gpt-4-1106-preview	1250	+3 / -3	-	command-r	1148	+3 / -3	+
claude-3-opus-20240229	1248	+2 / -2	-	claude-1	1148	+4 / -4	-
athene-70b-0725	1245	+8 / -6	+	mistral-medium	1147	+4 / -4	-
gpt-4-0125-preview	1245	+2 / -2	-	reka-flash-21b-20240226	1147	+3 / -4	-
llama-3.1-70b-instruct	1244	+8 / -9	+	qwen1.5-72b-chat	1147	+4 / -4	+
yi-large-preview	1239	+3 / -3	-	mixtral-8x22b-instruct-v0.1	1145	+2 / -3	+
gemini-1.5-flash-api-0514	1227	+3 / -3	-	claude-2.0	1131	+4 / -6	-
deepseek-v2-api-0628	1220	+6 / -6	+	gemini-pro-dev-api	1131	+4 / -3	-
gemma-2-27b-it	1218	+4 / -3	+	zephyr-orpo-141b	1127	+10 / -6	+
yi-large	1212	+4 / -5	-	gemma-2-2b-it	1126	+10 / -10	+
nemotron-4-340b-instruct	1209	+3 / -4	+	qwen1.5-32b-chat	1125	+3 / -3	+
bard-jan-24-gemini-pro	1208	+5 / -7	-	mistral-next	1124	+5 / -5	-
glm-4-0520	1206	+3 / -5	-	phi-3-medium-4k-instruct	1122	+4 / -4	+
llama-3-70b-instruct	1206	+2 / -2	+	starling-lm-7b-beta	1118	+4 / -5	+
claude-3-sonnet	1200	+2 / -2	-	claude-2.1	1118	+3 / -3	-
reka-core-20240501	1199	+3 / -3	-	gpt-3.5-turbo-0613	1116	+3 / -4	-
command-r-plus	1189	+2 / -2	+	mixtral-8x7b-instruct-v0.1	1114	+0 / -0	-

Tuning with SimPO significantly improves gemma2

a chat model [princeton-nlp/gemma-2-9b-it-SimPO](#) like 99

Rank* (UB)	Rank (StyleCtrl)	Model
35	30	Gemma-2-27b-it
35	31	Gemma-2-9b-it-SimPO
35	33	Deepseek-Coder-v2-0724
35	33	Command_R+_(08-2024)
35	35	Yi-Large
35	48	Gemini-1.5-Flash-8B-001
Cont.....		
50	46	Command_R+_(04-2024)
50	46	Qwen2-72B-Instruct
50	49	Gemma-2-9b-it

Starting from Gemma-2-9b-it

- Closed Pre-training
- Closed RLHF

Continued preference learning

- On-policy UltraFeedback data (50k data!)
- Annotated by ArmoRM-Llama3-8B-v0.1

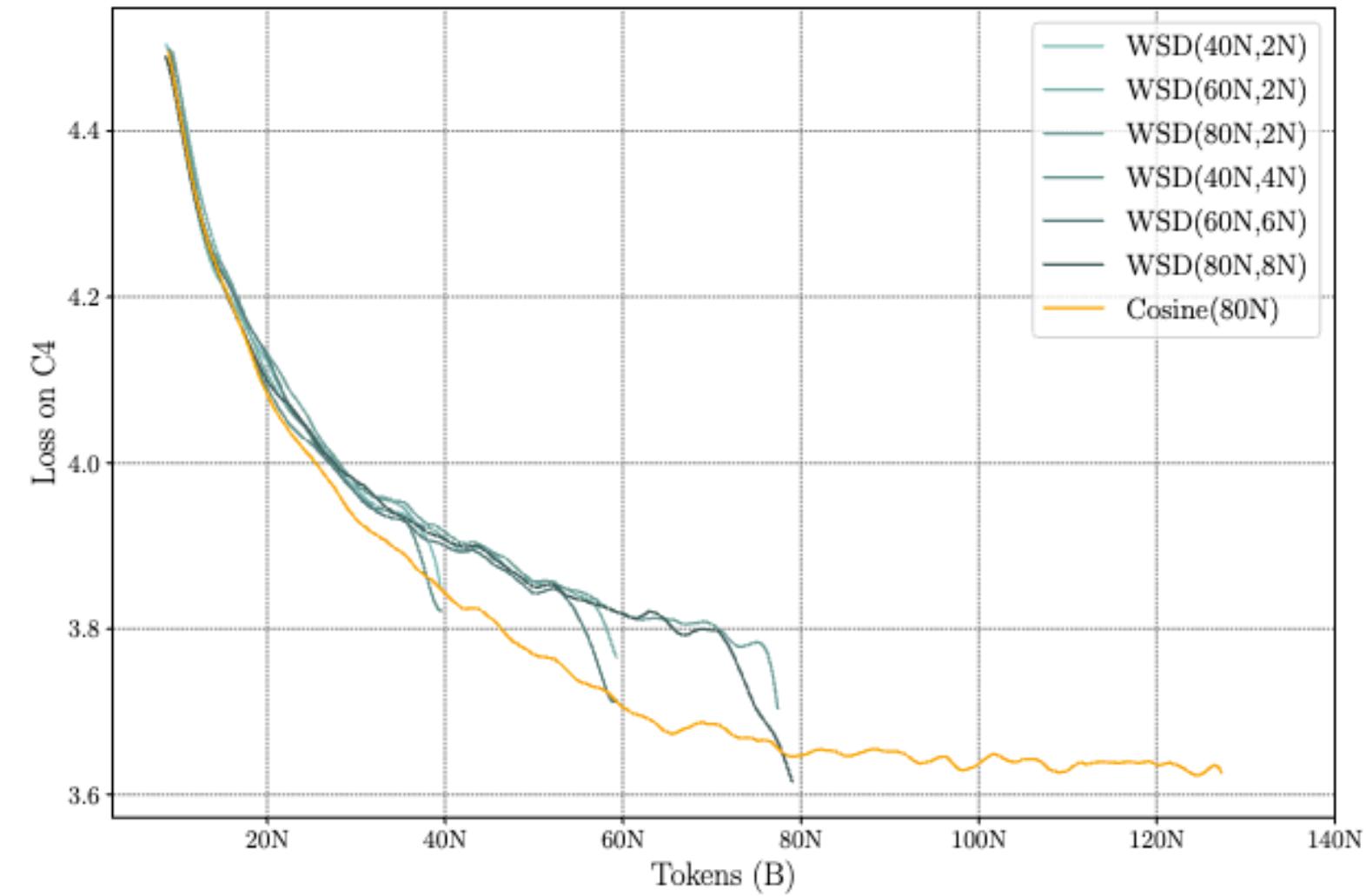
Gemma-2-9b-it-SimPO

- Strongest <10 B model on Chatbot Arena
- Completely within academic budget (16 H100 GPU hours!)

Where does the gain actually come from?

MiniCPM - Two important training techniques

$$WSD(T; s) = \begin{cases} \frac{s}{W}\eta, & s < W \\ \eta, & W < s < T \\ f(s - T)\eta, & T < s < S \end{cases}$$



Two stage training

	C-Eval	CMMLU	MMLU	GSM8K	MATH	HumanEval	MBPP
A-1	40.0	41.5	44.6	27.7	5.1	27.7	24.4
A-2	52.6	51.1	50.9	42.3	5.4	30.4	30.3
B-1	40.9	41.5	47.9	34.2	7.9	43.9	30.5
B-2	41.2	42.0	47.9	34.4	7.3	43.9	29.8
B-3	49.1	46.8	49.6	31.8	10.5	44.5	32.8

- Warmup-Stable-Decay (WSD) learning rate scheduler
- Sudden drop in loss when the lr starts to decay

- A1, B1, B2: standard pre-training, finetuning
- A2, B3: add high quality and SFT data to the decay stage during pretraining, the finetune

Conclusion

- Small models are getting stronger, open-weight, but not open-source :(
- Techniques that are effective for building small LMs
 - **High-quality data:** generally important for building foundation LMs
 - **Pruning:** reuse compute of existing models by directly removing weights from a model
 - **Distillation:** transfer knowledge from a large teacher model to a compact student model

mengzhou@princeton.edu

Thank you!