Democratization of Large Language Models (LLMs)

— Making LLMs Affordable for Everyone ;)

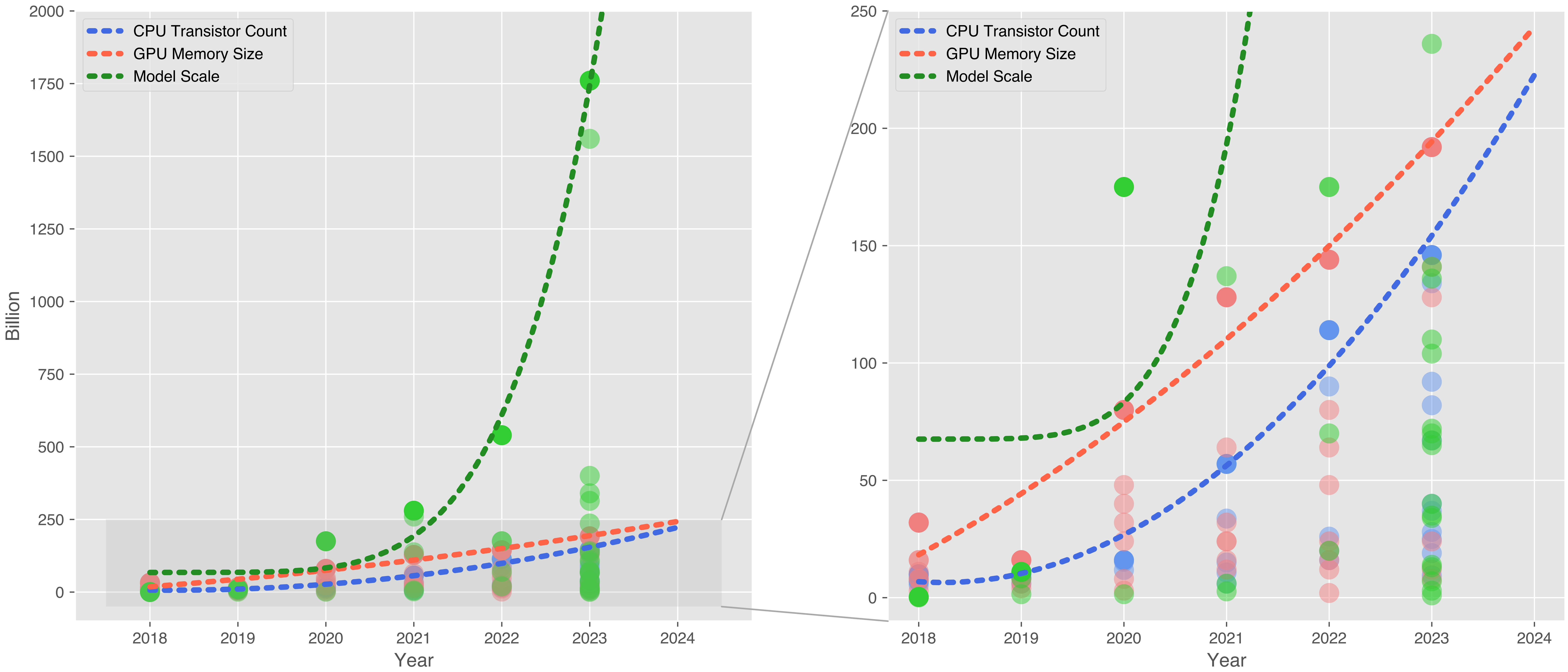
Chen Zhang https://genezc.github.io

2024/8/19

Why Democratization

Moore’s Law and Beyond

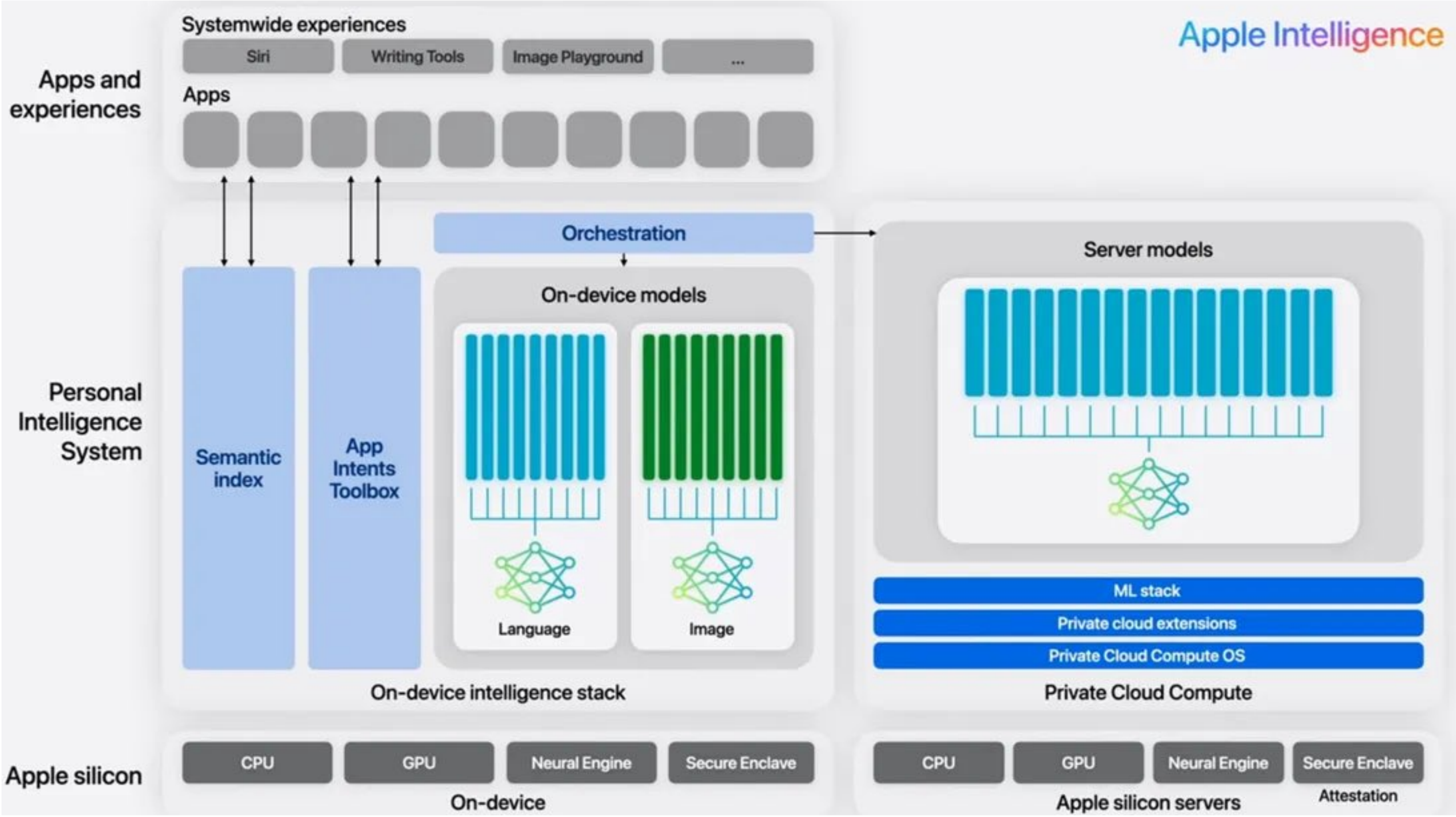
- Moore’s law and Huang’s law have respectively indicated that CPU and GPU performance will exponentially increase. However, this increase lags far behind that of model scale.



Why Democratization

An Exemplar: Apple Intelligence

- Apple has invented a concept termed *Apple Intelligence* that highlights the use of small language models (SLMs) to aid the drawbacks of serving LLMs.



Democratization in Size

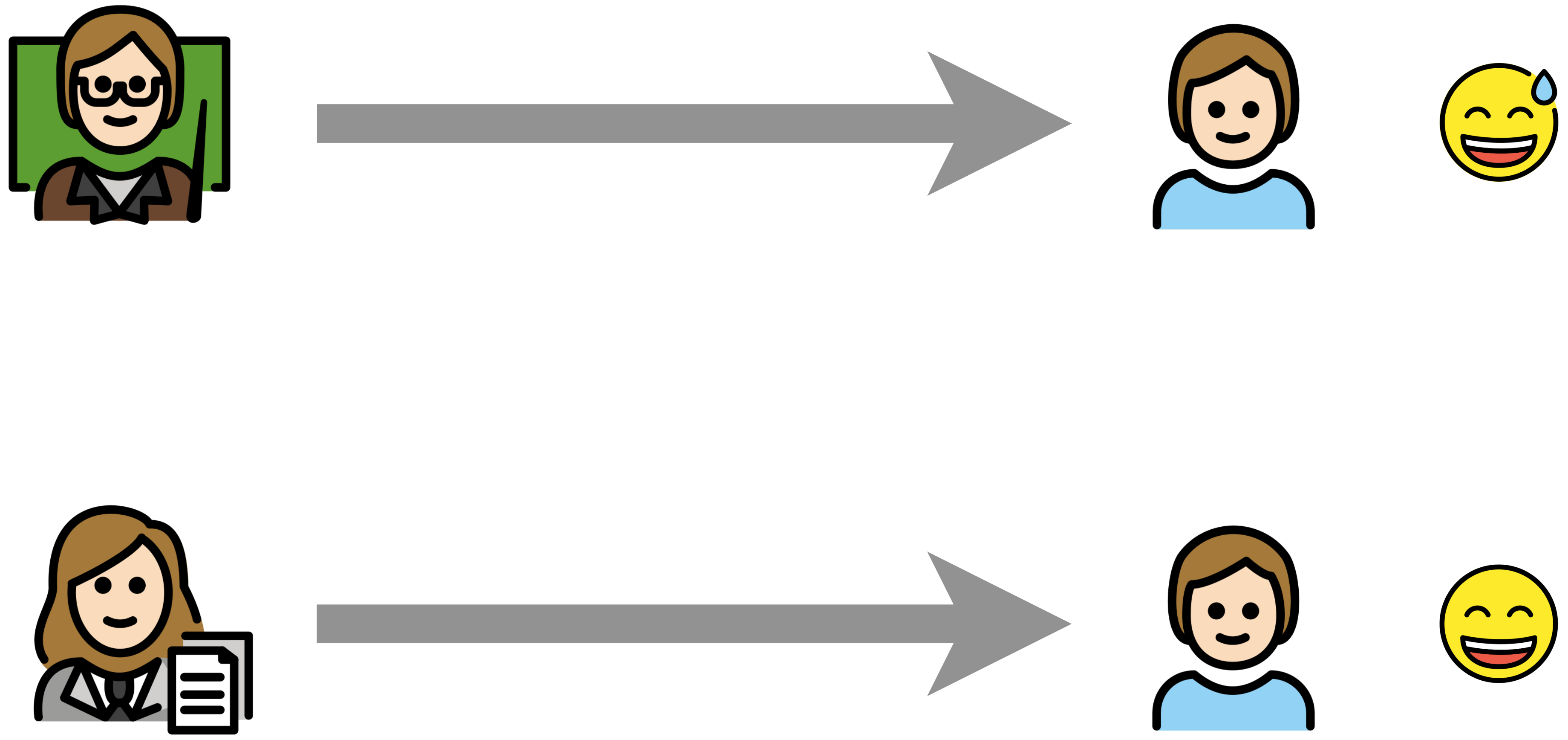
Distillation

- Language model (LM) distillation aims at reducing inference compute by distilling a large LM into a small LM under a teacher-student paradigm.

Democratization in Size

Capacity Gap: Intuition

- Distillation directly from the teacher to the student may suffer from the *capacity gap*.
- It is intuitive to recognize that a student at primary level might not be taught very well by a teacher at graduate level.



Democratization in Size

Capacity Gap: Theory I

- Here comes a minor theory of why *capacity gap* should be crucial.
- The approximation error in **Proposition 1** is negatively correlated with the *teacher size*.
- The approximation error in **Proposition 2** is negatively correlated with the *student size*, thus positively with the *teacher-student size discrepancy* given the teacher size is fixed.

Proposition 1 (VC dimension theory, Vapnik, 1998). *Assuming that the teacher function is $f_{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}$, the labeling function is $f \in \mathcal{F}$, and the data is $\mathcal{D}$, we have:*

$$r(f_{\mathcal{T}}) - r(f) \leq \epsilon_{\mathcal{T}} + o(\frac{|\mathcal{F}_{\mathcal{T}}|_c}{|\mathcal{D}|}),$$

where $r(\cdot)$ is the risk function, $|\cdot|_c$ is the function class capacity measure, and $|\cdot|$ is the data scale measure. It should be highlighted that the approximation error $\epsilon_{\mathcal{T}}$ is negatively correlated with the capacity of the teacher model while the estimation error $o(\cdot)$ is correlated with the learning optimization.

Proposition 2 (Generalized distillation theory, Lopez-Paz et al., 2016). *Additionally providing that the student function is $f_{\mathcal{S}} \in \mathcal{F}_{\mathcal{S}}$, we have:*

$$r(f_{\mathcal{S}}) - r(f_{\mathcal{T}}) \leq \epsilon_{\mathcal{G}} + o(\frac{|\mathcal{F}_{\mathcal{S}}|_c}{|\mathcal{D}|^{\alpha}}),$$

where the approximation error $\epsilon_{\mathcal{G}}$ is positively correlated with the capacity gap between the teacher and the student models, and $1/2 \leq \alpha \leq 1$ is a factor correlated to the learning rate.

Democratization in Size

Capacity Gap: Theory II

- Combining **Proposition 1 and 2** gives the ultimate theory.
- The student performance is proven to be concerned with both *the teacher performance* and *the capacity gap*.

Theorem 1. *The bound for the student function at a learning rate can be written as:*

$$r(f_{\mathcal{S}}) - r(f) \leq \epsilon_{\mathcal{T}} + \epsilon_{\mathcal{G}} + o(\frac{|\mathcal{F}_{\mathcal{T}}|_c}{|\mathcal{D}|}) + o(\frac{|\mathcal{F}_{\mathcal{S}}|_c}{|\mathcal{D}|^{\alpha}})$$
$$\leq \epsilon_{\mathcal{T}} + \epsilon_{\mathcal{G}} + o(\frac{|\mathcal{F}_{\mathcal{T}}|_c + |\mathcal{F}_{\mathcal{S}}|_c}{|\mathcal{D}|^{\alpha}}),$$

Proof. The proof is rather straightforward by combining Proposition 1 and 2. □

Remark 1. *Under the same distillation setting, we can ignore the estimation error. When we compare two students of different capacities distilled from a teacher of the same capacity, the student of a smaller capacity has a larger $\epsilon_{\mathcal{G}}$ thus lower performance. When we compare two students of the same capacities distilled from teachers of different capacities, the student distilled from the teacher of a larger capacity has a smaller $\epsilon_{\mathcal{T}}$ yet a larger $\epsilon_{\mathcal{G}}$ thus a tradeoff.*

Democratization in Size

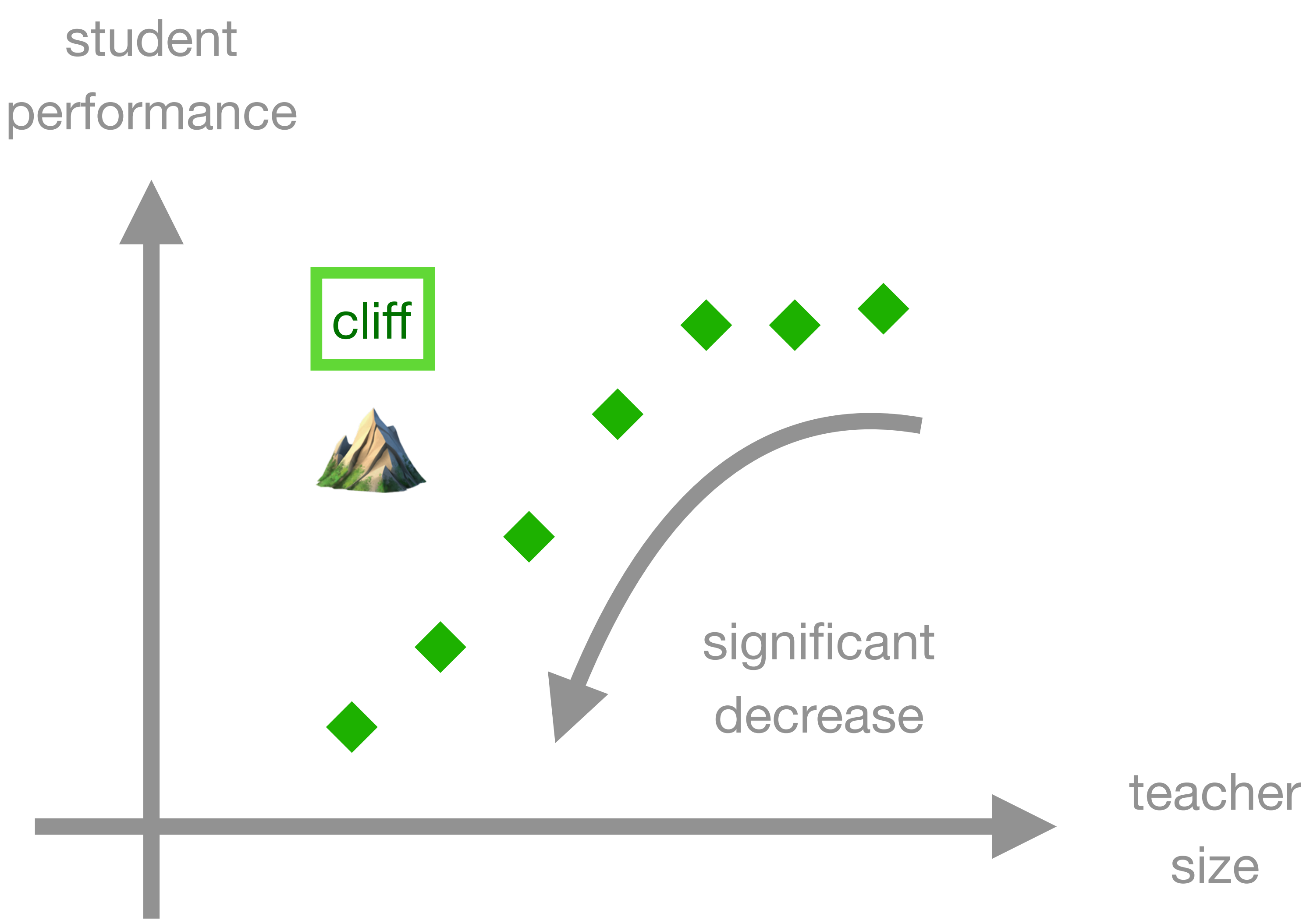
Capacity Gap: Observation

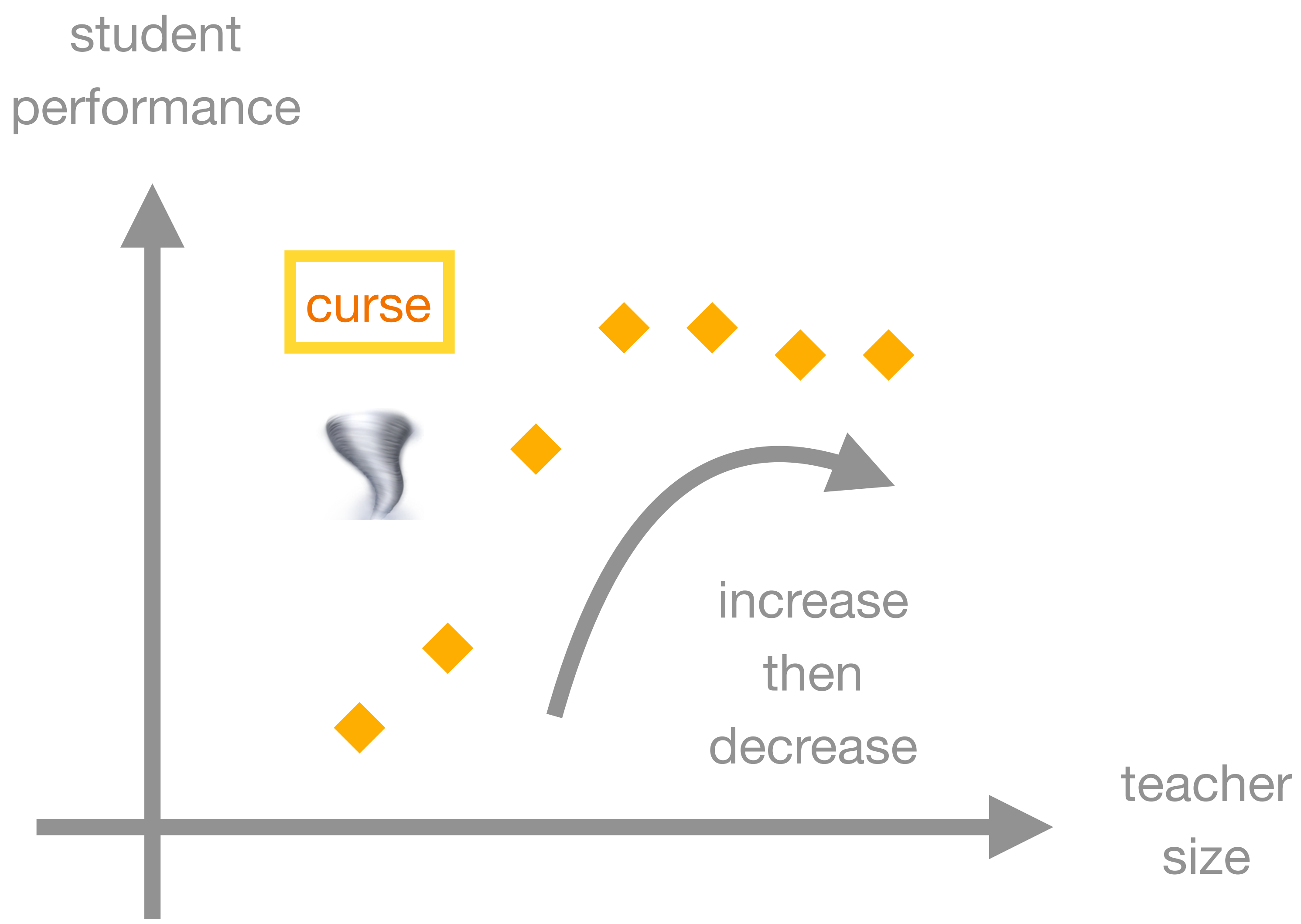
- The impact of capacity gap can be observed from three aspects, i.e., respectively the teacher perspective, the student perspective, and the compute perspective

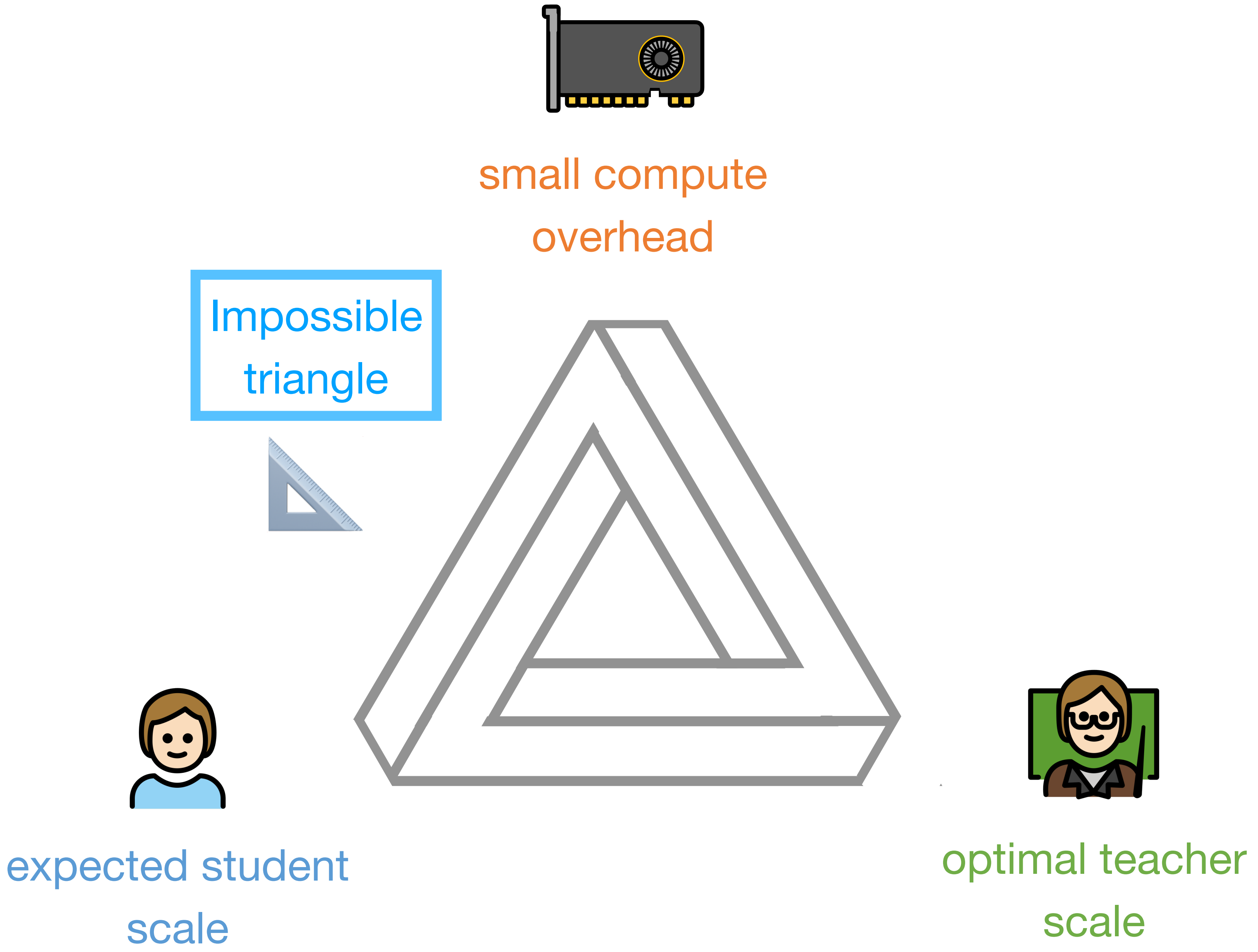
teacher perspective:
fixed teacher size
(models at few sizes)

student perspective:
fixed student size
(models at many sizes)

compute perspective:
fixed compute
(models at huge sizes)



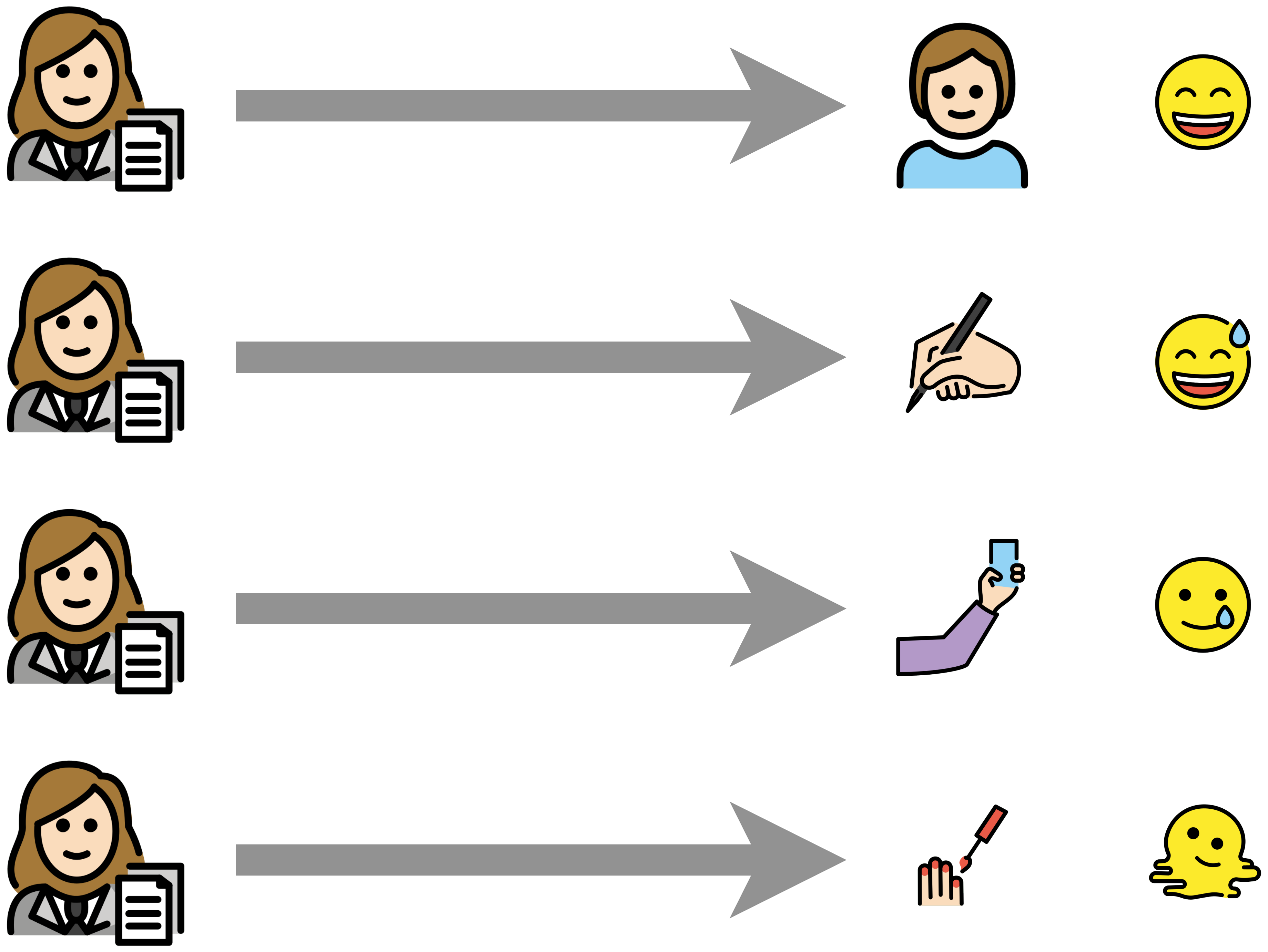




Democratization in Size

Cliff of Capacity Gap

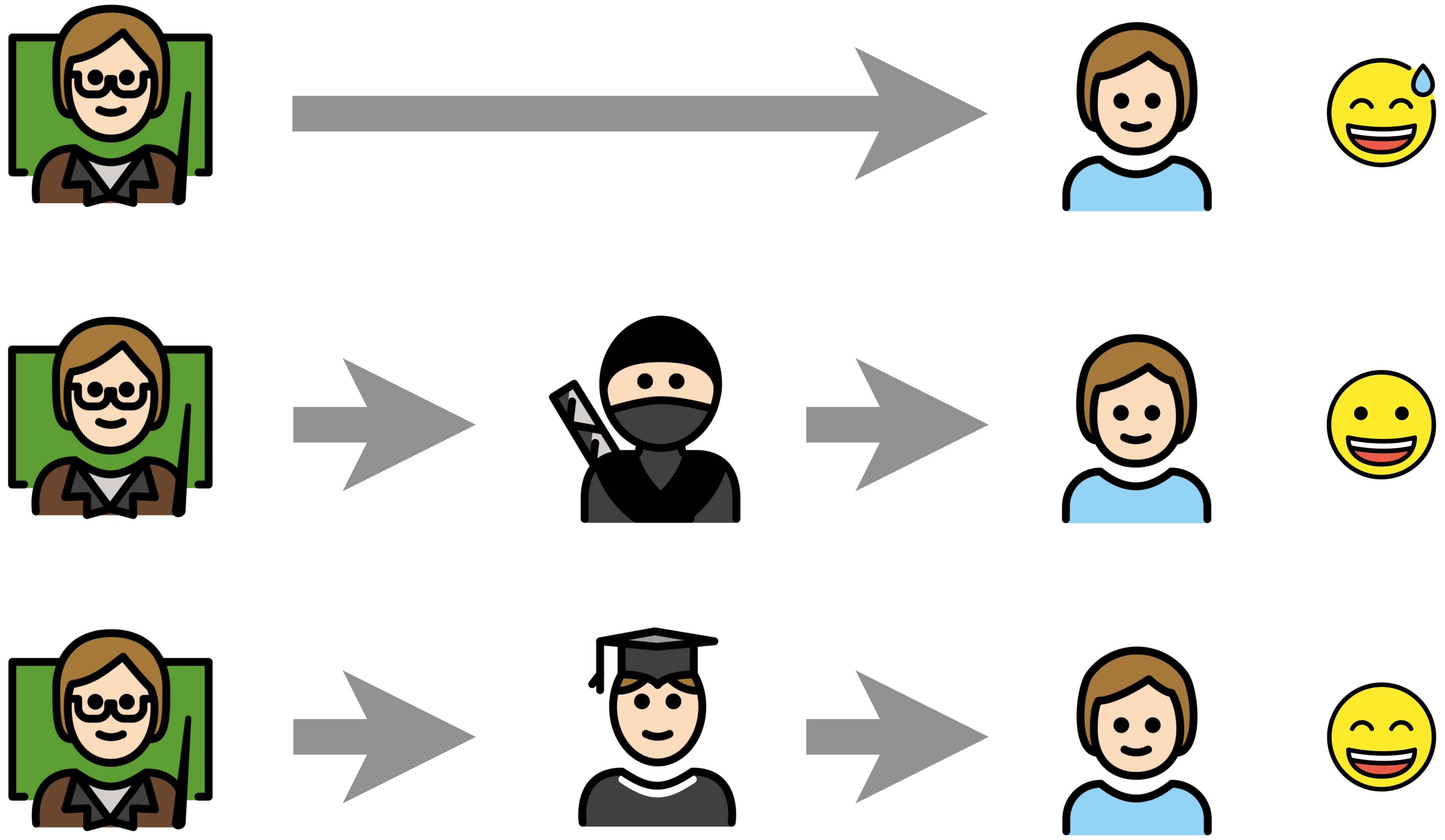
- From the teacher perspective, the student may suddenly suffer a significant performance decline while decreasing size, which we term as *cliff of capacity gap.*



Democratization in Size

Cliff of Capacity Gap: Optimality of Assistant

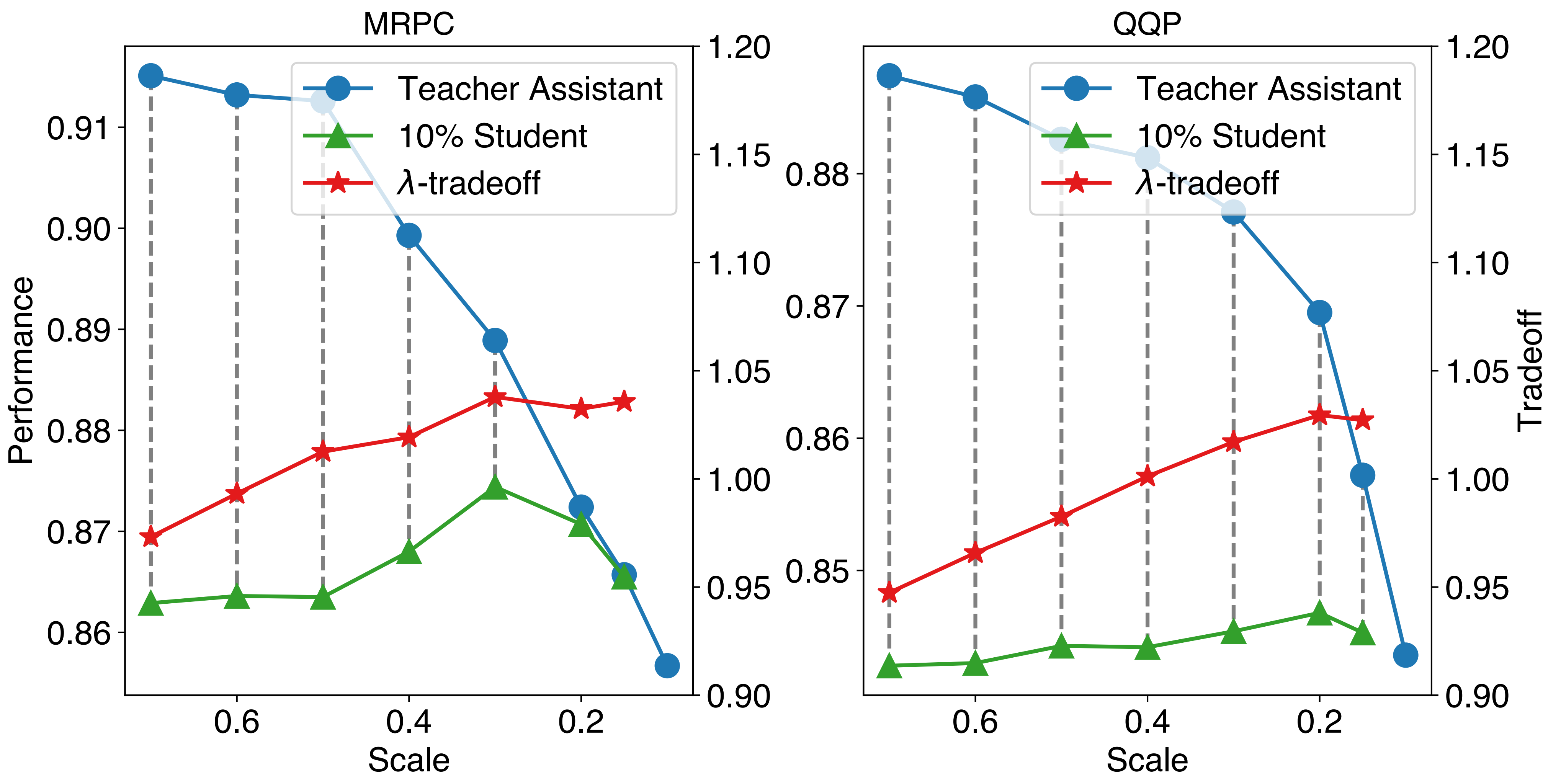
- A pioneering study to alleviate the cliff is teacher assistant-based distillation, which inserts an intermediate-size model between the teacher and student as a teacher assistant.
- We find that the student performance is very sensitive to the selected teacher assistant size and exhaustively uncover the optimal teacher assistant size could be very time-consuming with a brute-force search.

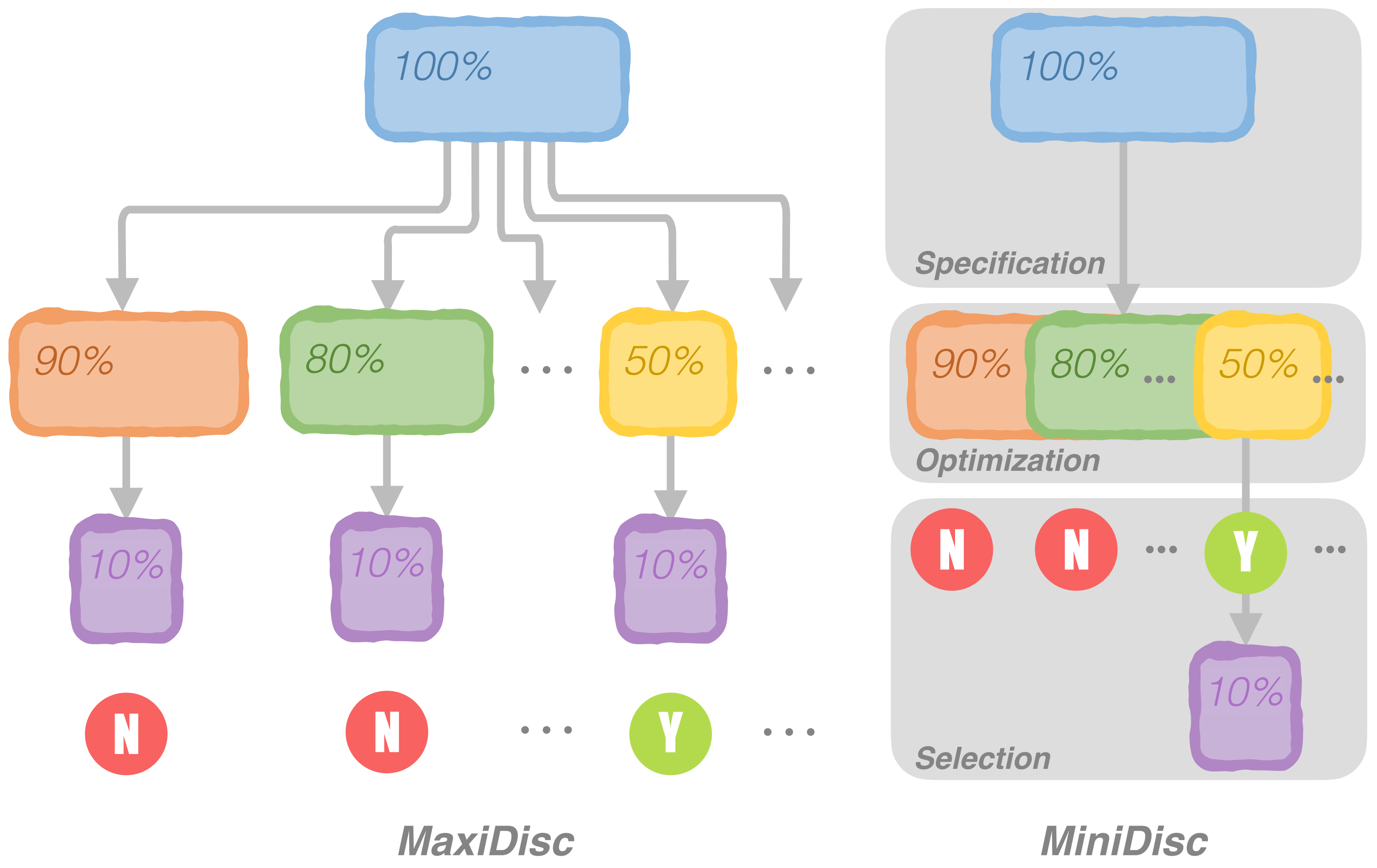


Democratization in Size

Cliff of Capacity Gap: Optimality of Assistant

- Scale-performance tradeoff is a good indicator of teacher assistant optimality, which is quantitatively defined by a *lambda-tradeoff* without an individual run to the student.
  - *t* = *m* + *lambda* * (*1* - *s*), where *m* is the performance and *(1 - s)* is the sparsity w.r.t. the teacher scale.
- Sparsity is a trivial measure while the performance is a measure that requires training, which is efficiently estimated by a *sandwich framework*:
  - *parameter sharing* among these candidate assistants admits one-run and more efficient training of all these assistants.





Democratization in Size

Cliff of Capacity Gap: Optimality of Assistant

- Main results

Table 4: The results of task-agnostic distillation upon $\text{BERT}_{\text{base}}$. The results of TinyBERT are reproduced based on their released checkpoints without additional task-specific distillation for a fair comparison. The GPU hours of teacher assistant-based methods are estimated with respect to their conventional counterparts.

Method	FLOPs	SST-2	MRPC	STS-B	QQP	MNLI-m/mm	QNLI	RTE	Average	GPUs
$\text{BERT}_{\text{base}}$	10.9G	93.8	91.5	87.1	88.4	84.9/84.9	91.9	71.5	86.7	–
Conventional Distillation										
$\text{FT}_{10\%}$ (2017)	1.1G	84.6	83.1	83.8	84.5	75.3/75.4	83.2	56.7	78.3	1×
$\mathcal{L}_{\text{TSD}10\%}$	1.1G	90.7	89.0	87.0	85.9	78.4/78.2	86.0	66.4	82.7	1×
$\text{MiniLM}_{\text{4L;384H}}$ (2021)	0.9G	90.0	88.6	87.2	86.1	80.0/80.3	87.9	67.2	83.4	1×
$\mathcal{L}_{\text{TAD}10\%}$	1.1G	92.0	90.1	87.9	86.6	80.0/80.3	88.0	67.2	**84.0**	1×
$\text{FT}_{5\%}$ (2017)	0.5G	84.1	82.4	81.8	83.7	74.4/74.9	82.5	57.0	77.6	1×
$\text{TinyBERT}_{\text{4L;312H}}$ (2020)	0.6G	88.5	87.9	86.6	85.6	78.9/79.2	87.3	67.2	82.7	1×
$\text{MiniLM}_{\text{3L;384H}}$ (2021)	0.7G	89.1	89.1	86.6	85.4	77.8/78.4	87.2	66.1	82.5	1×
$\mathcal{L}_{\text{TAD}5\%}$	0.5G	90.9	89.4	87.7	85.8	79.2/79.8	87.3	65.7	**83.2**	1×
Teacher Assistant-based Distillation										
$\text{TA}_{10\%}$ (2020)	0.9G	90.0	88.5	87.3	86.3	80.1/80.7	88.0	66.4	83.4	2×
$\text{MAXIDISC}_{10\%}$	1.1G	91.5	90.3	87.8	86.6	80.0/80.1	88.6	67.2	**84.0**	40×
$\text{MINIDISC}_{10\%}$	1.1G	91.4	90.0	87.5	86.6	79.8/80.0	88.0	67.2	83.8	4×
$\text{TA}_{5\%}$ (2020)	0.7G	89.8	85.9	86.0	85.5	77.6/78.5	86.8	66.1	82.0	2×
$\text{MAXIDISC}_{5\%}$	0.5G	90.1	89.7	87.4	85.6	79.3/79.7	87.1	67.9	**83.4**	40×
$\text{MINIDISC}_{5\%}$	0.5G	89.3	89.7	87.4	85.9	79.2/79.4	86.9	69.7	**83.4**	4×

Table 3: The results of task-specific distillation upon $\text{LLaMA2}_{\text{7B}}$. The Alpaca dataset (Taori et al., 2023) is utilized as the distillation data.

Method	MMLU
$\text{LLaMA2}_{\text{7B}}$	46.0
$\text{KD}_{15\%}$	25.6
$\text{TA}_{15\%}$	26.1
$\text{MAXIDISC}_{15\%}$	26.8
$\text{MINIDISC}_{15\%}$	26.9

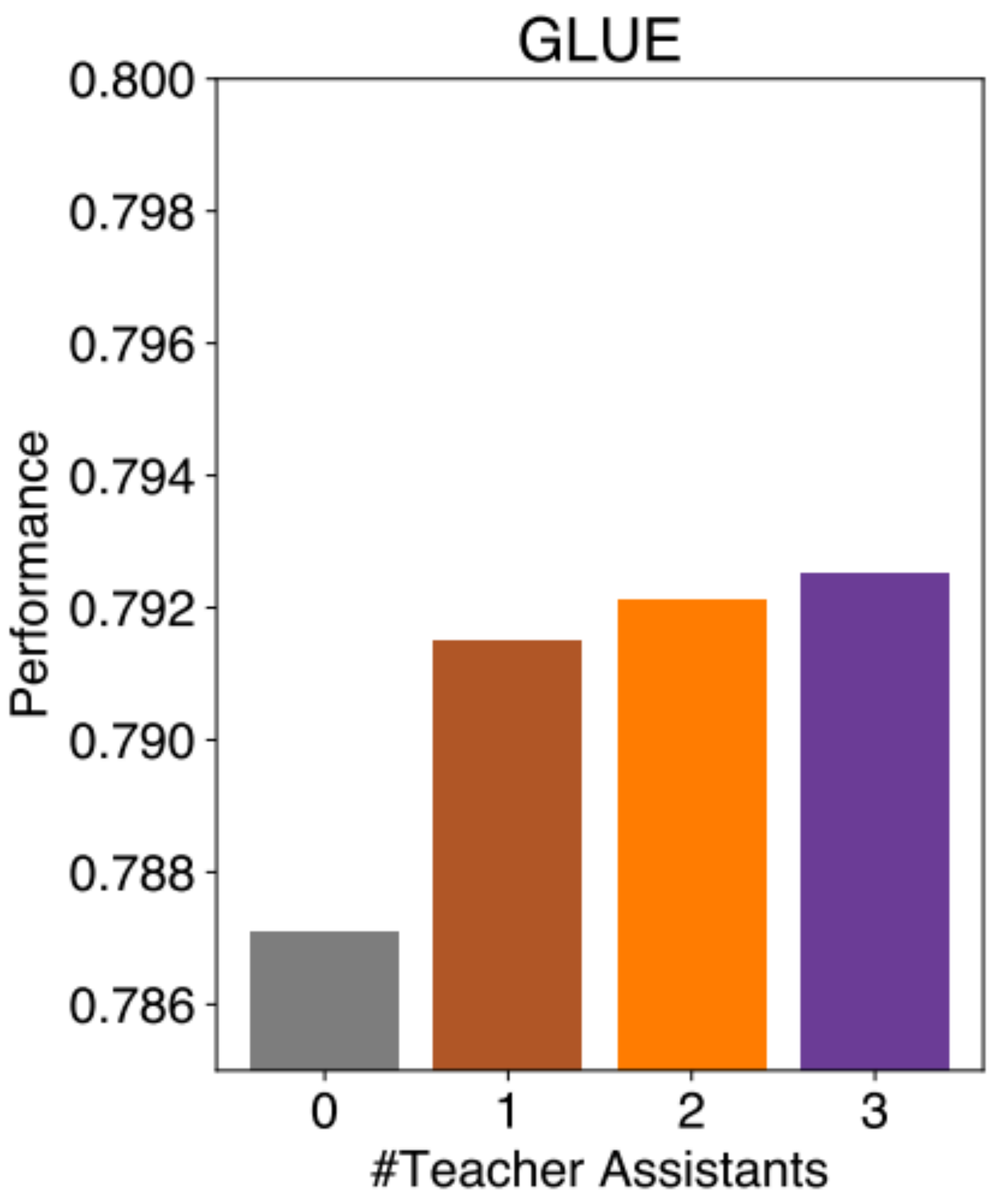
Democratization in Size

Cliff of Capacity Gap: Optimality of Assistant

- Ablation results

Table 5: The ablation study upon distilling $\text{BERT}_{\text{base}}$ to $\text{BERT}_{10\%}$.

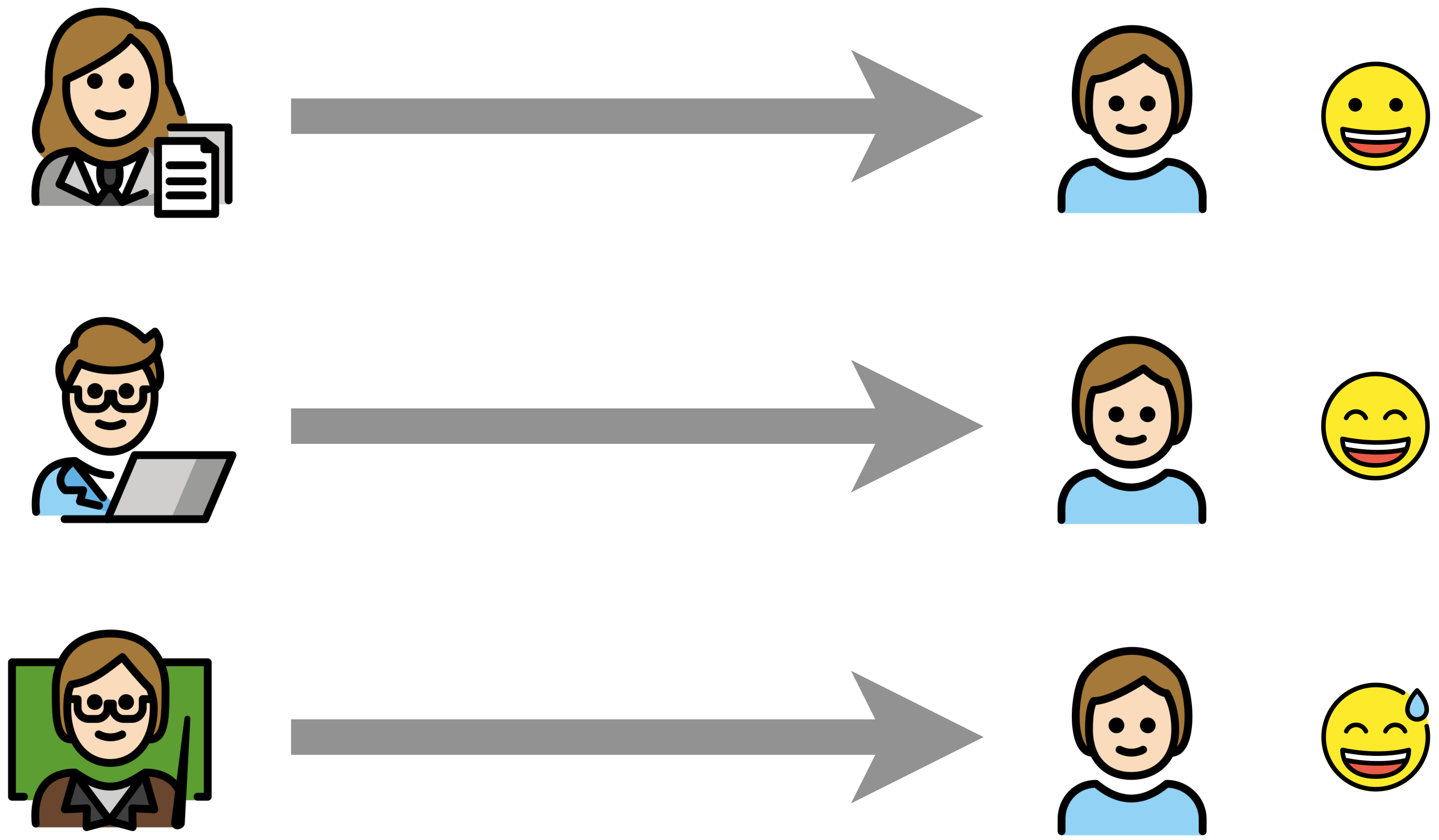
Method	GPU hours	MRPC	QQP
$\mathcal{L}_{\text{TSD}10\%}$	1×	87.8	84.6
$\text{MAXIDISC}_{10\%}$	40×	88.2	84.8
w/ λ-*tradeoff*	21×	88.2	84.8
w/ *sandwich framework*	23×	88.4	84.9
$\text{MINIDISC}_{10\%}$	4×	88.4	84.9



Democratization in Size

Curse of Capacity Gap

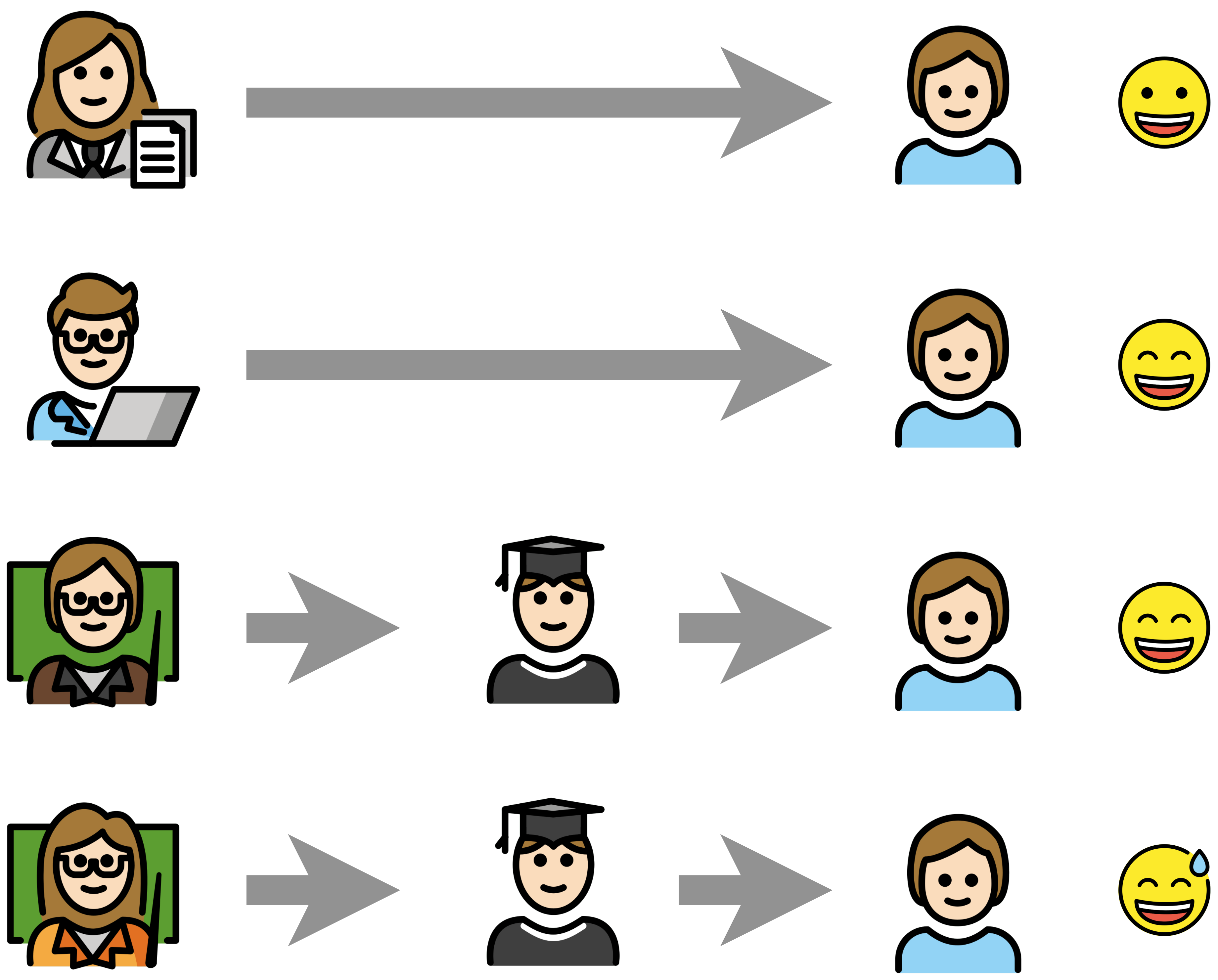
- From the student perspective, the student may first enjoy a improved performance then suffer a performance decline while teacher is increasing size, which we term as *curse of capacity gap.*



Democratization in Size

Curse of Capacity Gap: Generalization

- Teacher assistant-based distillation is also a good choice to circumvent the curse.
- We discover that even teacher assistant-based distillation can only lift the curse in a limited scope.



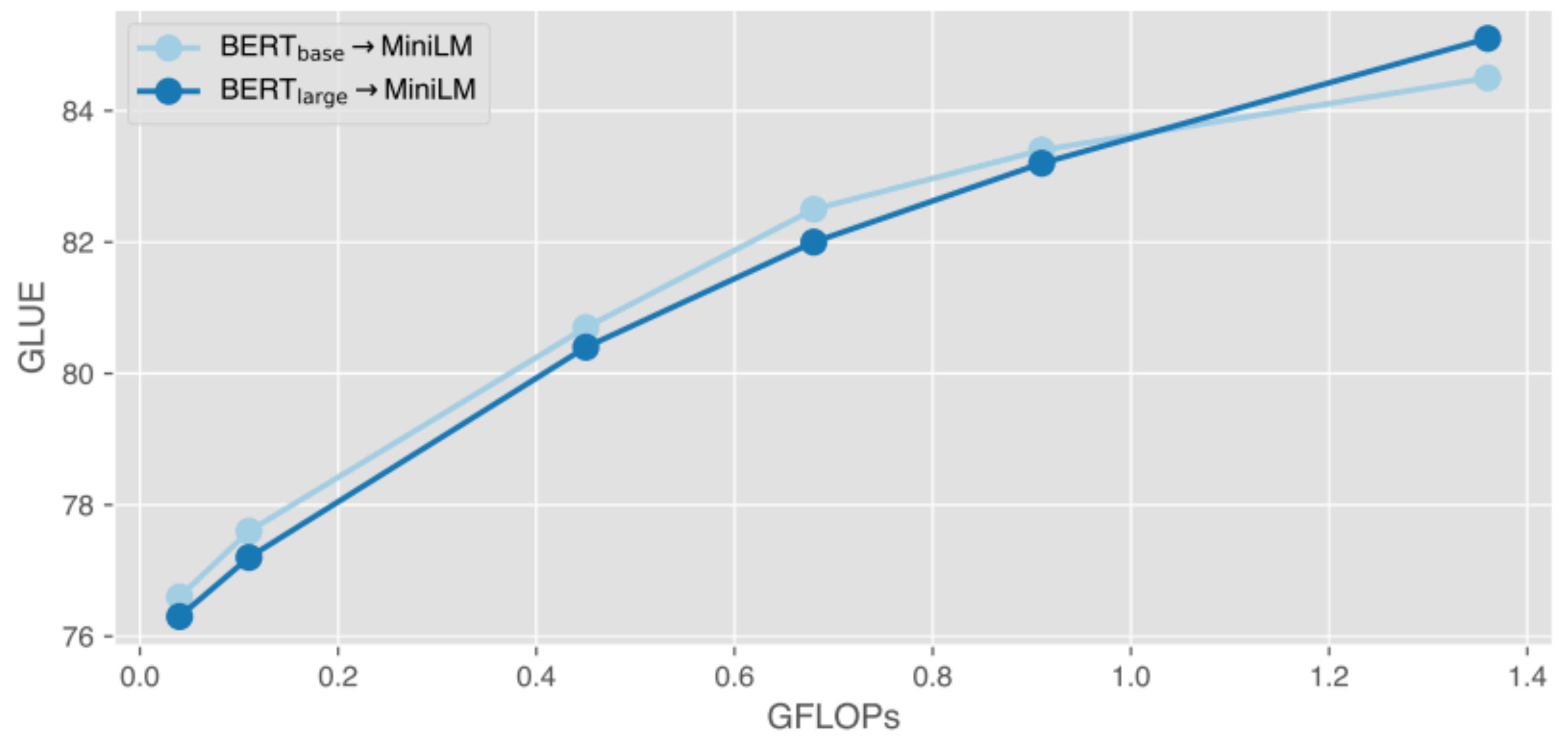
Democratization in Size

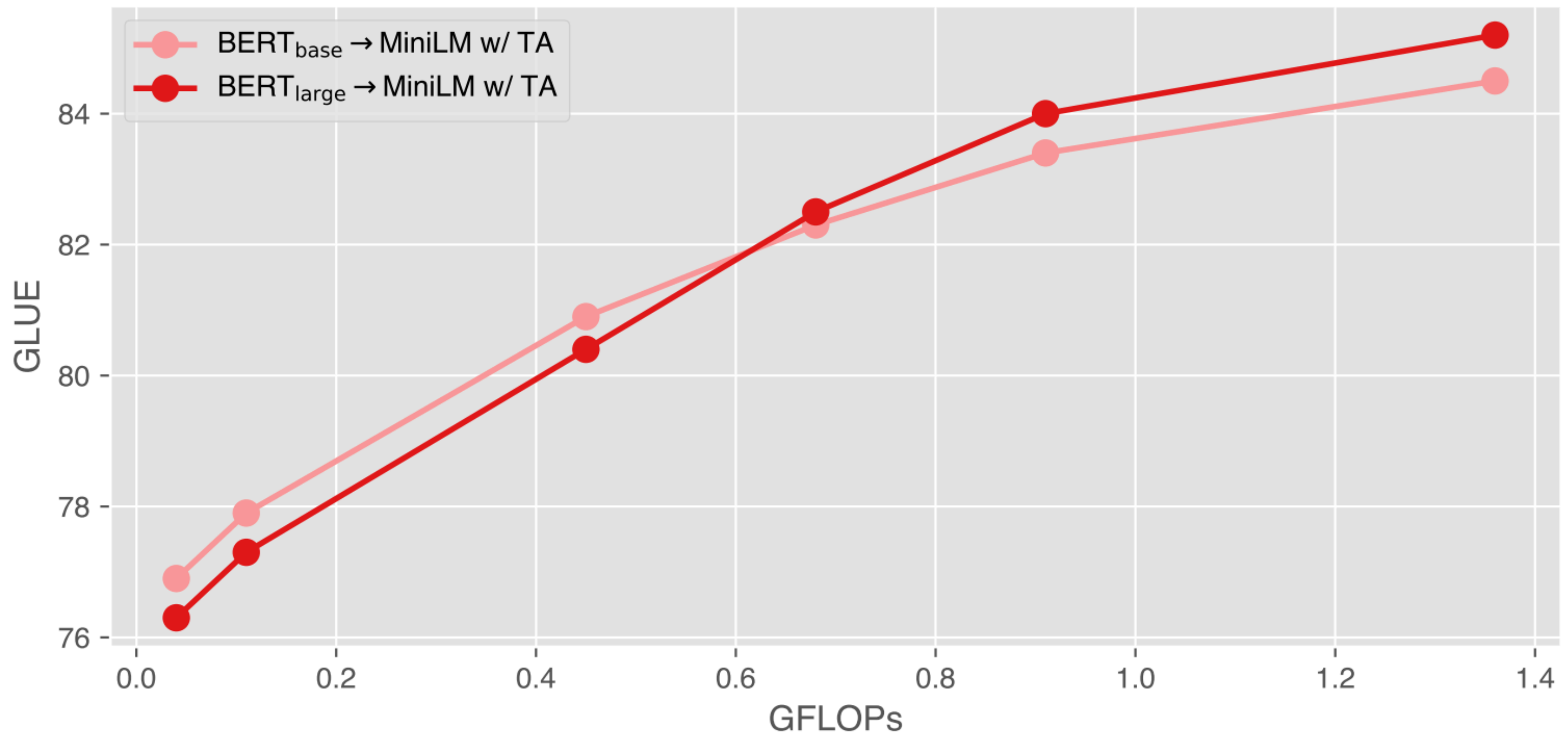
Curve of Capacity Gap: Generalization

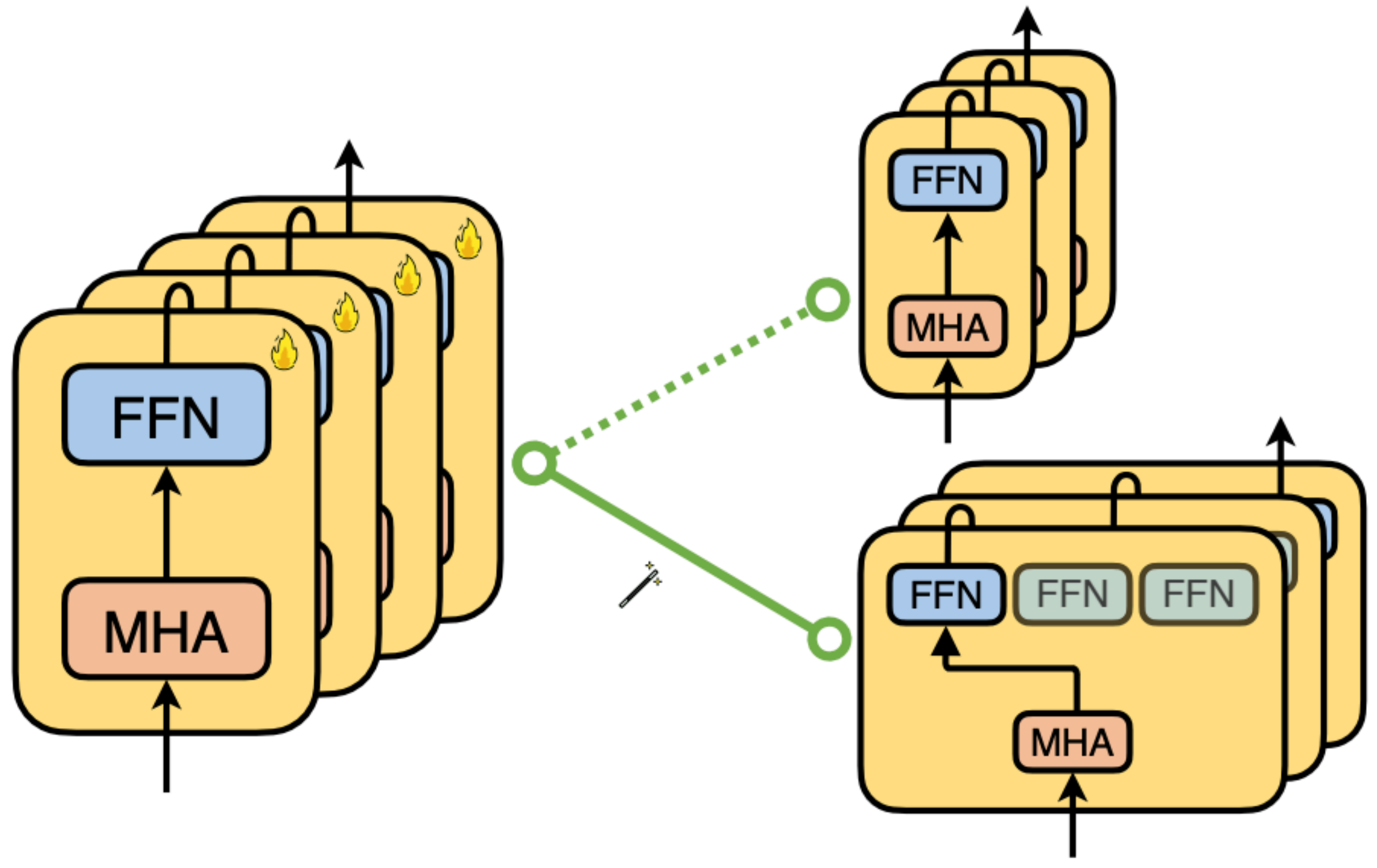
- An intuitive guideline is enlarging student capacity without increasing inference compute. *Mixture of experts (MoE)* student, enlarging student capacity with sparse experts.
- We incorporate the merits of MoE in the design of the distillation.

Table 1: The *curse of the capacity gap* in terms of GLUE (Wang et al., 2019). The △ denotes the performance difference of preceding two numbers. To ensure students at similar scales, the student/teacher scale ratios are properly reduced for some methods.

Method	$\text{BERT}_{\text{base}}$	$\text{BERT}_{\text{large}}$	△
Teacher	86.7	88.3	+1.6
$\text{KD}_{10\%/5\%}$ (2015)	81.3	80.8	−0.5
$\text{DynaBERT}_{15\%/5\%}$ (2020)	81.1	79.2	−1.9
$\text{MiniDisc}_{10\%/5\%}$ (2022a)	82.4	82.1	−0.3
$\text{TinyBERT}_{4L;312H}$ (2020)	82.7	82.5	−0.2
$\text{MiniLM}_{3L;384H}$ (2021b)	82.5	82.0	−0.5
$\text{MiniMoE}_{3L;384H}$ (**ours**)	82.6	83.1	+0.5







Democratization in Size

Curse of Capacity Gap: Generalization

- Main results

Table 3: The results of comparison between distilling BERT$_{\text{base}}$ and BERT$_{\text{large}}$.

Method	Teacher	SST-2 Acc	MRPC F1	STS-B SpCorr	QQP F1	MNLI-m/mm Acc	QNLI Acc	RTE Acc	GLUE Score	CoNLL F1
MiniLM$_{\text{6L;384H}}$	BERT$_{\text{base}}$	91.1	90.1	88.1	86.7	81.5/81.8	89.2	67.9	84.5	93.2
	BERT$_{\text{large}}$⇑	90.9	90.6	89.0	86.9	81.8/82.4	88.8	70.0	85.1	93.2
w/ TA	BERT$_{\text{base}}$	91.3	90.3	88.2	86.8	81.4/81.6	89.7	66.8	84.5	93.2
	BERT$_{\text{large}}$⇑	91.4	89.8	88.5	87.0	81.9/81.6	89.5	71.5	85.2	93.2
MiniMoE$_{\text{6L;384H}}$	BERT$_{\text{base}}$	91.3	90.2	88.6	86.5	81.6/81.5	89.5	68.6	84.7	93.3
	BERT$_{\text{large}}$⇑[1]	90.5	90.0	88.8	86.8	81.8/82.2	90.8	70.4	85.2	93.3
MiniLM$_{\text{4L;384H}}$	BERT$_{\text{base}}$	90.0	88.6	87.2	86.1	80.0/80.3	87.9	67.2	83.4	91.5
	BERT$_{\text{large}}$⇓	89.3	87.5	88.1	85.9	79.9/80.2	87.6	67.2	83.2	91.2
w/ TA	BERT$_{\text{base}}$	90.0	88.5	87.3	86.3	80.1/80.7	88.0	66.4	83.4	91.8
	BERT$_{\text{large}}$⇑	90.6	88.7	88.1	86.3	80.5/80.7	87.9	69.0	84.0	92.2
MiniMoE$_{\text{4L;384H}}$	BERT$_{\text{base}}$	90.8	88.1	88.2	85.9	79.8/80.4	88.6	69.3	83.9	92.3
	BERT$_{\text{large}}$⇑	90.5	88.0	88.7	86.7	80.9/80.9	89.2	69.0	84.2	92.4
MiniLM$_{\text{3L;384H}}$	BERT$_{\text{base}}$	89.1	89.1	86.6	85.4	77.8/78.4	87.2	66.1	82.5	90.1
	BERT$_{\text{large}}$⇓	89.1	86.1	87.1	85.1	78.6/78.5	86.0	65.7	82.0	87.3
w/ TA	BERT$_{\text{base}}$	89.8	87.8	86.0	85.5	77.6/78.5	86.8	66.1	82.3	90.4
	BERT$_{\text{large}}$⇓	89.7	84.9	87.2	85.2	78.5/79.1	86.6	66.4	82.2	90.2
MiniMoE$_{\text{3L;384H}}$	BERT$_{\text{base}}$	89.3	87.4	87.8	85.6	78.2/78.7	87.2	67.0	82.6	90.7
	BERT$_{\text{large}}$⇑	89.1	88.4	87.6	86.2	78.8/79.5	87.5	67.9	83.1	91.6

[1] ⇑ is used to indicate the deficiency is tackled on both GLUE and CoNLL, otherwise ⇓ is used.

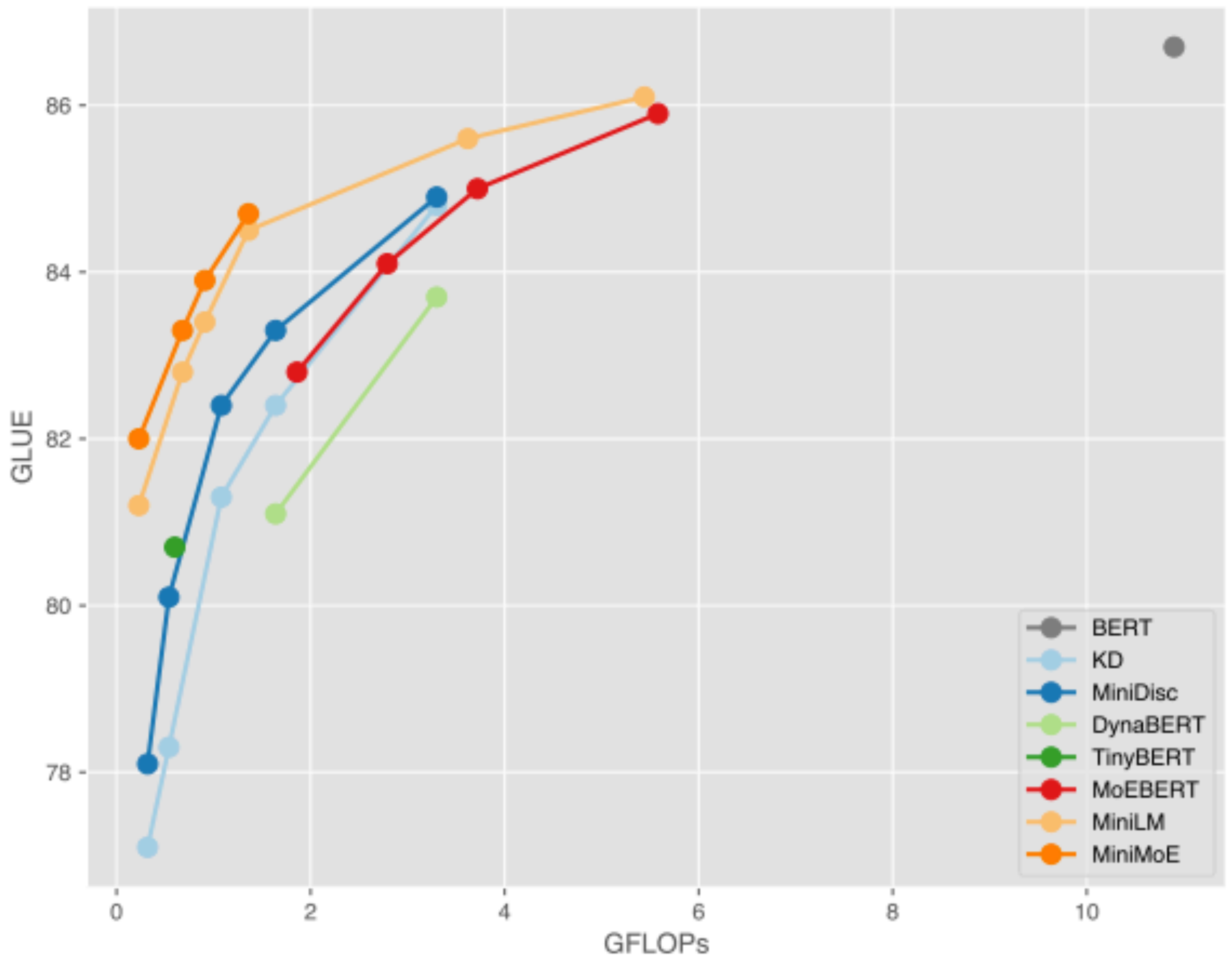
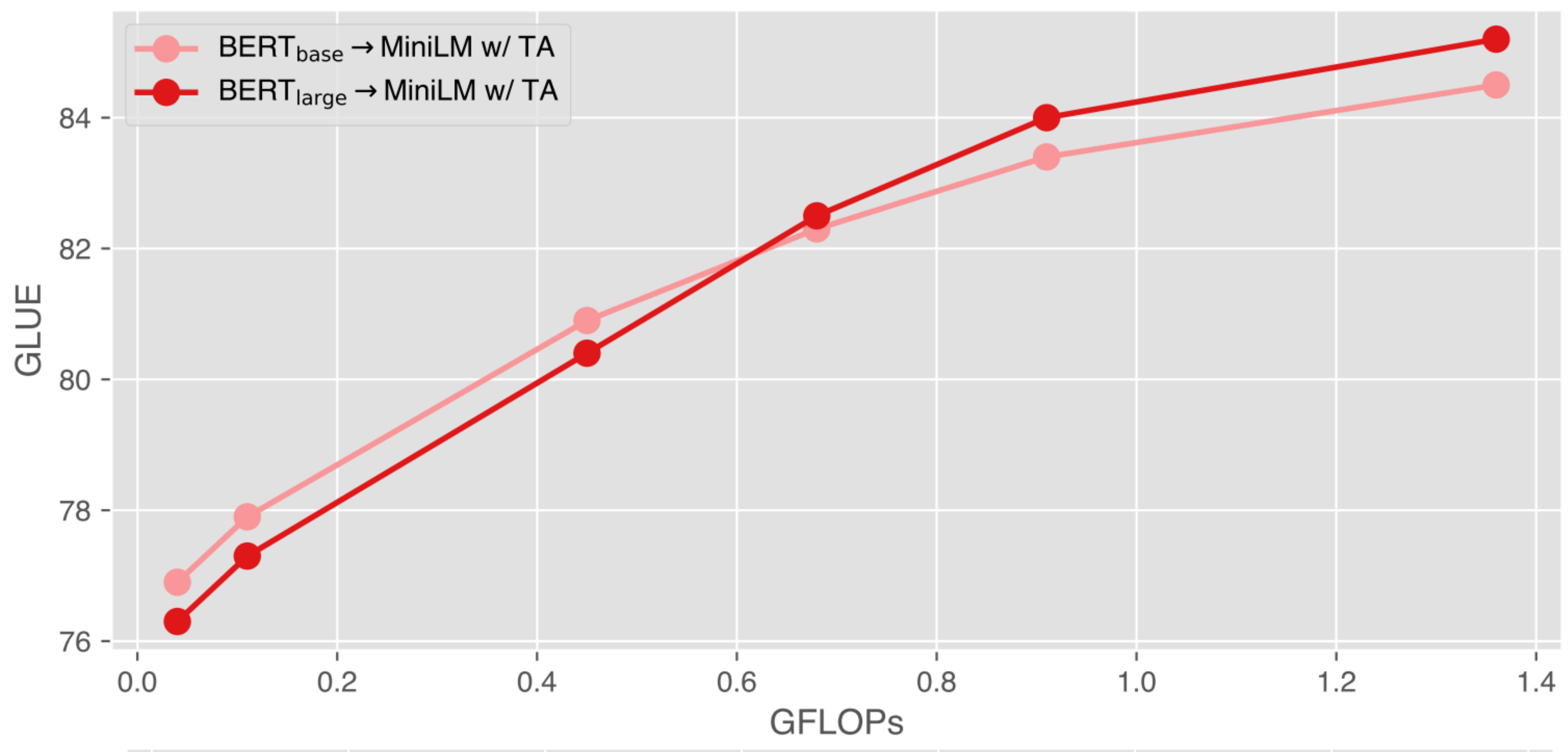


Figure 1: GLUE v.s. GFLOPs.

Democratization in Size

Curse of Capacity Gap: Generalization

- Ablation results



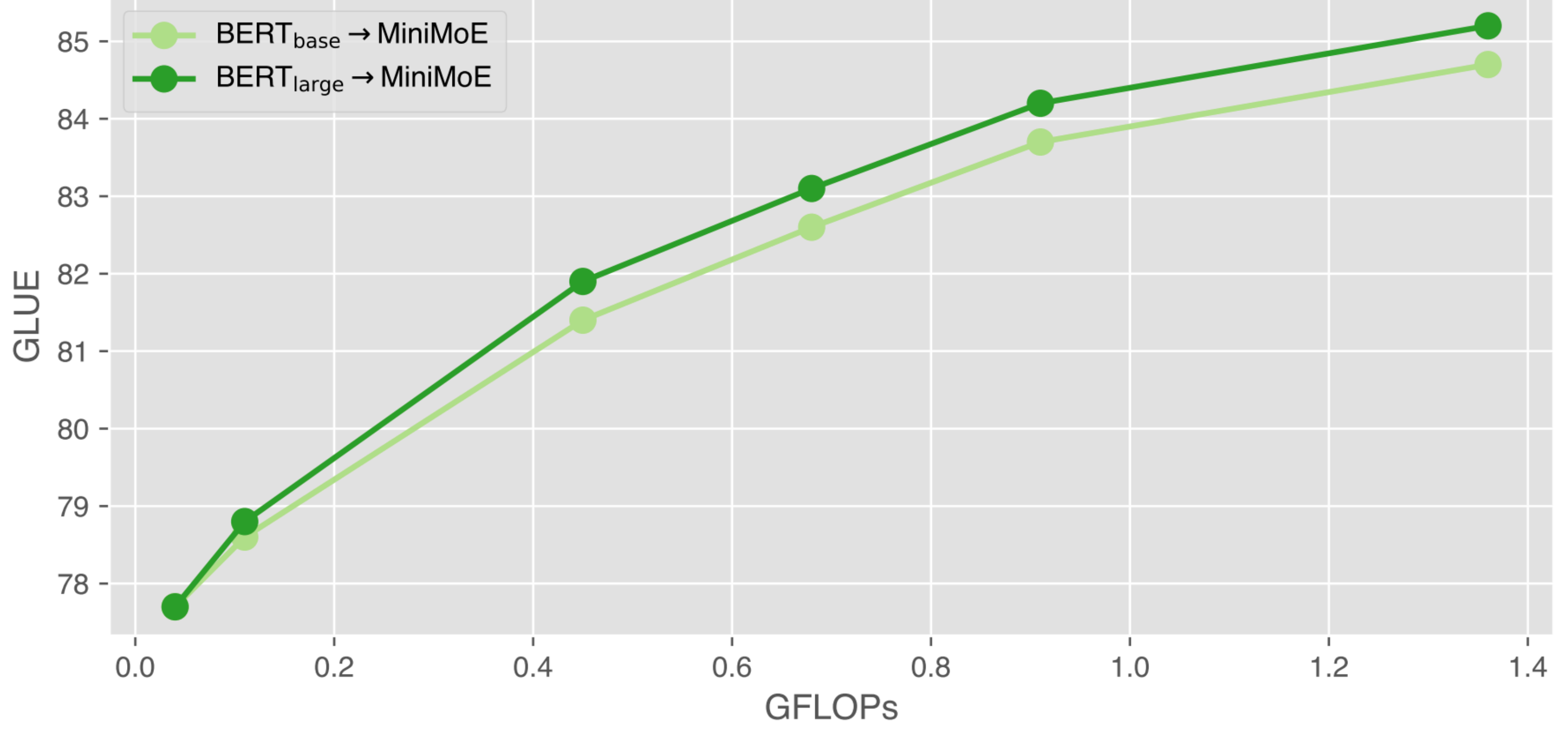


Table 5: Practical inference compute with reference to BERT$_{base}$.

Method	GFLOPs	Throughput	Params
BERT$_{base}$	10.9	80.8 tokens/ms	109.5 M
KD$_{5\%}$	0.54	544.7 tokens/ms	28.7 M
MiniLM$_{3L;384H}$	0.68	485.3 tokens/ms	17.2 M
MiniMoE$_{3L;384H}$	0.68	433.1 tokens/ms	28.3 M

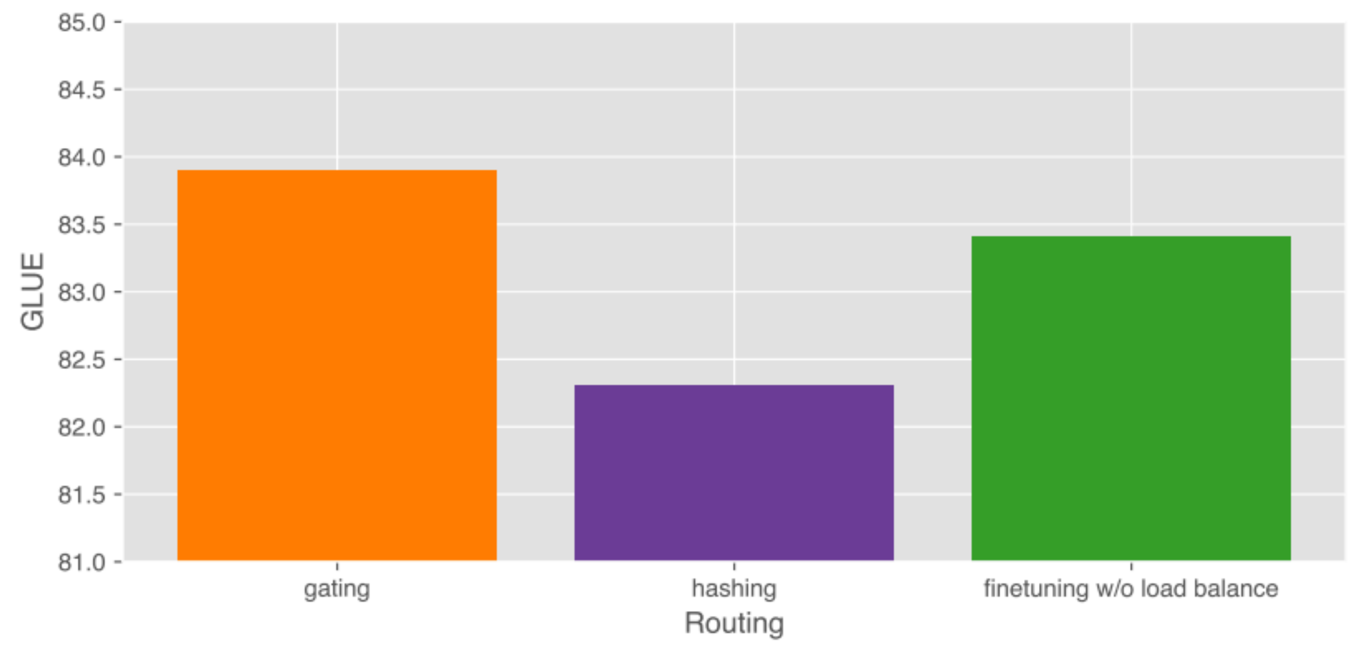


Figure 5: The performance of different routing choices with MiniMoE$_{4L;384H}$ upon distilling BERT$_{base}$.

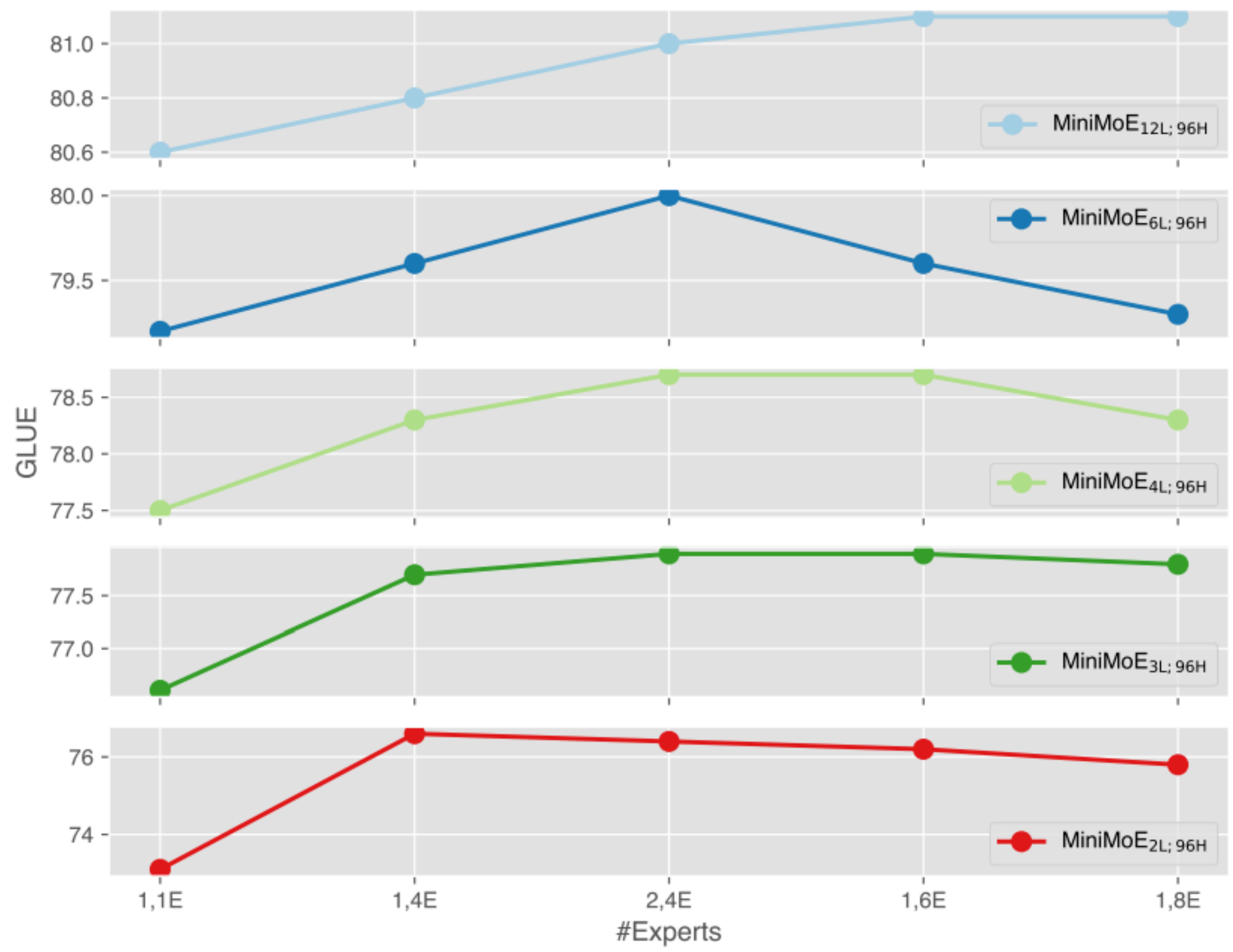
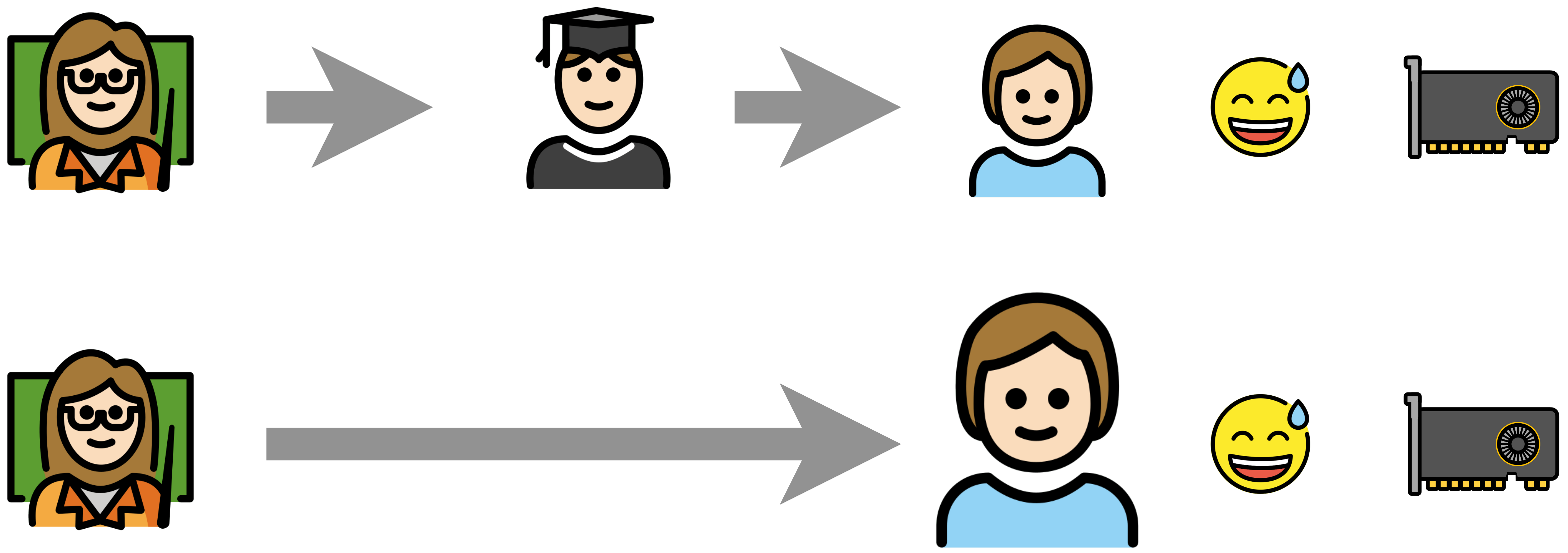


Figure 6: The impact of expert number on the performance upon distilling BERT$_{base}$, where x,yE denotes x experts in each MHA and y experts in each FFN. For example, 1,1E is the original dense model, and 1,4E is the MoE model used in Table 4.

Democratization in Size

Impossible Triangle of Capacity Gap

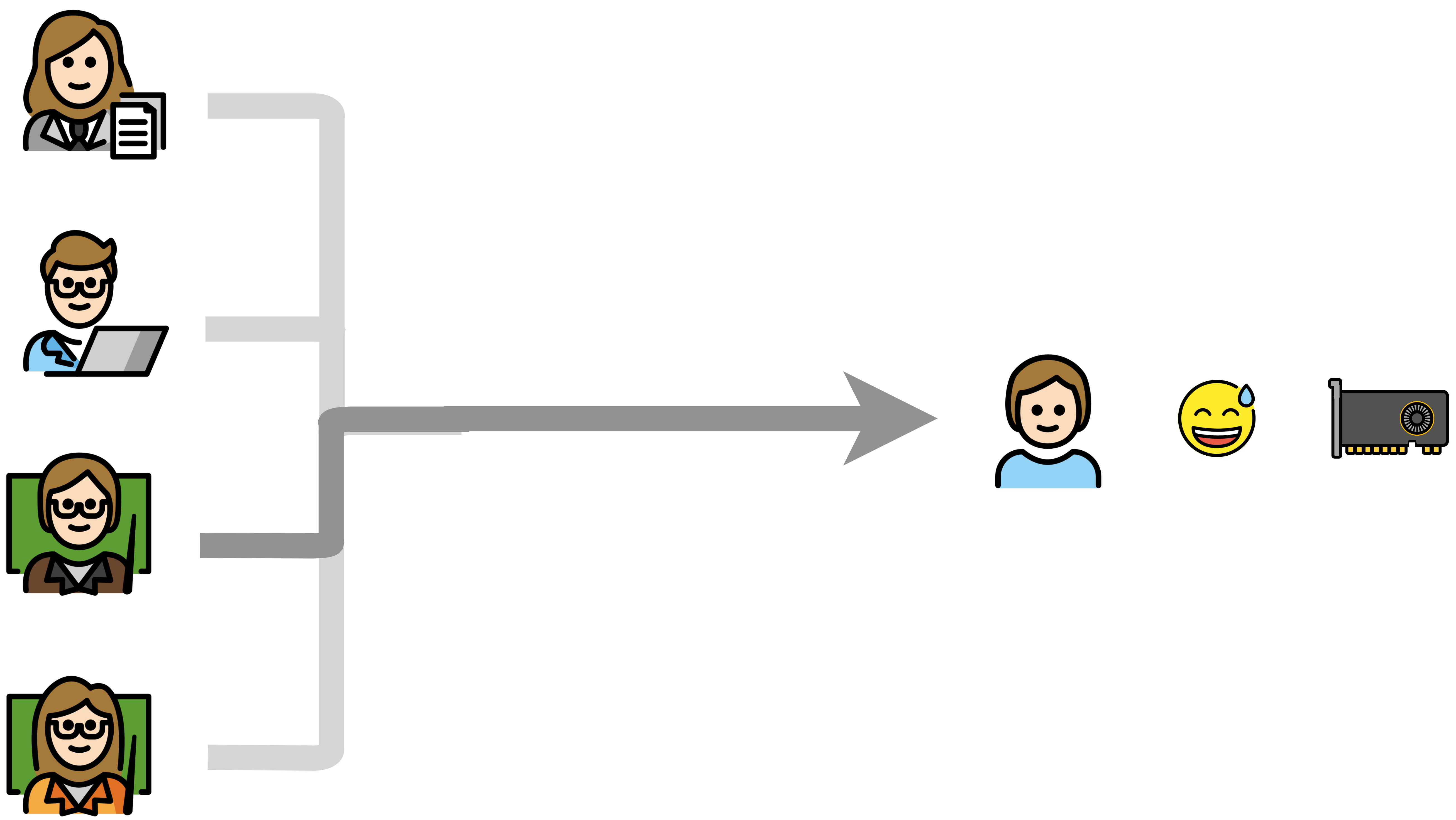
- Besides the teacher and student perspectives, we can also take a third-party perspective from the compute side.
- The optimal teacher size, the expected student size, and the small compute overhead besides the teacher and student consumptions form an *impossible triangle of capacity gap*.



Democratization in Size

Impossible Triangle of Capacity Gap: Sustainability

- Determining the optimal teacher would also be consuming, then how?

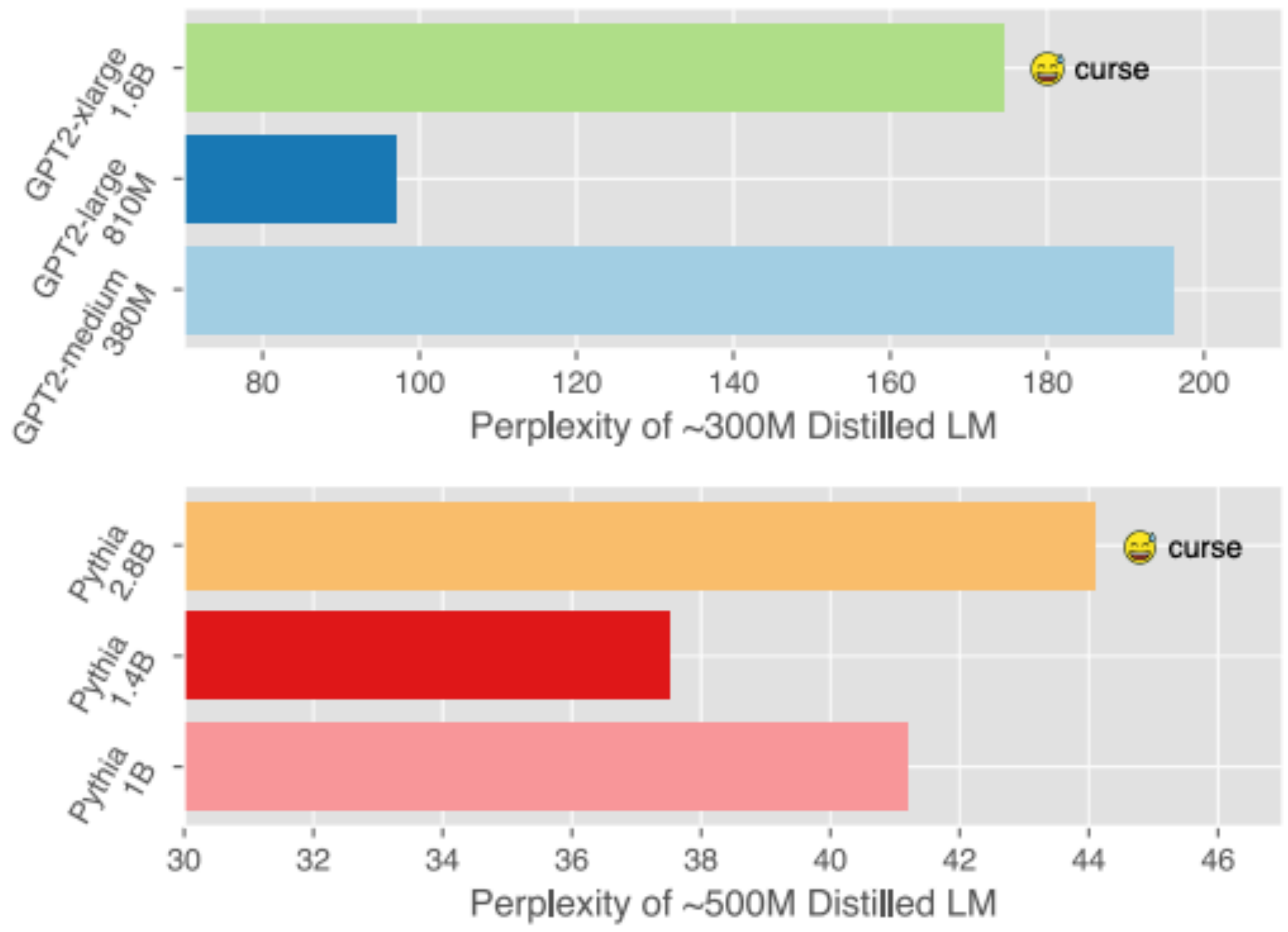


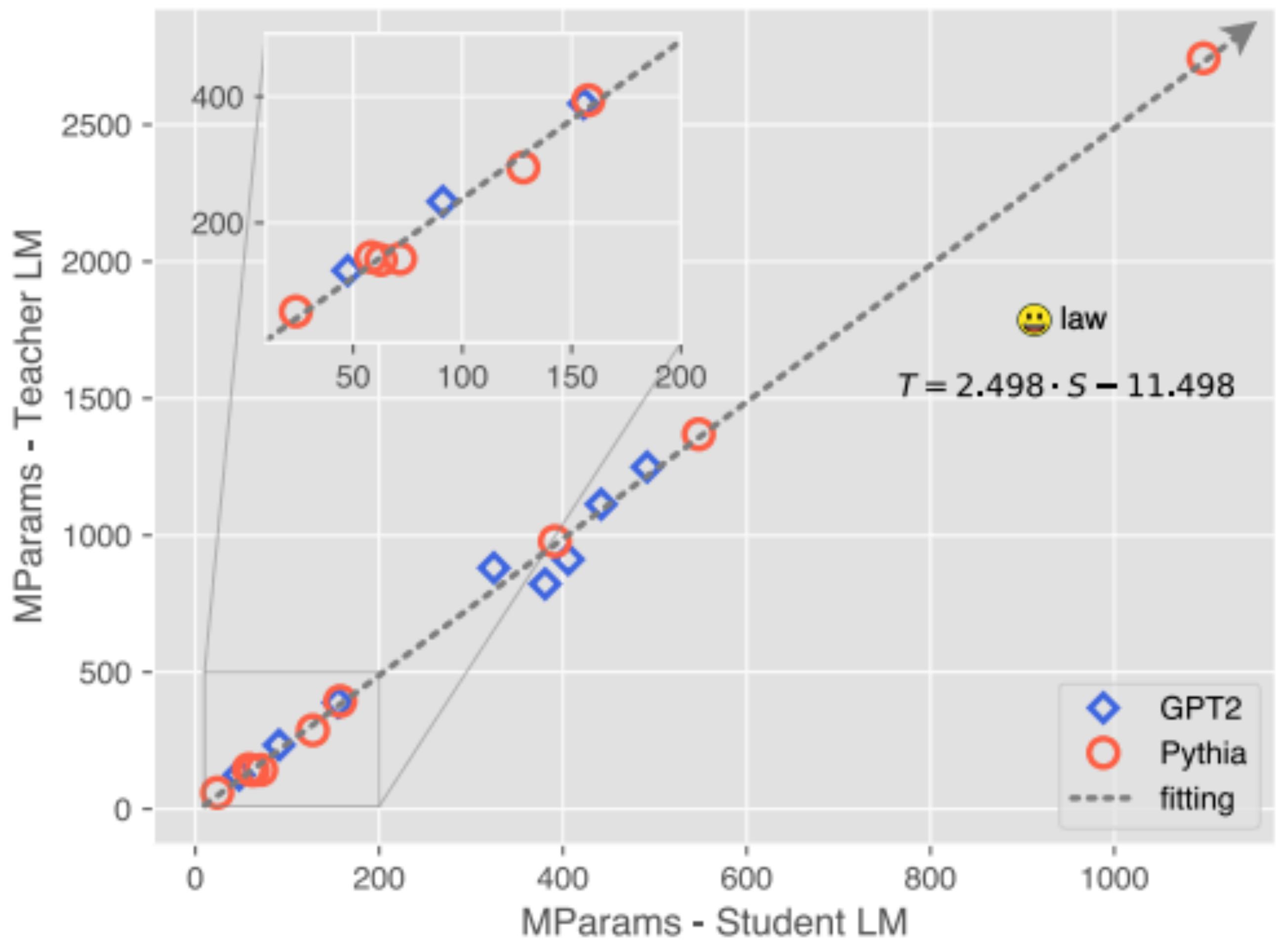
Democratization in Size

Impossible Triangle of Capacity Gap: Sustainability

- Inspired by the spirits of *scaling law*, we attempt to show a law of how the optimal teacher will be sized provided an expected student size.

Method	Compute Overhead
TA (2020)	
- FLOPs	$> \gamma \cdot 8 \cdot N \cdot D$
- Memory	$> \gamma \cdot 20 \cdot N$
MiniMoE (2023b)	
- FLOPs	0
- Memory	$> \gamma \cdot 16 \cdot N$
Search	
- FLOPs	$> \gamma \cdot 6 \cdot N \cdot D$
- Memory	$> \gamma \cdot 16 \cdot N$

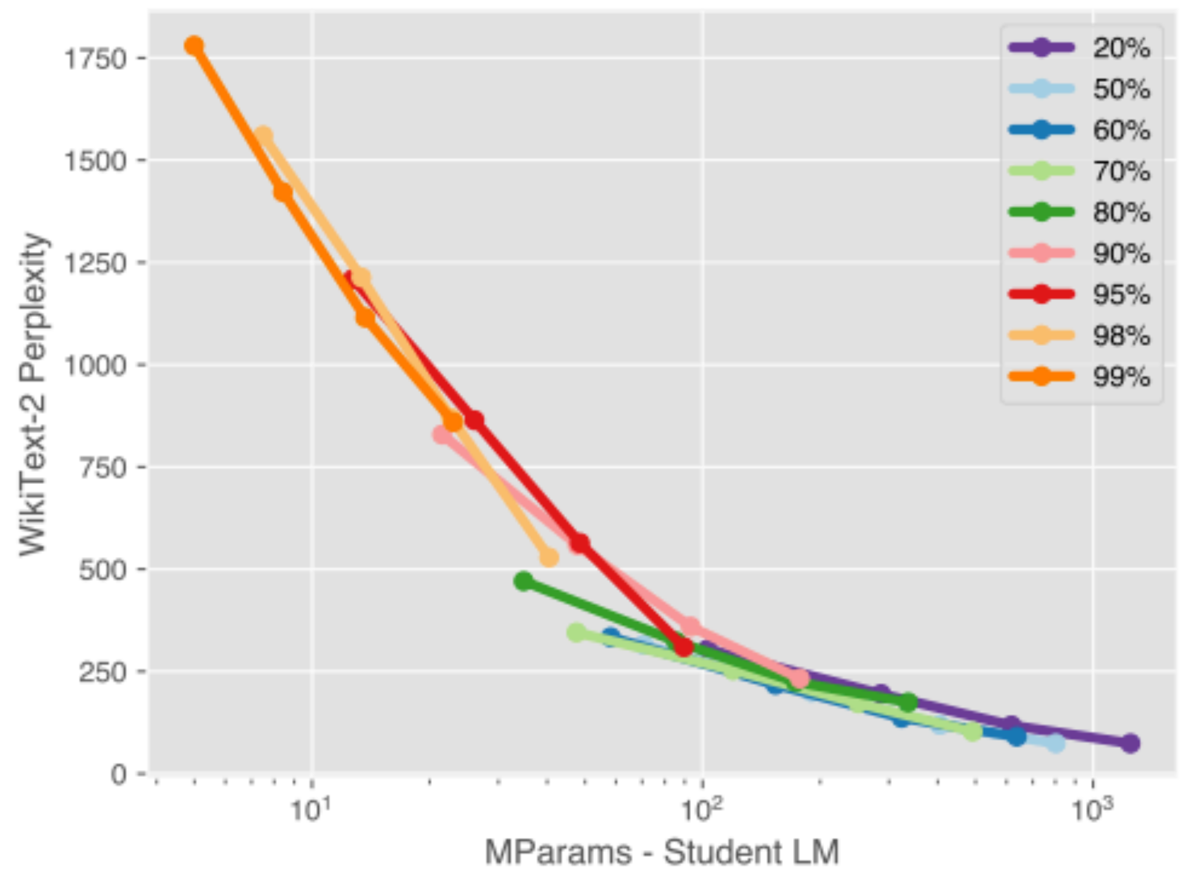




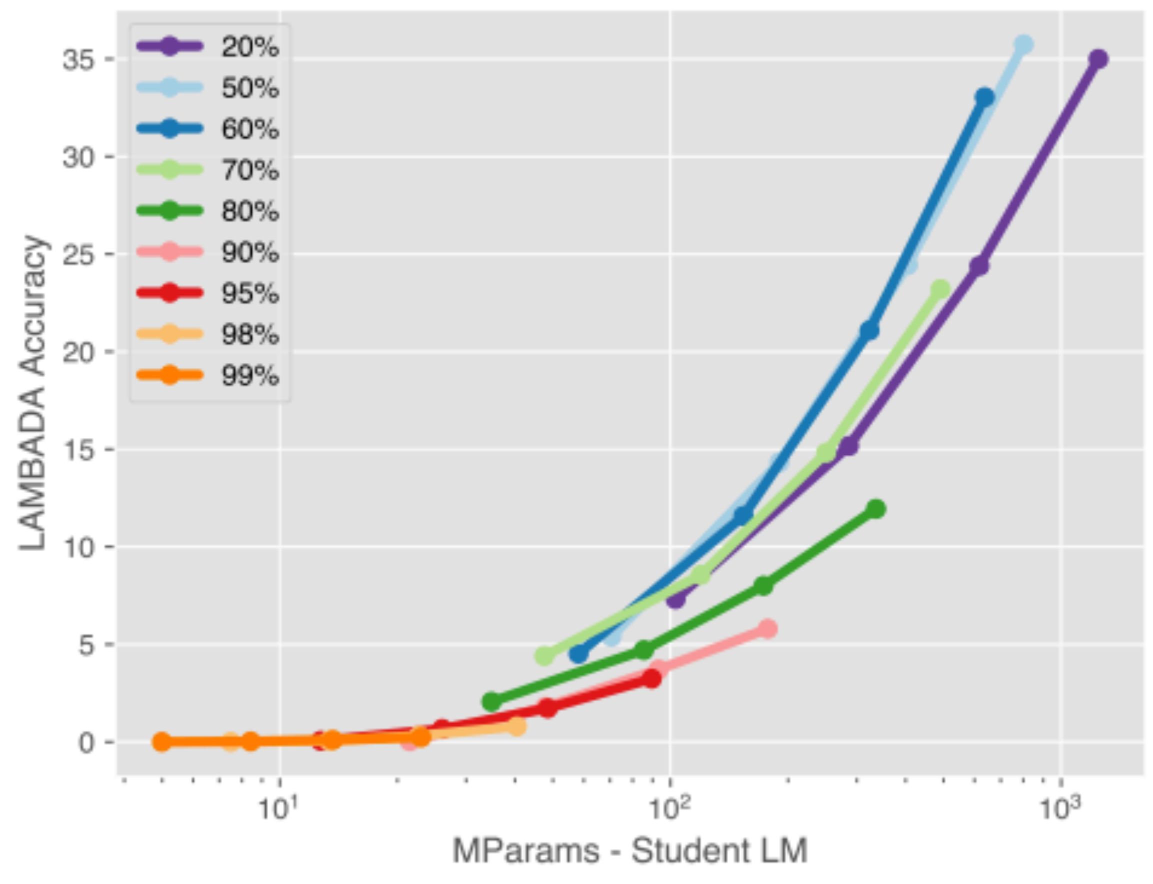
Democratization in Size

Impossible Triangle of Capacity Gap: Sustainability

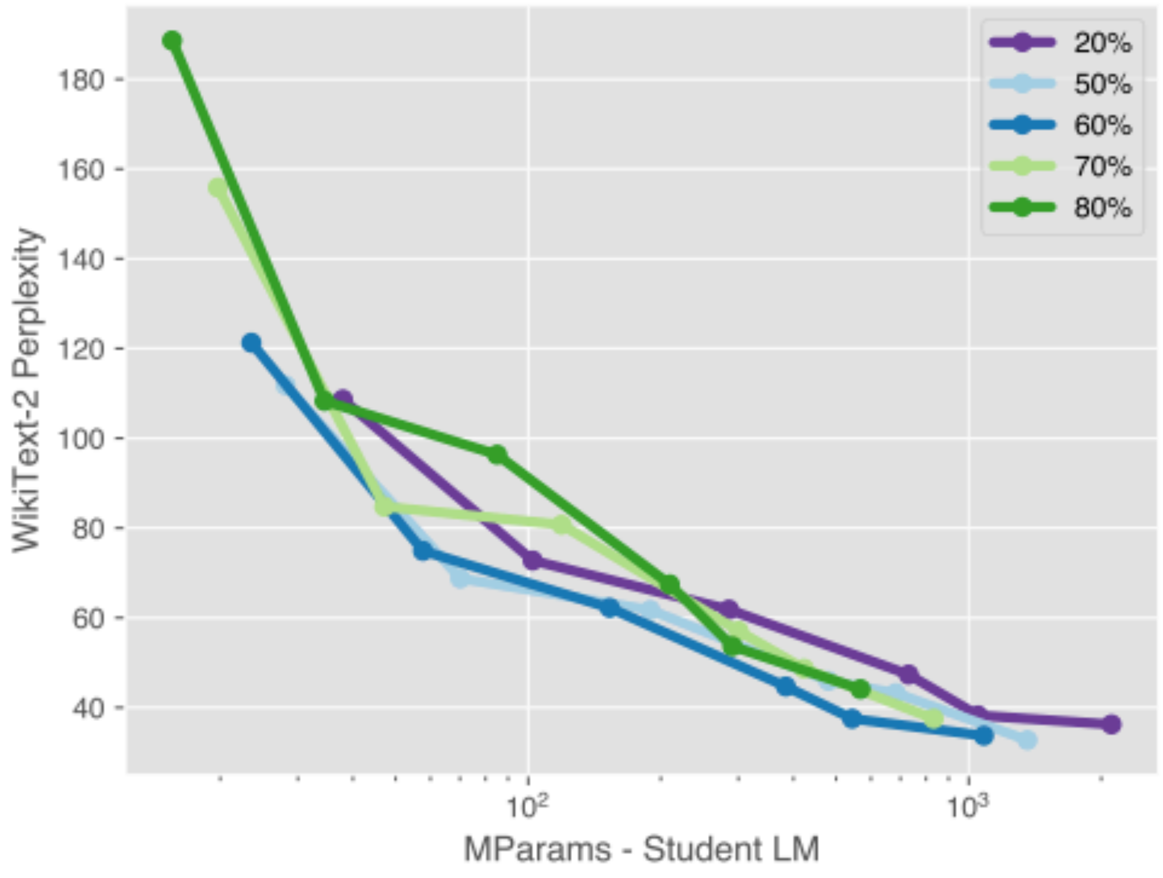
- Main results



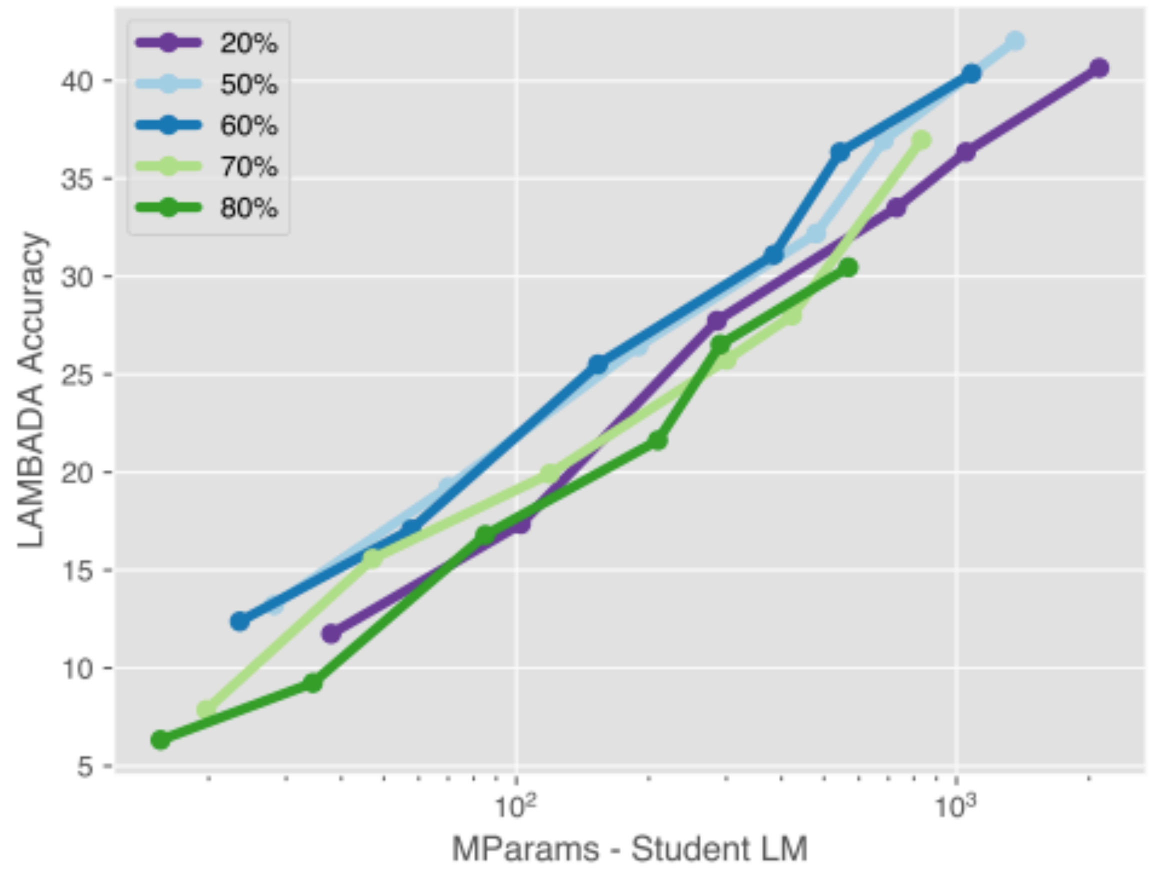
(a) GPT2 on WikiText2.



(b) GPT2 on LAMBADA.



(c) Pythia on WikiText2.



(d) Pythia on LAMBADA.

Democratization in Size

Impossible Triangle of Capacity Gap: Sustainability

- Main results

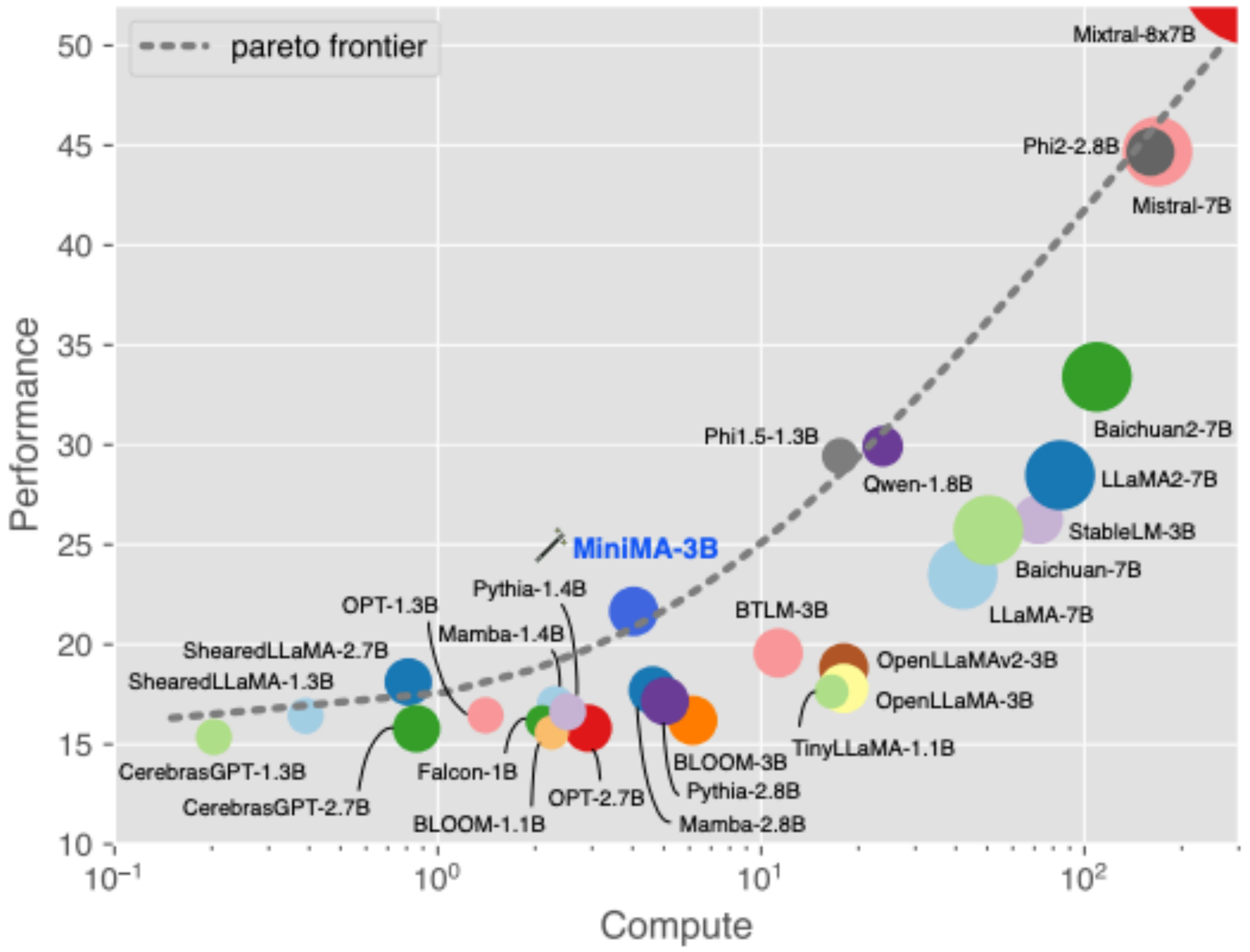


Figure 3: The new compute-performance pareto frontier is yielded by MINIMA, namely MINIMA is more compute-efficient given any compute budget. The radius of each circle stands for the model scale. *performance*: average value across different tasks in percent. *compute*: estimated training compute in $\times 10^9$ TFLOPs as detailed in Appendix A.

Table 3: The results of MINIMA on standard benchmarks. 5-shot direct prompting on MMLU and CEval. 3-shot direct prompting on DROP and BBH. 8-shot chain-of-thought prompting on GSM8K. 0-shot prompting on HumanEval. The result of CEval and DROP are reported on validation sets and those of the others on test sets. The results of 7B LMs are used as sanity checks of these benchmarks. The best and second best results among 3B LMs are underlined and **boldfaced** respectively.

LM	Tokens	MMLU Acc	CEval Acc	DROP EM Score	BBH EM Score	GSM8K Maj1@1	HumanEval Pass@1
LLaMA-7B (2023a)	1,000 B	35.10	28.00	27.46	30.93	9.17	10.37
LLaMA2-7B (2023b)	2,000 B	46.00	34.40	31.57	32.02	14.10	12.80
Baichuan-7B (2023)	1,200 B	42.60	43.50	19.82	31.94	8.57	7.93
Baichuan2-7B (2023)	2,600 B	54.31	55.27	25.97	35.21	13.19	16.46
Mistral-7B (2023)	–	62.67	45.91	46.59	43.88	41.02	28.05
Mamba-2.8B (2023)	300 B	25.58	24.74	15.72	29.37	3.49	7.32
ShearedLLaMA-2.7B (2023)	50 B	26.97	22.88	19.98	30.48	3.56	4.88
CerebrasGPT-2.7B (2023a)	53 B	24.66	23.18	11.46	29.32	2.43	3.66
OPT-2.7B (2022b)	180 B	26.02	24.52	13.70	28.71	1.90	0.00
BLOOM-3B (2022)	341 B	26.60	23.77	14.32	29.84	2.04	0.61
Pythia-2.8B (2023)	300 B	26.28	23.11	16.04	29.30	2.73	5.49
OpenLLaMA-3B (2023)	1,000 B	26.70	26.30	20.14	30.56	3.11	0.00
OpenLLaMAv2-3B (2023)	1,000 B	26.36	25.41	18.19	30.45	4.62	7.93
BTLM-3B (2023b)	627 B	27.20	26.00	17.84	30.87	4.55	**10.98**
StableLM-3B (2023)	4,000 B	44.75	31.05	**22.35**	32.59	10.99	15.85
MINIMA	126 B	**28.51**	**28.23**	22.50	**31.61**	**8.11**	**10.98**

Democratization in Size

Impossible Triangle of Capacity Gap: Sustainability

- Main results

Table 5: The results of MINICHAT on GPT4 assessments. Macro average scores are reported across fields in these two datasets. The better results are **boldfaced**.

LM Pair	Vicuna-Bench Macro Avg	BELLE-Bench Macro Avg
MINICHAT		
v.s. OpenBuddy-3B (2023)	**7.64**: 5.42	**7.77**: 6.81
v.s. BiLLa-7B (2023)	7.24: **7.41**	**7.73**: 7.49
v.s. ChatGLM-6B (2023b)	**7.63**: 5.63	**7.44**: 7.23
v.s. Phoenix-7B (2023)	**7.35**: 6.95	**7.62**: 7.21
v.s. ChatGLM2-6B (2023a)	**7.35**: 7.30	7.40: **8.00**

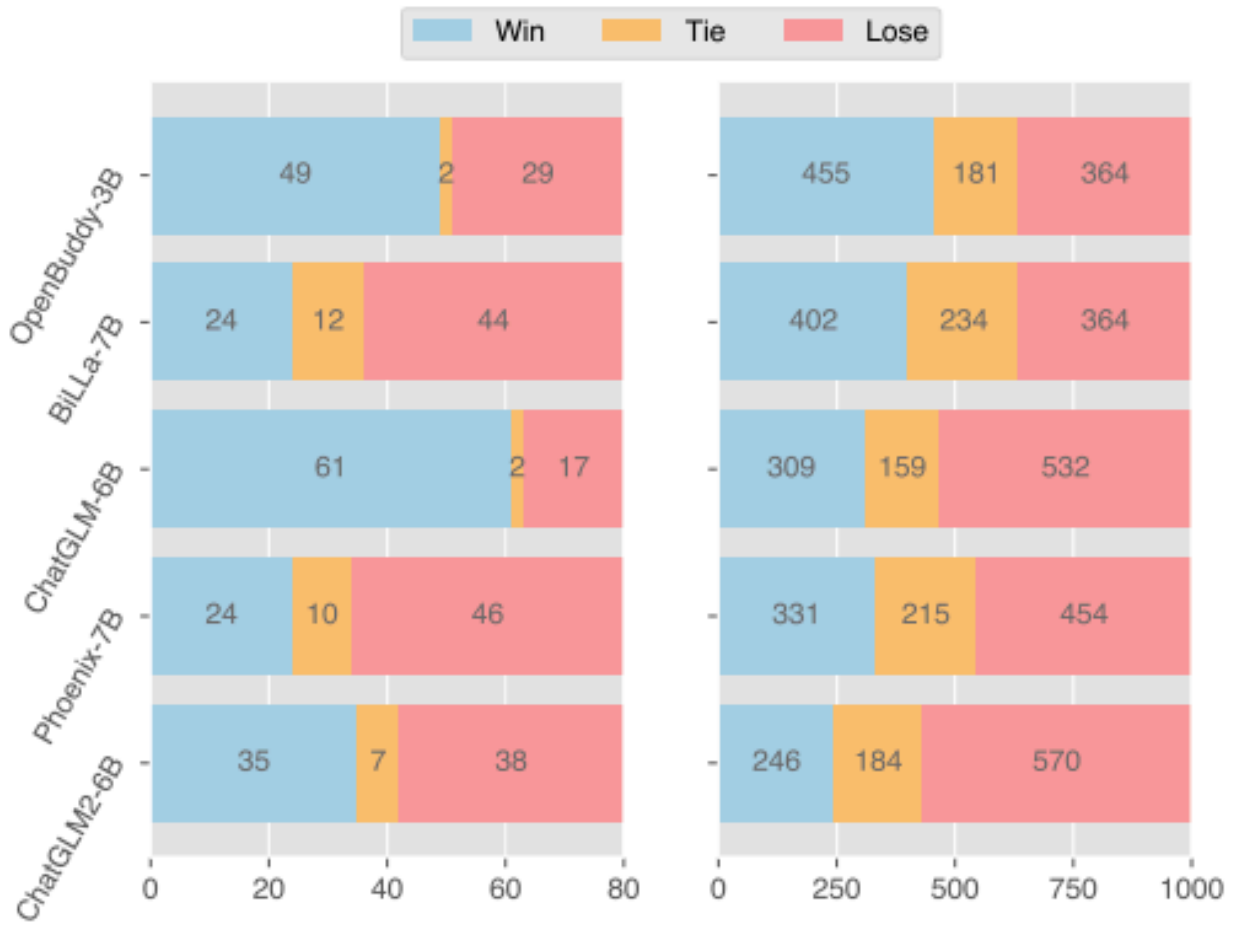


Figure 6: The win-lose-tie portions of MINICHAT versus its competitors. The left is reported on the Vicuna-Bench while the right is reported on the BELLE-Bench.

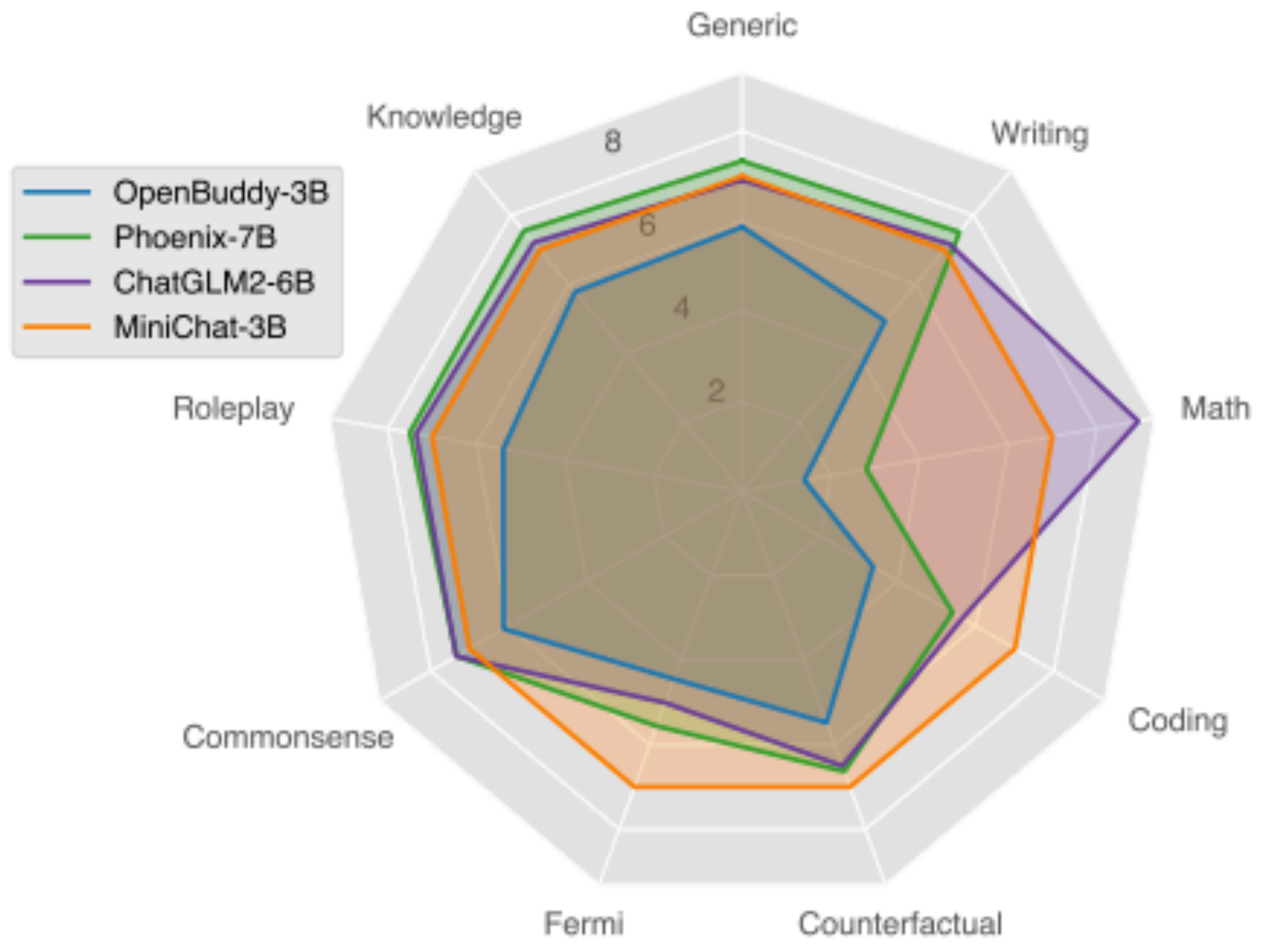


Figure 4: The reasonable ability radar is realized by MINICHAT. The head-to-head comparison scores on Vicuna-Bench of each baseline are rebased according to MINICHAT.

Democratization in Size

Impossible Triangle of Capacity Gap: Sustainability

- Ablation results

Table 4: The results of ablation studies.

LM	MMLU Acc	CEval Acc	DROP EM Score	BBH EM Score	GSM8K Maj1@1	HumanEval Pass@1
MINIMA	28.51	28.23	22.50	31.61	8.11	10.98
- global pruning priority	28.92	28.52	19.55	32.89	7.26	11.37
- from LLaMA2-13B	26.82	28.23	18.92	30.01	5.31	10.37
- from LLaMA2-70B	26.91	24.89	19.03	28.55	3.56	7.32
- without distillation	25.39	27.79	20.41	30.21	5.08	8.54
- with teacher assistant	27.69	27.49	21.31	30.02	7.72	9.58

Table 13: The results of MINICHAT on standard benchmarks.

LM	MMLU Acc	CEval Acc	DROP EM Score	BBH EM Score	GSM8K Maj1@1	HumanEval Pass@1
OpenLLaMA-3B (2023)	26.70	26.30	20.14	30.56	3.11	0.00
OpenBuddy-3B (2023)	23.88	24.67	15.89	29.08	11.07	3.66
Δ	-2.82	-1.63	-4.25	-1.48	+7.96	+3.66
MINIMA	28.51	28.23	22.50	31.61	8.11	10.98
MINICHAT	38.40	36.48	22.58	31.36	29.72	18.29
Δ	+9.89	+8.25	+0.08	-0.25	+21.61	+7.31

Democratization in Size

Impossible Triangle of Capacity Gap: Sustainability

- Cases

Question: Implement a program to find the common elements in two arrays without using any extra data structures.

OpenBuddy-3B: To find the common elements in two arrays without using any extra data structures, you can use the following algorithm:

1. Sort the two arrays in ascending order (using an appropriate sorting function).
2. Compare the first element of each array. If it's the same, then the rest of the array is common.
3. If the first element is different, then the rest of the array is common.
4. Repeat steps 2 and 3 until either all the elements in the first array are different or all the elements in the second array are different.

Here is the Python code for the algorithm:

```
def find_common_elements(arr1, arr2):
    sorted_arr1 = sorted(arr1)
    sorted_arr2 = sorted(arr2)
    for i in range(len(sorted_arr1)):
        if sorted_arr1[i] == sorted_arr2[i]:
            return sorted_arr1
    return []
```

This algorithm can be used to find the common elements in any two arrays.

ChatGLM2-6B: One way to implement a program to find the common elements in two arrays without using any extra data structures is to use a hash table. A hash table is a data structure that stores key-value pairs and allows for fast lookup, insertion, and deletion of elements.

To use a hash table to find the common elements in two arrays, you would first need to create a hash table with a key as the index and a value as the key-value pair. Then, for each element in the first array, you would use the hash table to look up the element and compare it to the elements in the second array. If a match is found, the element and its index would be added to the hash table.

Here is a simple example of how this could be implemented in Python:

```
class Element:
    def __init__(self, value):
        self.value = value

class HashTable:
    def __init__(self):
        self.table = {}

    def put(self, key, value):
        self.table[key] = value

    def get(self, key):
        return self.table.get(key)

    def contains(self, key):
        return self.table.contains(key)
```

```
# Example usage
element1 = Element('a')
element2 = Element('b')
table = HashTable()
table.put('a', element1)
table.put('b', element2)
print(table.get('a')) # prints the Element object for 'a'
print(table.contains('a')) # prints True
print(table.get('z')) # raises a KeyError
```

Note that this approach has a time complexity of O(n * m), where n and m are the lengths of the arrays, which may be slow for large arrays. There are other data structures that could be used to implement this, such as a linked list or a tree-based data structure, but a hash table is a simple and efficient solution for this problem.

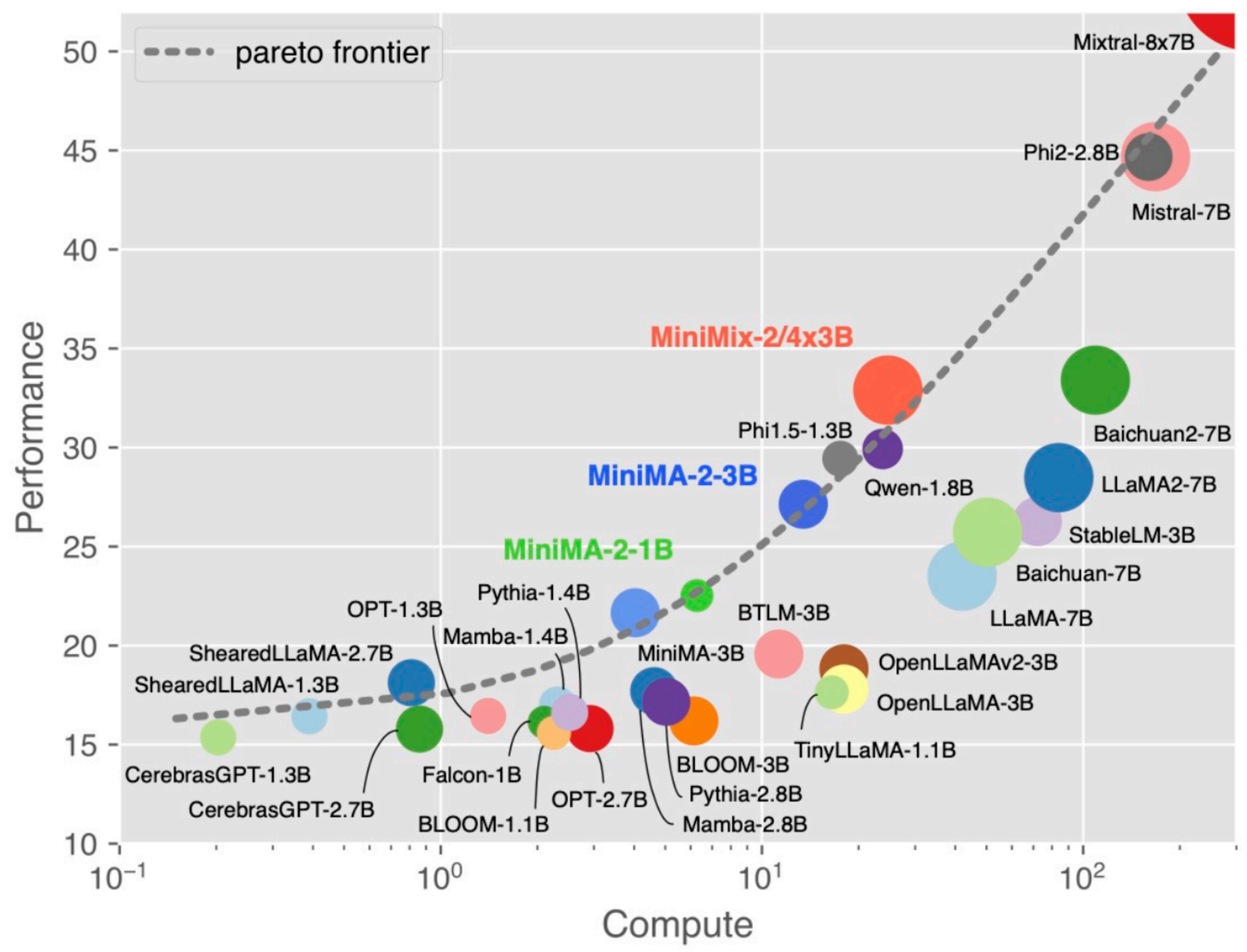
MiniChat: Here is a Python implementation:

```
def common_elements(arr1, arr2):
    if len(arr1) == 0:
        return []
    if len(arr2) == 0:
        return arr1
    common_elements = []
    for element in arr1:
        if element in arr2:
            common_elements.append(element)
    return common_elements
```

Democratization in Size

Impossible Triangle of Capacity Gap: Sustainability

- Extensions



Standard Benchmarks

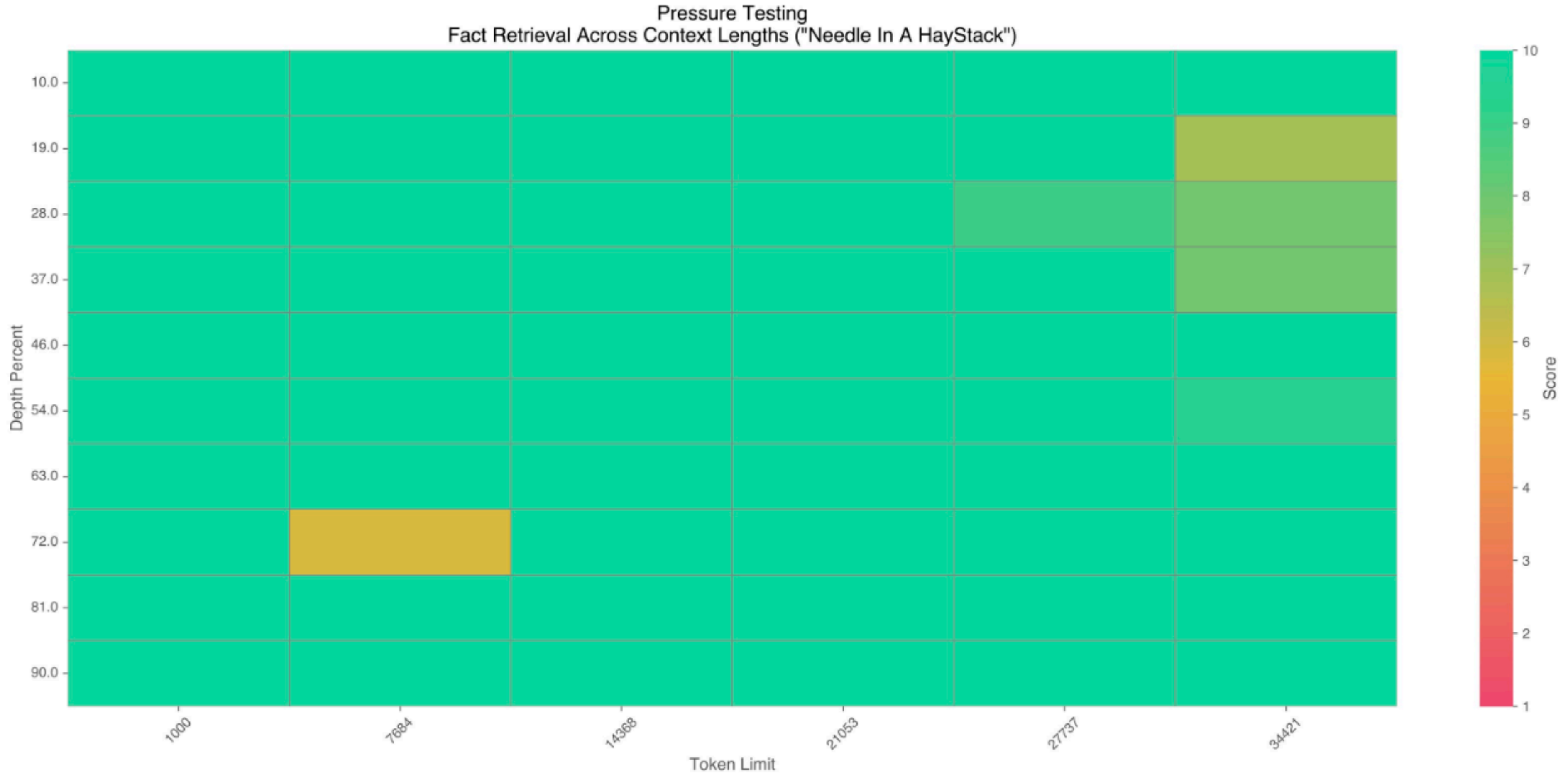
Method	TFLOPs	MMLU (5-shot)	CEval (5-shot)	DROP (3-shot)	HumanEval (0-shot)	BBH (3-shot)	GSM8K (8-shot)
Mamba-2.8B	4.6E9	25.58	24.74	15.72	7.32	29.37	3.49
ShearedLLaMA-2.7B	0.8E9	26.97	22.88	19.98	4.88	30.48	3.56
BTLM-3B	11.3E9	27.20	26.00	17.84	10.98	30.87	4.55
StableLM-3B	72.0E9	44.75	31.05	22.35	15.85	32.59	10.99
Qwen-1.8B	23.8E9	44.05	54.75	12.97	14.02	30.80	22.97
Phi-2-2.8B	159.9E9	56.74	34.03	30.74	46.95	44.13	55.42
LLaMA-2-7B	84.0E9	46.00	34.40	31.57	12.80	32.02	14.10
MiniMA-3B	4.0E9	28.51	28.23	22.50	10.98	31.61	8.11
MiniMA-2-1B	6.3E9	31.34	34.92	20.08	10.37	31.16	7.28
MiniMA-2-3B	13.4E9	40.14	44.65	23.10	14.63	31.43	8.87
MiniMix-2/4x3B	25.4E9	44.35	45.77	33.78	18.29	33.60	21.61
MiniChat-3B	4.0E9	38.40	36.48	22.58	18.29	31.36	29.72
MiniChat-2-3B	13.4E9	46.17	43.91	30.26	22.56	34.95	38.13

Instruction-following Benchmarks

Method	AlpacaEval	MT-Bench	MT-Bench-ZH
GPT-4	95.28	9.18	8.96
Zephyr-7B-Beta	90.60	7.34	6.27#
Vicuna-7B	76.84	6.17	5.22#
LLaMA-2-Chat-7B	71.37	6.27	5.43#
Qwen-Chat-7B	-	-	6.24
Phi-2-DPO	81.37	-	1.59#$
StableLM-Zephyr-3B	76.00	6.64	4.31#
Rocket-3B	79.75	6.56	4.07#
Qwen-Chat-1.8B	-	-	5.65
MiniChat-3B	48.82	-	-
MiniChat-2-3B	77.30	6.23	6.04

specialized mainly for English.

$ finetuned without multi-turn instruction data.



Democratization in Size

Law of Capacity Gap: External Evidence

- There are evident cases where similar law is implicitly applied yet not explicitly shown.
- Apple: Apple Foundation Model (https://arxiv.org/pdf/2407.21075)
- Google: Gemma-2 (https://arxiv.org/pdf/2408.00118)
- Nvidia: Minitron (https://arxiv.org/pdf/2407.14679v1)

AFM-on-device: For the on-device model, we found that knowledge distillation [Hinton et al., 2015] and structural pruning are effective ways to improve model performance and training efficiency. These two methods are complementary to each other and work in different ways. More specifically, before training AFM-on-device, we initialize it from a pruned 6.4B model (trained from scratch using the same recipe as AFM-server), using pruning masks that are learned through a method similar to what is described in [Wang et al., 2020; Xia et al., 2023]. The key differences are: (1) we only prune the hidden dimension in the feed-forward layers; (2) we use Soft-Top-K masking [Lei et al., 2023] instead of HardConcrete masking [Louizos et al., 2018]; (3) we employ the same pre-training data mixture as the core phase to learn the mask, training for 188B tokens. Then, during the core pre-training of AFM-on-device, a distillation loss is used by replacing the target labels with a convex combination of the true labels and the teacher model's top-1 predictions, (with 0.9 weight assigned to the teacher's labels), training for a full 6.3T tokens. We observe

Model	Tokens	Hellaswag	MMLU
4B-Random-Init	150B*	46.22	24.36
4B-Random-Init	400B	48.23	26.24
4B-Pruned (prune Nemotron-4 15B)	150B*	50.85	24.57
4B-Pruned-Distill (prune Nemotron-4 15B)	**100B***	**51.04**	**37.81**
4B-Pruned-Distill (prune MINITRON 8B)	**100B***	**52.04**	**42.45**

Table 12: Accuracy comparison for 4B using the 8T blend. * Indicates settings with iso-compute.

3.2. Knowledge Distillation

Given a large model used as a teacher, we learn smaller models by distilling from the probability given by the teacher of each token x given its context x_c, i.e., $P_T(x \mid x_c)$. More precisely, we minimize the negative log-likelihood between the probabilities from the teacher and the student:

$$\min_{P_S} \sum_x -P_T(x \mid x_c) \log P_S(x \mid x_c),$$

where P_S is the parameterized probability of the student. Note that knowledge distillation was also used in Gemini 1.5 (Gemini Team, 2024).

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

Table 7 | Perplexity measured on a validation set of models of different sizes trained with or without distillation. The teacher has 7B parameters.

Future Work

- Law of capacity gap with data amount considered.
- Vocabulary-agnostic distillation (e.g., from LLaMA-3.1-405B).
- Multi-teacher distillation (e.g., from generalist/code/math models).
- RLHF with distillation (e.g., DPO with a large teacher).
- etc.

Thanks