

Less Data, Faster Training

repeating smaller datasets speed up learning



With: Jingwen Liu



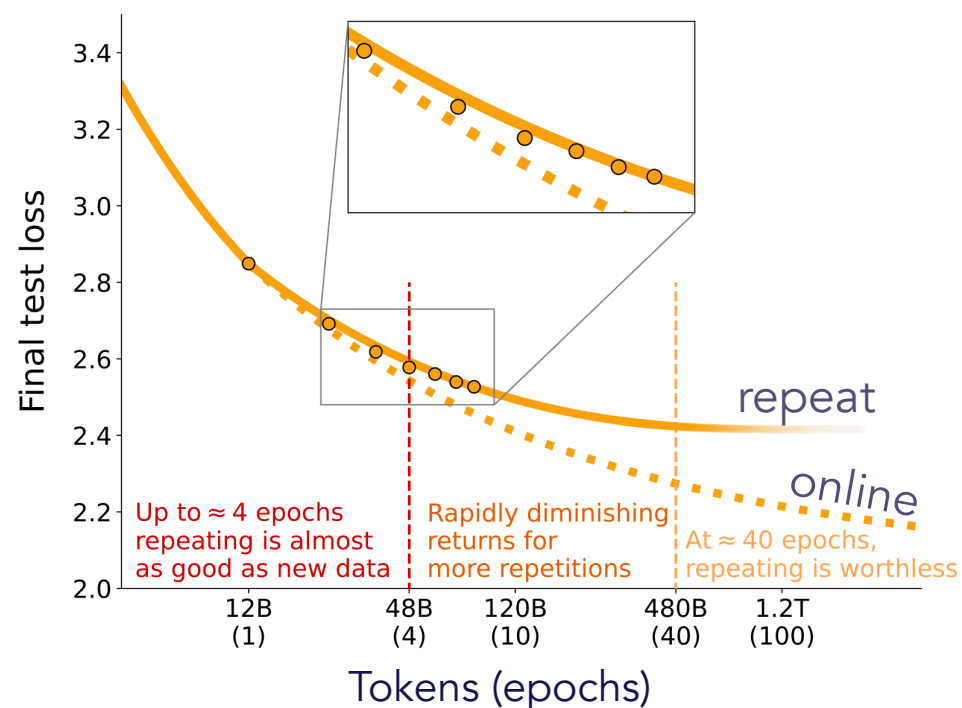
Ezra Edelman



Surbhi Goel

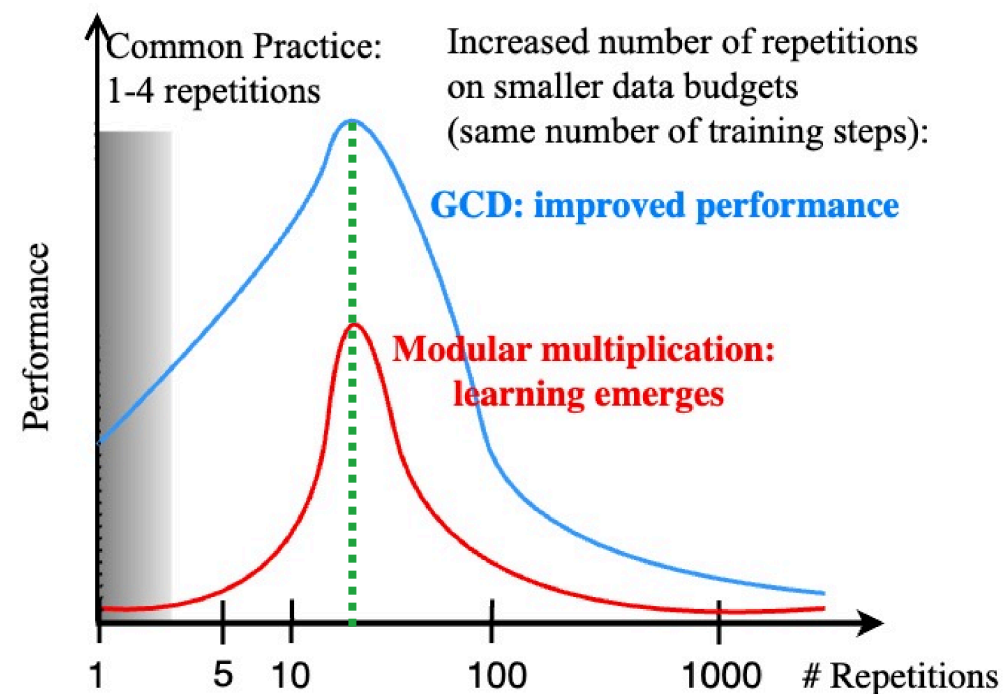
Less data means...

... worse test loss.



Muennighoff et al. 23

... but *can* be **better** under same compute?



Charton & Kempe 24

Less data, faster training



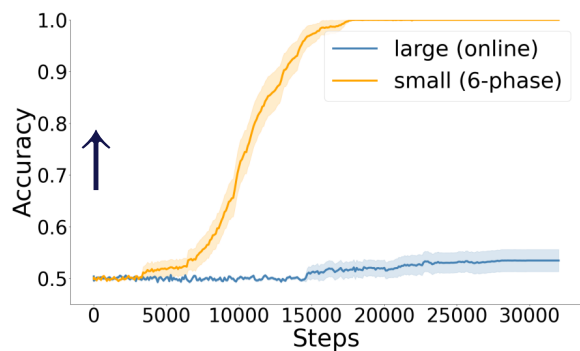
1. Observed across setups. ... *task, architecture, optimizer.*
2. Prior explanations are insufficient.
3. Our explanation: sampling biases from finite samples.

* *Data repetition as favorable optimization bias*

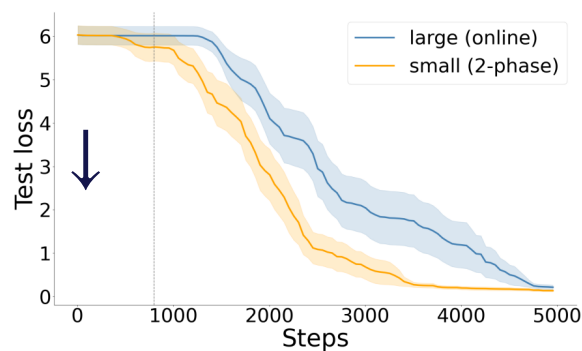
Small-vs-large gap in various settings

→ *random* subset

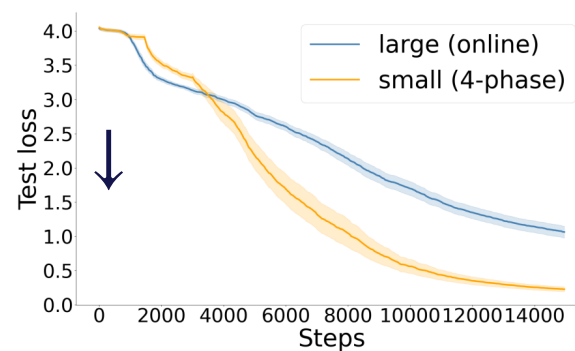
Transformer + AdamW on mini-batches



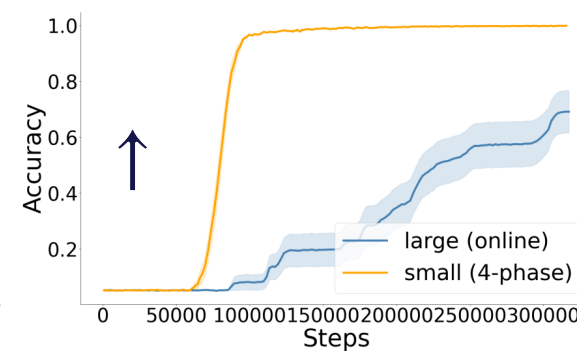
(a) Sparse parity



(b) Single-index model



(c) ICL linear regression

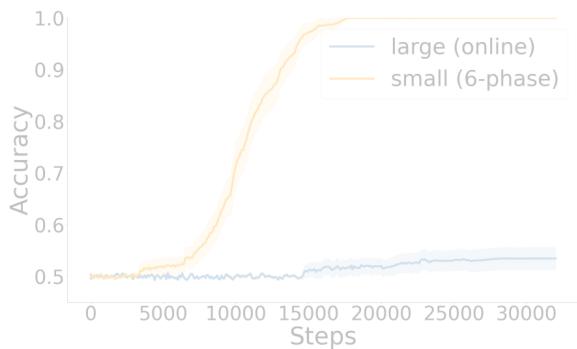
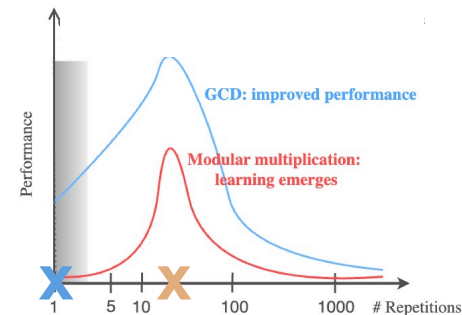


(d) Modular addition

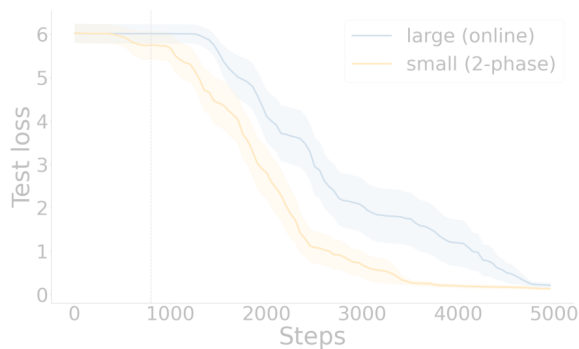
Small-vs-large gap

→ *random* subset

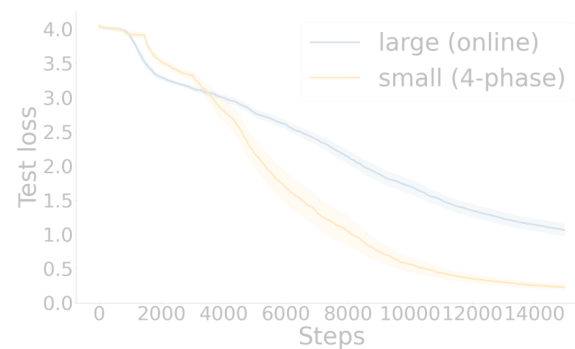
Transformer + AdamW on mini-batches



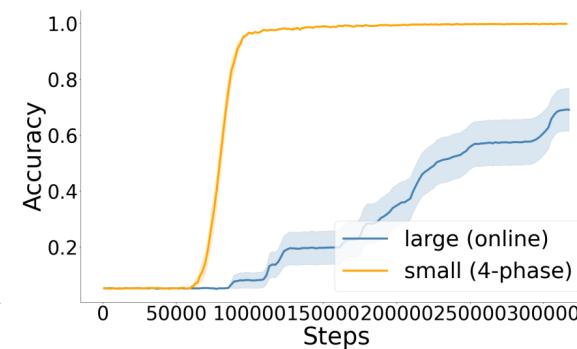
(a) Sparse parity



(b) Single-index model



(c) ICL linear regression



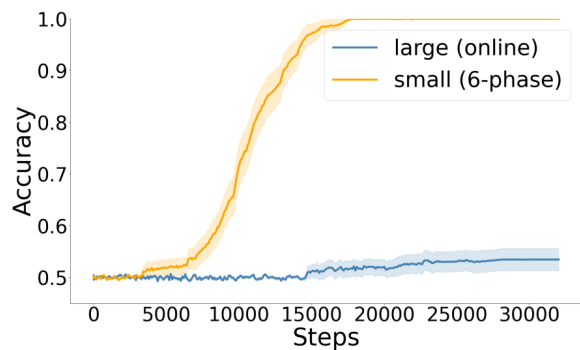
(d) Modular addition

$$x, z \sim \text{Unif}([M]), \quad y = (x + z) \bmod p$$

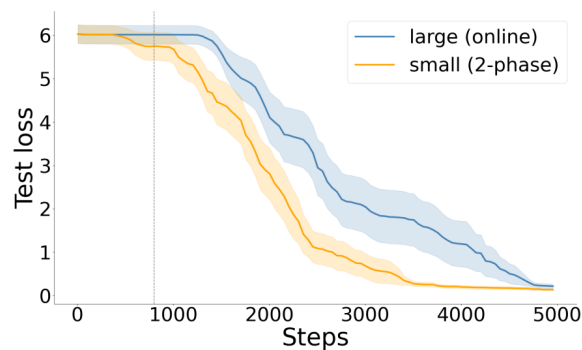
Small-vs-large gap

→ *random* subset

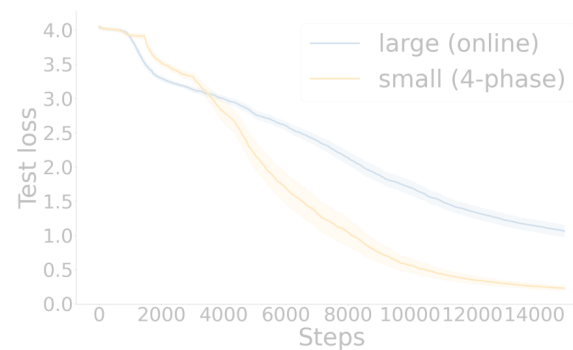
Transformer + AdamW on mini-batches



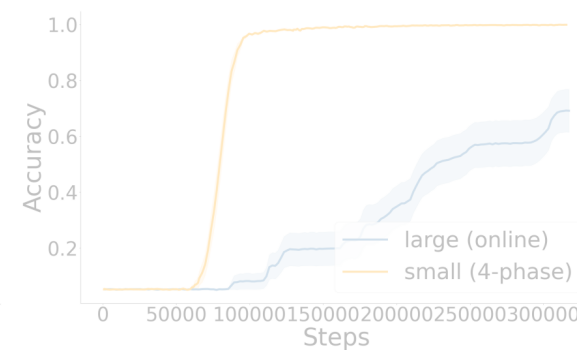
(a) Sparse parity



(b) Single-index model



(c) ICL linear regression



(d) Modular addition

$$x \in \{\pm 1\}^d, \quad y = \prod_{i \in S} x_i \in \{\pm 1\}$$

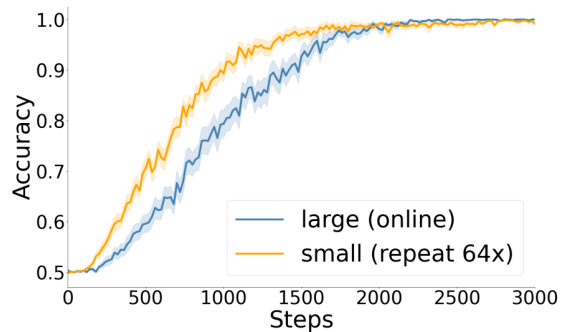
$$x \sim \mathcal{N}(0, I_d), \quad y = \sigma(\langle w^*, x \rangle)$$

Small-vs-large gap

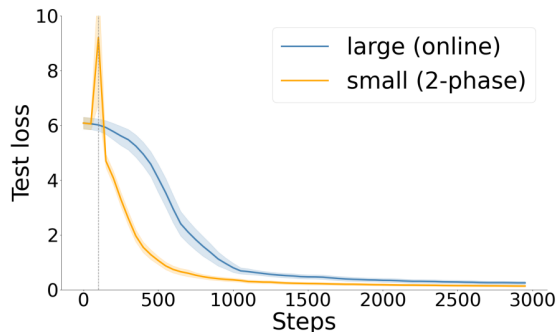
→ *random* subset

MLP + SGD optimizer

mini batches

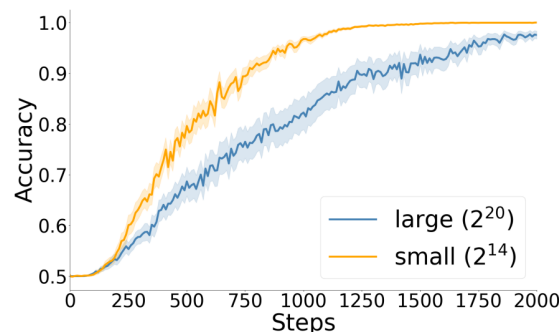


sparse parity

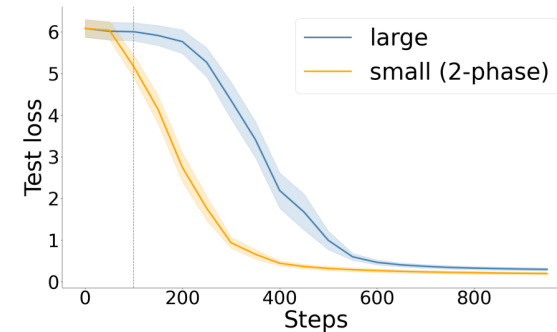


single-index model

full batches



sparse parity

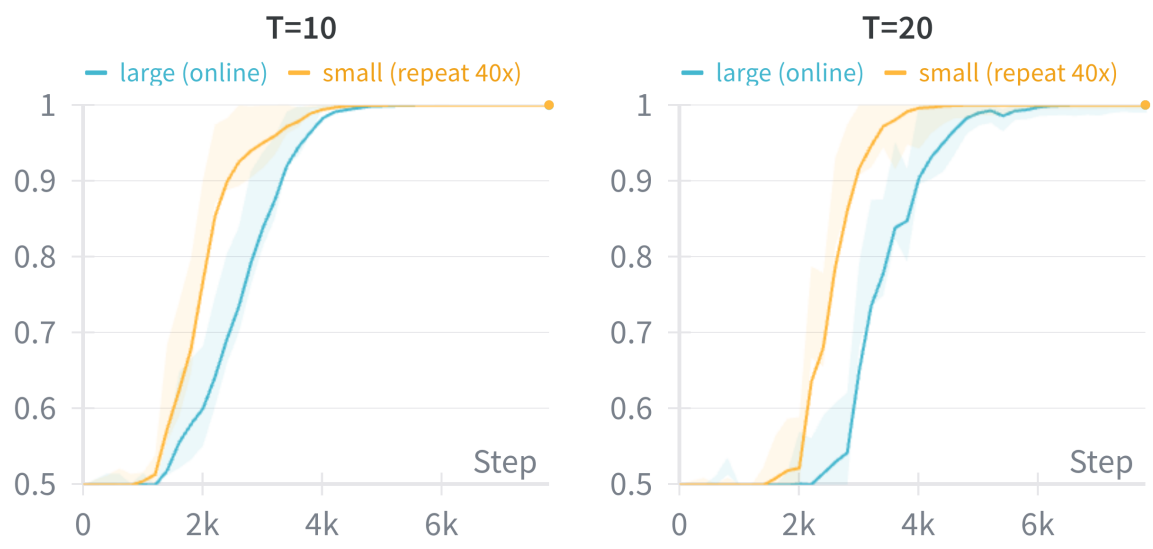
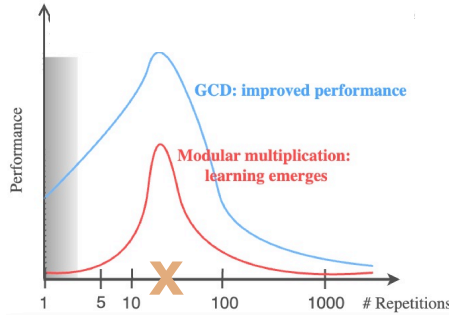


single-index model

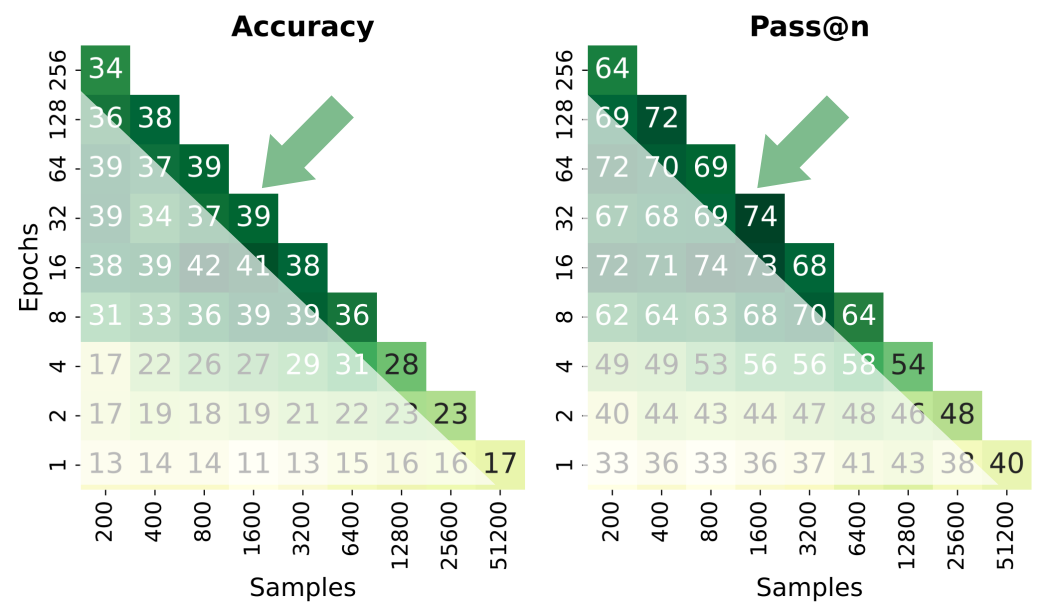
Small-vs-large gap ... *why?*

→ *random* subset

LLM finetuning



Qwen3 SFT on full parity



Olmo3 long-CoT SFT for AIME24/25, GPQA
(Kopiczko et al. 26)

Less data, faster training

“small-vs-large gap” in compute

1. Observed across setups. ... *task, architecture, optimizer.*
- 2. Prior explanations are insufficient. ... *thoughts?* 🤔**
3. Our explanation: sampling biases from finite samples.
theory + empirical evidence

Attempt 1: CSQ-SQ distinction

- SQ (statistical query): oracle returns a noisy version of $\mathbb{E}[\phi(x, y)]$.
- CSQ (correlational SQ): $\dots \dots \dots \mathbb{E}[\phi(x)y]$.

Attempt 1: CSQ-SQ distinction

Idea: SGD gets more powerful when taking >1 gradient steps on the same batch.

(i.e. CSQ $\rightarrow$ SQ)

$\mathbb{E}[\phi(x)y]$ vs $\mathbb{E}[\phi(x, y)]$

Why online SGD is CSQ:

$$\nabla_{\mathbf{w}} \mathbb{E}_{\mathbf{x}, y} (f_{\mathbf{w}}(\mathbf{x}) - y)^2 \propto \underbrace{-\mathbb{E}_{\mathbf{x}, y} [y \cdot \nabla_{\mathbf{w}} f_{\mathbf{w}}(\mathbf{x})]}_{\text{correlational query}} + \mathbb{E}_{\mathbf{x}} [\underbrace{f_{\mathbf{w}}(\mathbf{x}) \cdot \nabla_{\mathbf{w}} f_{\mathbf{w}}(\mathbf{x})}_{\text{can be evaluated without } y}].$$

Attempt 1: CSQ-SQ distinction

Idea: SGD gets more powerful when taking >1 gradient steps on the same batch.

(i.e. CSQ $\rightarrow$ SQ)

$\mathbb{E}[\phi(x)y]$ vs $\mathbb{E}[\phi(x, y)]$

SGD moves to SQ with >1 gradient steps: for the 2nd step,

$$\boldsymbol{w}^2 = \boldsymbol{w}^1 + \eta \cdot y \sigma'(\langle \boldsymbol{x}, \boldsymbol{w}^1 \rangle) \boldsymbol{x} = \eta \sigma'(0) \underbrace{y \cdot \boldsymbol{x}}_{\text{CSQ term}} + \underbrace{\eta y \sigma'(\eta \sigma'(0) \|\boldsymbol{x}\|^2 \cdot y) \boldsymbol{x}}_{\text{non-CSQ term}}.$$

Attempt 1: CSQ-SQ distinction

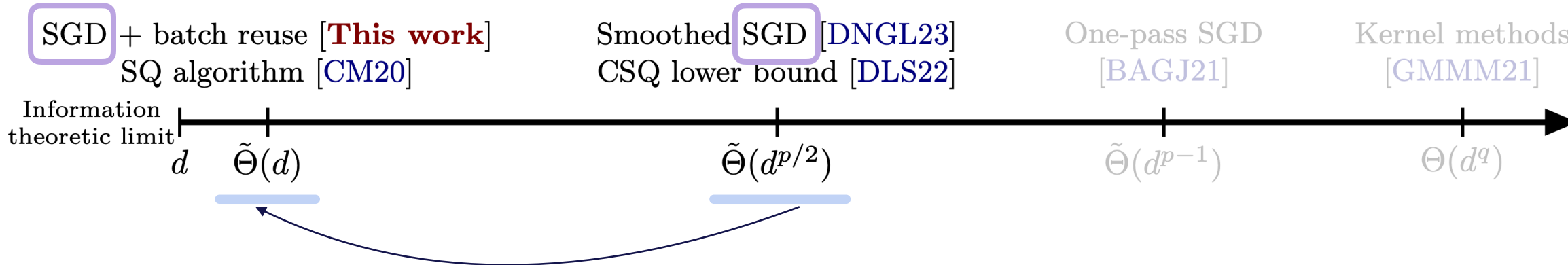
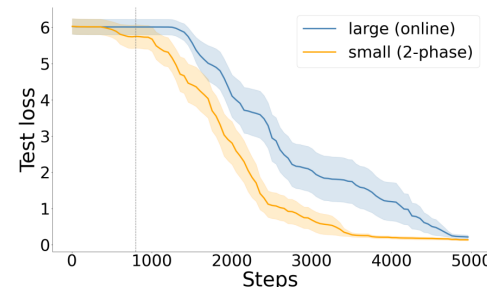
Idea: SGD gets more powerful when taking >1 gradient steps on the same batch.

(i.e. CSQ $\rightarrow$ SQ)

$$\mathbb{E}[\phi(x)y] \text{ vs } \mathbb{E}[\phi(x, y)]$$

Single-index model

$$x \sim \mathcal{N}(0, I_d), \quad y = \sigma(\langle w^*, x \rangle)$$

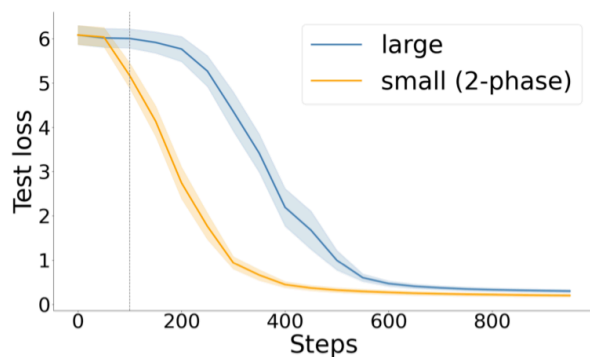


Attempt 1: CSQ-SQ distinction

Idea: SGD gets more powerful when taking >1 gradient steps on the same batch.
(i.e. CSQ $\rightarrow$ SQ)

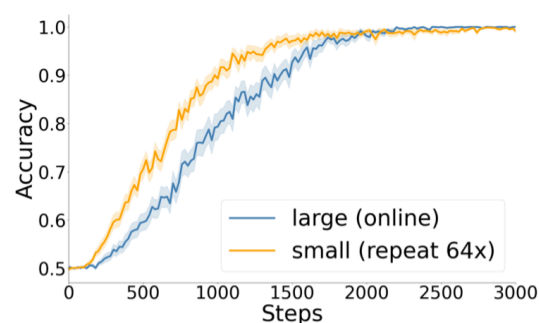
Fail to explain the small-vs-large gap for:

1) Full-batch updates (GD);

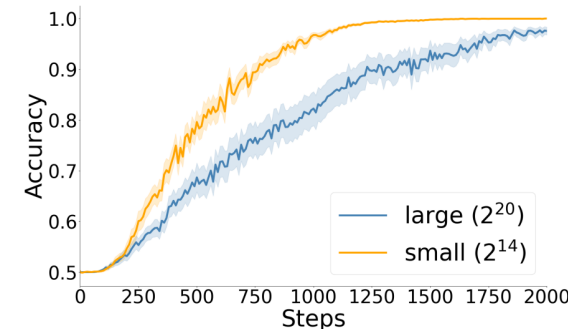


single-index model

2) Sparse parity.



mini-batch

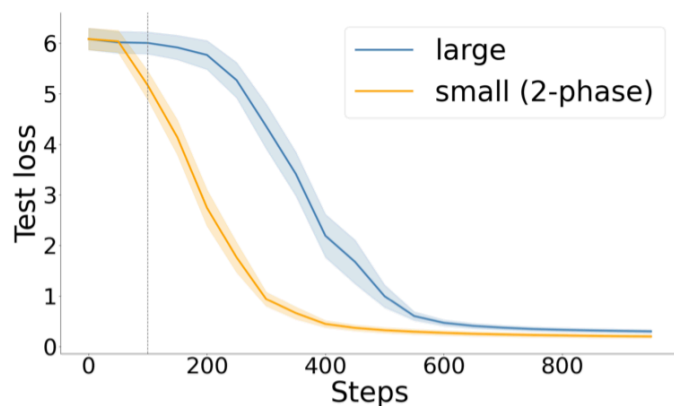


full-batch

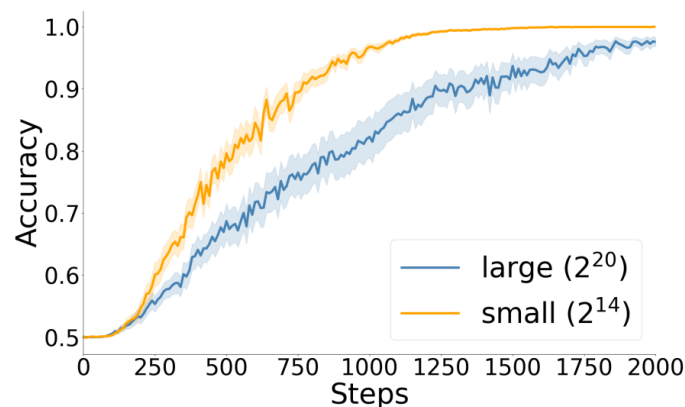
Attempt 2: Gradient variance

Idea: smaller datasets induce smaller gradient variances from batch sampling.

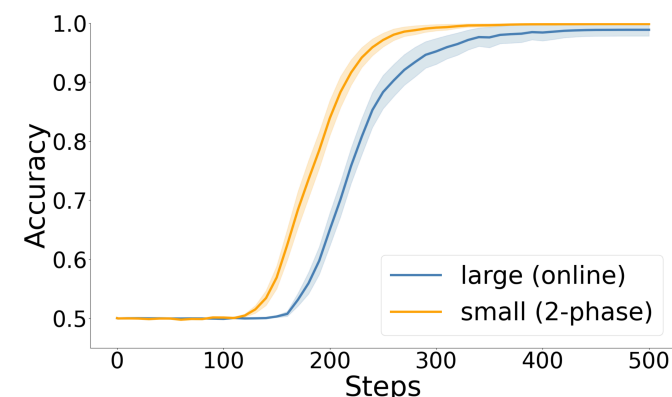
Fail to explain: the small-vs-large gap on **full-batch** training.



single-index model, MLP



(20, 6)-parity, MLP



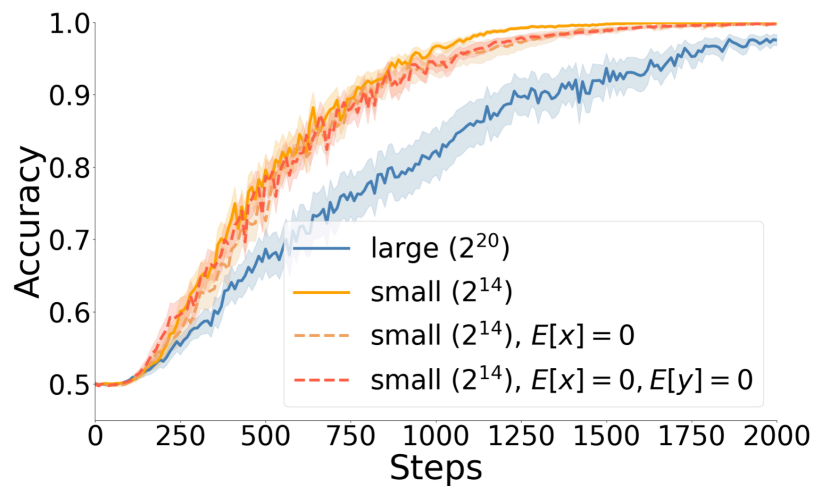
(10, 6)-parity, Transformer

Attempt 3: biased inputs

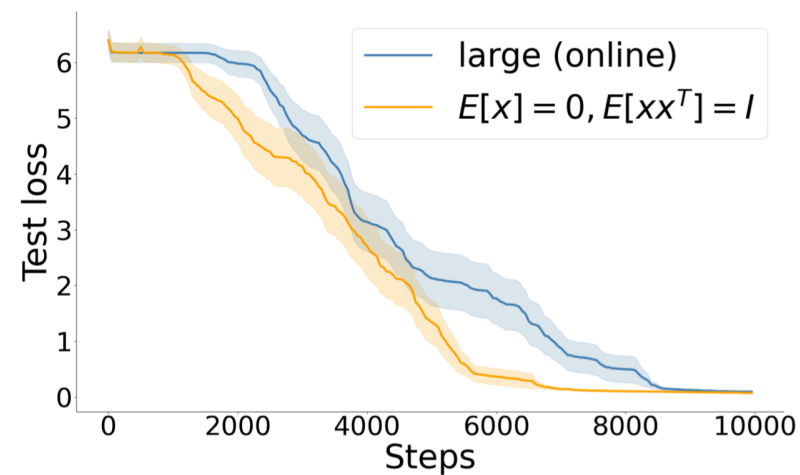
Idea (parity): biased distribution (i.e., $x_i \sim \text{Ber}(p)$ for $p \neq 0.5$) is helpful.

→ Small subsets help because the empirical distribution has $\hat{p} \neq 0.5$?

Fail to explain: the small-vs-large gap when 1) **ensuring** $\hat{p} = 0.5$. (spoiler: we use $\hat{\mathbb{E}}[yx^{\otimes k}]$'s bias)



(a) (20,6)-parity, accuracy



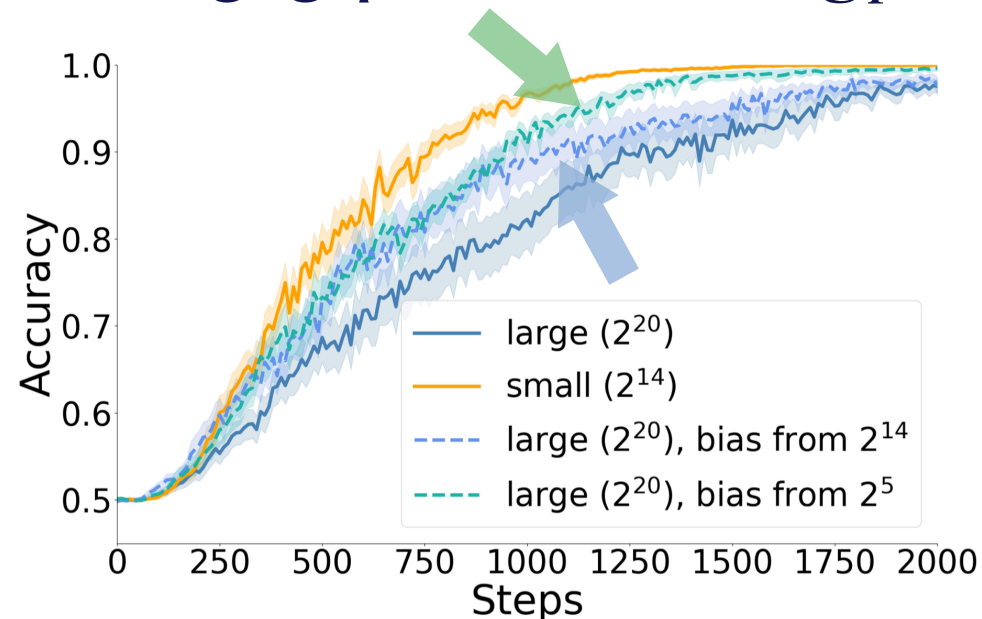
(b) SIM, loss

Attempt 3: biased inputs

Idea (parity): biased distribution (i.e., $x_i \sim \text{Ber}(p)$ for $p \neq 0.5$) is helpful.

→ Small subsets help because the empirical distribution has $\hat{p} \neq 0.5$?

Fail to explain: the small-vs-large gap when 2) **biasing p for large-set SGD.**
(i.e. online)



Attempt 3: biased inputs

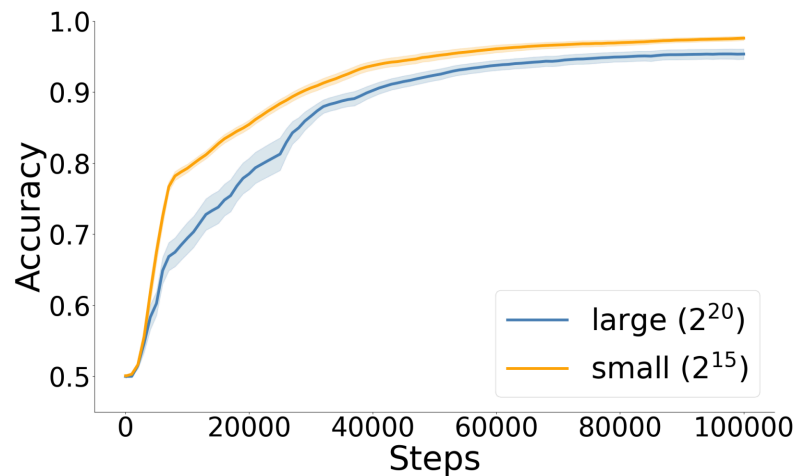
sparsity ← sample size

$$(\hat{p} - p)^k \text{ vs } N^{-1/2}$$

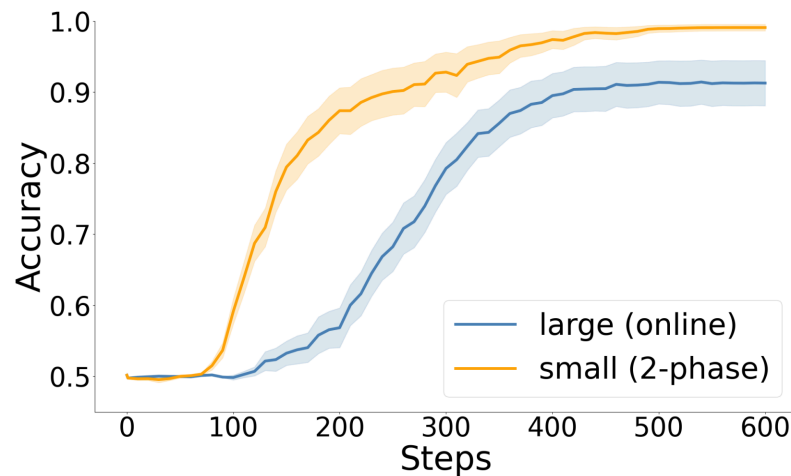
Idea (parity): biased distribution (i.e., $x_i \sim \text{Ber}(p)$ for $p \neq 0.5$) is helpful.

→ Small subsets help because the empirical distribution has $\hat{p} \neq 0.5$?

Fail to explain: the small-vs-large gap when 3) k is large (e.g. dense parity).



(a) (20, 20)-parity, MLP



(b) (10, 10)-parity, Transformer

Prior explanations are insufficient

1. **CSQ vs SQ:** can't explain **full-batch (GD)** or **parity**.
2. **Gradient variance reduction:** can't explain **full-batch (GD)**.
3. **Biased input distributions:** different **magnitude**; requiring **sparsity**.

Less data, faster training

“small-vs-large gap” in compute

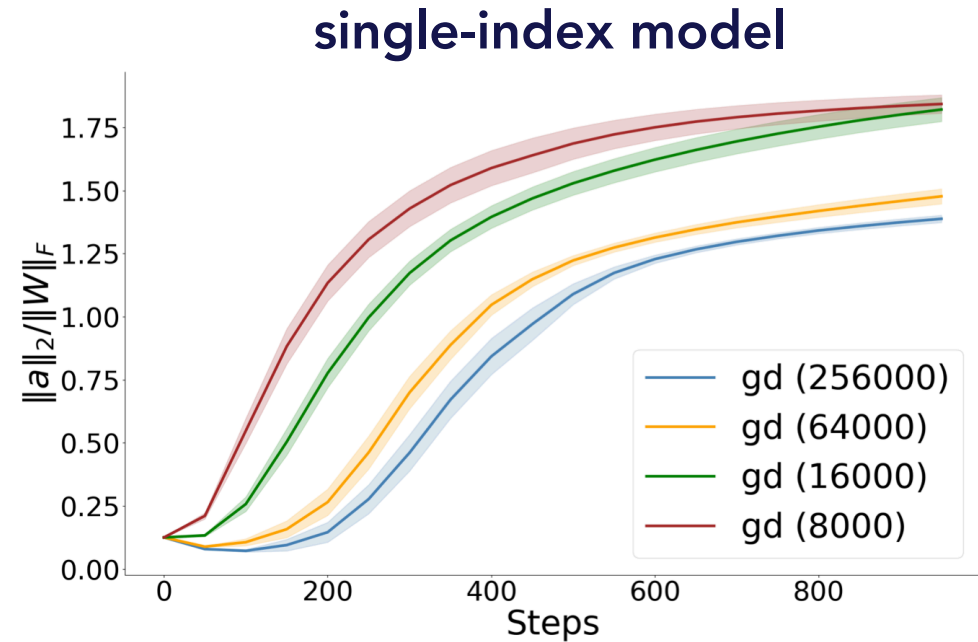
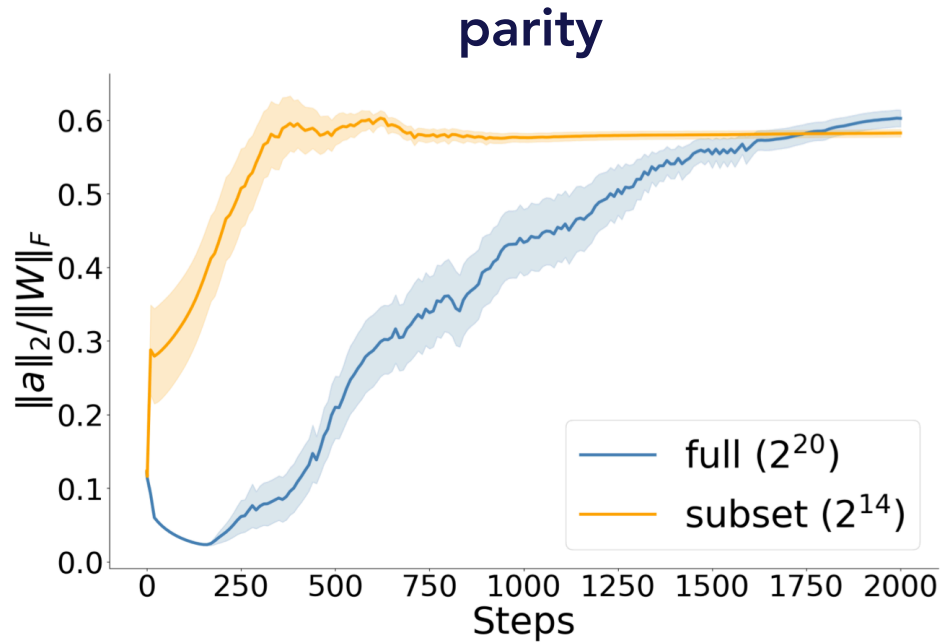
1. Observed across setups.
2. Prior explanations are insufficient.
3. Our explanation: **sampling biases** adjust layer-wise learning speed.

$$\asymp N^{-1/2}$$

Sampling bias adjusts layer-wise learning

Sampling biases ($\asymp N^{-1/2}$) lead to favorable *relative norm growth* to escape saddle.

- For $f(x) = a^\top \sigma(Wx)$, smaller N grows $\|a\|_2 / \|W\|_F$ faster.



Formalize with 2-phase training

Model: $f(x) = a\sigma(w^\top x) - 1$, where $\sigma(z) = \frac{1}{2}z^2$.

2-phase training on 2-sparse parity:
(projected updates with correlation loss)

- Phase 1: on a random size- N set ($d \leq N \leq d^2$);
- Phase 2: on the full population.

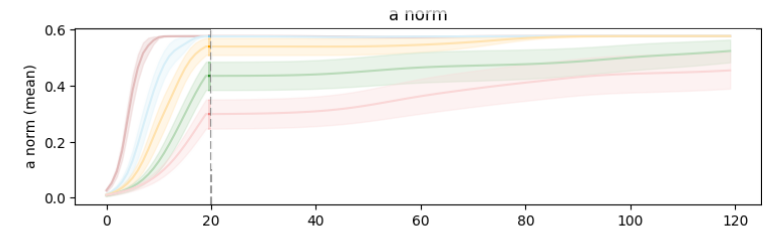
With proper transition, the total #steps is

$$\tilde{O}_{\epsilon,\delta}((Nd)^{1/4}).$$

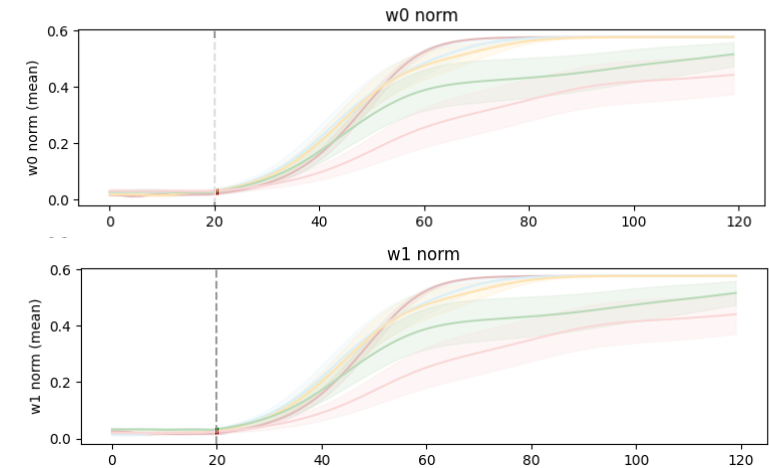
a 's gradient $\asymp N^{-1/2}$

due to $\hat{M} := \hat{\mathbb{E}}[yxx^\top]$

Phase 1: smaller $N \rightarrow a$ grows faster.



Phase 2: larger $a \rightarrow w$ grows faster.



Sampling bias adjusts layer-wise learning

Idea: *sampling biases* ($\asymp N^{-1/2}$) lead to favorable **relative norm growth**.

- For $f(x) = a^\top \sigma(Wx)$, smaller N grow $\|a\|_2 / \|W\|_F$ faster due to $\hat{M} := \hat{\mathbb{E}}[yxx^\top]$.

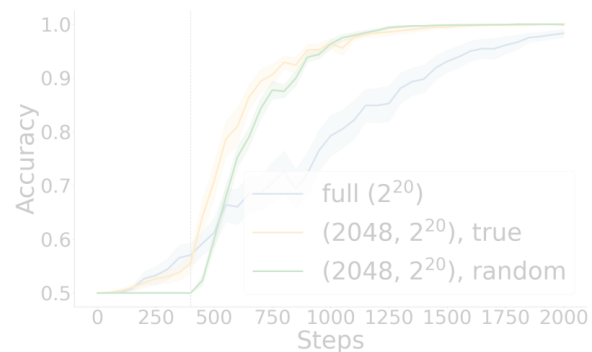
Predictions:

1. Small-set training of *random labels* exists small-vs-large gap.
2. Proper *layer-wise interventions* reduce/remove the gap.

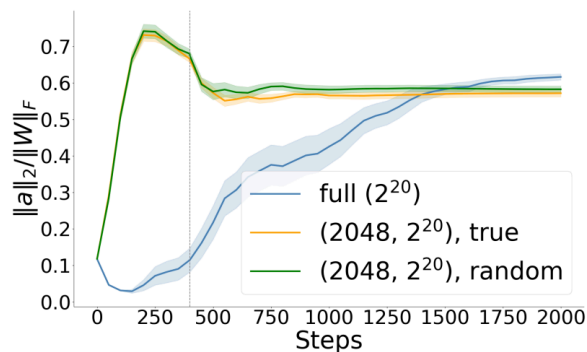
1. Small-vs-large gap from random labels

2-phase training: 1) small-set random labels; 2) large-set training.
(no signal & can only affect norms)

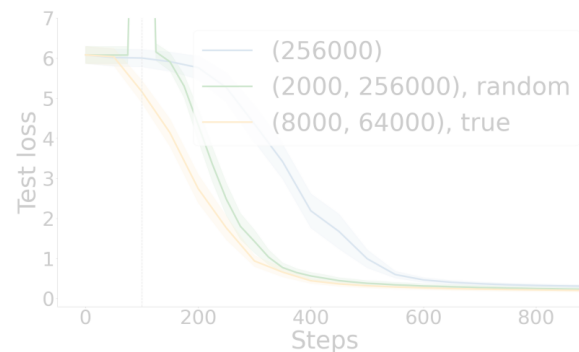
With small-set random labels: faster growth of $\|a\|_2/\|W\|_F$.



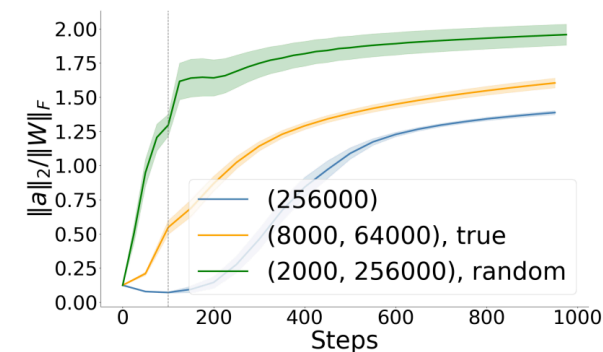
(20,6)-parity, acc.



(20,6)-parity, $\frac{\|a\|_2}{\|W\|_F}$



SIM, $d = 40$, loss

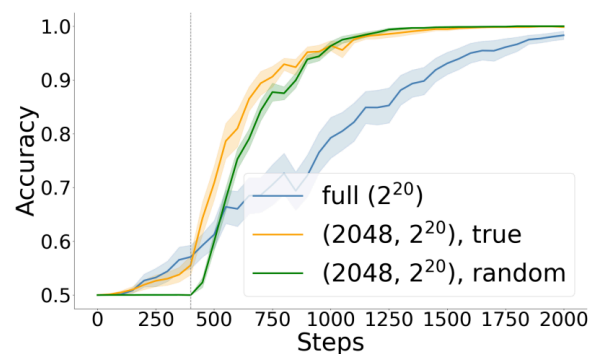


SIM, $d = 40$, $\frac{\|a\|_2}{\|W\|_F}$

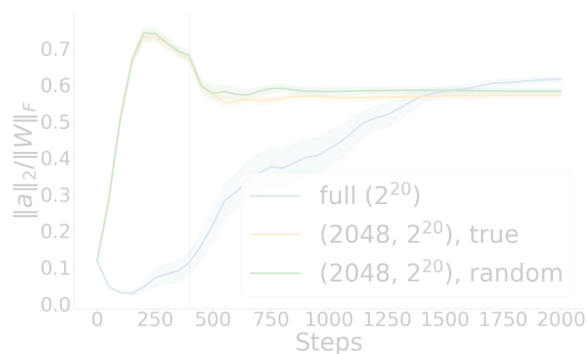
1. Small-vs-large gap from random labels

2-phase training: 1) small-set random labels; 2) large-set training.
(no signal & can only affect norms)

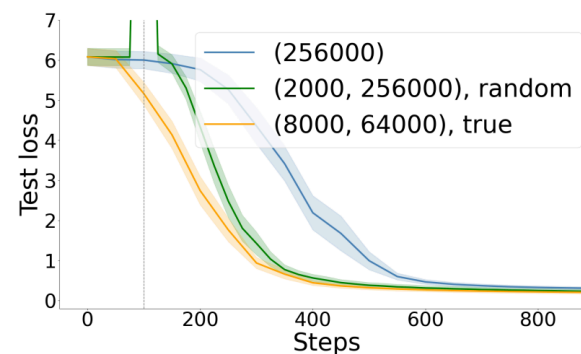
With small-set random labels: faster learning.



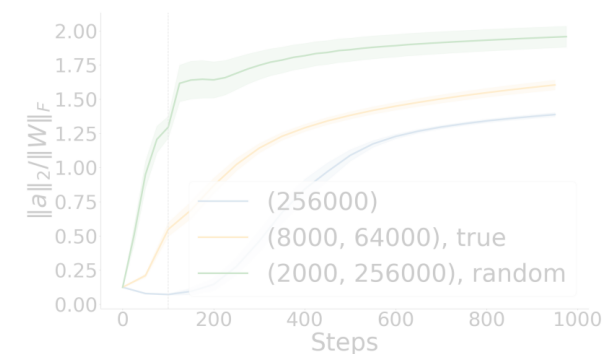
(20,6)-parity, acc.



(20,6)-parity, $\frac{\|a\|_2}{\|W\|_F}$



SIM, $d = 40$, loss

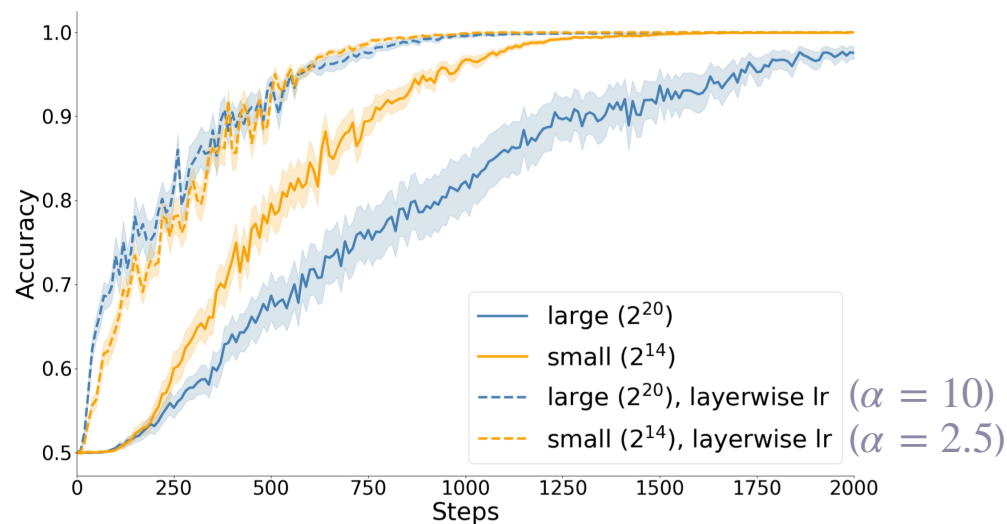


SIM, $d = 40$, $\frac{\|a\|_2}{\|W\|_F}$

2. Layer-wise interventions bridges the gap

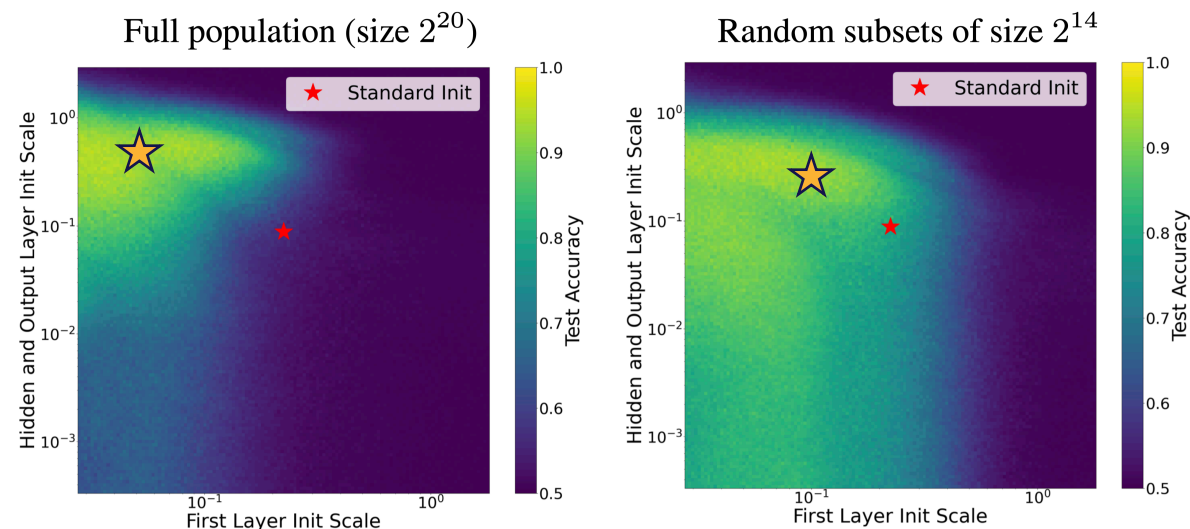
Learning rate

$$\eta_w = \eta \cdot \alpha, \eta_a = \eta / \alpha$$



Initialization

$$\tilde{w} = c_1 w, \tilde{a} = c_2 a$$

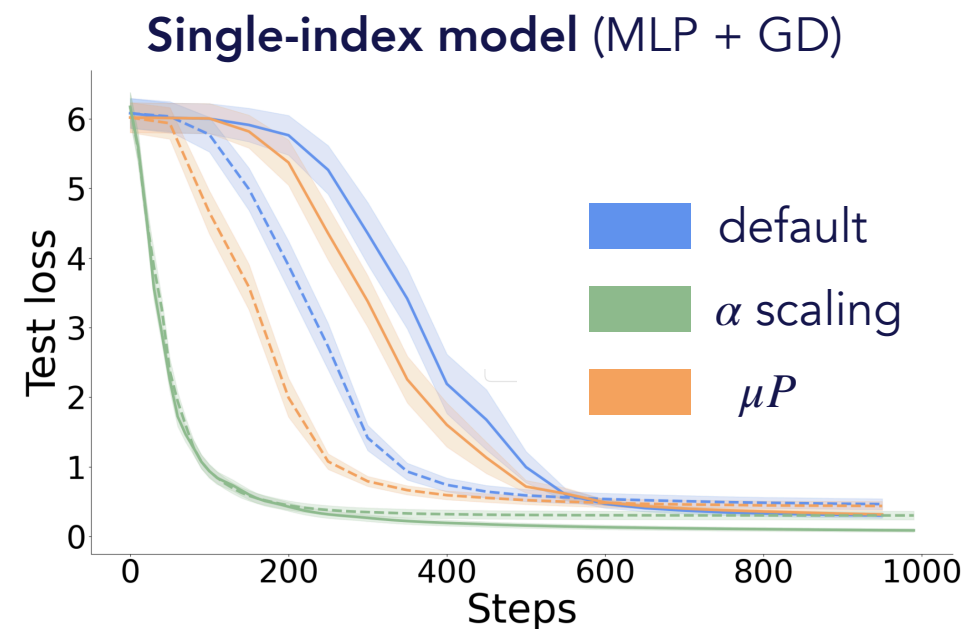
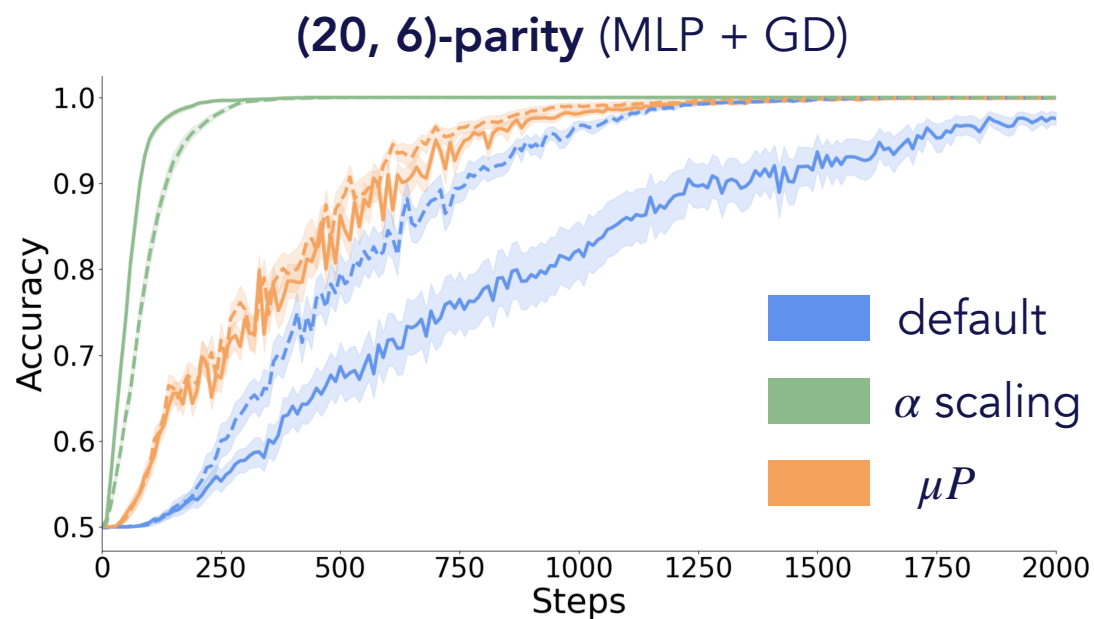


Small-set training is **more robust** to learning setup.

2. Layer-wise interventions reduce the gap

“Optimal” setup?

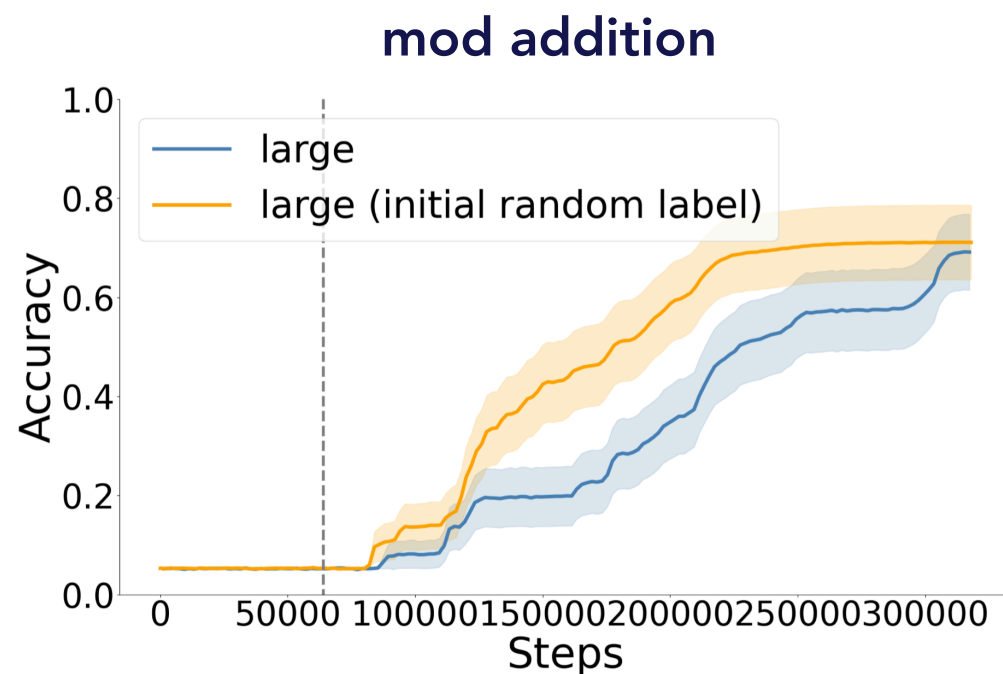
- e.g. μP ... balancing contributions from each layer.



— large - - - small

Transformer: 1) random labels

Speedup by training first on a small set with random labels.
(no signal & only affect norms)

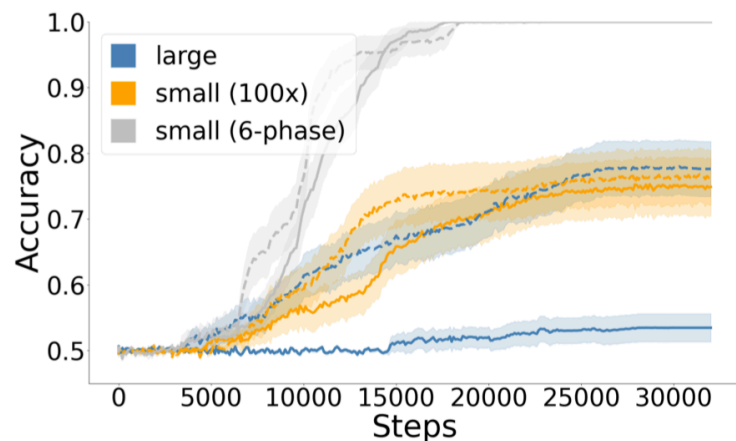


Transformer: 2) layer-wise interventions

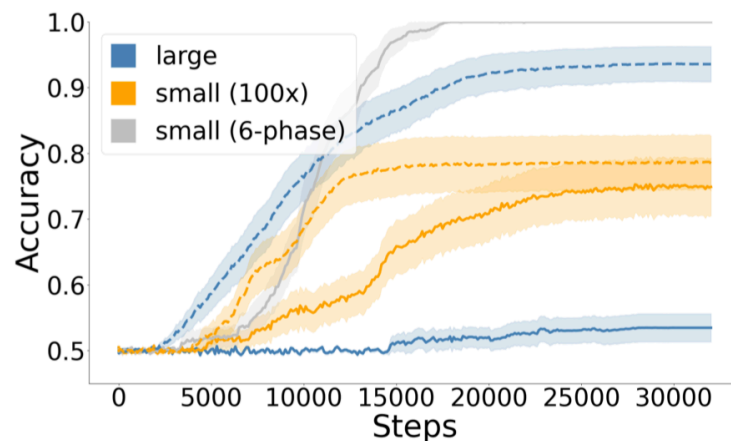
QK vs rest: intervene with learning rate, initialization, or QK normalization.

$$a_{ij} \propto \exp(\gamma^{-1} \text{RMS}(q_i)^\top \text{RMS}(k_j))$$

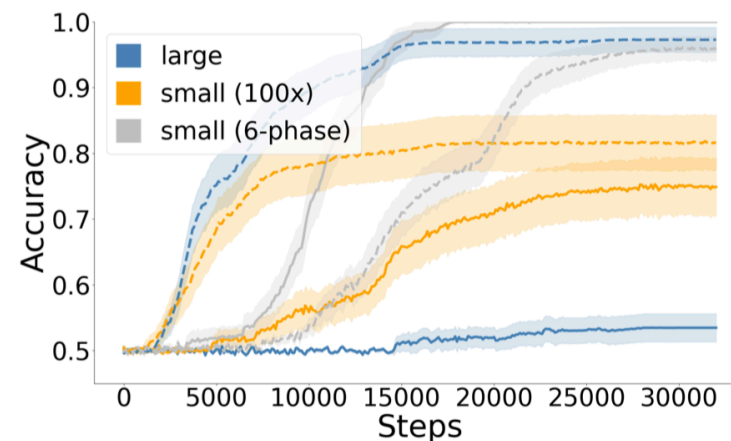
(20, 6)-parity on a subset



(a) QK learning rate



(b) QK initialization scale

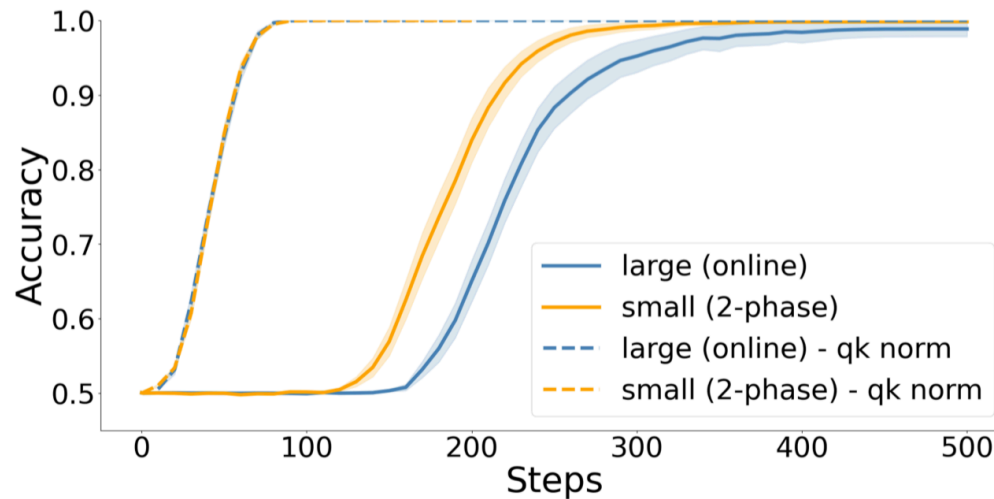


(c) QK normalization

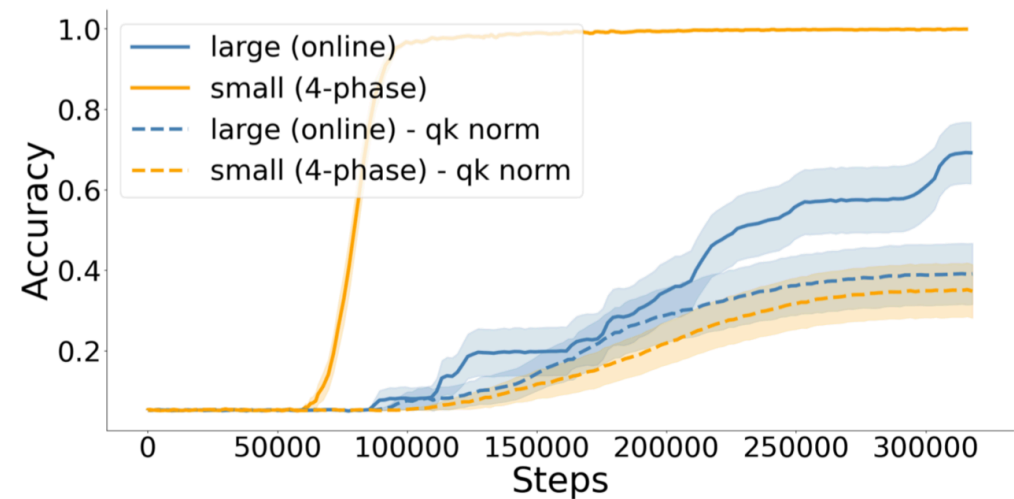
— default - - - intervention

Transformer: 2) layer-wise interventions

QK vs rest: intervene with learning rate, initialization, or **QK normalization**.
... *complicated effects*
(e.g. known to hurt retrieval)



(a) (20, 6)-parity, full-batch updates



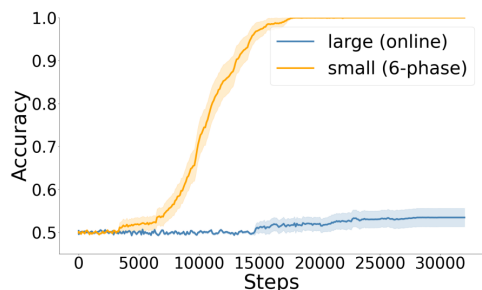
(b) mod addition

Less data, faster training

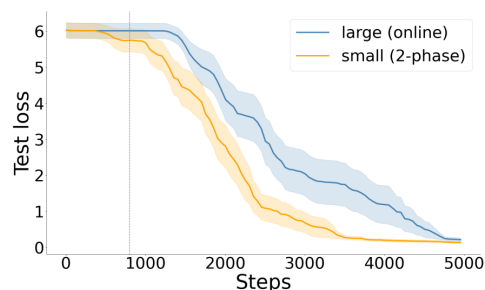
“small-vs-large gap” in compute

1. Observed across setups ... *task, architecture, optimizer.*
2. Prior explanations are insufficient ... *e.g. for parity with full-batch updates.*
3. Our explanation: sampling biases adjust **relative layer norms**.
 - Gap can be bridged by layerwise interventions,
but small-set training is more **robust** + provides **favorable biases**.

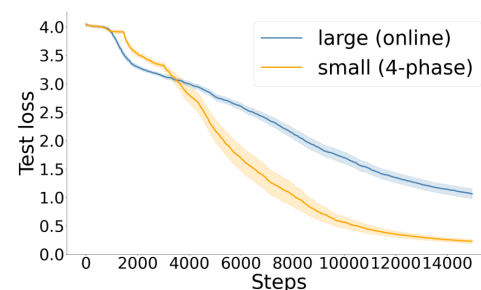
Small-vs-large gap: data repetition as *favorable optimization bias*.



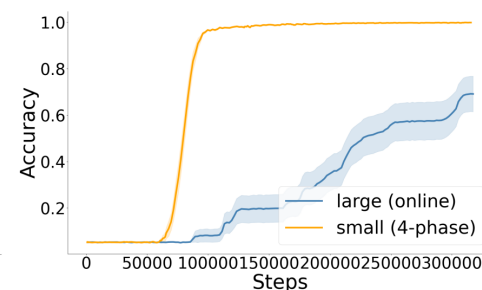
(a) Sparse parity



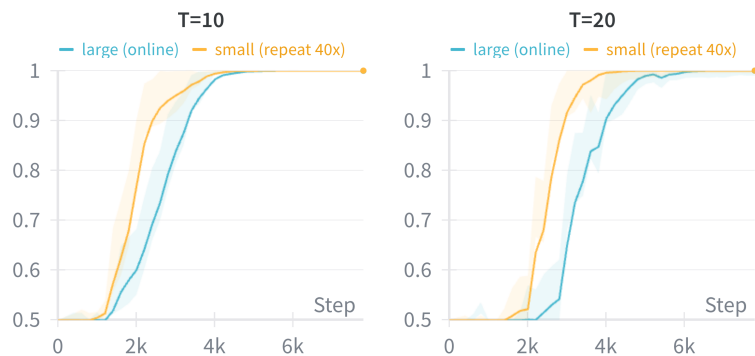
(b) Single-index model



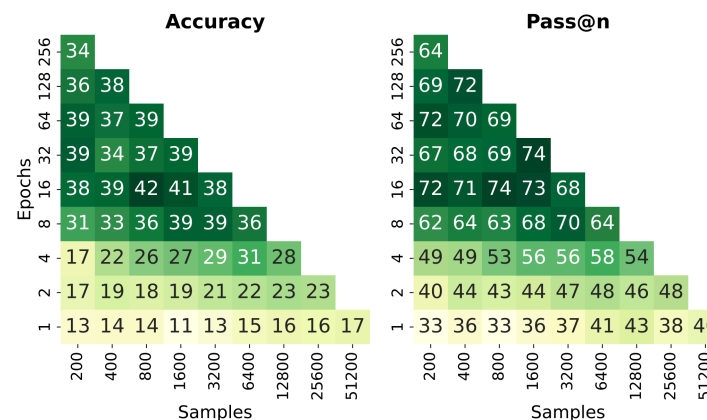
(c) ICL linear regression



(d) Modular addition



Qwen3 on full parity



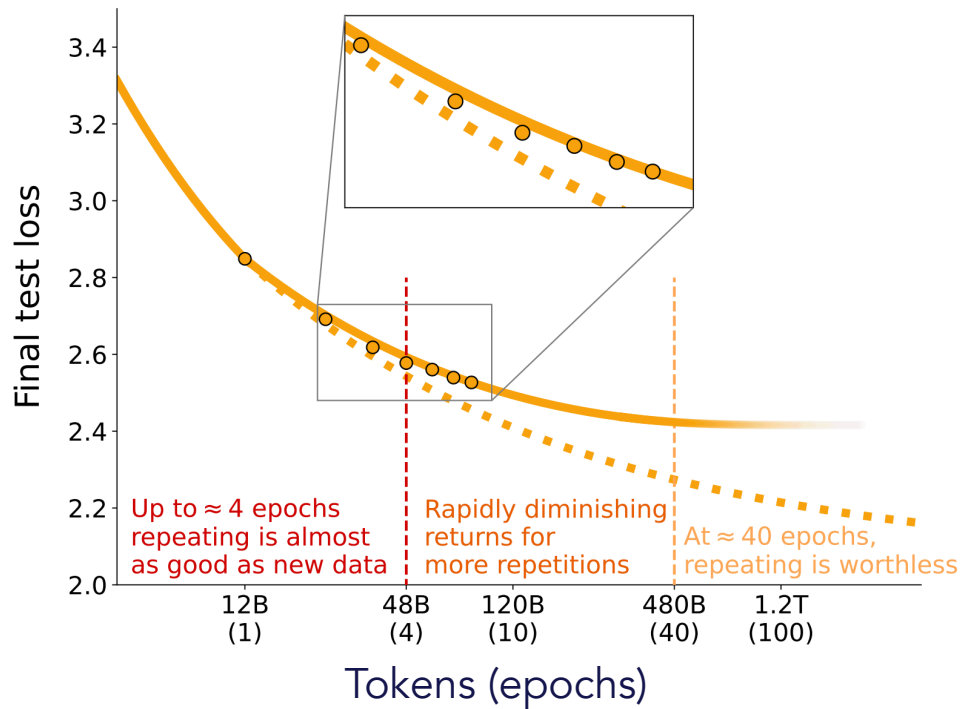
Olmo3 on long CoT (Kopiczko et al. 26)

Can less data be favorable?

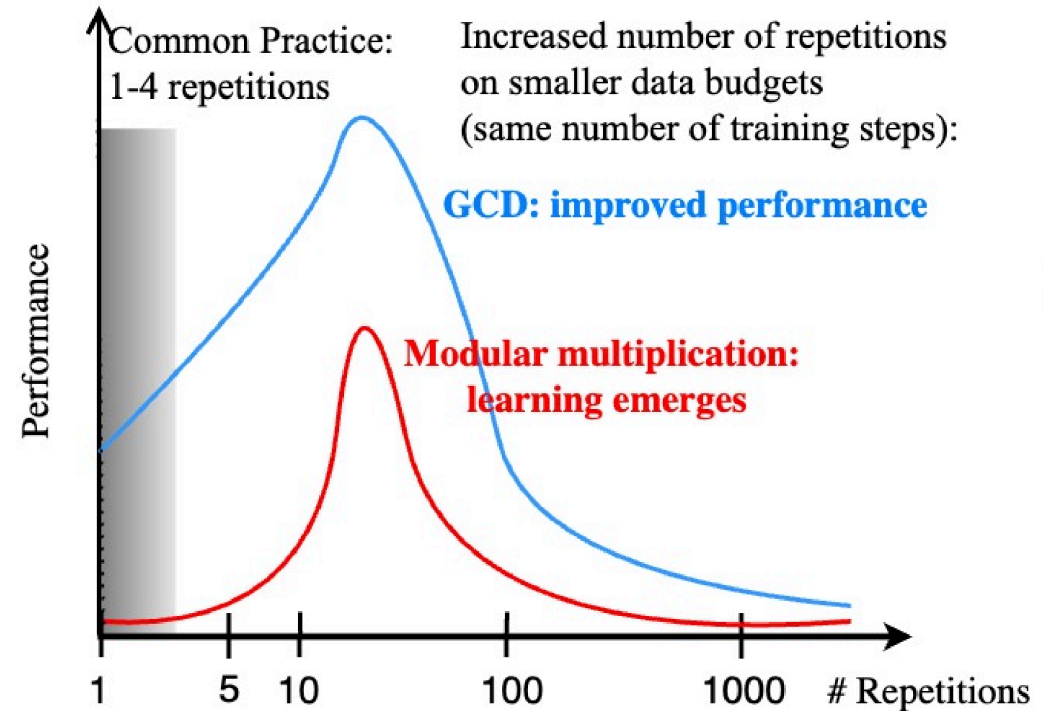
(statistical)

Which is correct?

(computational)



Muennighoff et al. 23



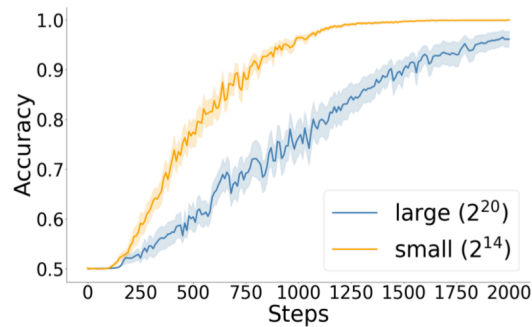
Charton & Kempe 24

Can less data be favorable?

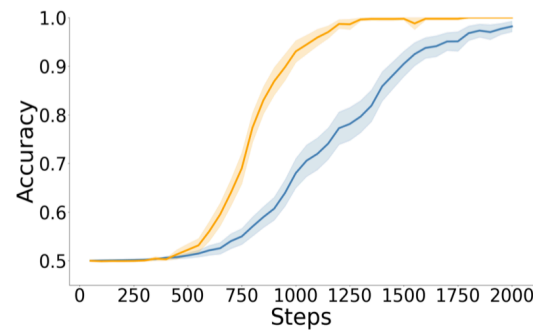
Yes, on problems with structures and symmetry, “reasoning” (e.g. SFT on CoT).

- More challenging optimization → wider gap ... deeper/narrower models, more complex tasks.

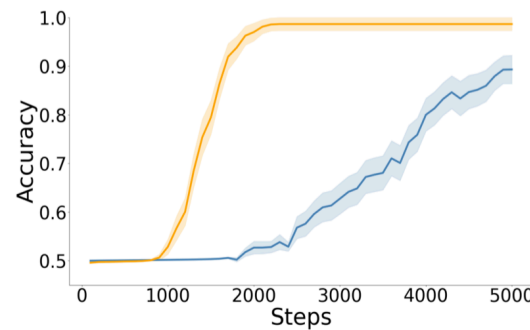
Depth scaling (parity, MLP + GD)



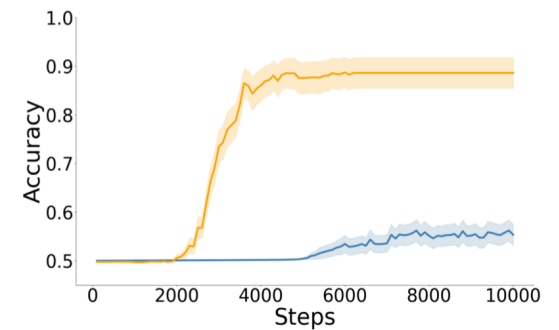
(a) Depth 2



(b) Depth 4



(c) Depth 6



(d) Depth 8

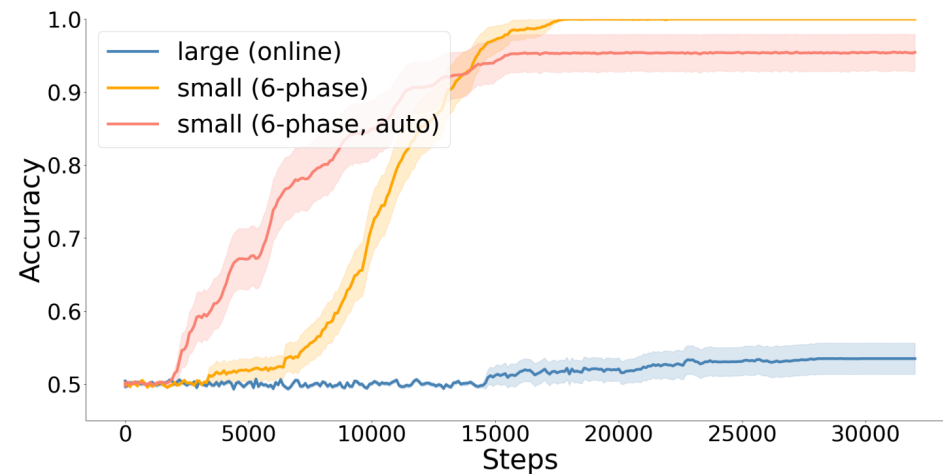
Can less data be favorable?

Yes, on problems with structures and symmetry, “reasoning” (e.g. SFT on CoT).

- More challenging optimization → wider gap ... deeper/narrower models, more complex tasks.
- “Computational-statistical gap” as a mental picture.

Use all data when compute allows.

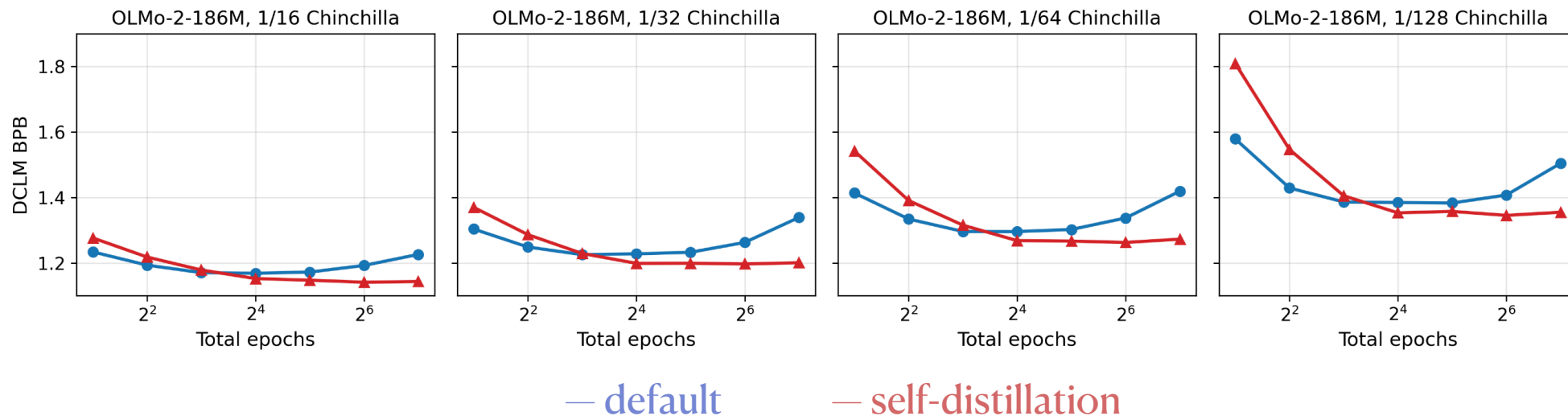
- e.g. automatic multi-phase curriculum.
- Further benefits from data selection?
... be mindful about the bitter lesson.



Can less data be favorable?

Likely not, with proper init, linear regression, redundant data, or pre-training.

- How to regularize pre-training?
 - e.g. explicit regularizers, synthetic data/distillation.



Discussions

- Good practices on model **initialization** / scaling?

- Residual stream ($1/\sqrt{L}$); per-block/module treatment (e.g. Complete^(d)); QK temperature.

```
# NanoChat: QK x1.2? q = q * 1.2 # sharper attention (split scale between Q and K), TODO think through better
k = k * 1.2
```

- **Stabilizing** w/o QK norm; weight projection / hyperball.
- Repetition in other optimization settings, e.g. **RL**.

Less data, faster training

“small-vs-large gap” in compute

- Observed across setups ... *task, architecture, optimizer*.
- Our explanation: sampling biases adjust relative layer norms.
- Data repetition as favorable optimization bias.

Thank you!



arXiv:2605.20314



Kempner
INSTITUTE



HARVARD
UNIVERSITY