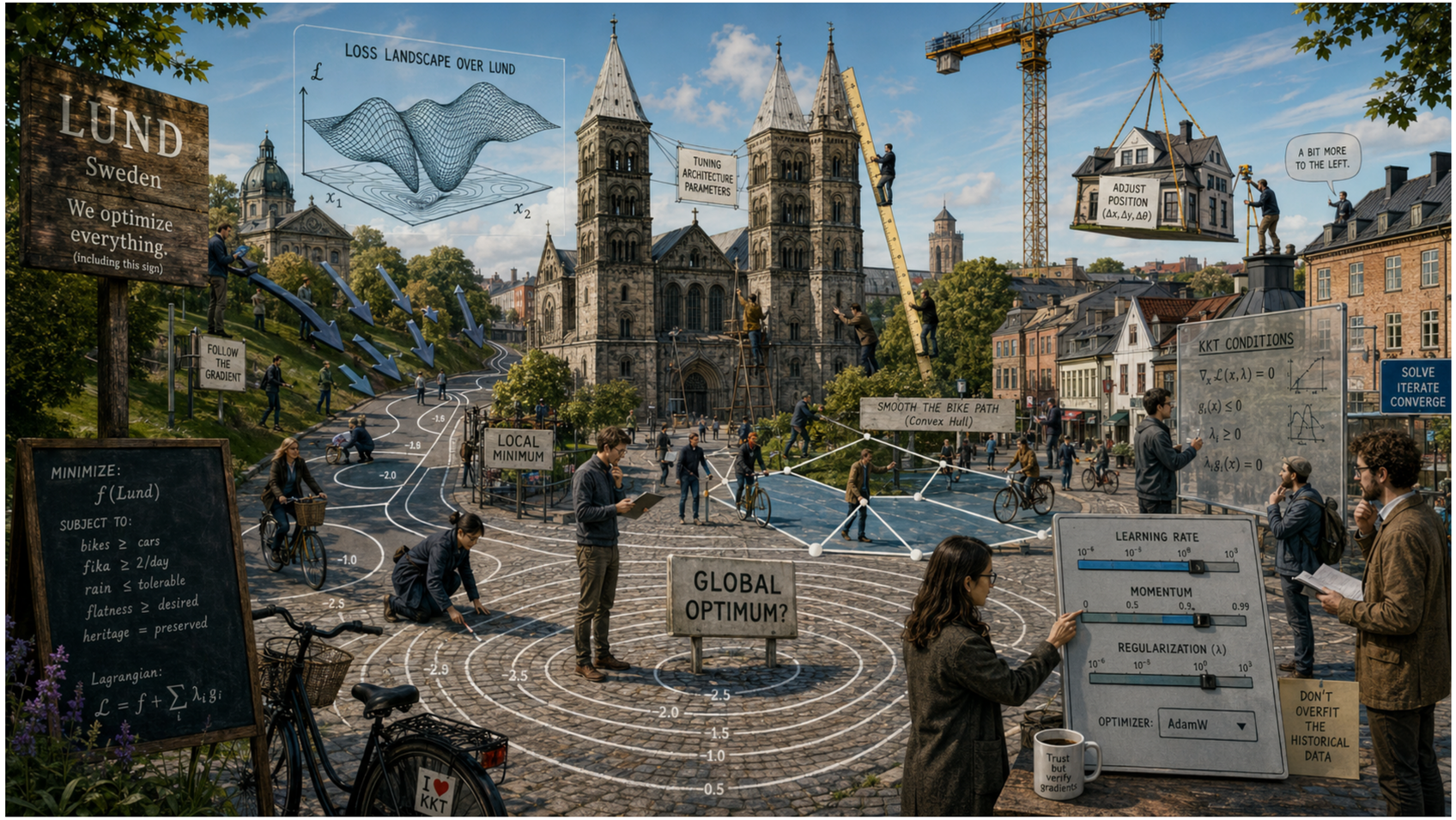
Training LLMs: do we understand our Optimizers?



Antonio Orvieto, Lund, 2026


THE JUNGLE WAS
JUST THE BEGINNING.
ADAM-SGD GAP
WHEN OPTIMIZATION EVOLVES,
ONLY ONE SURVIVES.
SUMMIT ENTERTAINMENT PRESENTS IN ASSOCIATION WITH FRONTIER PICTURES A VERDIGRIS / AGSRAS PRODUCTION MICHAELPARSONS • LENA MORALES DANIEL HAYES AND CLANCY BROWN
MUSIC BY J. PETERSON EDITOR KYLE MORRIS PRODUCTION DESIGNER RICK WHITLEY CREATURE & ANIMATRONIC ANIDIRECTION DIRECTION BY STAN WINSTON EXECUTIVE PRODUCERS TOM LEVY BRIAN HART WRITTEN BY JONATHAN REED & MARK VOLKER
COMING SUMMER 1996
DOLBY DIGITAL
IN SELECTED THEATRES
PG-13 PARENTS STRONGLY CAUTIONED
INTENSE SCIENCE FICTION VIOLENCE AND PERIL
SOME LANGUAGE
PALEO RESEARCH UNIT

Training LLMs: do we understand our Optimizers?

Training LLMs: do we understand our Optimizers?

Ok let’s put a twist on it..

Training ~~LLMs~~: do we understand our Optimizers?

Training ~~LLMs~~: do we understand our Optimizers?

Because:

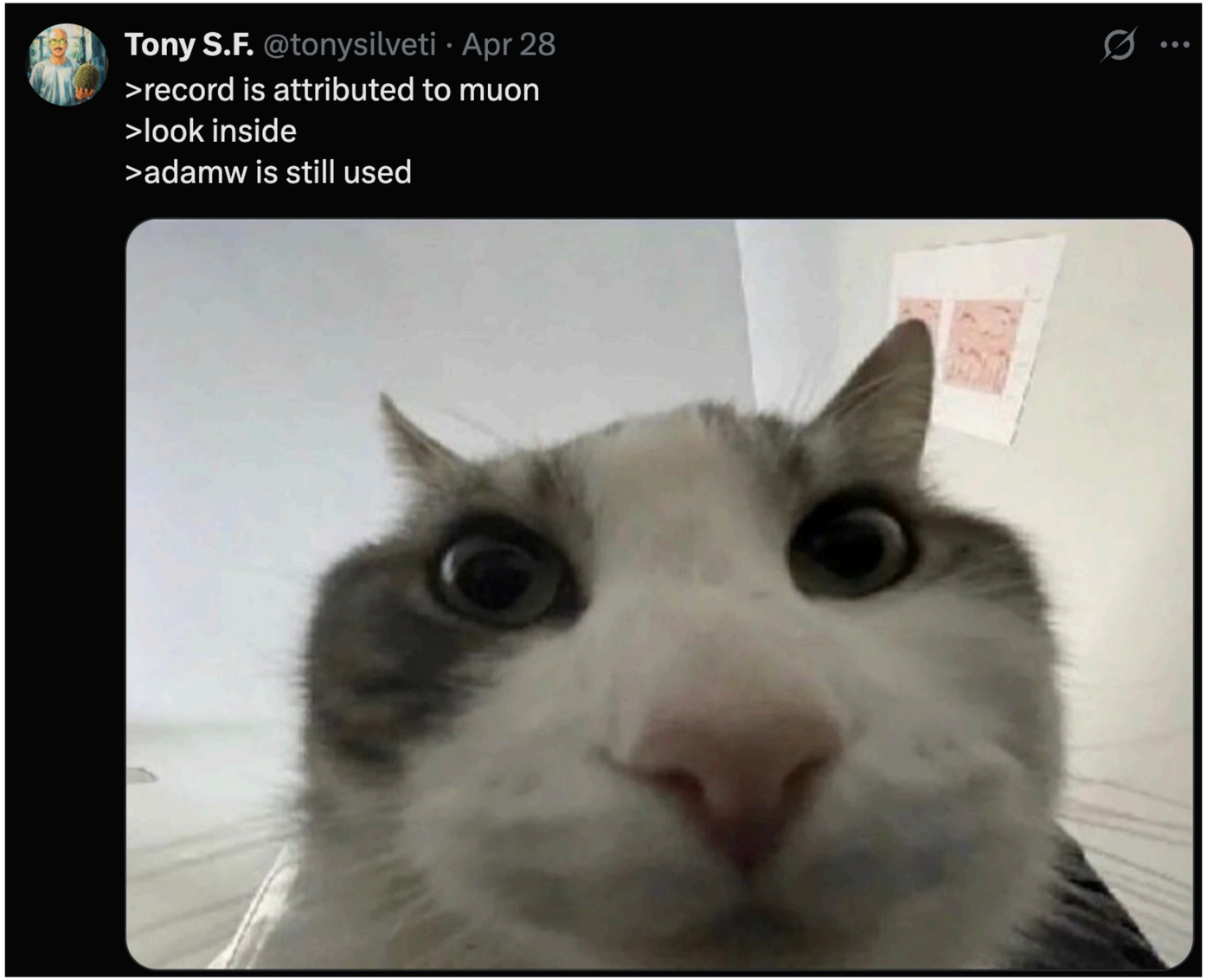
1) My point here is that this does not really matter..

2) Yes, LLMs are important

3) Yet, looking **only** at LLM pretraining is dangerous

4) But still, lots of stuff here will e still about LLMs

Why Adam?



PS: I actually love Muon and LMOs as well !!

Adam is used, is reliable, and is still a crucial component even in the most innovative pipelines

MUON IS SCALABLE FOR LLM TRAINING

TECHNICAL REPORT

Jingyuan Liu[1] Jianlin Su[1] Xingcheng Yao[2] Zhejun Jiang[1] Guokun Lai[1] Yulun Du[1]
Yidao Qin[1] Weixin Xu[1] Enzhe Lu[1] Junjie Yan[1] Yanru Chen[1] Huabin Zheng[1]
Yibo Liu[1] Shaowei Liu[1] Bohong Yin[1] Weiran He[1] Han Zhu[1] Yuzhi Wang[1]
Jianzhou Wang[1] Mengnan Dong[1] Zheng Zhang[1] Yongsheng Kang[1] Hao Zhang[1]
Xinran Xu[1] Yutao Zhang[1] Yuxin Wu[1] Xinyu Zhou[1] * Zhilin Yang[1]

[1] Moonshot AI [2] UCLA

We propose to match Muon's update RMS to be similar to that of AdamW. From empirical observations, AdamW's update RMS is usually around 0.2 to 0.4. Therefore, we scale Muon's update RMS to this range by the following adjustment:

$$\mathbf{W}_t = \mathbf{W}_{t-1} - \eta_t(0.2 \cdot \mathbf{O}_t \cdot \sqrt{\max(A, B)} + \lambda \mathbf{W}_{t-1}) \tag{4}$$

We validated this choice with empirical results (see Appendix A for details). Moreover, we highlighted that with this adjustment, Muon can directly **reuse** the learning rate and weight decay tuned for AdamW.

deepseek

DeepSeek-V4:
Towards Highly Efficient Million-Token Context Intelligence

DeepSeek-AI
research@deepseek.com

Basic Configurations. We maintain the AdamW (Loshchilov and Hutter, 2017) optimizer for the embedding module, the prediction head module, the static biases and gating factors of *m*HC modules, and the weights of all RMSNorm modules. All other modules are updated with Muon. Following Liu et al. (2025), we also apply weight decay to Muon parameters, use the Nesterov (Jordan et al., 2024; Nesterov, 1983) trick, and rescale the Root Mean Square (RMS) of the update matrix for reutilization of our AdamW hyper-parameters. Different from them, we use hybrid Newton-Schulz iterations for orthogonalization.

+ Studying Adam-SGD gap is fundamental for understanding the Muon-Adam gap.
(Its actually much more general, about norms gap)

$$m_i^k = \text{EMA}_{\beta_1}\left[\nabla_i L(w^k)\right], \quad v_i^k = \text{EMA}_{\beta_2}\left[\nabla_i L(w^k)^2\right]$$

$$w_i^{k+1} = w_i^k - \frac{\eta}{\sqrt{v_i^k + \epsilon}}\, m_i^k$$

EMA: exponential moving average

Setting β_1 to 0
we get **RMSprop**
(Tieleman, Hinton, 2012)

$$w_i^{k+1} = w_i^k - \frac{\eta}{\sqrt{v_i^k + \epsilon}}\, \nabla_i L(w_i^k)$$

Setting β_1, β_2
to 0 we get
SignSGD

$$w_i^{k+1} = w_i^k - \eta \frac{\nabla_i L(w_i^k)}{\sqrt{[\nabla_i L(w_i^k)]^2 + \epsilon}}$$

$$\approx \text{sign}(\nabla_i L(w_i^k))$$

But there are many more explanations and GREAT works! (just a few below)

NOISE IS NOT THE MAIN FACTOR BEHIND THE GAP BETWEEN SGD AND ADAM ON TRANSFORMERS, BUT SIGN DESCENT MIGHT BE

Frederik Kunstner, Jacques Chen, J. Wilder Lavington & Mark Schmidt†
University of British Columbia, Canada CIFAR AI Chair (Amii)†
{kunstner,jola2372,schmidtm}@cs.ubc.ca
jacquesc@students.cs.ubc.ca

Heavy-Tailed Class Imbalance and Why Adam Outperforms Gradient Descent on Language Models

Frederik Kunstner [1] **Robin Yadav** [1] **Alan Milligan** [1] **Mark Schmidt** [1 2] **Alberto Bietti** [3]

Toward Understanding Why Adam Converges Faster Than SGD for Transformers

Yan Pan YPAN2@ANDREW.CMU.EDU
Yuanzhi Li YUANZHIL@ANDREW.CMU.EDU
Carnegie Mellon University

Why are Adaptive Methods Good for Attention Models?

Jingzhao Zhang
MIT
jzhzhang@mit.edu

Sai Praneeth Karimireddy
EPFL
sai.karimireddy@epfl.ch

Andreas Veit
Google Research
aveit@google.com

Seungyeon Kim
Google Research
seungyeonk@google.com

Sashank Reddi
Google Research
sashank@google.com

Sanjiv Kumar
Google Research
sanjivk@google.com

Suvrit Sra
MIT
suvrit@mit.edu

Why Transformers Need Adam: A Hessian Perspective

Yushun Zhang[12], **Congliang Chen**[12], **Tian Ding**[2], **Ziniu Li**[12], **Ruoyu Sun**[12*], **Zhi-Quan Luo**[12]
[1]The Chinese University of Hong Kong, Shenzhen, China
[2]Shenzhen Research Institute of Big Data
{yushunzhang,congliangchen,ziniuli}@link.cuhk.edu.cn
dingtian@sribd.cn, sunruoyu@cuhk.edu.cn, luozq@cuhk.edu.cn

But there are many more explanations and GREAT works! (just a few below)

Noise Is Not the Main Factor Behind the Gap Between SGD and Adam on Transformers, But Sign Descent Might Be

Frederik Kunstner, Jacques Chen, J. Wilder Lavington & Mark Schmidt[†]
University of British Columbia, Canada CIFAR AI Chair (Amii)[†]
{kunstner,jola2372,schmidtm}@cs.ubc.ca
jacquesc@students.cs.ubc.ca

Heavy-Tailed Class Imbalance and Why Adam Outperforms Gradient Descent on Language Models

Frederik Kunstner [1] **Robin Yadav** [1] **Alan Milligan** [1] **Mark Schmidt** [1 2] **Alberto Bietti** [3]

Why are Adaptive Methods Good for Attention Models?

Jingzhao Zhang
MIT
jzhzhang@mit.edu

Sai Praneeth Karimireddy
EPFL
sai.karimireddy@epfl.ch

Andreas Veit
Google Research
aveit@google.com

Seungyeon Kim
Google Research
seungyeonk@google.com

Sashank Reddi
Google Research
sashank@google.com

Sanjiv Kumar
Google Research
sanjivk@google.com

Suvrit Sra
MIT
suvrit@mit.edu

Toward Understanding Why Adam Converges Faster Than SGD for Transformers

Yan Pan YPAN2@ANDREW.CMU.EDU
Yuanzhi Li YUANZHIL@ANDREW.CMU.EDU
Carnegie Mellon University

Why Transformers Need Adam: A Hessian Perspective

Yushun Zhang[12], **Congliang Chen**[12], **Tian Ding**[2], **Ziniu Li**[12], **Ruoyu Sun**[12*], **Zhi-Quan Luo**[12]
[1]The Chinese University of Hong Kong, Shenzhen, China
[2]Shenzhen Research Institute of Big Data
{yushunzhang,congliangchen,ziniuli}@link.cuhk.edu.cn
dingtian@sribd.cn, sunruoyu@cuhk.edu.cn, luozq@cuhk.edu.cn

Published as a conference paper at ICLR 2023

NOISE IS NOT THE MAIN FACTOR BEHIND THE GAP BETWEEN SGD AND ADAM ON TRANSFORMERS, BUT SIGN DESCENT MIGHT BE

Frederik Kunstner, Jacques Chen, J. Wilder Lavington & Mark Schmidt[†]
University of British Columbia, Canada CIFAR AI Chair (Amii)[†]
{kunstner,jola2372,schmidtm}@cs.ubc.ca
jacquesc@students.cs.ubc.ca

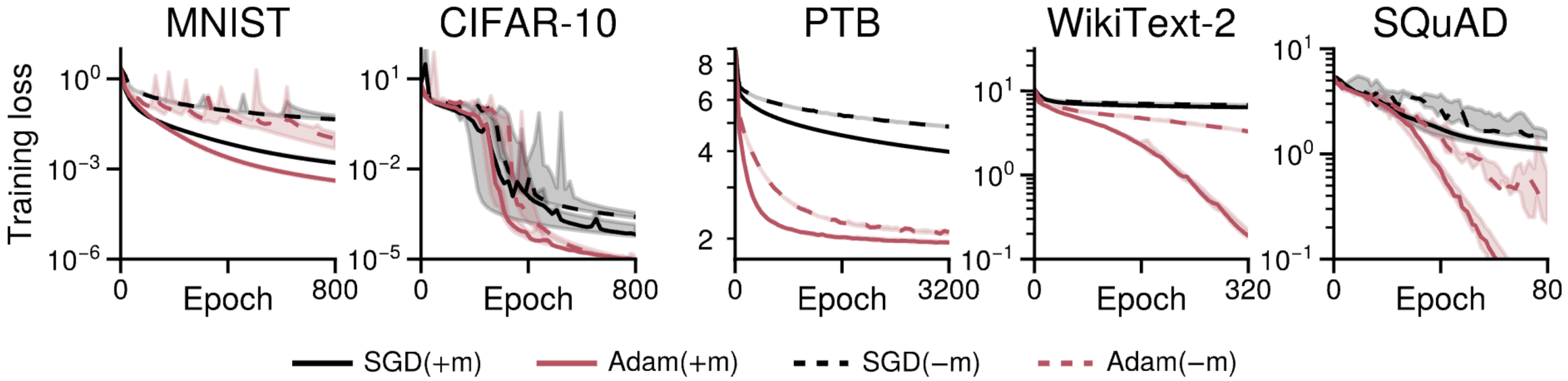


Figure 2: **The gap does not disappear when training in full batch.**

Published as a conference paper at ICLR 2023

NOISE IS NOT THE MAIN FACTOR BEHIND THE GAP BETWEEN SGD AND ADAM ON TRANSFORMERS, BUT SIGN DESCENT MIGHT BE

Frederik Kunstner, Jacques Chen, J. Wilder Lavington & Mark Schmidt[†]
University of British Columbia, Canada CIFAR AI Chair (Amii)[†]
{kunstner,jola2372,schmidtm}@cs.ubc.ca
jacquesc@students.cs.ubc.ca

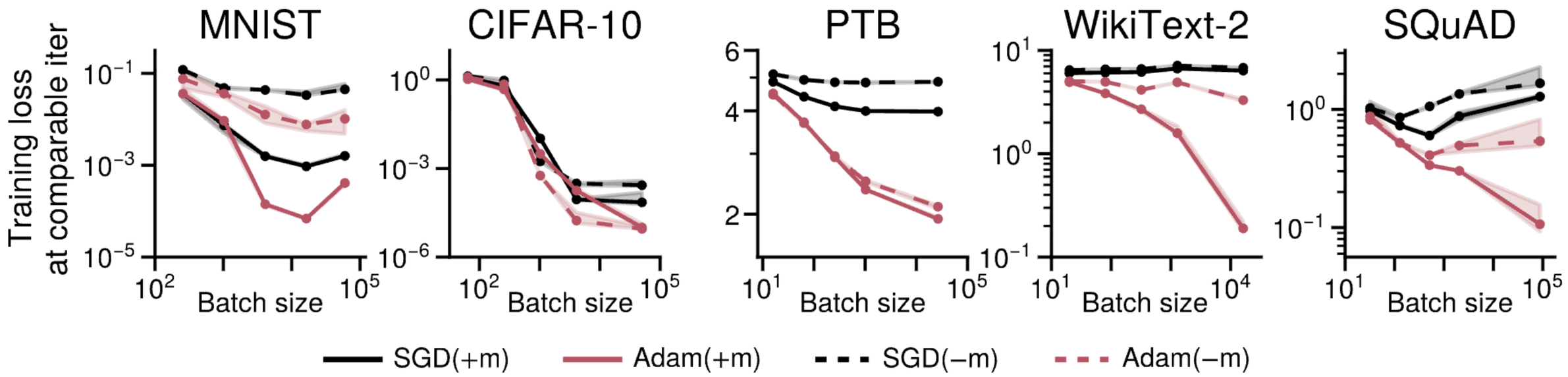


Figure 3: **The gap between SGD and Adam increases with batch size.**

Is this true at a reasonable scale?

Simplified Models

$$m_i^k = \text{EMA}_{\beta_1}\left[\nabla_i L(w^k)\right], \qquad v_i^k = \text{EMA}_{\beta_2}\left[\nabla_i L(w^k)^2\right]$$

$\beta_1 = 0$

$$w_i^{k+1} = w_i^k - \frac{\eta}{\sqrt{v_i^k + \epsilon}}\, m_i^k$$

Adam

$\beta_2, \epsilon = 0$

$$w_i^{k+1} = w_i^k - \frac{\eta}{\sqrt{v_i^k + \epsilon}}\, \nabla_i L(w_i^k)$$

RMSprop

$$w_i^{k+1} = w_i^k - \eta \text{sign}(\nabla_i L(w_i^k))$$

SignSGD

Momentum reintroduced

$$w_i^{k+1} = w_i^k - \eta \text{sign}(m_i^k)$$

Signum

In Search of Adam's Secret Sauce

Antonio Orvieto *
ELLIS Institute Tübingen, MPI-IS
Tübingen AI Center, Germany

Robert M. Gower
CCM, Flatiron Institute, Simons Foundation
New York, US

We heavily tune all methods claiming a connection to Adam. SignSGD + momentum closes 96% gap

Table 1: ***(Signum closes 96% of the perplexity gap between Adam and SGD)*** *Validation perplexity comparison of widely used optimizers that interpolate between* `SGD` *and* `Adam`*, evaluated on a language modeling task (160M parameters, 3.2B SlimPajama tokens, sequence length 2048, batch size 256 – Chinchilla optimal). We report the mean and 2-sigma interval of validation perplexity (on 100M held-out tokens) across 3 initialization seeds. Weight decay is always decoupled [Loshchilov and Hutter, 2019] and set to* 0.1 *[Biderman et al., 2023, Liu et al., 2024] except for* `SGD` *where we further tune (§B).* `RMSprop` *does not use momentum, and Gclip is global norm clipping to* 1 *(before applying momentum), Cclip is coordinate-wise clipping (after applying momentum). Other hyperparameters, for all other methods, are carefully tuned, see e.g. Figure 2 and §3.*
To optimally tune hyperparameters (e.g. Figure 2), we performed a total of 582 full training runs.

	`Adam`	`Signum`	`RMSprop`	`SGD+Cclip`	`SignSGD`	`SGD+Gclip`	`SGD`
Val ppl.	**21.86± 0.21**	23.23± 0.16	27.04± 0.34	33.40± 0.39	36.78± 0.57	37.76± 0.61	53.62± 5.14

We do heavy tuning for each method, e.g. RMSprop:

$$w_i^{k+1} = w_i^k - \text{schedule}_k \cdot \frac{\eta}{\sqrt{\text{EMA}_{\beta_2}(g_i^k) + \epsilon}} \nabla_i L(w_i^k)$$

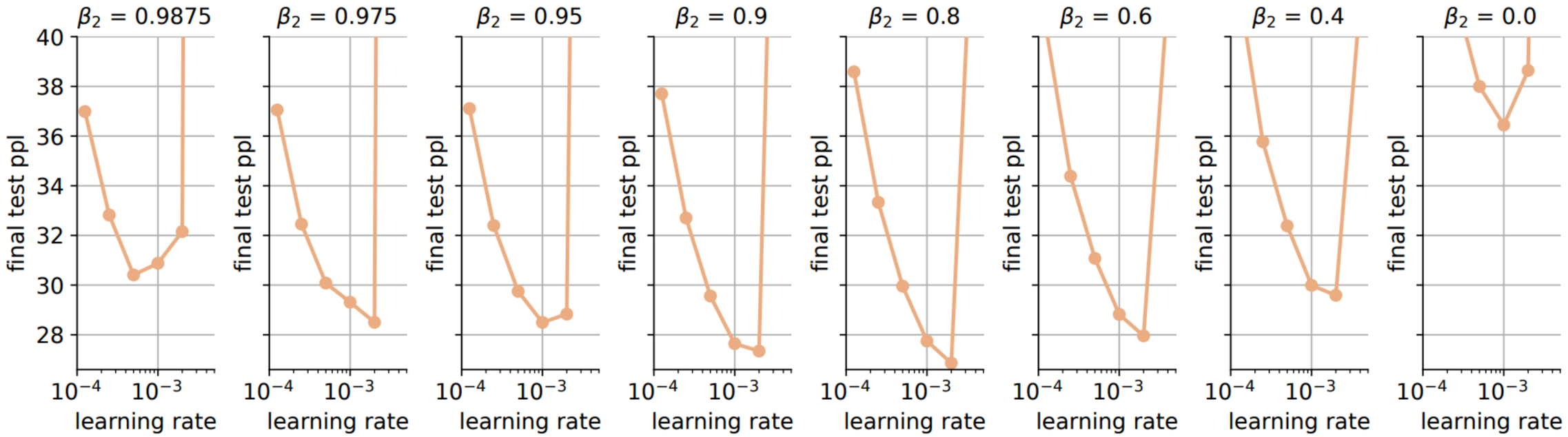
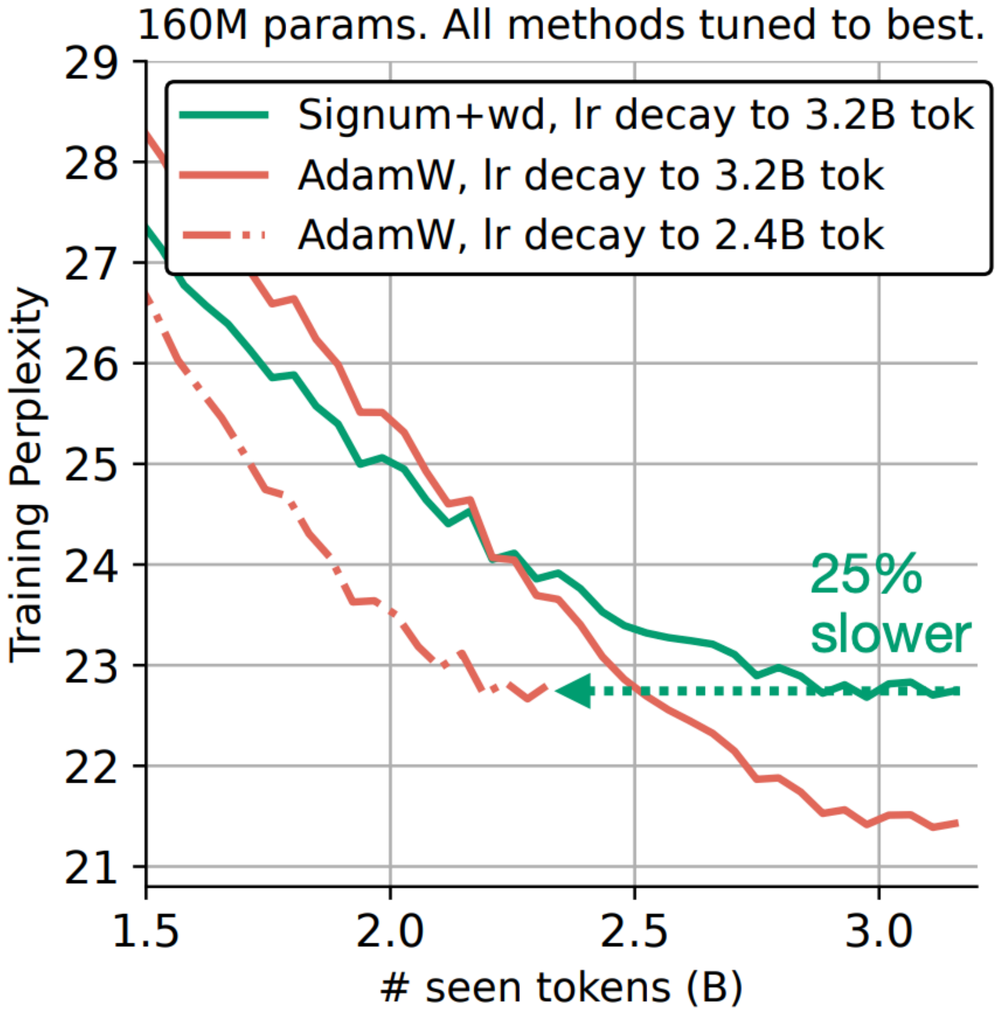


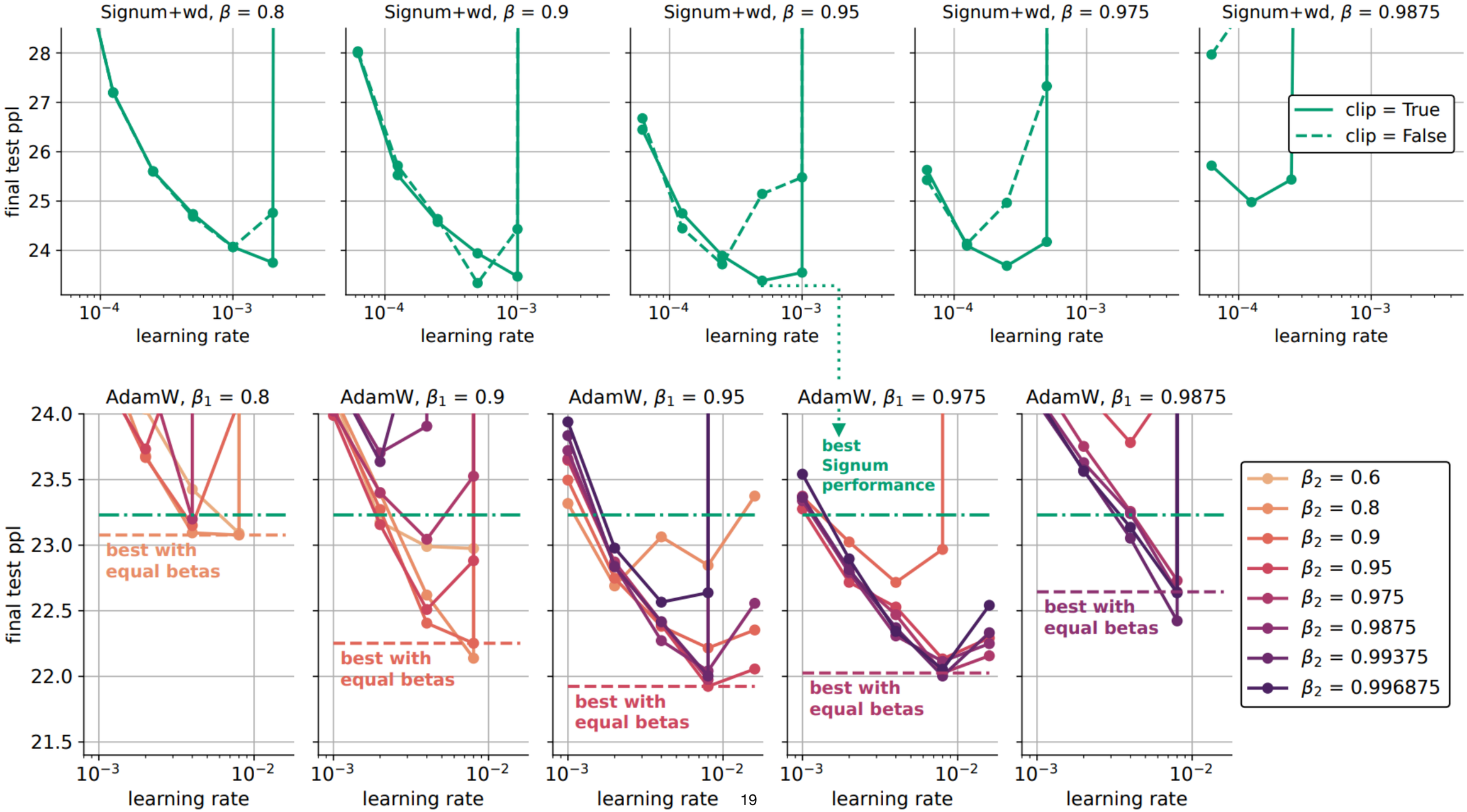
Figure 10: ***RMSprop with decoupled weight decay*** *0.1. Implemented with Pytorch AdamW setting* $\beta_1 = 0$.

Signum: a good model, but there is still something more..

It is 25% slower at optimal tuning!



Actually, $\beta_1 = \beta_2$ works very well in Adam!



410M parameters, Chinchilla-optimal

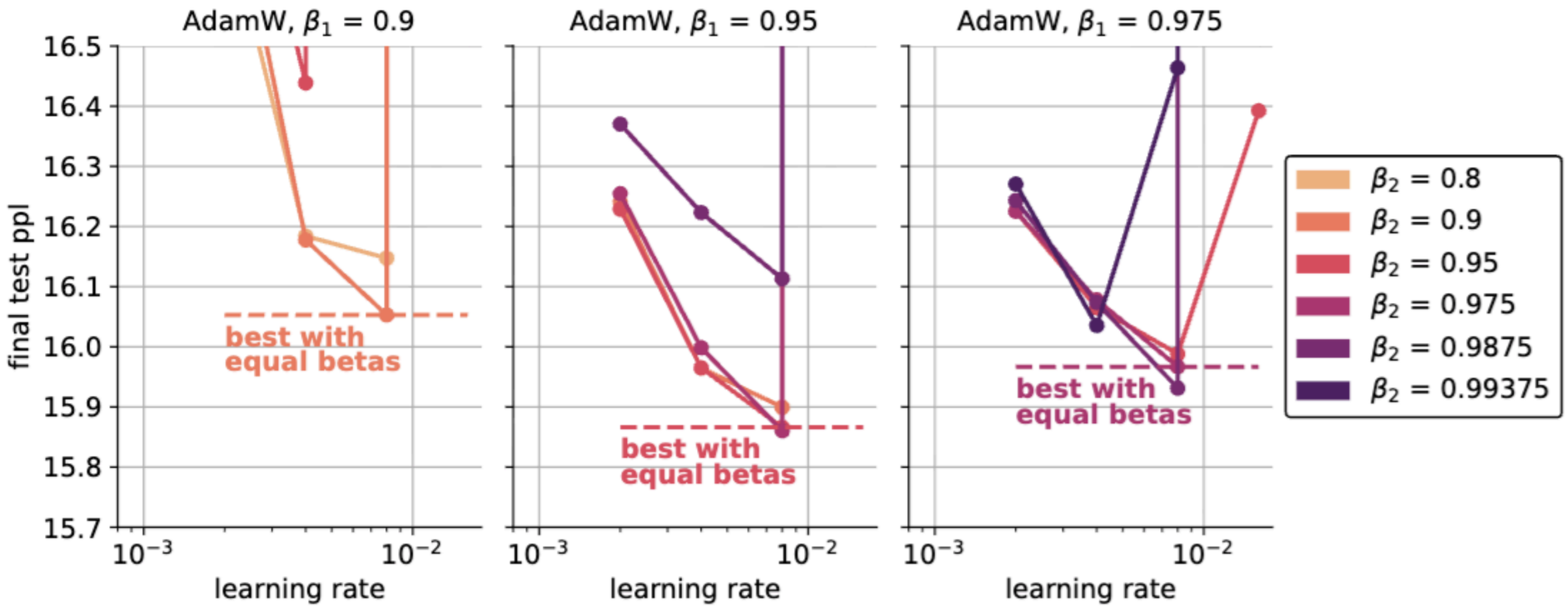


Figure 4: *The final validation performance (100M held-out tokens) for 44 trained LMs with 420M parameters trained on 8.2 B SlimPajama tokens (Chinchilla-optimal).* ***Equal betas yields near-optimal performance****. We use gradient clipping and a batch size of 512 (scaled by 2 compared to Figure 2, as suggested by Zhang et al. [2025]). Sequence length is 2048, weight decay is* 0.1*. Note that the standard setting* $(0.9, 0.95)$ *is quite suboptimal here.*

This is your Adam LM
Default. Be careful ;)

410M parameters, Chinchilla-optimal

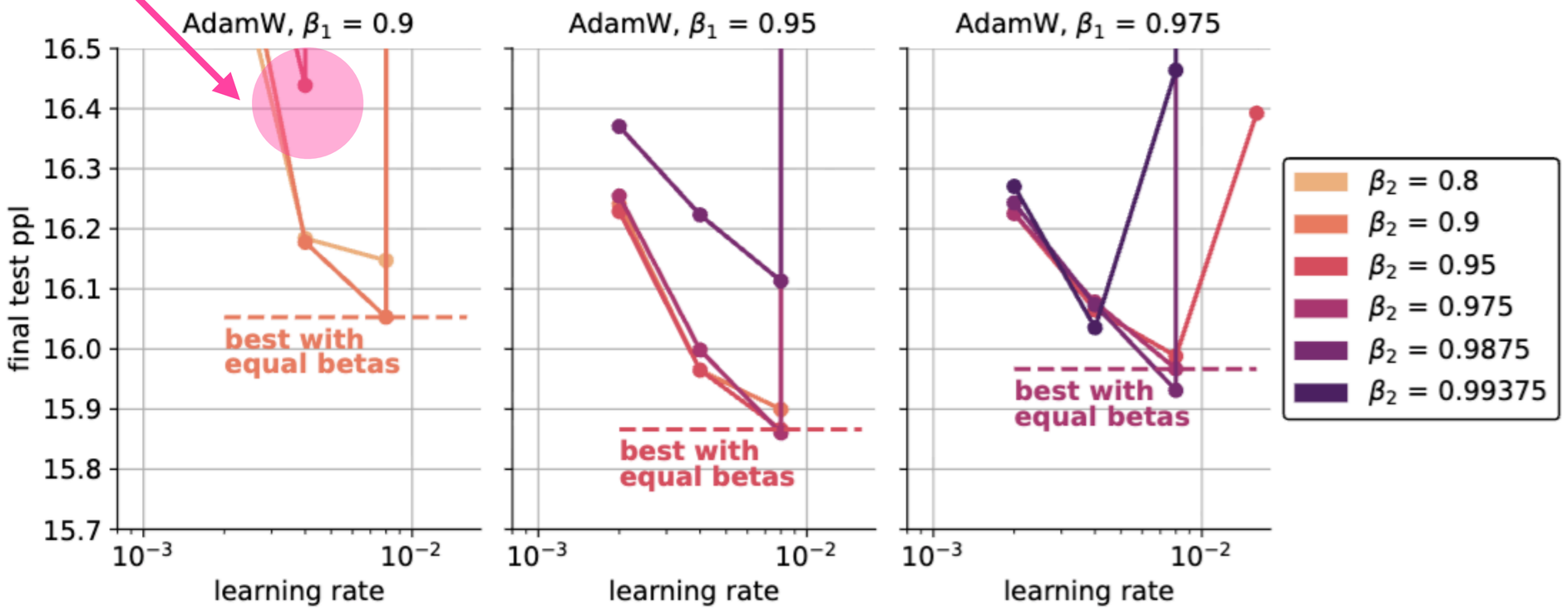
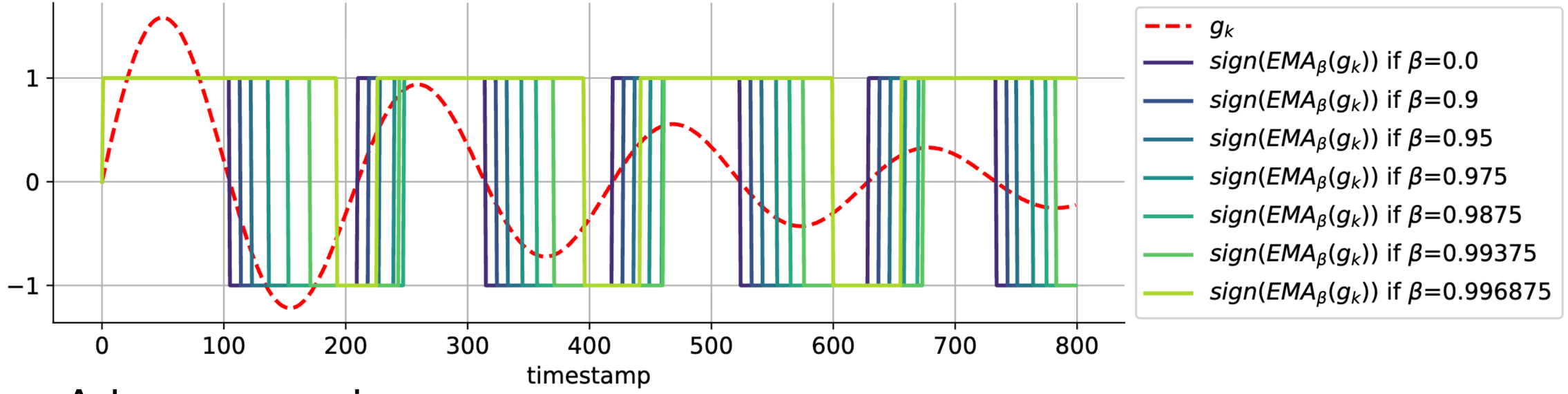
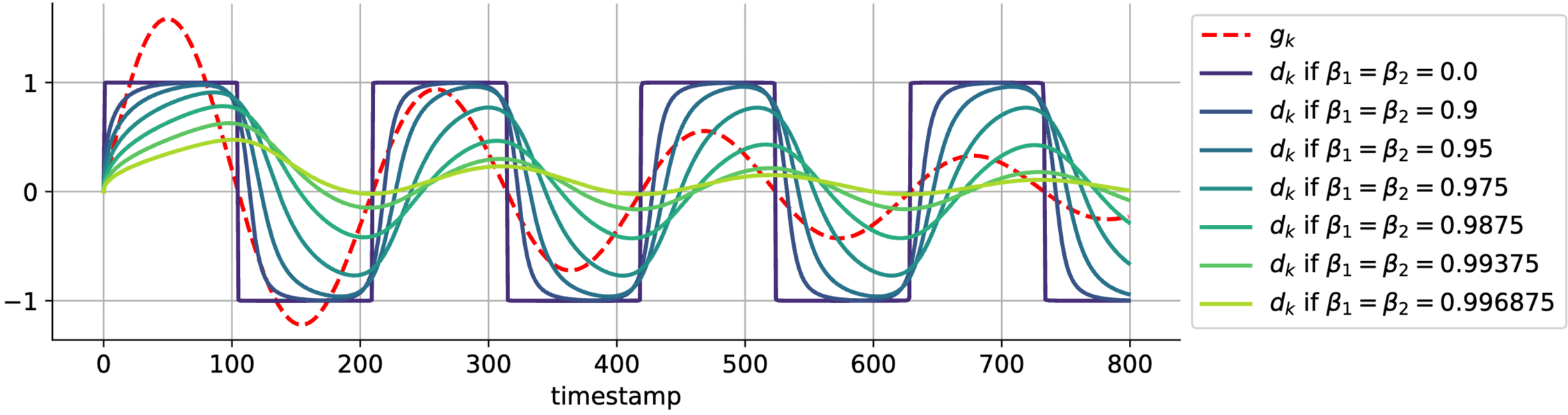


Figure 4: *The final validation performance (100M held-out tokens) for 44 trained LMs with 420M parameters trained on 8.2 B SlimPajama tokens (Chinchilla-optimal).* ***Equal betas yields near-optimal performance****. We use gradient clipping and a batch size of 512 (scaled by 2 compared to Figure 2, as suggested by Zhang et al. [2025]). Sequence length is 2048, weight decay is* 0.1*. Note that the standard setting* $(0.9, 0.95)$ *is quite suboptimal here.*

SignGD processing



Adam processing



Sounds familiar? Yes! Was already done in 2018!

Dissecting Adam: The Sign, Magnitude and Variance of Stochastic Gradients

Lukas Balles [1] Philipp Hennig [1]

2018!!

$$\frac{m_t}{\sqrt{v_t}} = \frac{\operatorname{sign}(m_t)}{\sqrt{\frac{v_t}{m_t^2}}} = \sqrt{\frac{1}{1+\frac{v_t-m_t^2}{m_t^2}}} \odot \operatorname{sign}(m_t)$$

The missing piece in Balles and Hennig (2018) was to show when and if the term $\sigma_k^2 := v_k - m_k^2$ is a measure of variance.

We show: $v_k - m_k^2$ only has a precise variance interpretation for the case $\beta_1 = \beta_2$.

Proposition: Adam's update d_k can be represented as

$$d_k = \frac{m_k}{\sqrt{m_k^2 + \gamma\,\mathsf{EMA}_\tau[(am_{k-1} - bg_k)^2]}}$$

for some $a, b, \gamma \in \mathbb{R}$ and $\tau \in (0,1)$ if and only if $\beta_1 = \beta_2$.

So, to summarize:

- Yes, SignGD+m is very good
- Yes, Adam is slightly better
- But in the end.. yeah **sign (which btw is an LMO) is enough!**

So, to summarize:

- Yes, SignGD+m is very good
- Yes, Adam is slightly better
- But in the end.. yeah **sign (which btw is an LMO) is enough!**

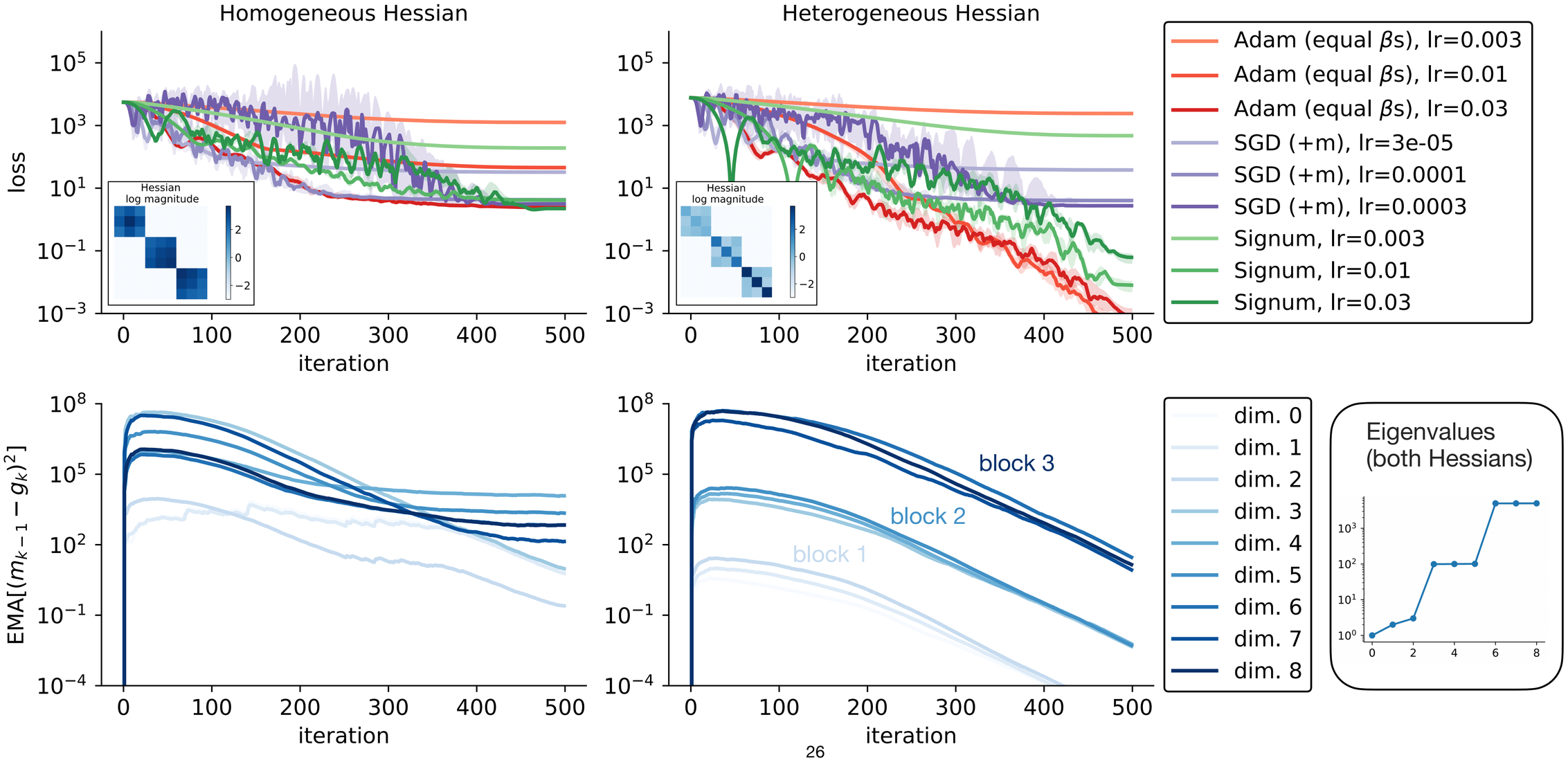
But Why?!

Heterogeneity in the landscape

We reproduced this

Why Transformers Need Adam: A Hessian Perspective

Yushun Zhang[12], **Congliang Chen**[12], **Tian Ding**[2], **Ziniu Li**[12], **Ruoyu Sun**[12*], **Zhi-Quan Luo**[12]
[1]The Chinese University of Hong Kong, Shenzhen, China
[2]Shenzhen Research Institute of Big Data
{yushunzhang,congliangchen,ziniuli}@link.cuhk.edu.cn
dingtian@sribd.cn, sunruoyu@cuhk.edu.cn, luozq@cuhk.edu.cn



Which is also linked to data distribution $\Longrightarrow$ Hessian!

Scaling Laws for Gradient Descent and Sign Descent for Linear Bigram Models under Zipf's Law

Frederik Kunstner
frederik.kunstner@inria.fr

Francis Bach
francis.bach@inria.fr

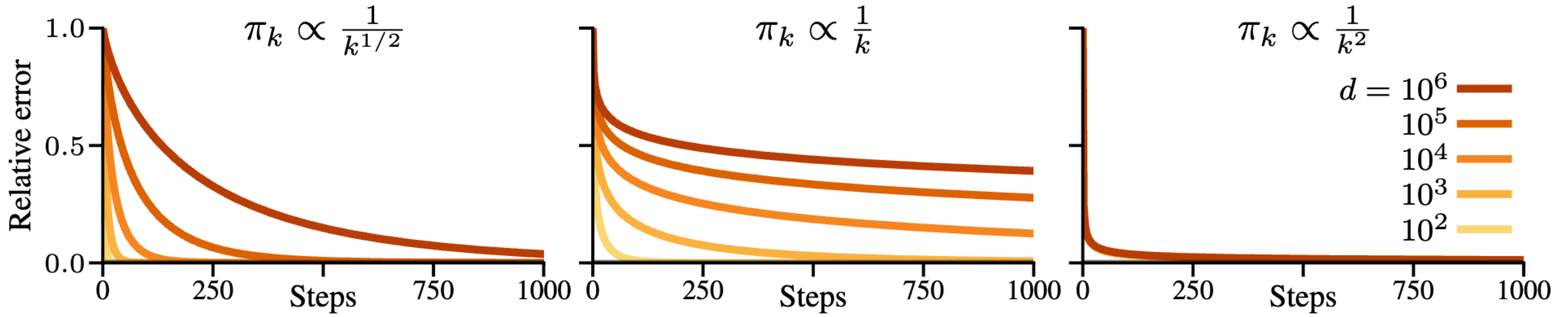


Figure 1: **Gradient descent (GD) scales badly with vocabulary size when the data is Zipfian.**

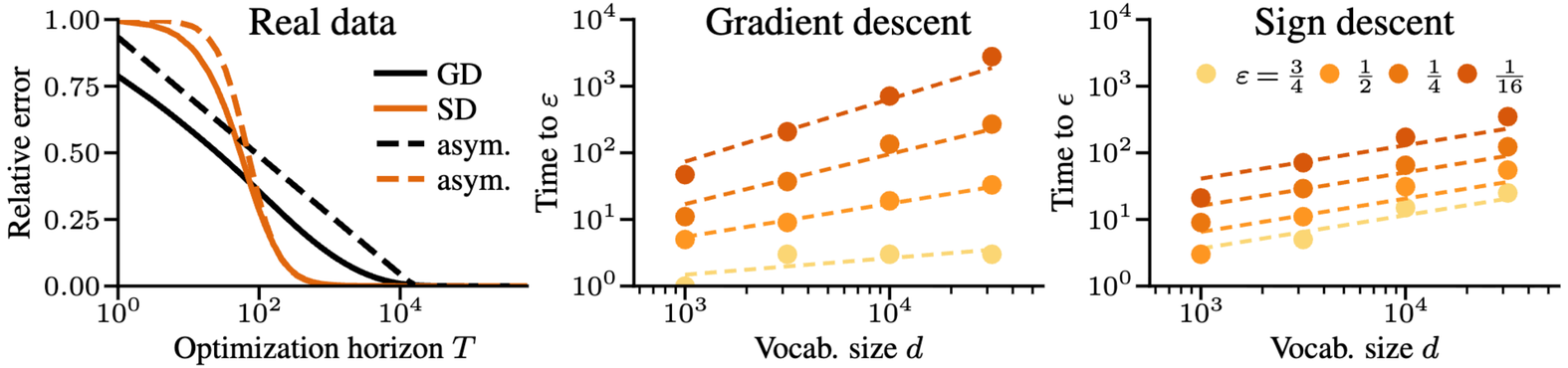


Figure 2: **Our scaling predicts the behavior of gradient descent and sign descent on real data.**

But aren't we forgetting something?

Published as a conference paper at ICLR 2023

NOISE IS NOT THE MAIN FACTOR BEHIND THE GAP BETWEEN SGD AND ADAM ON TRANSFORMERS, BUT SIGN DESCENT MIGHT BE

Frederik Kunstner, Jacques Chen, J. Wilder Lavington & Mark Schmidt†
University of British Columbia, Canada CIFAR AI Chair (Amii)†
{kunstner,jola2372,schmidtm}@cs.ubc.ca
jacquesc@students.cs.ubc.ca

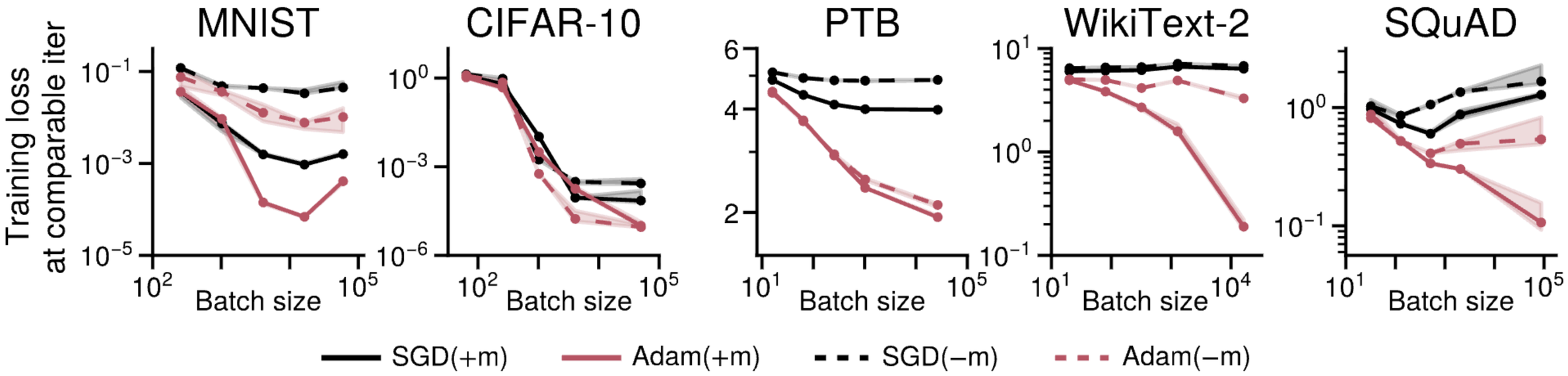


Figure 3: **The gap between SGD and Adam increases with batch size.**

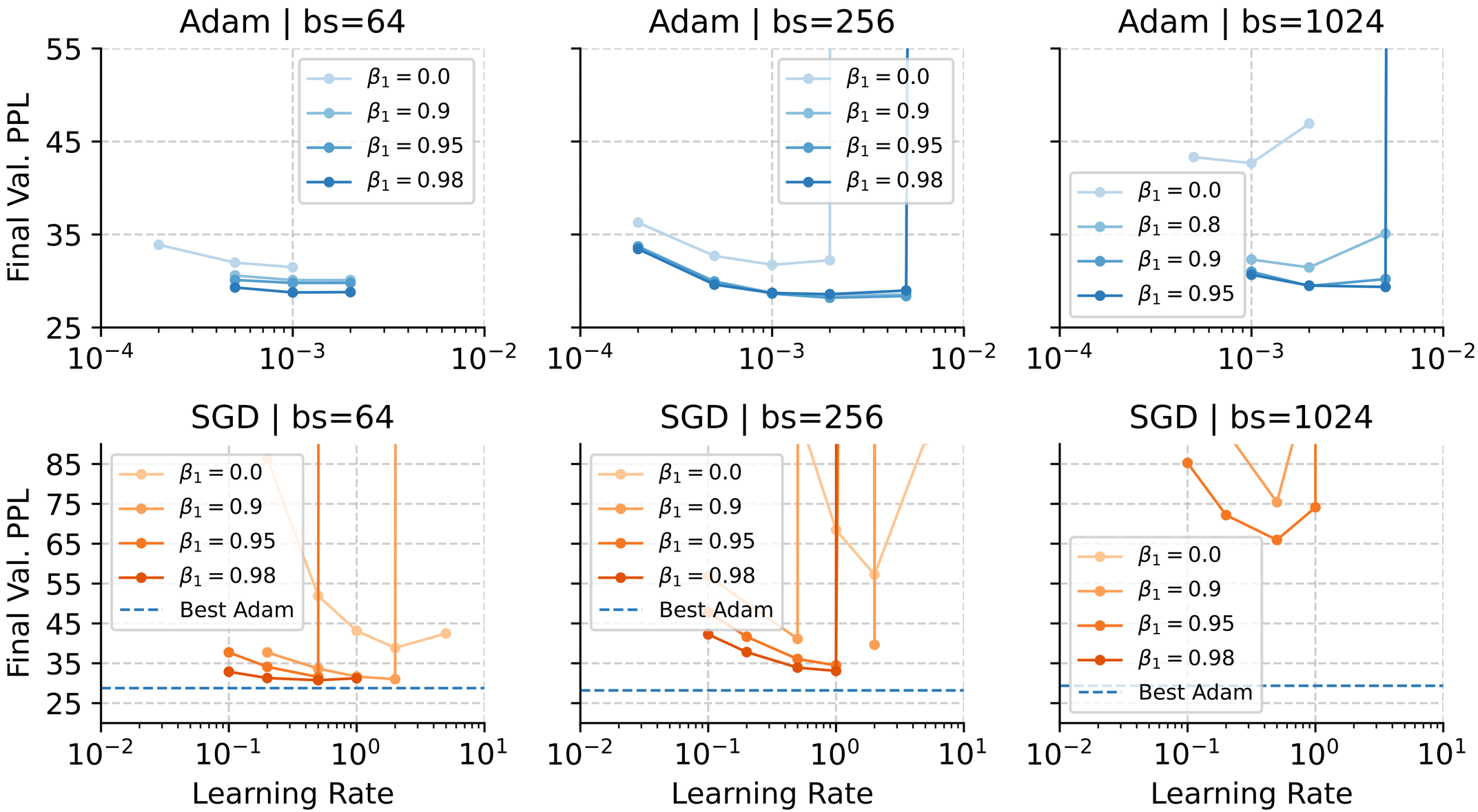
But aren't we forgetting something?

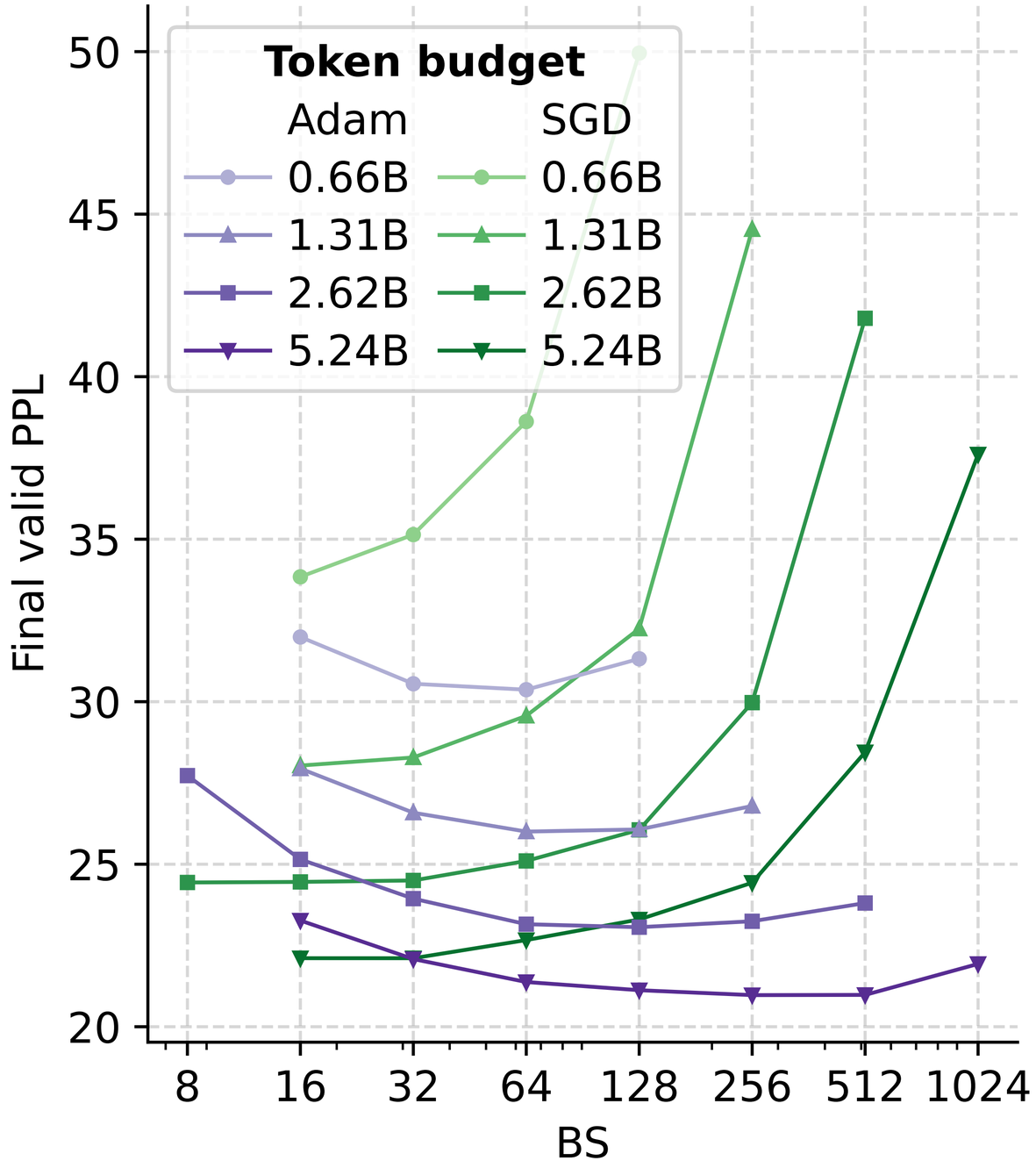
The gap depends on BS
let's dive more into that!

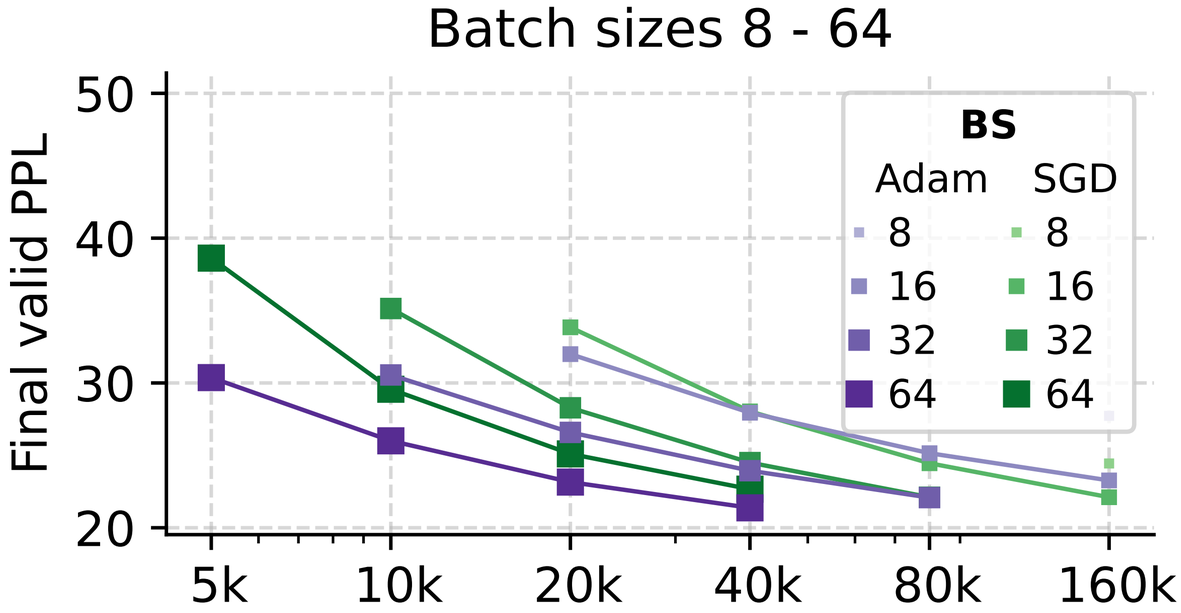
Is your batch size the problem? Revisiting the Adam-SGD gap in language modeling

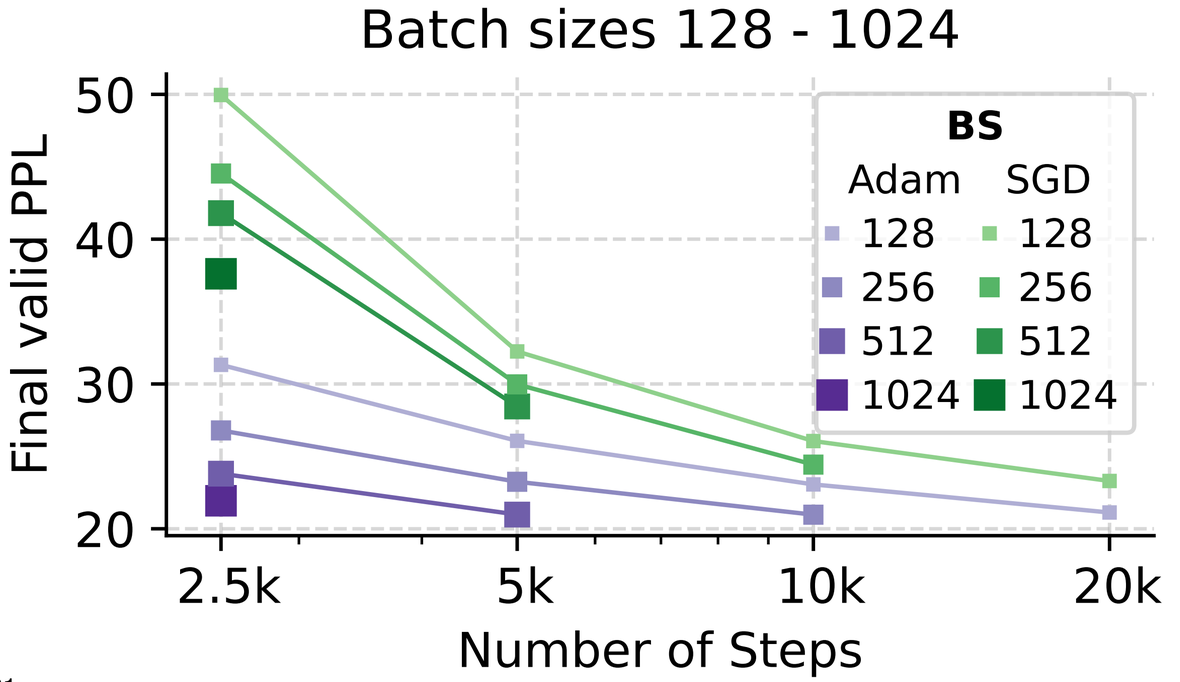
Teodora Srećković TEODORASREC@GMAIL.COM
Jonas Geiping JONAS@TUE.ELLIS.EU
Antonio Orvieto ANTONIO@TUE.ELLIS.EU
Max Planck Institute for Intelligent Systems, ELLIS Institute Tübingen, Tübingen AI Center

Adam vs. SGD training a 160M parameter transformer (1.2 B tokens budget)

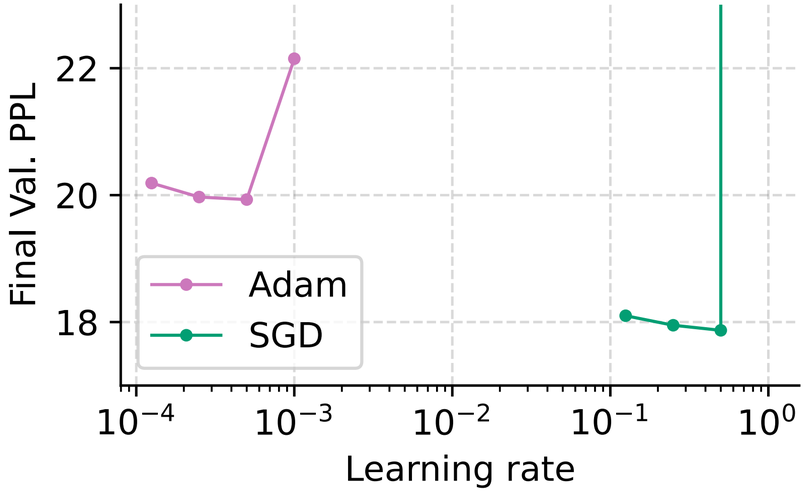


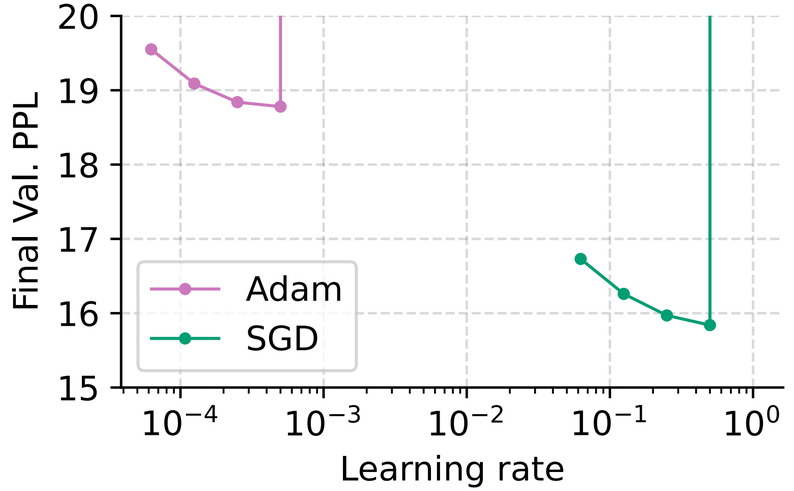

Token budget
Adam
SGD
0.66B
1.31B
2.62B
5.24B
0.66B
1.31B
2.62B
5.24B
Final valid PPL
BS
8
16
32
64
128
256
512
1024
20
25
30
35
40
45
50

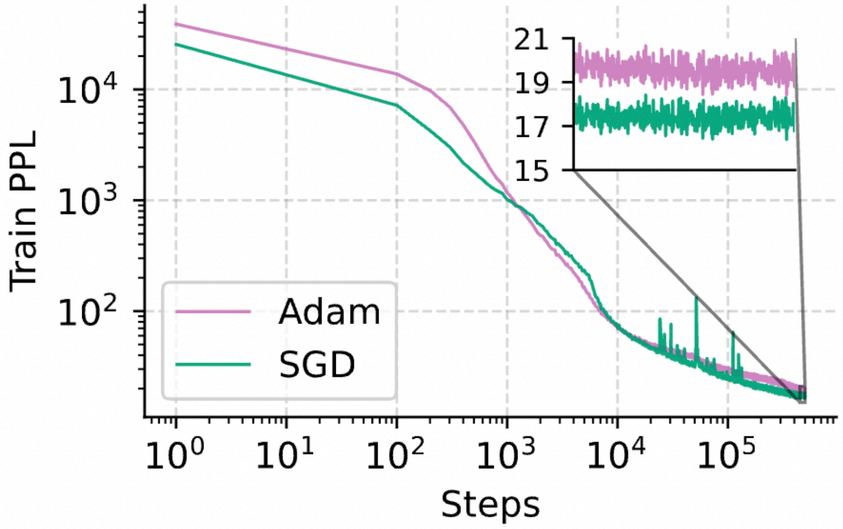

Batch sizes 8 - 64
BS
Adam
SGD
8
16
32
64
8
16
32
64
Final valid PPL
5k
10k
20k
40k
80k
160k
20
30
40
50

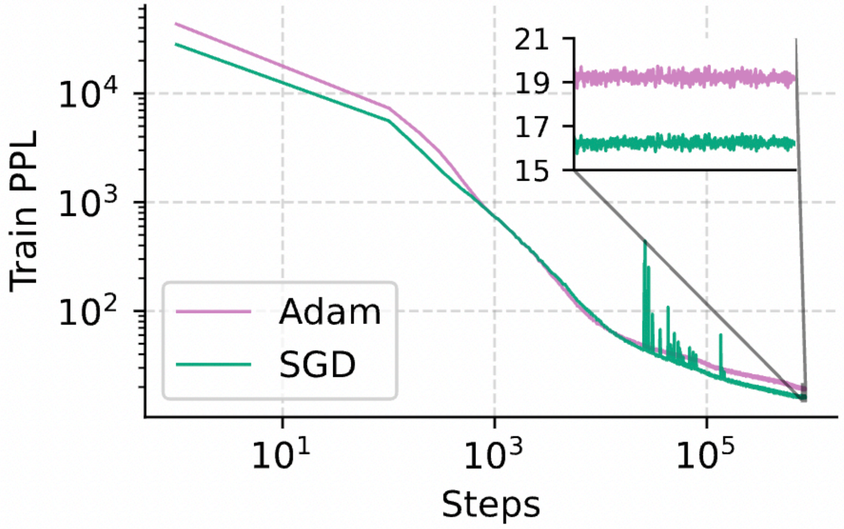

Batch sizes 128 - 1024
BS
Adam
SGD
128
256
512
1024
128
256
512
1024
Final valid PPL
2.5k
5k
10k
20k
Number of Steps
20
30
40
50

Adam < SGD at very low batch sizes, even at larger scales (tuned) !









(a) 410M model on SlimPajama (seq. length 2048, batch size 8, 500k steps) – 1.5 days of training.

(b) 1B model on FineWeb (seq. length 1024, batch size 16, 850k steps) – 5 days of training.

Another paper independently confirmed this!

And yes, Fred already said it ~4 years ago!

Small Batch Size Training for Language Models: When Vanilla SGD Works, and Why Gradient Accumulation Is Wasteful

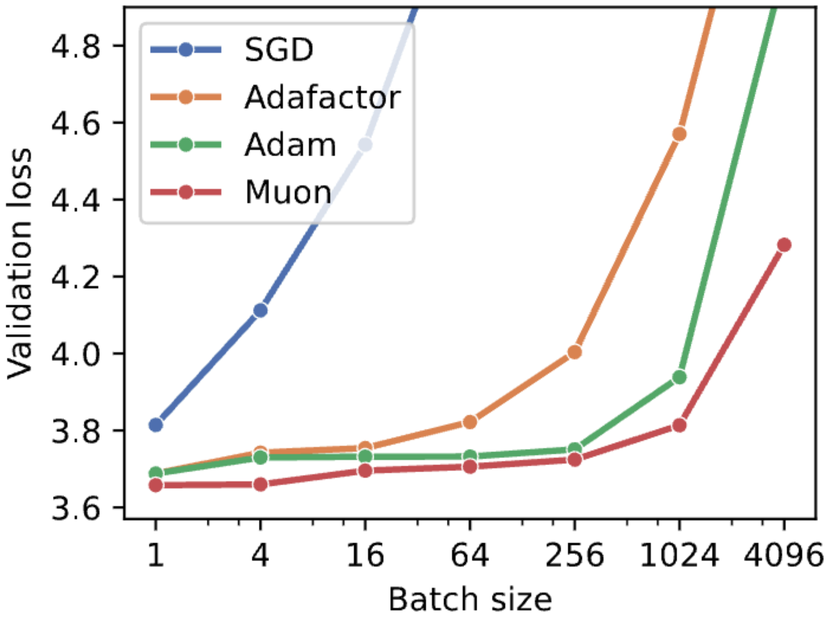
Martin Marek
New York University
martin.m@nyu.edu

Sanae Lotfi
New York University

Aditya Somasundaram
Columbia University

Andrew Gordon Wilson
New York University

Micah Goldblum
Columbia University



(a) Large batches require more sophisticated optimizers

But Why?!

A bit more specifically, the real question is…

Is the superiority of Adam in large batch due to Attention / Transformers / Language modeling?

A bit more specifically, the real question is…

Is the superiority of Adam in large batch due to Attention / Transformers / Language modeling?

The answer is no, no, and no.

Revisiting the Adam–SGD Gap Beyond Single Factors

Chenxiang Zhang, Rustem Islamov, Enea Monzio Compagnoni,
Jun Pang, Aurelien Lucchi, Antonio Orvieto

So, finally (I have been dreaming about this forever)
we did **all the confounders ablations** — with good tuning

Architecture	Dataset	Modality	Transformer
GPT [Radford et al., 2019]	TinyStories [Eldan and Li, 2023]	Language	✓
GPT [Radford et al., 2019]	Fineweb [Penedo et al., 2024]	Language	✓
GPT [Radford et al., 2019]	HG38 [Genome Reference Consortium, 2013]	Genomics	✓
ViT [Dosovitskiy et al., 2021]	HT-I1K [Kunstner et al., 2024]	Vision	✓
ViT [Dosovitskiy et al., 2021]	I21K [Deng et al., 2009]	Vision	✓
GRIT [Ma et al., 2023]	ZINC250K [Irwin et al., 2012]	Graphs	✓
GCNN [Dauphin et al., 2017]	TinyStories [Eldan and Li, 2023]	Language	✗
GCNN [Dauphin et al., 2017]	Fineweb [Penedo et al., 2024]	Language	✗
ConvNext [Liu et al., 2022]	I21K [Deng et al., 2009]	Vision	✗
ResNet [He et al., 2016]	I21K [Deng et al., 2009]	Vision	✗
GAT [Veličković et al., 2018]	ZINC250K [Irwin et al., 2012]	Graphs	✗

Δ^* = optimal gap.

- Positive = SGD better
- Negative = Adam better

Baseline: Softmax Attention. All like Teodora's paper

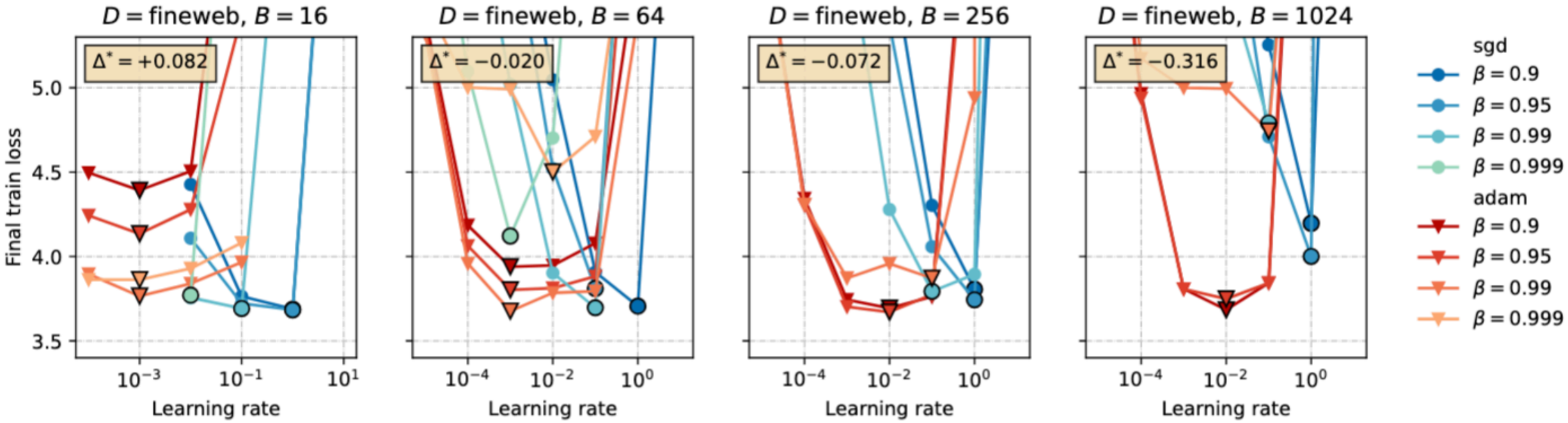


Figure 8: GPT 50M parameters trained on FineWeb 1B tokens.

Baseline: ~~Softmax Attention~~. All like Teodora's paper

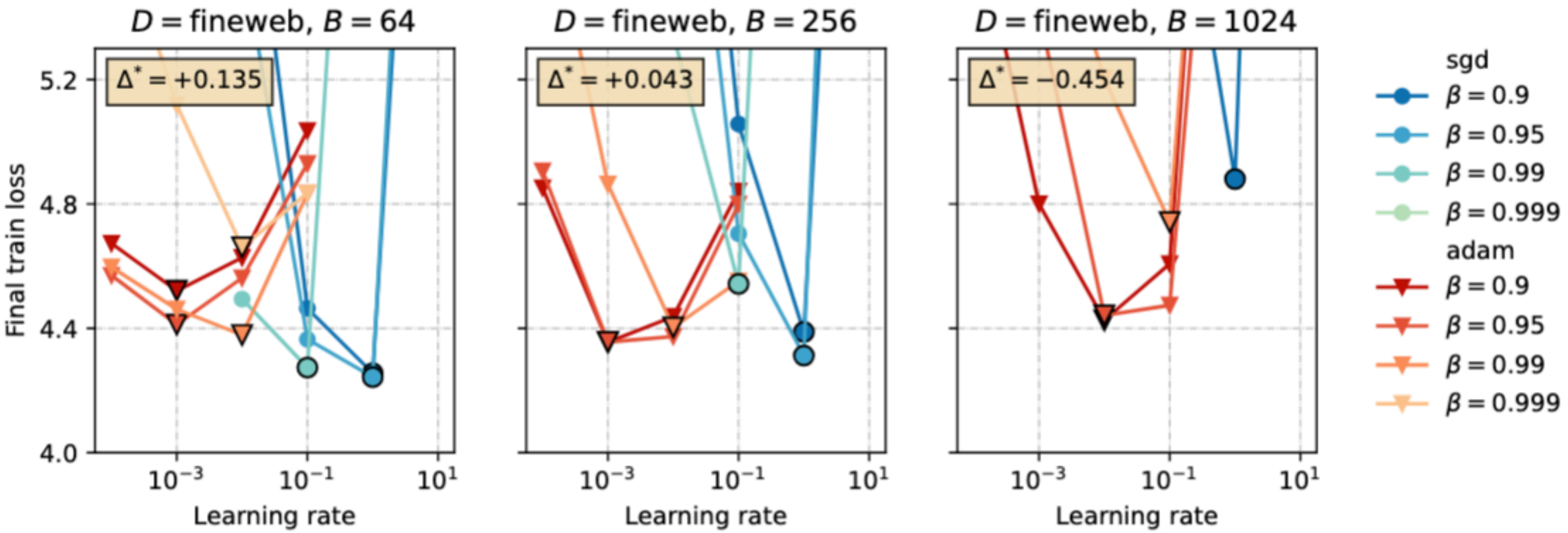
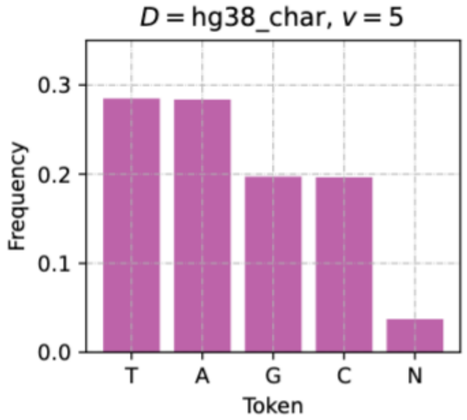


Figure 12: GCNN 53M parameters trained on FineWeb 1B.

Gated convolution network has same exact behavior
So not really about Softmax..



Softmax Attention, next token prediction but now… we have the **human genome as dataset (HG38) !**

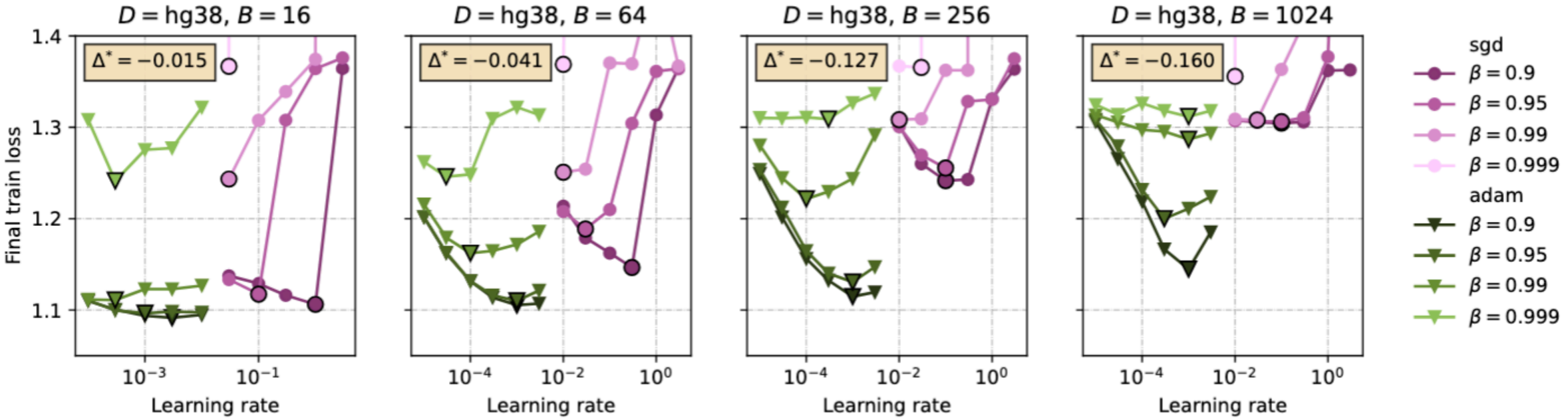


Figure 10: GPT 10M parameters trained on HG38 2.7B tokens, preprocessed with `char` tokenizer.

Adam is still WOW. So mmh, does not seem to be related to natural language!

Is this then about in some way **classification?**
Take the ZINC dataset for molecular property prediction (graph regression)

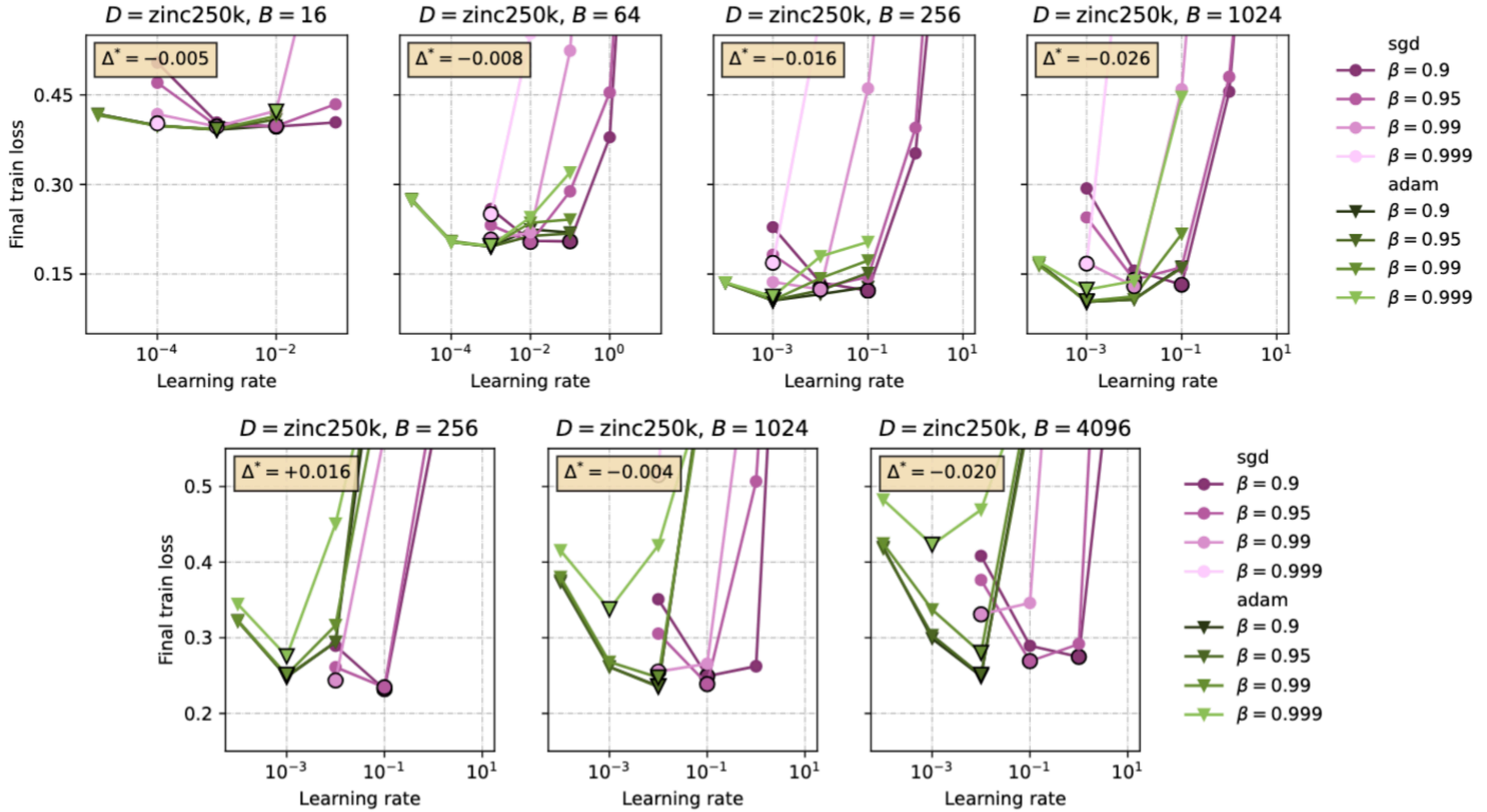
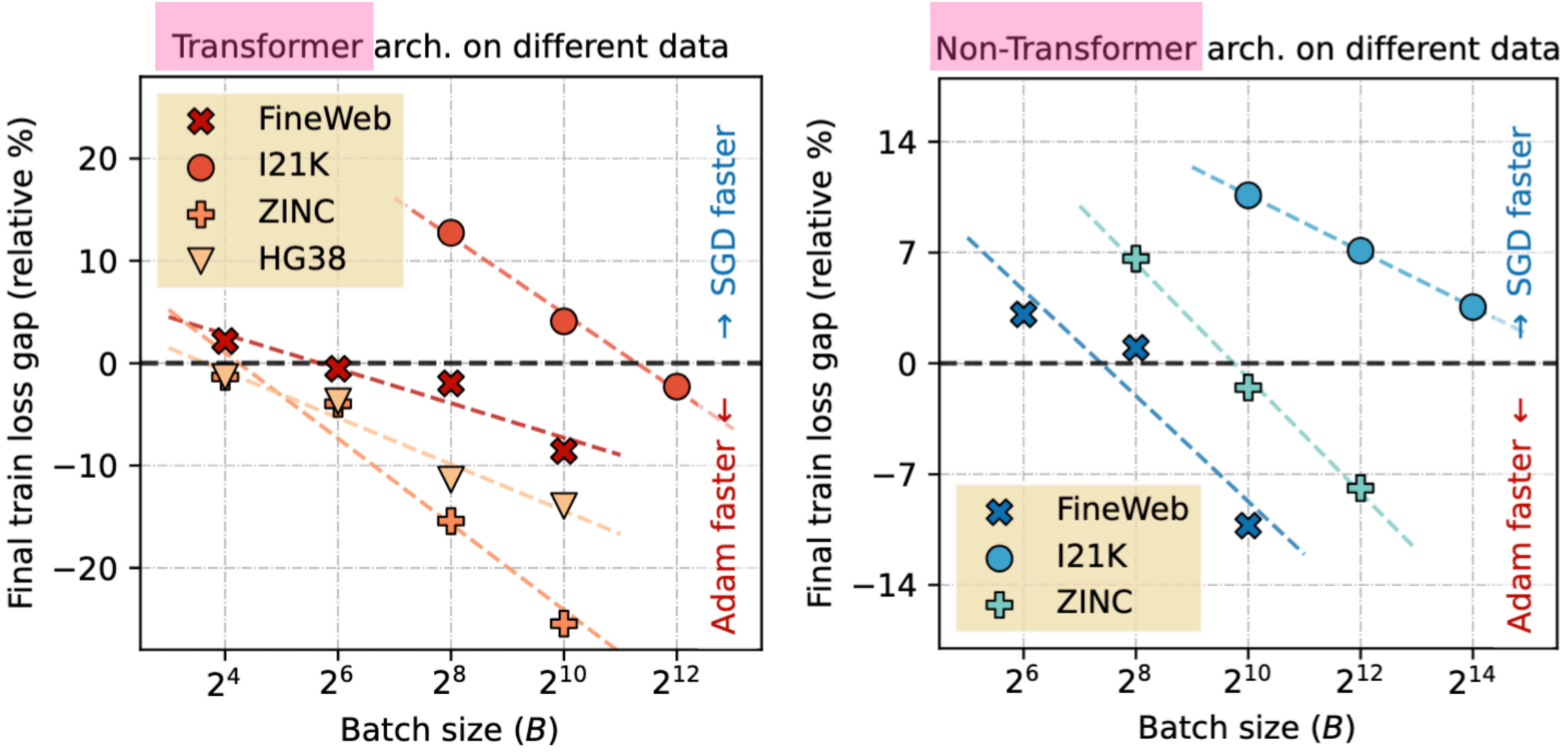


Figure 11: GRIT (top) and GAT (bottom) trained on ZINC250K.

Both GRIT and GAT (first is transformer, second message passing, have same trend)


Transformer arch. on different data
FineWeb
I21K
ZINC
HG38
Final train loss gap (relative %)
20
10
0
−10
−20
2^4
2^6
2^8
2^{10}
2^{12}
Batch size (B)
→ SGD faster
Adam faster ←
Non-Transformer arch. on different data
FineWeb
I21K
ZINC
Final train loss gap (relative %)
14
7
0
−7
−14
2^6
2^8
2^{10}
2^{12}
2^{14}
Batch size (B)
→ SGD faster
Adam faster ←

So, how can we understand this?

Intuition is simple!!!

Assumption: gradient noise is i.i.d. with constant 1-sample covariance Σ

For **SGD**, the following is a weak-first-order-approx. :

drift diffusion

$$dX_t = -\nabla f(X_t)dt + \sqrt{\frac{\eta\Sigma}{B}}dW_t$$

Adaptive Methods through the Lens of SDEs: Theoretical Insights on the Role of Noise
Enea Monzio Compagnoni[1], Tianlin Liu[1], Rustem Islamov[1], Frank Norbert Proske[2], Antonio Orvieto[3,4,5], and Aurelien Lucchi[1]

For **SignSGD**, the following (Compagnoni et al. 24) is a weak-first-order-approx. :

drift diffusion

$$dX_t = -\operatorname{erf}\left(\sqrt{\frac{B}{2}}\Sigma^{-\frac{1}{2}}\nabla f(X_t)\right)dt + \sqrt{\eta}\left[I_d - \operatorname{diag}\left(\operatorname{erf}\left(\frac{\sqrt{B}\Sigma^{-\frac{1}{2}}\nabla f(X_t)}{\sqrt{2}}\right)\right)^2\right]^{1/2} dW_t$$

* i.e., algorithms follow flow for small η

UNDERSTANDING GRADIENT ORTHOGONALIZATION FOR DEEP LEARNING VIA NON-EUCLIDEAN TRUST-REGION OPTIMIZATION

A PREPRINT

Dmitry Kovalev
Yandex Research
dakovalev1@gmail.com

Deriving Hyperparameter Scaling Laws via Modern Optimization Theory

Egor Shulgin, Jörg K.H. Franke, Dimitri von Rütte, Tianyue H. Zhang, Niccolò Ajroldi, Korbinian Pöppel, Bernhard Schölkopf, Aaron Klein, Peter Richtárik, Antonio Orvieto

OPEN EURO LLM

Kovalev:

$$\min_{1 \leq k \leq K} \mathbb{E}\left[\|\nabla f(x^k)\|_\star\right] \leq \frac{\Delta_0}{\eta K} + \frac{2\rho\sigma}{\alpha\sqrt{b}K} + 2\rho\sigma\sqrt{\frac{\alpha}{b}} + \frac{7L\eta}{2} + \frac{2L\eta}{\alpha}$$

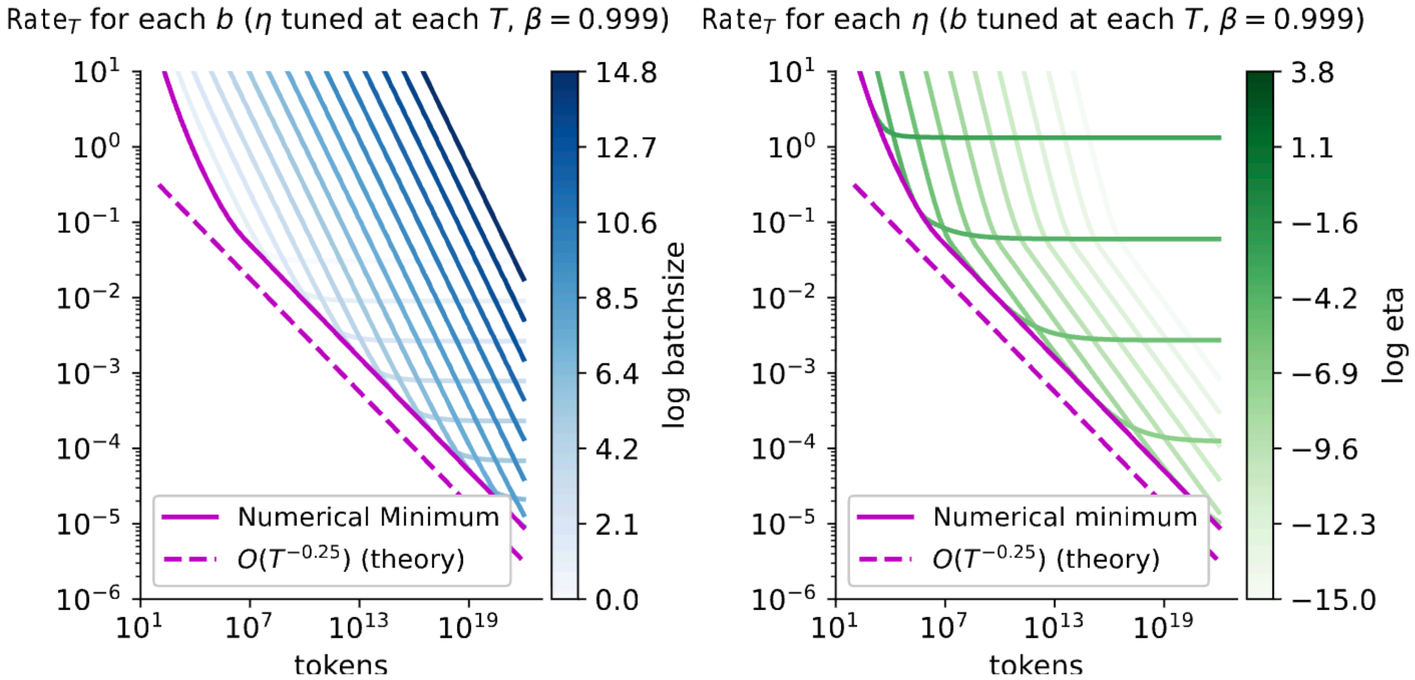
Minimize over everything for each token budget $T = Kb$.

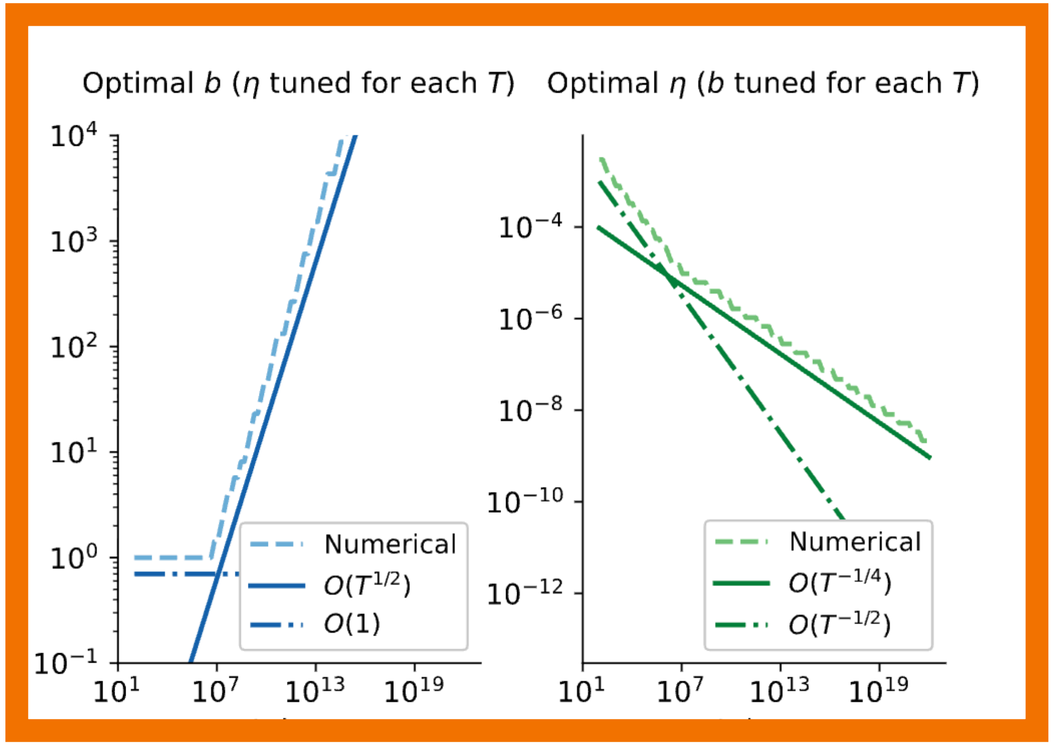
Kovalev:

$$\min_{1\le k\le K} \mathbb{E}\left[\|\nabla f(x^k)\|_\star\right] \le \frac{\Delta_0}{\eta K} + \frac{2\rho\sigma}{\alpha\sqrt{bK}} + 2\rho\sigma\sqrt{\frac{\alpha}{b}} + \frac{7L\eta}{2} + \frac{2L\eta}{\alpha}$$

Minimize over everything for each token budget $T = Kb$.

There exist indeed an optimal batch size for LMO methods





But for SGD, batch size 1 is already optimal!

Also check paper by Rustem and Tony!

On the Role of Batch Size in Stochastic Conditional Gradient Methods

Rustem Islamov[1,4,†], Roman Machacek[2], Aurelien Lucchi[1], Antonio Silveti-Falls[3]
Eduard Gorbunov[4,⋆], Volkan Cevher[5,⋆]

[1]University of Basel, Switzerland, [2]University of Bern, Switzerland

[3]CentraleSupélec, France, [4]MBZUAI, UAE, [5]EPFL, Switzerland

Thanks!!!