The Road Less Scheduled

Aaron Defazio

Research Scientist

FAIR

Meta

JOINT WORK WITH

Xingyu (Alice) Yang

Harsh Mehta

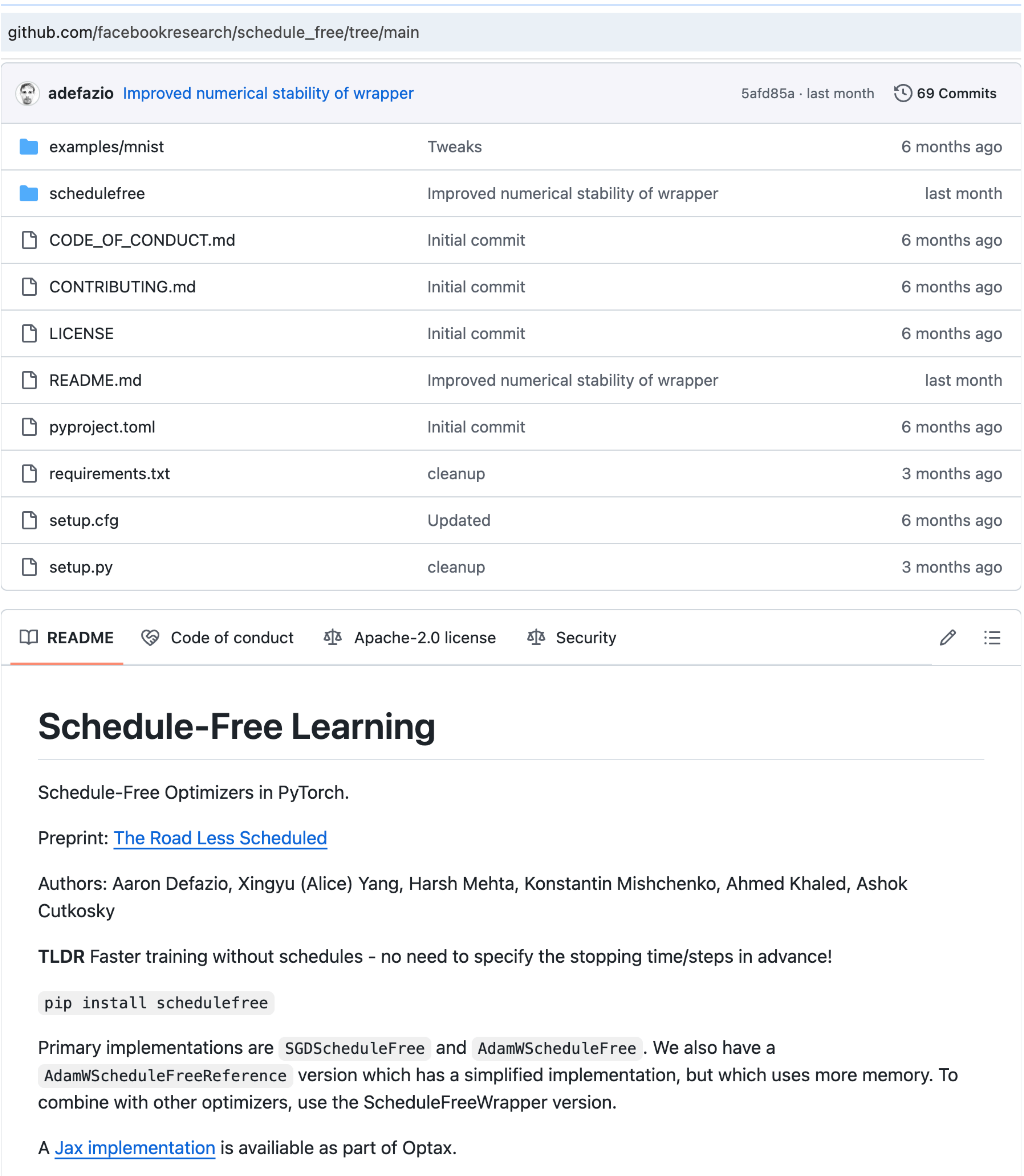
Konstantin Mishchenko

Ahmed Khaled

Ashok Cutkosky

1.9k stars

downloads 351k | downloads/month 118k



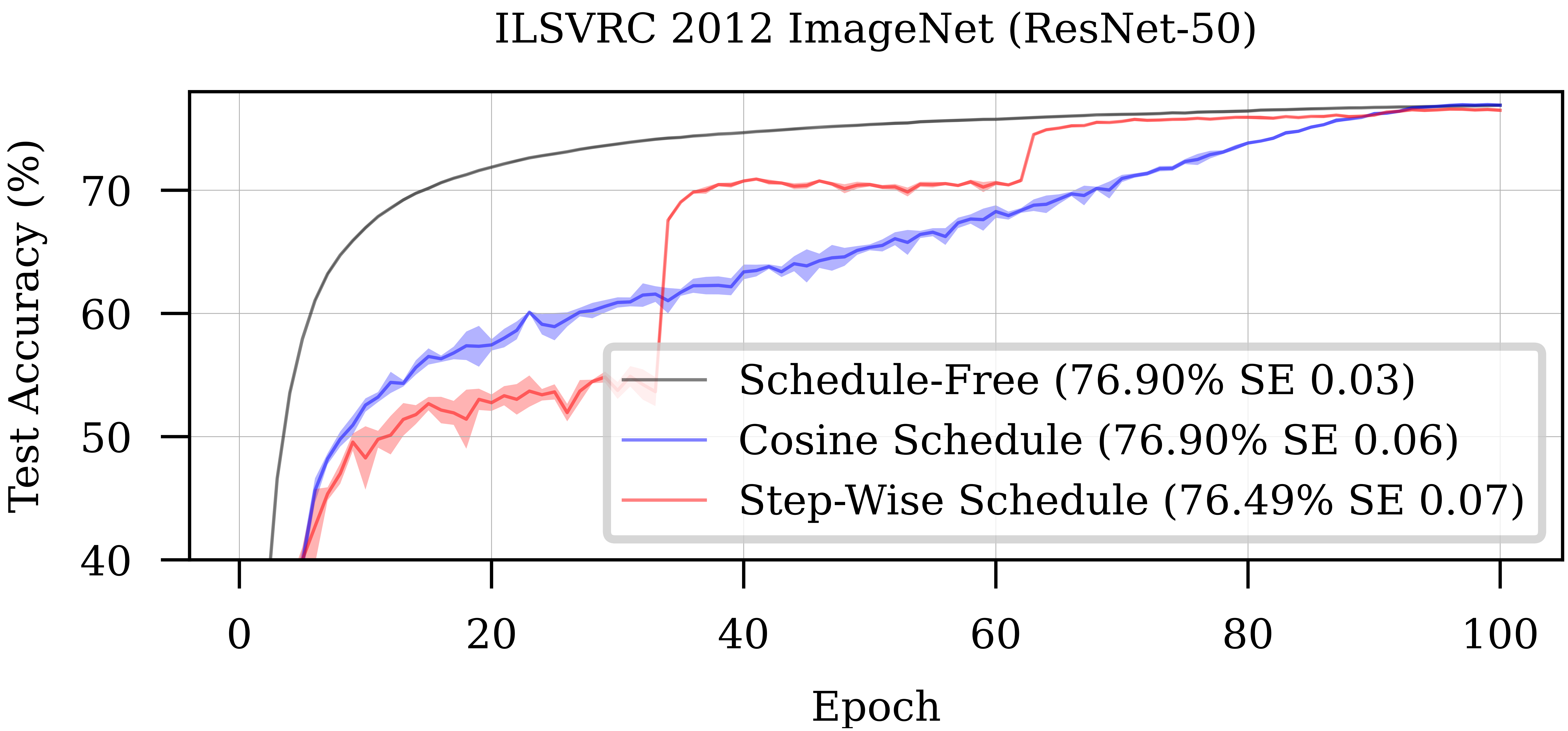
Schedule-Free Learning

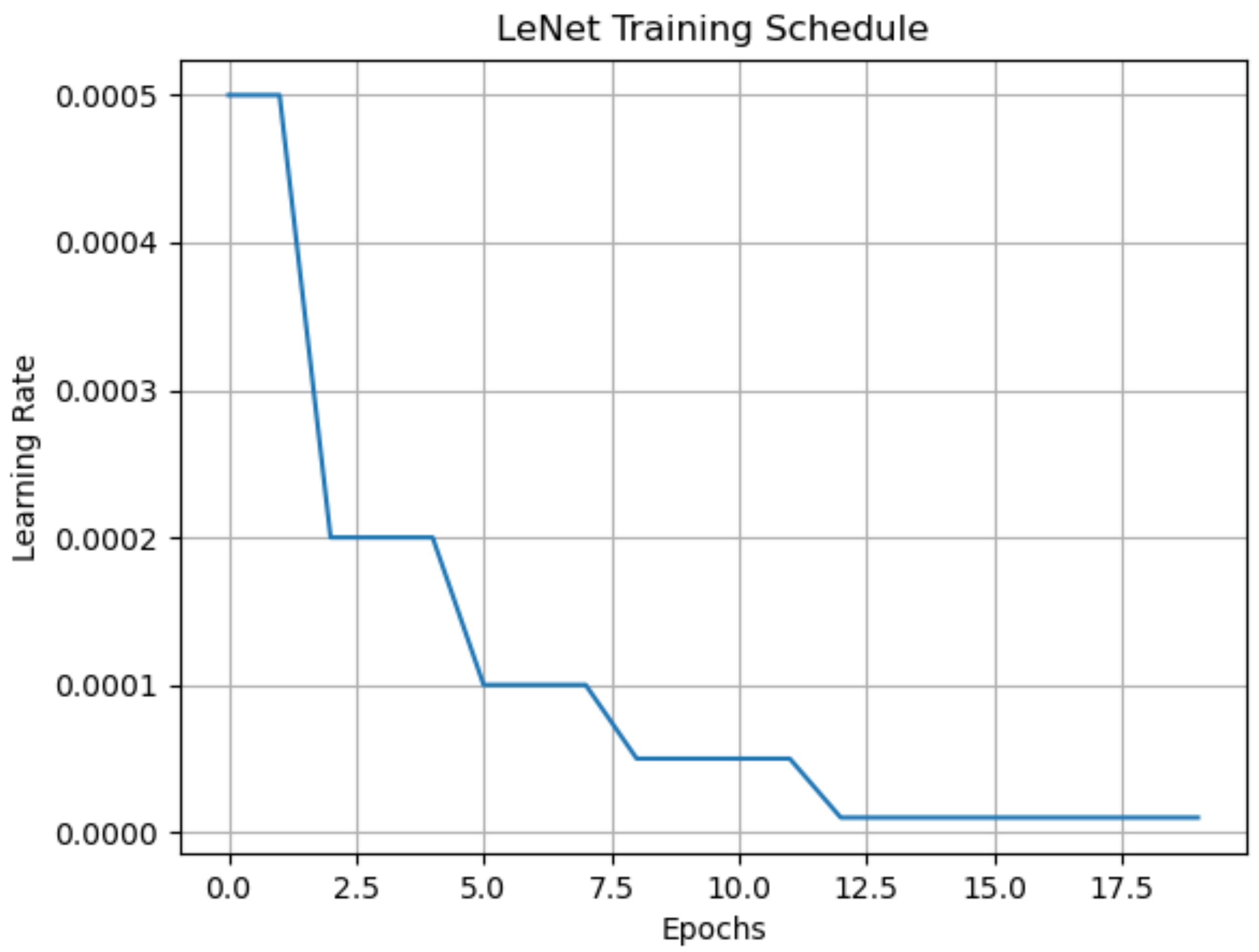
1. An alternative to schedules that doesn't need to know the stopping time T in advance (supports anytime stopping)
2. Obtains the theoretically optimal rate of convergence for Lipschitz convex problems
3. Works in practice: matches or outperforms cosine schedules!

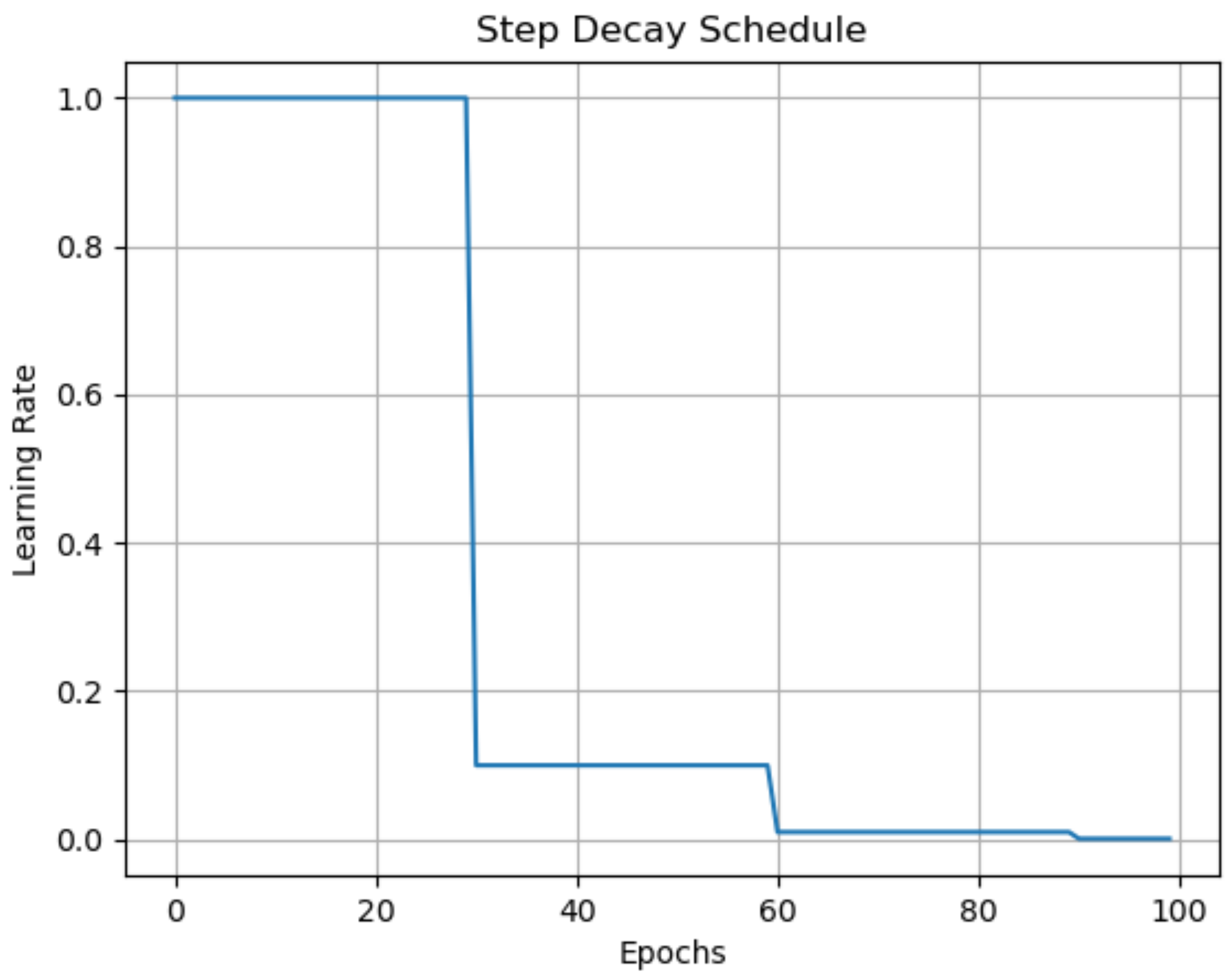
   Proof: AlgoPerf Challenge self-tuning track **winner**

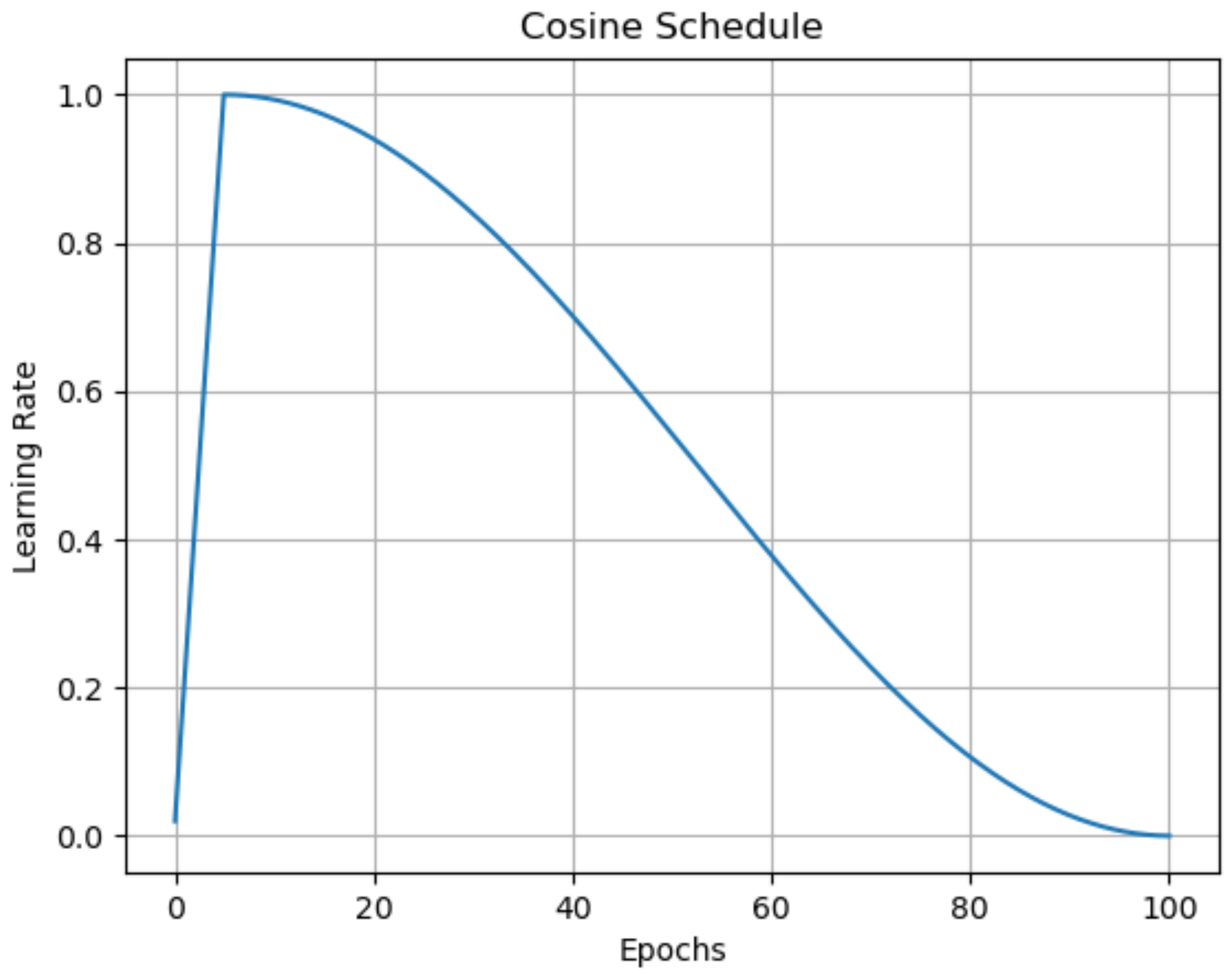
Faster early convergence!

Smooth loss curves!









1989 - 1998

Handwritten Digit Recognition with a Back-Propagation Network

Y. Le Cun, B. Boser, J. S. Denker, D. Henderson,
R. E. Howard, W. Hubbard, and L. D. Jackel
AT&T Bell Laboratories, Holmdel, N. J. 07733

Gradient-Based Learning Applied to Document Recognition

YANN LECUN, MEMBER, IEEE, LÉON BOTTOU, YOSHUA BENGIO, AND PATRICK HAFFNER

2012

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

"we adjusted manually throughout training. The heuristic which we followed was to divide the learning rate by 10 when the validation error rate stopped improving with the current learning rate"

2017

Published as a conference paper at ICLR 2017

SGDR: STOCHASTIC GRADIENT DESCENT WITH WARM RESTARTS

Ilya Loshchilov & Frank Hutter
University of Freiburg
Freiburg, Germany,
{ilya,fh}@cs.uni-freiburg.de

THEORY-PRACTICE MISMATCH #1

We never use SGD in the precise form as we analyze!

In practice we return the **last iterate,** whereas we analyze the **average** iterate.

$$x_{t+1} = x_t - \gamma_t g_t$$
$$\bar{x}_T = \frac{1}{T} \sum_{t=1}^{T} x_t$$

SGD with averaging gives exactly worst-case optimal rates for several complexity classes, notably the convex + Lipschitz setting.

Without averaging you get a log(T) worse rate

But what do experiments suggest?

Folk-law: Averaging is bad and unnecessary, it’s an artifact of the analysis not reflective of real world problems.

Folk-law: The $\gamma_t = D/G\sqrt{t}$ schedule is bad, use one of the empirically better schedules we found via trial and error.

A **flat schedule is even worse**!

A Perspective on Scheduling

$$x_{t+1} = x_t - \gamma_t g_t$$

$$\bar{x}_T = \frac{1}{T}\sum_{t=1}^{T} x_t$$

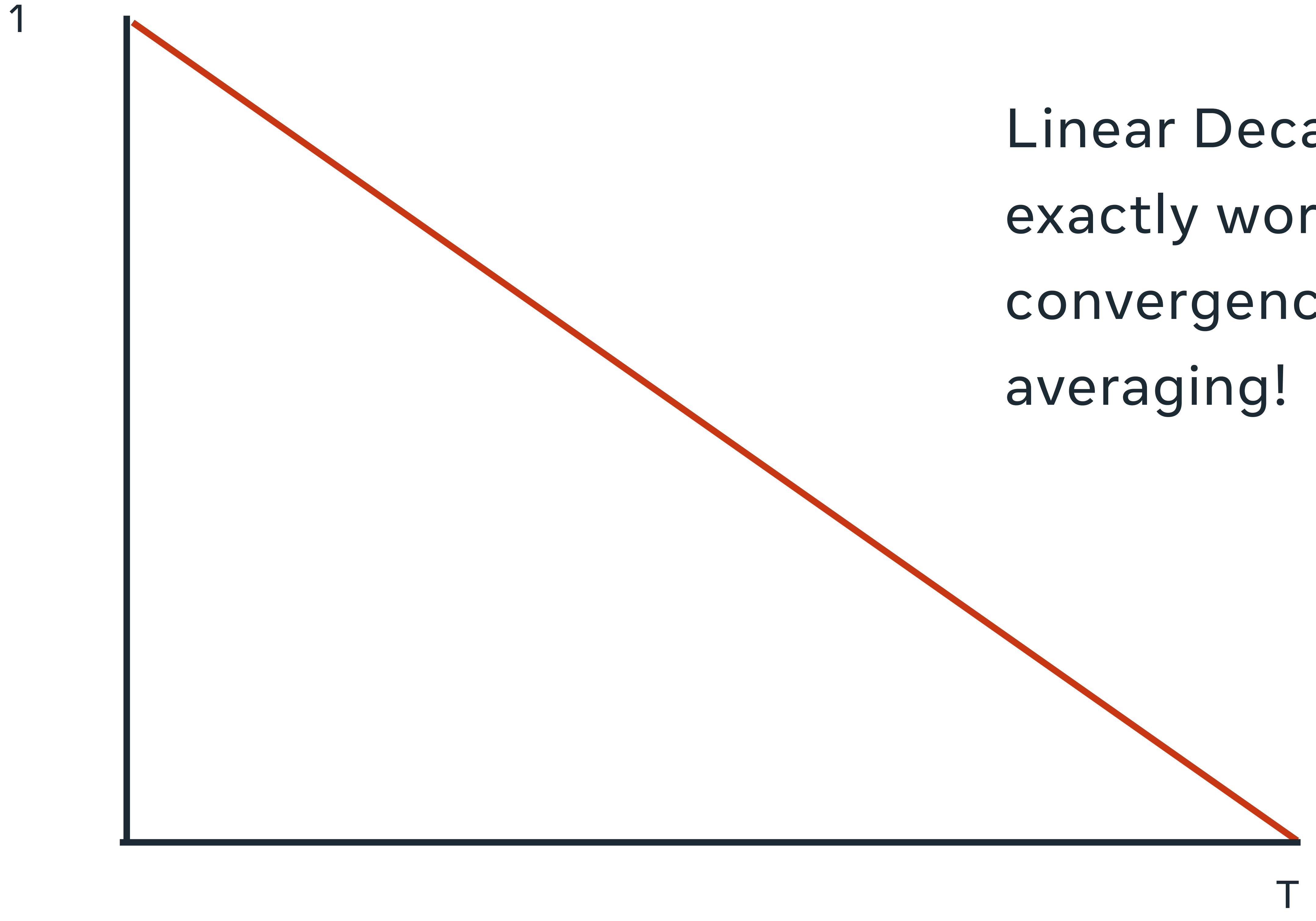
The schedules used by experimentalists are not replacing this $D/G\sqrt{T}$ part!

They are actually replacing **averaging.**

High-Performance schedules arise naturally from **theory** by analyzing the last iterate x_T rather than $\bar{x}_T$

Theoretically Optimal Schedules
(for convex problems)

$$\gamma_t = \frac{D}{G\sqrt{T}} \cdot \left(1 - \frac{t}{T}\right)$$



Linear Decay Schedule

Linear Decay Schedules give exactly worst-case optimal convergence rates without averaging!

$$f(\bar{x}_T) - f_* \leq \frac{DG}{\sqrt{T}}$$

EXACT CONVERGENCE RATE OF THE LAST ITERATE IN SUBGRADIENT METHODS

MOSLEM ZAMANI* AND FRANÇOIS GLINEUR †

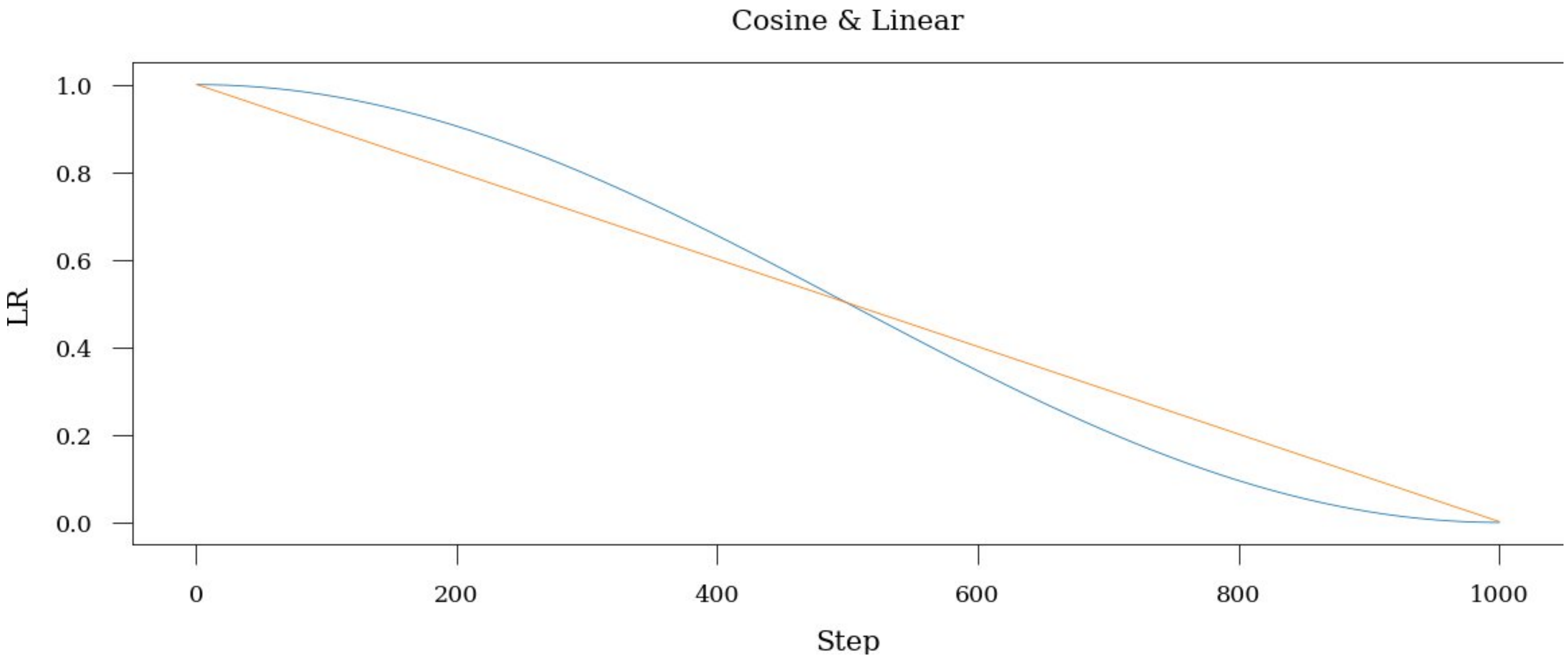
When, Why and How Much? Adaptive Learning Rate Scheduling by Refinement

Aaron Defazio ADEFAZIO@META.COM
FAIR, Meta
Ashok Cutkosky ASHOK@CUTKOSKY.COM
Boston University
Harsh Mehta HARSHM@GOOGLE.COM
Google Research
Konstantin Mishchenko KONSTA.MISH@GMAIL.COM
Samsung AI Center

Linear Decay works **extremely well** in practice (when combined with warmup) …

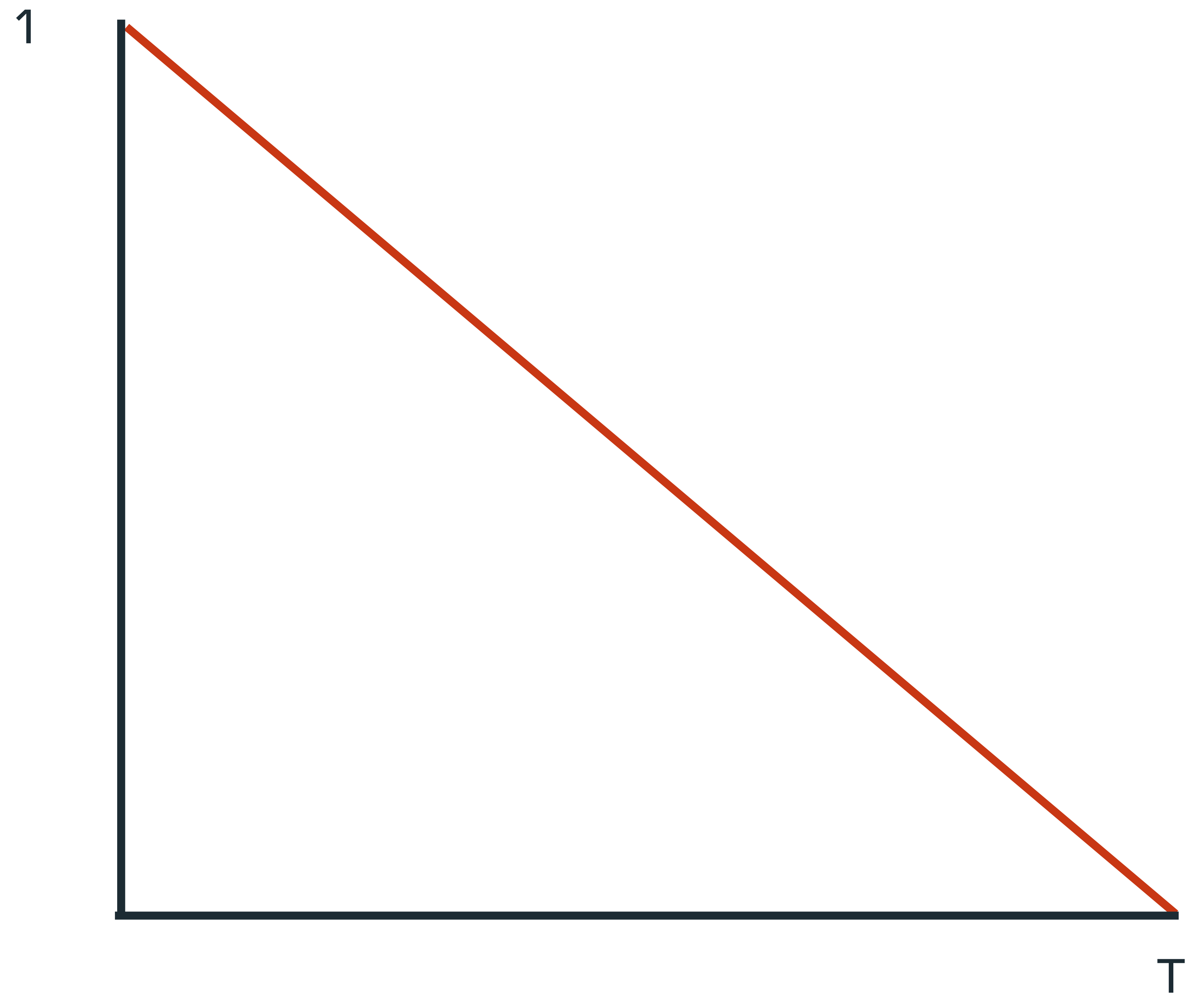
Almost always better than cosine.

Cosine largely wastes the last 5% of the run by using too small a learning rate



Stop using cosine! It is complete nonsense

Linear Decay emulates Averaging



$$\bar{x}_T = \frac{1}{T}\sum_{t=1}^{T} x_t$$

Gradient from t=1 appears in all terms in the average: **weight 1**

Gradient from t=T/2 appears in half the terms in the in the average: **weight 1/2**

Gradient from t=3T/4 appears in 1/4 of the terms in the in the average: **weight 1/4**

.... same weighting as for linear decay

THEORY-PRACTICE MISMATCH #2

If linear decay emulates averaging … and works so well …

….. **why doesn't averaging work**?

Averaging needs momentum (done right)

Schedule-Free Learning Paradigm

Interpolation beta=0.9

(A kind of momentum)

$$y_t = (1-\beta)z_t + \beta x_t$$

$$z_{t+1} = z_t - \gamma \nabla f(y_t)$$

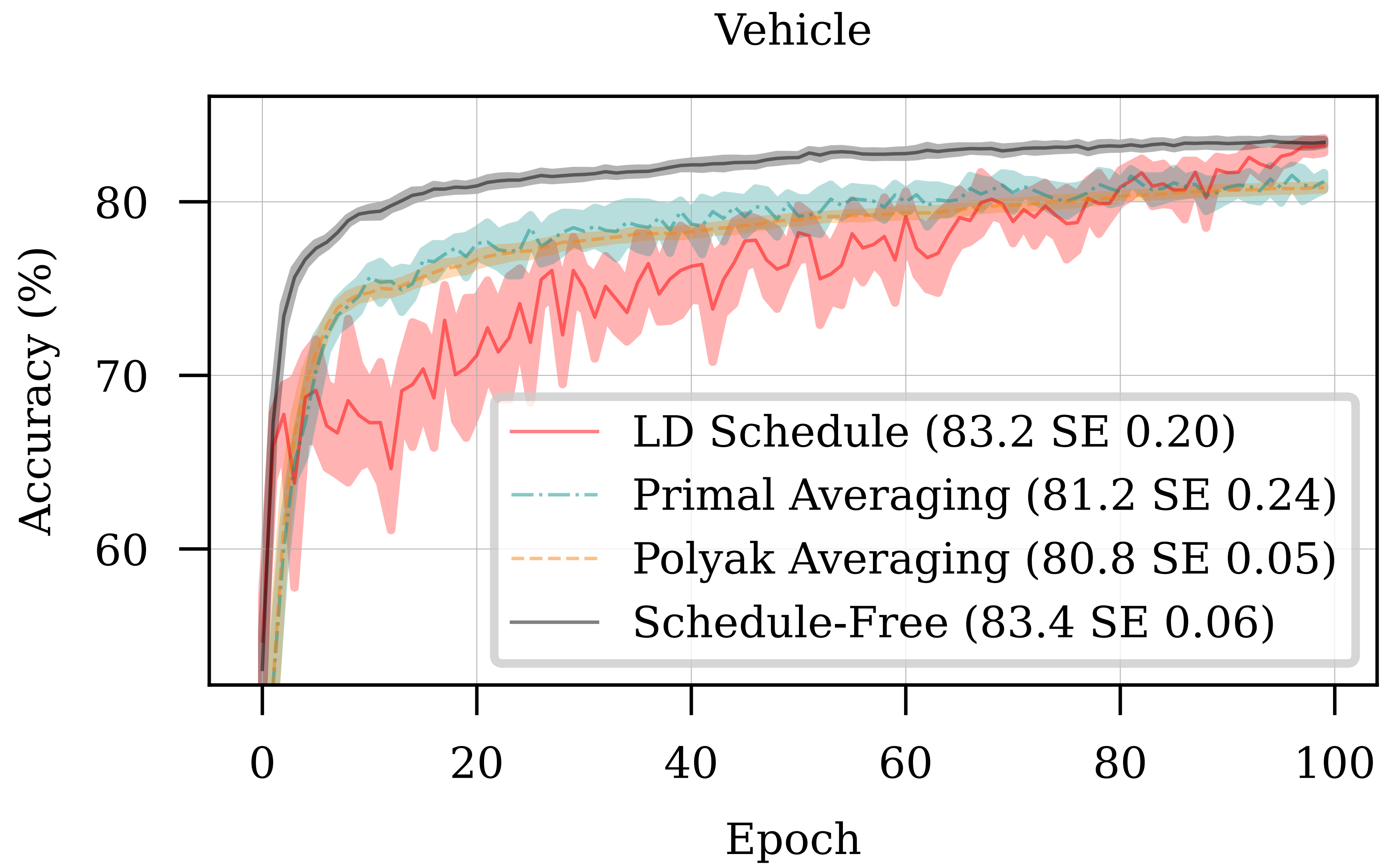
Running Average

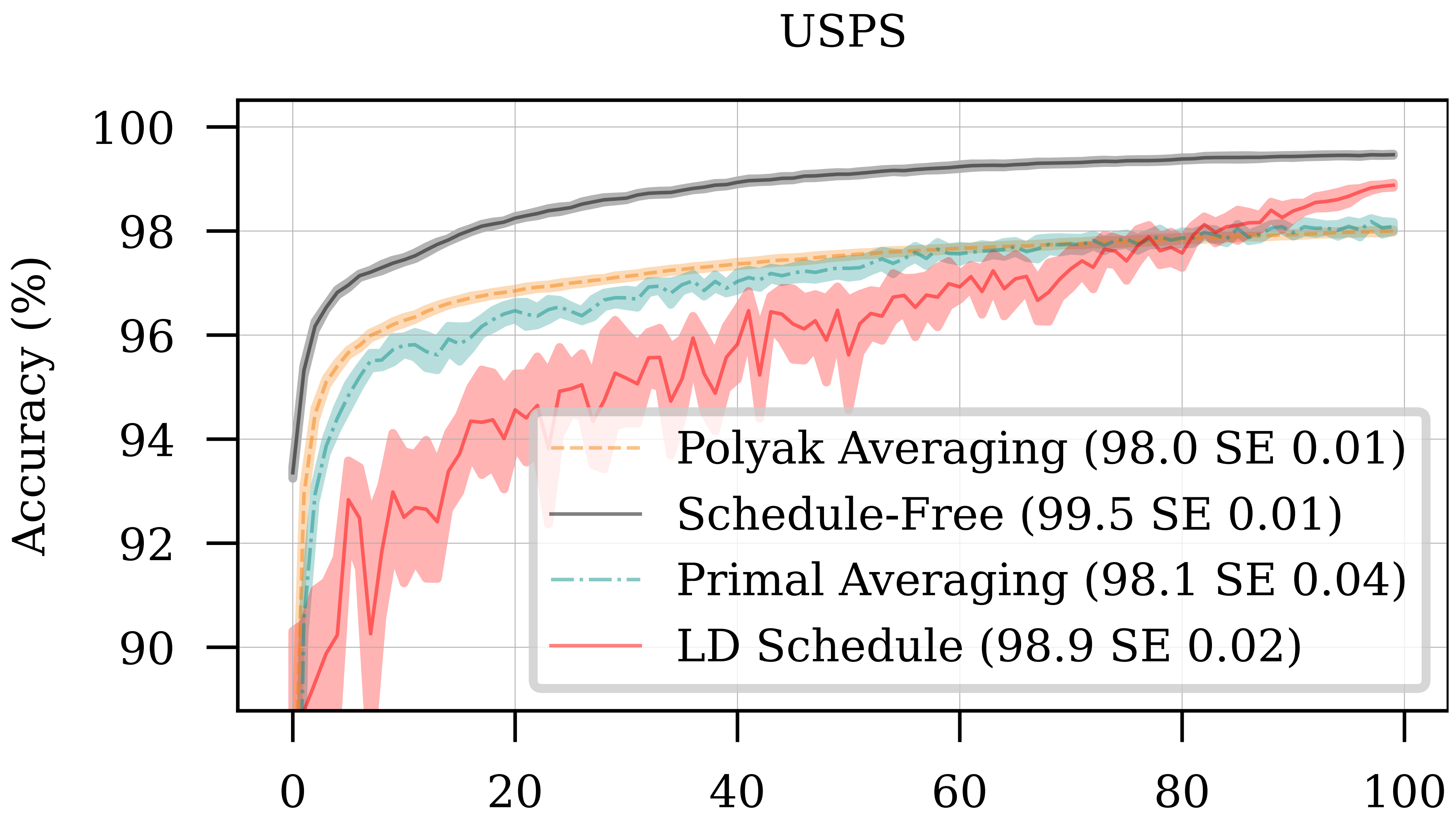
Equivalent to:

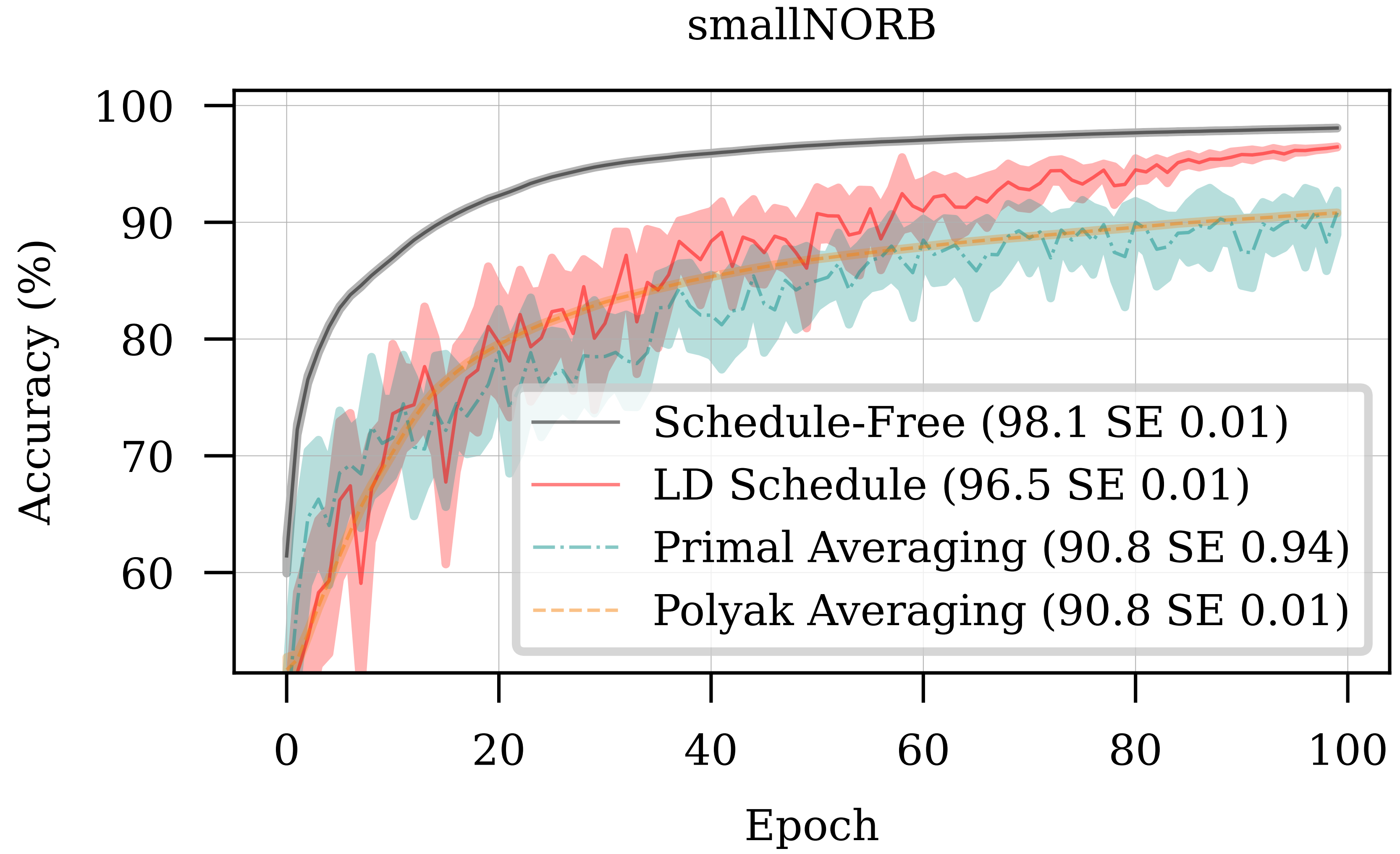
$$x_{t+1} = \left(1 - \frac{1}{t+1}\right) x_t + \frac{1}{t+1} z_{t+1}$$

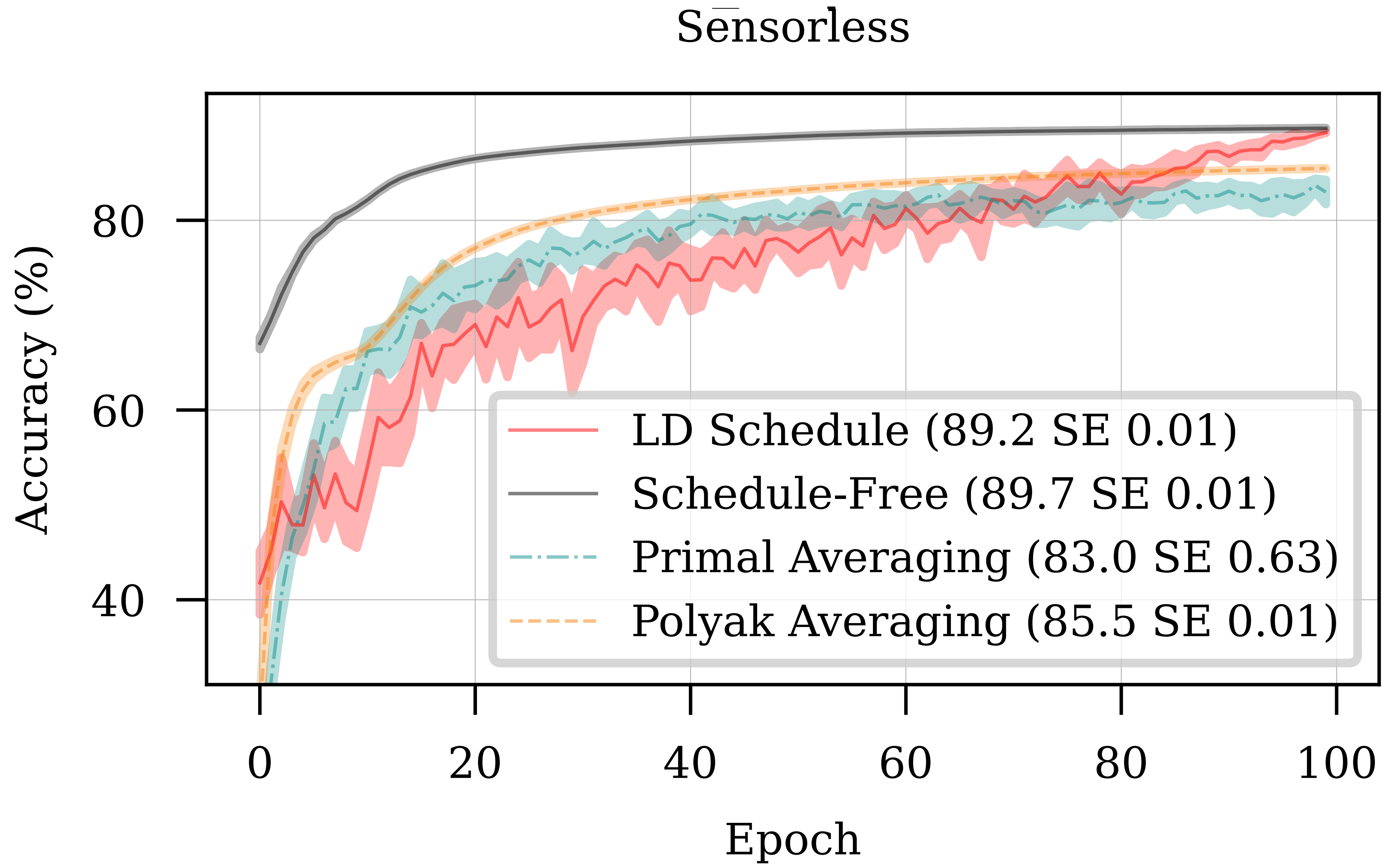
$$x_t = \frac{1}{t} \sum_{i=1}^{t} z_i$$

Even for convex problems, Schedule-Free outperforms classical averaging and linear decay schedules!









Schedule-Free does momentum in a different, more gradual way….

$$y_t = (1-\beta)z_t + \beta x_t$$

$$z_{t+1} = z_t - \gamma \nabla f(y_t)$$

$$x_{t+1} = \left(1 - \frac{1}{t+1}\right)x_t + \frac{1}{t+1}z_{t+1}$$

$\beta = 0.9$ results in the current gradient evaluation point y containing 0.1 of the most recent gradient g_{t-1}

Classical momentum does the same thing! 0.1 of the most recent gradient is included in the step

$$m_{t+1} = \beta m_t + \left(1-\beta\right)\nabla f(x_t)$$
$$x_{t+1} = x_t - \alpha_t m_{t+1}$$

But classical momentum incorporates the rest of the gradient over the next ~10 steps, whereas Schedule-Free incorporates it much **slower**, of the reminder of optimization

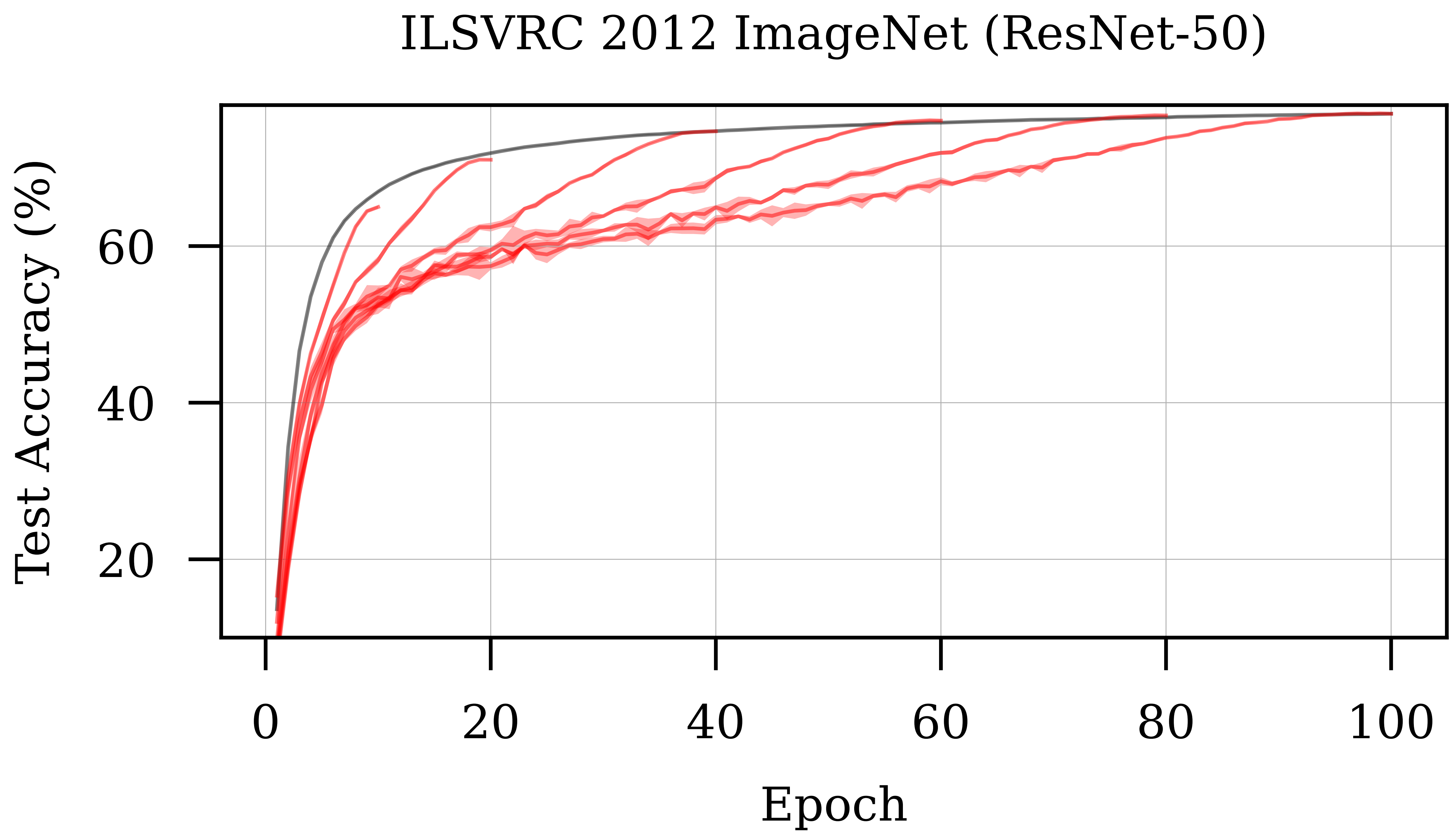
For general convex Lipschitz functions

Schedule-Free gives exactly optimal worst-case rates for ANY beta, whereas classical momentum for any fixed beta gives worse rates.

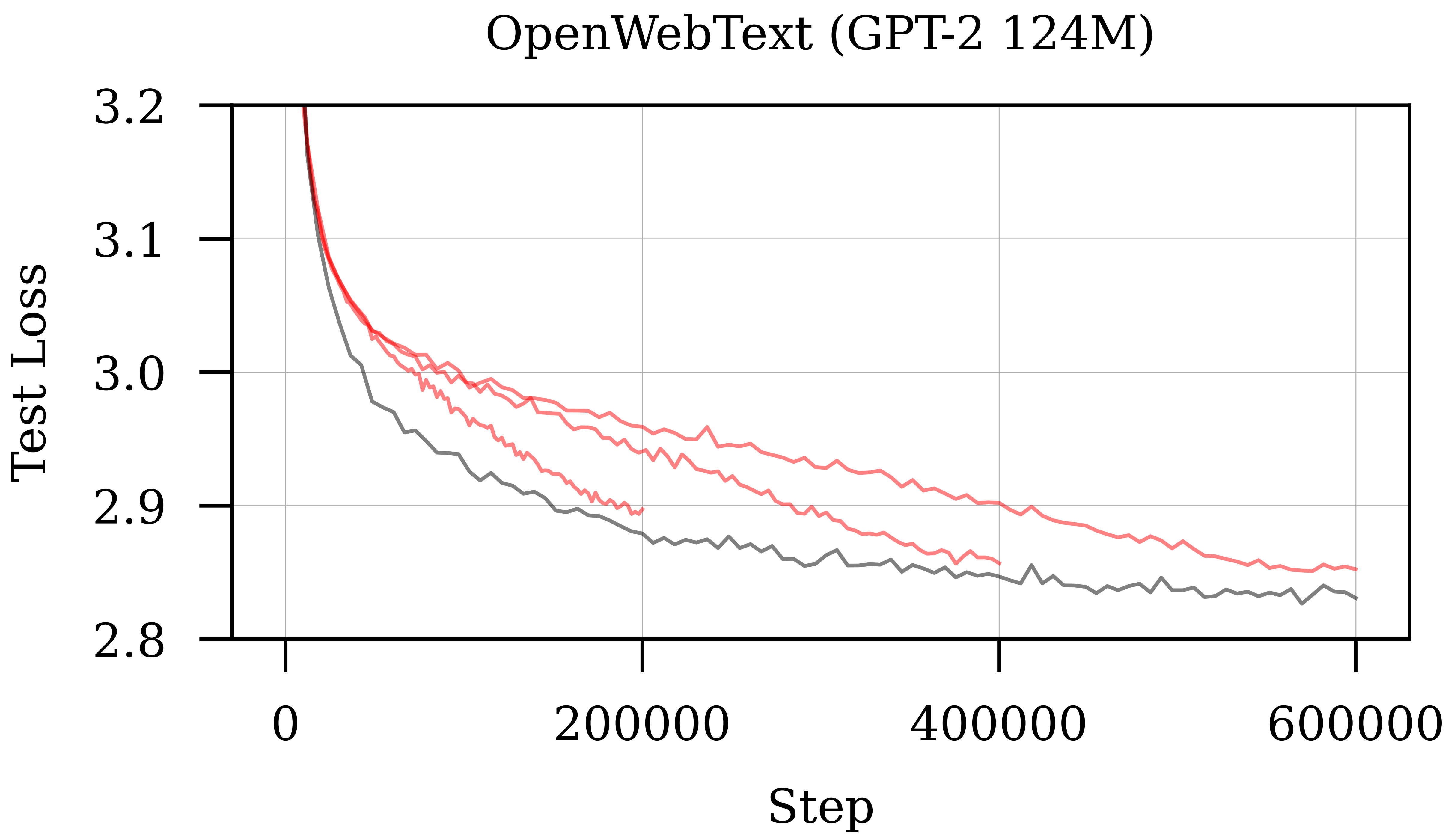
Theorem 1. *Suppose F is a convex function, and $\zeta_1, \ldots, \zeta_T$ is an i.i.d. sequence of random variables such that $F = \mathbb{E}[f(x, \zeta)]$ for some function f that is G-Lipschitz in x. For any minimizer $x_\star$, define $D = \|x_1 - x_\star\|$ and $\gamma = D/(G\sqrt{T})$. Then for any $\beta \in [0, 1]$, Schedule-Free SGD ensures:*

$$\mathbb{E}[F(x_T) - F(x_\star)] \leq \frac{DG}{\sqrt{T}}$$

Varying cosine-schedule length shows how Schedule-Free closely tracks the Pareto frontier of Loss v.s. training time.

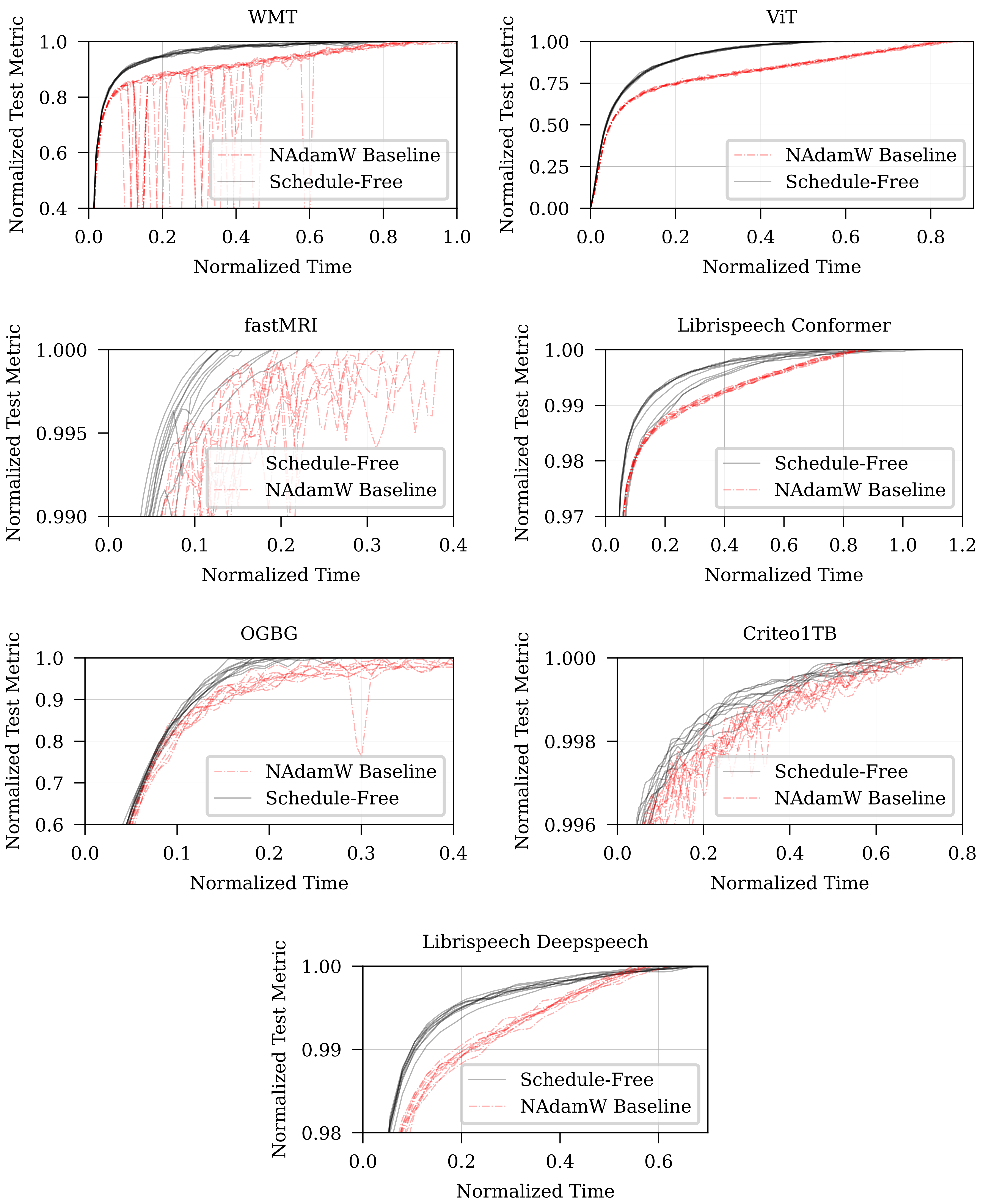


Schedule-Free SGD



Schedule-Free AdamW

Schedule-Free AdamW also **won** the MLCommons AlgoPerf 2024 Algorithmic Efficiency Challenge Self Tuning Track!



Schedule-Free runs have much smoother loss curves and faster convergence

Thank you!