When, Where, and Why to Average Weights?

Niccolò Ajroldi[1,2], Antonio Orvieto[1,2,3], Jonas Geiping[1,2,3]

[1] Ellis Institute Tübingen, [2] Max Planck Institute for Intelligent Systems, [3] Tübingen AI Center

INSTITUTE
TÜBINGEN

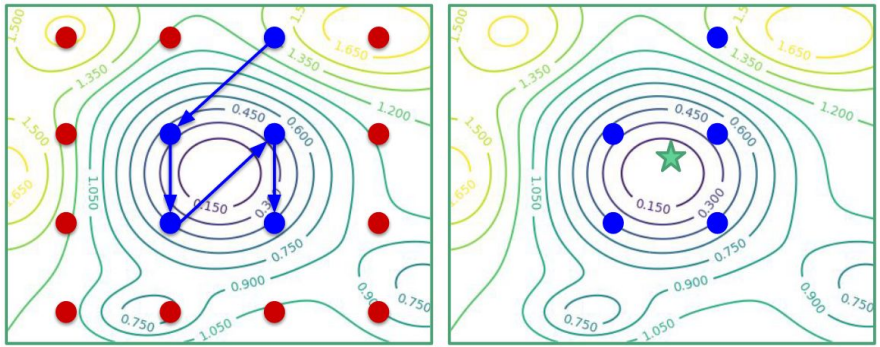
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

Weight Averaging

- Average model parameters from separate training runs → model soups
- Average weights from the **same training trajectory**
  - “Checkpoint Averaging”
  - LAtest Weight Averaging - LAWA
  - Exponential Moving Average - EMA

$$\bar{\theta}_t \leftarrow \frac{1}{N}\sum_{j=1}^{N}\theta_{t-j}$$

$$\bar{\theta}_t \leftarrow \beta\bar{\theta}_t + (1-\beta)\theta_t$$



Yang, G., et al. (2019). *SWALP: Stochastic weight averaging in low-precision training*. arXiv.

Motivation

Simple, principled approach (Polyak, 1990; Ruppert, 1988)

- Stochastic approximation (Polyak & Juditsky, 1992; Bach & Moulines, 2013)
- Convex optimization (Garrigos & Gower, 2023)
- Deep Learning:
  - + accelerate convergence (Athiwaratkun, 2018; Sanyal, 2023)
  - + improve generalization (Szegedy, 2016; Merity, 2017; Kaddour, 2022; Melis, 2023)
  - + robustness (Morales-Brotons, 2024)
  - + smooth loss landscape (Izmailov, 2019)
  - + proxy for LR decay (Sandler, 2023; Schaipp, 2025)

Contributions

Large-scale evaluation of Weight Averaging (WA) on **AlgoPerf**

1. Can it speed-up training?
2. Improve generalization?
3. Replace LR schedule?

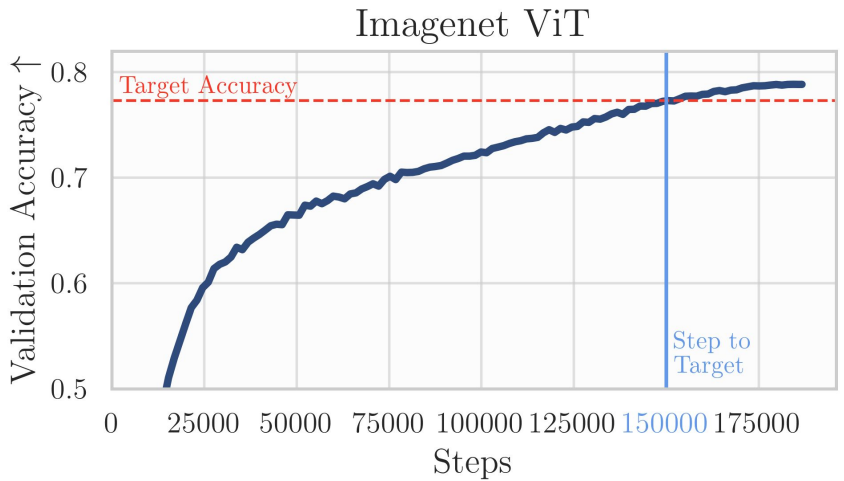
Enter AlgoPerf

A Benchmark for Optimization Algorithms

AlgoPerf: a Benchmark for Optimization Algorithms

- Open-source project [1,2]
- Only allowed to change the optimizer
- **8 workloads**: datasets & models
  - Fixed!
  - Each with a target validation score (loss / accuracy…)

Goal: be the **fastest** to the target!



[1] Dahl, G. E., et al. (2023). *Benchmarking neural network training algorithms*. *arXiv*. 2306.07179
[2] Kasimbeg, P., et al. (2025). Accelerating neural network training: An analysis of the AlgoPerf competition. In *The Thirteenth International Conference on Learning Representations*.

Benchmarking WA on AlgoPerf

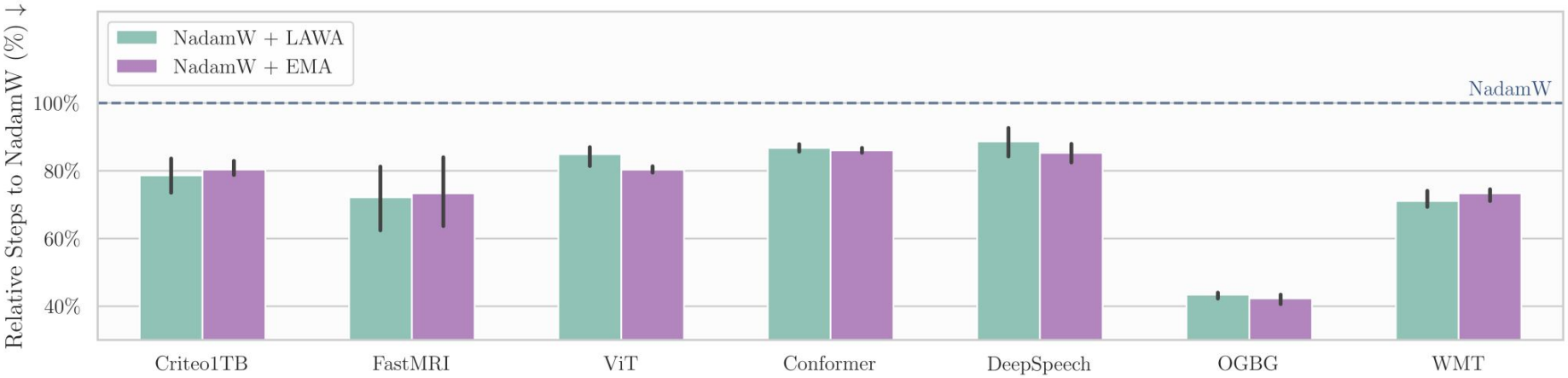
Strategy:

- Take the best optimization algorithm: **NadamW** [1,2] + best hyperparam
  - → strong baseline

- **NadamW +LAWA** or **+EMA**
  - → can we beat the baseline?
  - → can we reach the target sooner?

[1] Dozat, T. (2016). Incorporating Nesterov momentum into Adam.
[2] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization.

Efficiency Gains of Averaging



Use average to trigger **early stopping**:
→ reduce computational costs
→ **~15%** reduction of GPU hours on AlgoPerf

	NadamW	+LAWA	+EMA
GPU-Hours	636	548	540

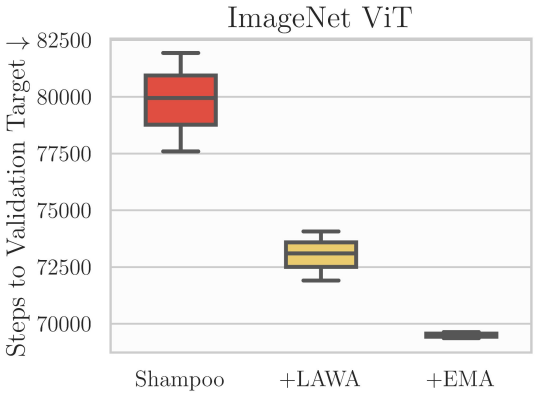
Changing Optimizer

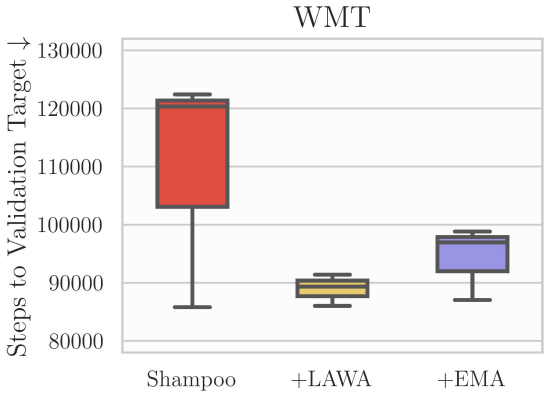
What if we change the baseline optimization algorithm?

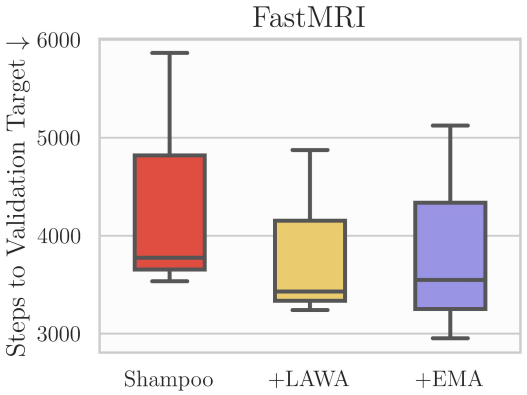
Is averaging just compensating for a slower optimizer?

Changing Optimizer

Can we accelerate Distributed Shampoo [1,2,3] (fastest optimizer on AlgoPerf)?





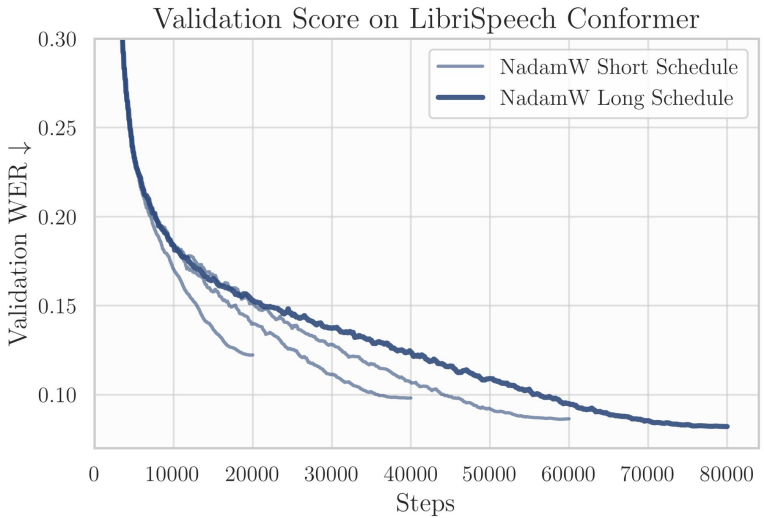


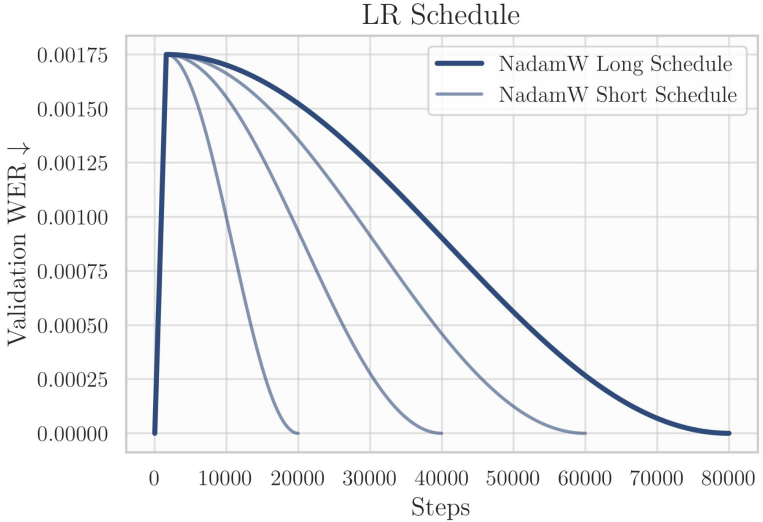
[1] Gupta, V., Koren, T., & Singer, Y. (2018). Shampoo: Preconditioned stochastic tensor optimization. *arXiv*, 1802.09568.
[2] Anil, R., Gupta, V., Koren, T., Regan, K., & Singer, Y. (2021). Scalable second-order optimization for deep learning. *arXiv*, 2002.09018.
[3] Shi, H.-J. M., Lee, T.-H., Iwasaki, S., et al (2023). A distributed data-parallel PyTorch implementation of the distributed Shampoo optimizer for training neural networks at scale. *arXiv*, 2309.06497.

Why is Averaging Faster?

Observation

- Performance *during training* strongly depends on the **LR schedule**
- A short schedule brings better performance *in the short term*

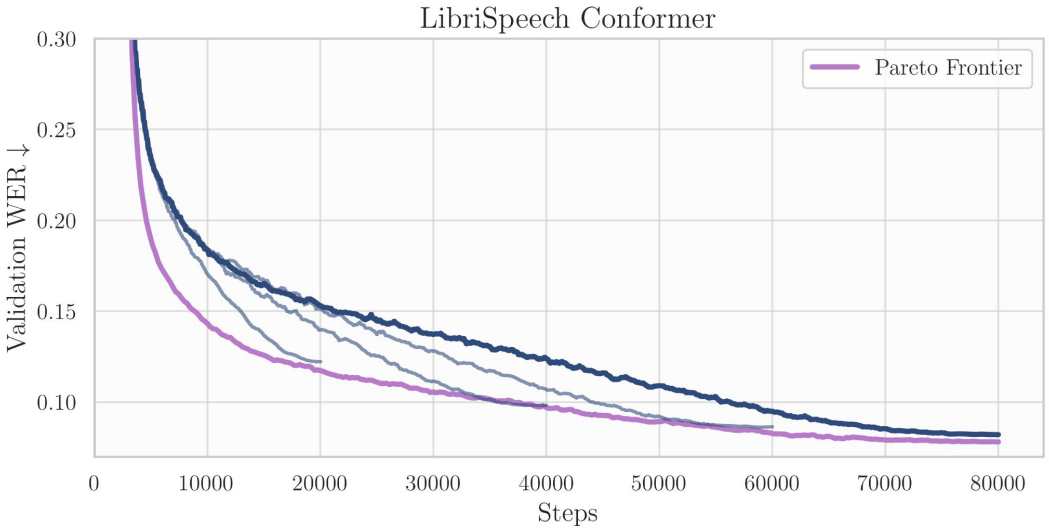




Question

How to achieve best performance *at any time* without prematurely decaying LR?

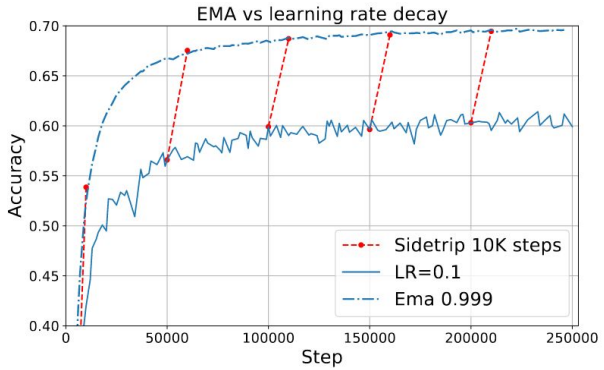
How to achieve **Pareto-optimality** of loss vs training time?



Averaging ≈ LR Decay

Theoretical **equivalence** between **Averaging** and **LR decay** for SGD [1]

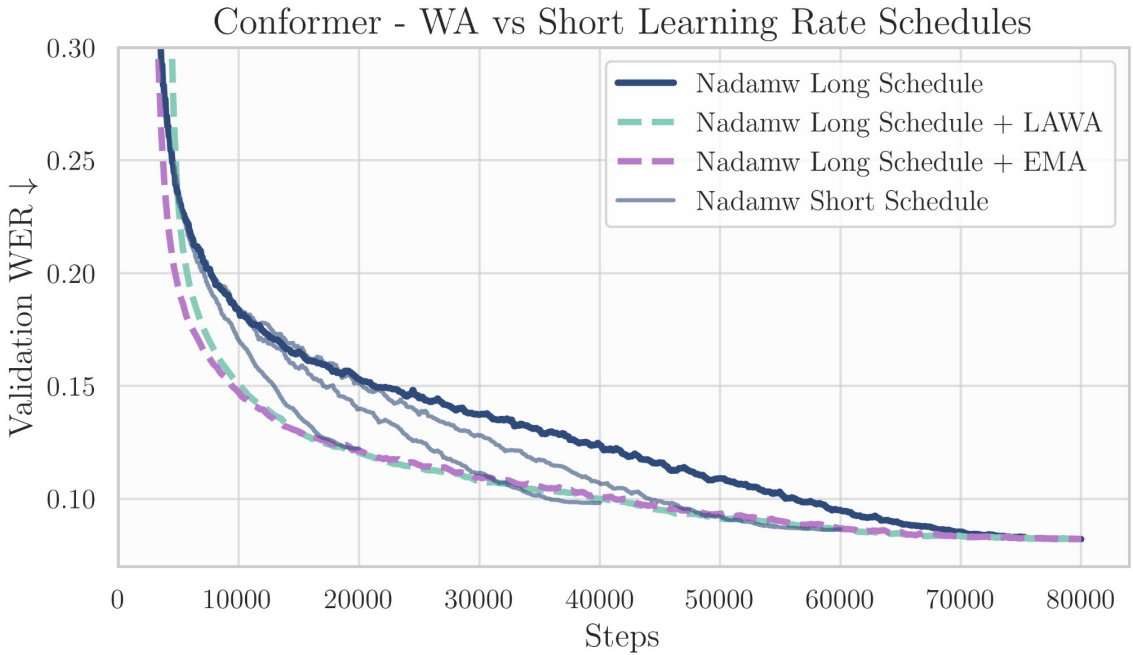
- Under a *Noisy Quadratic Loss* model [2], they derive LR schedules equivalent to averaging
- They verify empirical equivalence on SGD



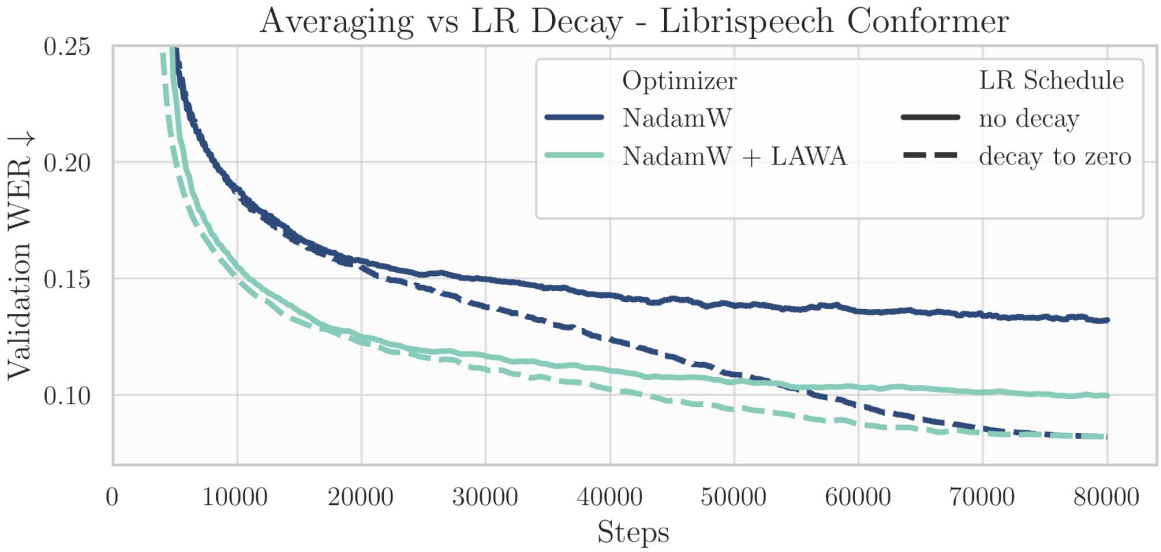
[1]: Sandler, M., Zhmoginov, A., Vladymyrov, M., & Miller, N. (2023). Training trajectories, mini-batch losses and the curious role of the learning rate. *arXiv*, 2301.02312.
[2]: Schaul, T., Zhang, S., & LeCun, Y. (2013). No more pesky learning rates. In *International Conference on Machine Learning* (pp. 343–351). PMLR.

Averaging brings us **closer** to the Pareto Frontier

Averaging reduces the gap between the current model and one with a cooled-down LR.



Averaging **cannot** fully **replace** LR decay



Replacing the LR schedule with either LAWA or EMA yields worse result.
A more sophisticated Averaging Scheme might be needed (Defazio et al. 2025).

Averaging **cannot** fully **replace** LR decay

We validate this across AlgoPerf workloads.

	Criteo1TB	FastMRI	ViT	Conformer	DeepSpeech	OGBG	WMT
	Loss ↓	SSIM ↑	Accuracy ↑	WRT ↓	WRT ↓	MAP ↑	BLEU ↑
NadamW + LR Decay	$\mathbf{0.1237}_{\pm 0.00003}$	$0.7238_{\pm 0.00029}$	$\mathbf{0.771}_{\pm 0.05905}$	$\mathbf{0.0848}_{\pm 0.01508}$	$\mathbf{0.1191}_{\pm 0.00233}$	$\mathbf{0.2818}_{\pm 0.00286}$	$\mathbf{30.7023}_{\pm 0.15644}$
NadamW + LAWA - No Decay	$0.1238_{\pm 0.00009}$	$\mathbf{0.7250}_{\pm 0.00175}$	$0.6475_{\pm 0.00013}$	$0.1071_{\pm 0.01188}$	$0.1371_{\pm 0.00097}$	$0.2748_{\pm 0.00312}$	$29.7648_{\pm 0.21885}$

When, Where, and Why to Average?

When and Where?

- When we have a target score in mind
- If annealing is not possible or not completed yet [1]
- To slightly improve final performance

Why?

- It’s free!
- Materialize a better model *without* cooling down the LR

[1]: Geiping, J., McLeish, S., Jain, N., et al. (2025). Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv*, 2502.05171.

A recent Large-Scale application

DeepSeek-V3 Technical Report

DeepSeek-AI

Exponential Moving Average in CPU. During training, we preserve the Exponential Moving Average (EMA) of the model parameters for early estimation of the model performance after learning rate decay. The EMA parameters are stored in CPU memory and are updated asynchronously after each training step. This method allows us to maintain EMA parameters without incurring additional memory or time overhead.

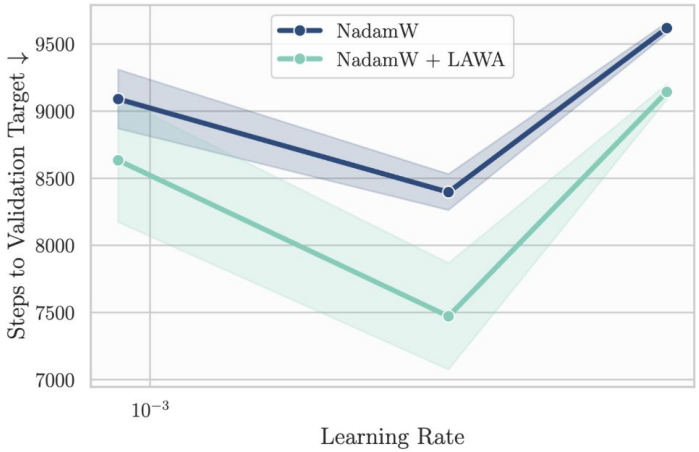
[1]: DeepSeek-AI, (2025). DeepSeek-V3 technical report. *arXiv*, 2412.19437.

Thank you!

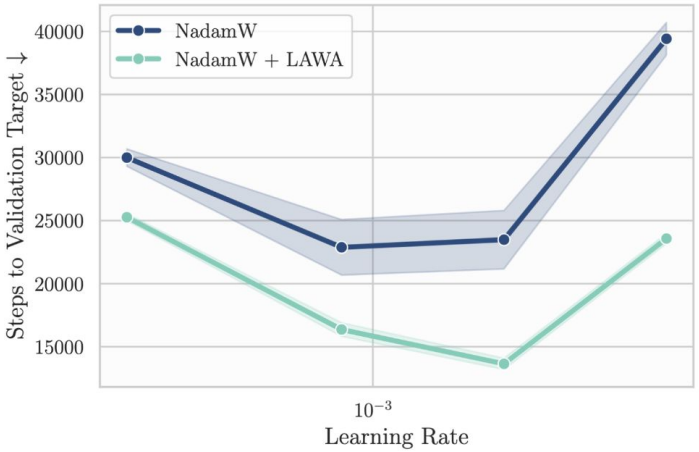
What if a strong baseline is not available?

Efficiency gains at **peak tuning** → do they hold in generic hyperparameter setting?

→ sweep LR of baseline → apply LAWA

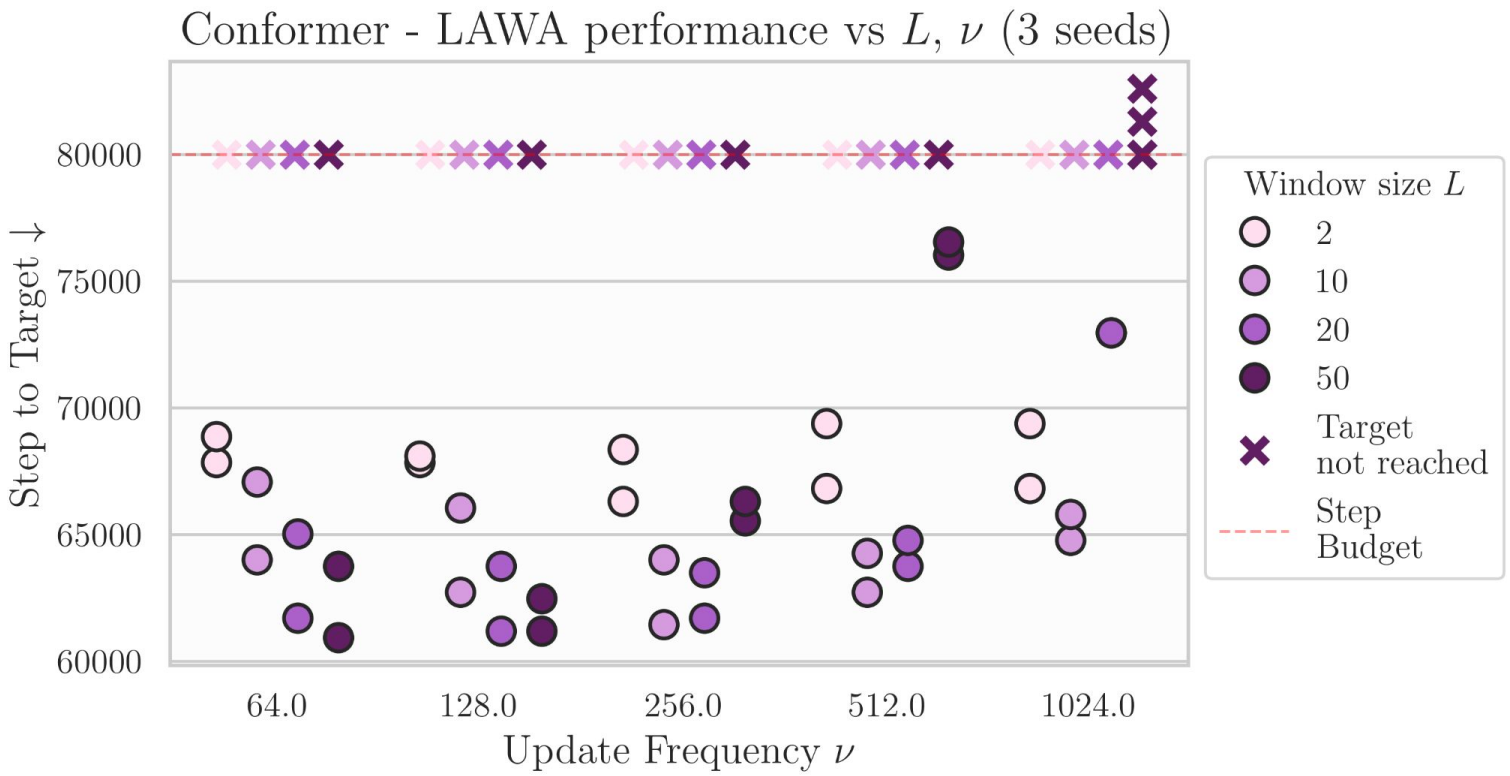


(a) Criteo1TB

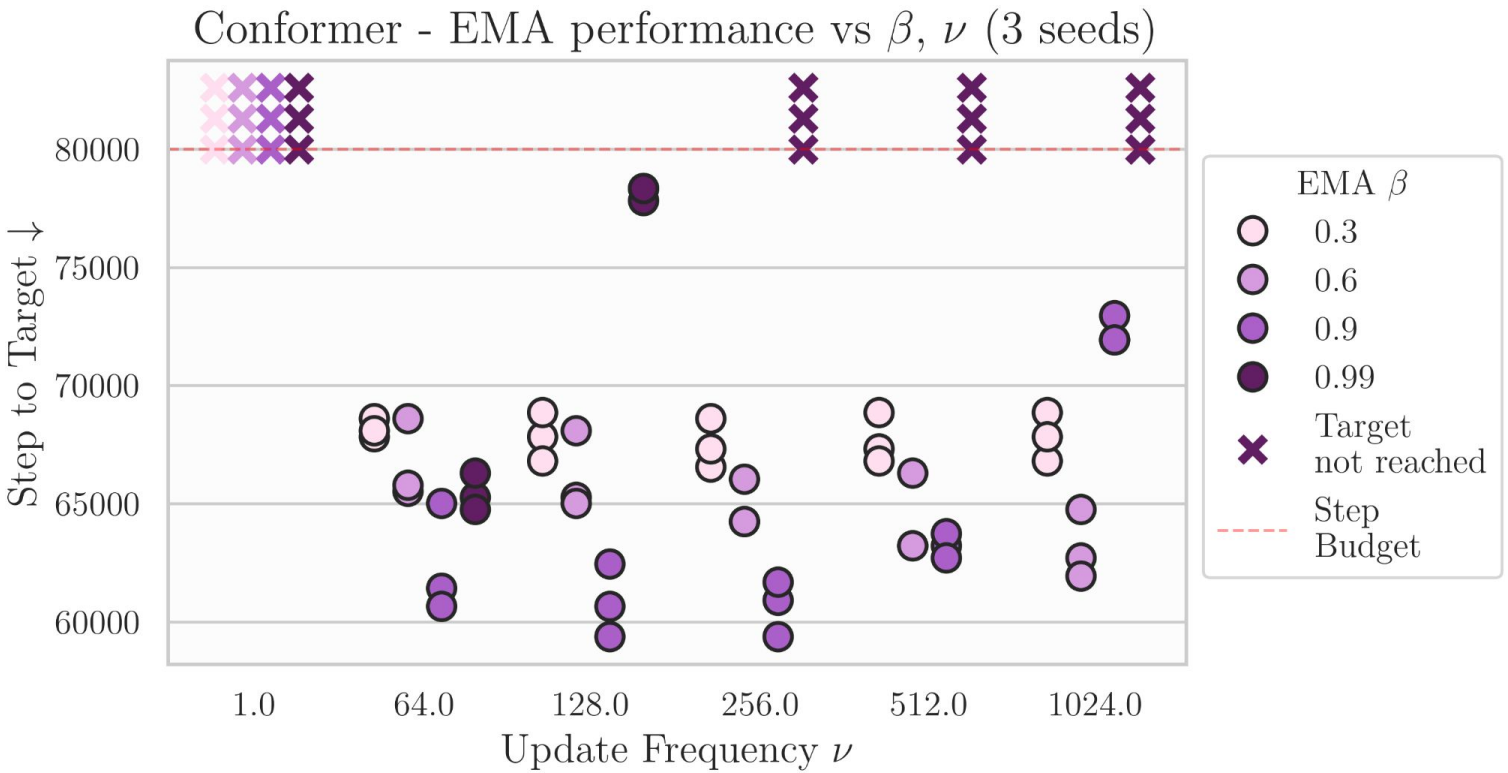


(b) OGBG

LAWA Hyperparameters



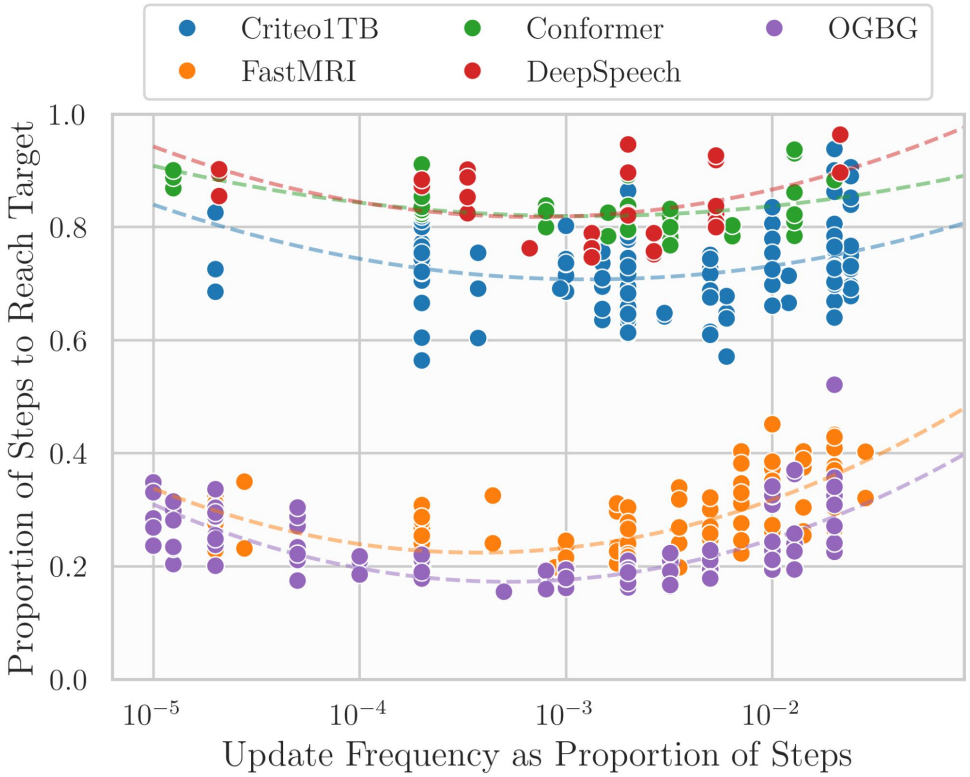
EMA hyperparameters



Optimal averaging horizon

LAWA, window size=10

How often to collect checkpoints?



Overhead of a naive EMA implementation on AlgoPerf-1