The Results of the Inaugural AlgoPerf Competition

How to train a neural net 30% faster

Frank Schneider

February 11, 2025

EBERHARD KARLS UNIVERSITÄT TÜBINGEN
MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

erc
Federal Ministry of Education and Research
imprs-is

The state of deep learning training methods

A confusingly crowded field of methods & hyperparameters

A huge number of optimizers...

Name	Ref.	Name	Ref.	Name	Ref.
AcceleGrad	[illegible]	C-ADAM	[illegible]	PAGE	[illegible]
ACClip	[illegible]	CADA	[illegible]	PAL	[illegible]
AdaAlter	[illegible]	Cool Momentum	[illegible]	PolyAdam	[illegible]
AdaBatch	[illegible]	CProp	[illegible]	Polyak	[illegible]
AdaBayes/AdaBayes-SS	[illegible]	Curveball	[illegible]	PowerSGD/PowerSGDM	[illegible]
AdaBelief	[illegible]	Dadam	[illegible]	Probabilistic Polyak	[illegible]
AdaBlock	[illegible]	DeepMemory	[illegible]	ProxLS	[illegible]
AdaBound	[illegible]	DGNOpt	[illegible]	PStorm	[illegible]
AdaComp	[illegible]	DiffGrad	[illegible]	QHAdam/QHM	[illegible]
Adadelta	[illegible]	EAdam	[illegible]	RAdam	[illegible]
Adafactor	[illegible]	EKFAC	[illegible]	Ranger	[illegible]
AdaFix	[illegible]	Eve	[illegible]	RangerLars	[illegible]
AdaFom	[illegible]	Expectigrad	[illegible]	RMSProp	[illegible]
AdaFTRL	[illegible]	FastAdaBelief	[illegible]	RMSterov	[illegible]
Adagrad	[illegible]	FRSGD	[illegible]	S-SGD	[illegible]
ADAHESSIAN	[illegible]	G-AdaGrad	[illegible]	SAdam	[illegible]
Adai	[illegible]	GADAM	[illegible]	Sadam/SAMSGrad	[illegible]
AdaLoss	[illegible]	Gadam	[illegible]	SALR	[illegible]
Adam	[illegible]	GOALS	[illegible]	SAM	[illegible]
Adam+	[illegible]	GOLS-I	[illegible]	SC-Adagrad/SC-RMSProp	[illegible]
AdamAL	[illegible]	Grad-Avg	[illegible]	SDProp	[illegible]
AdaMax	[illegible]	GRAPES	[illegible]	SGD	[illegible]
AdamBS	[illegible]	Gravilon	[illegible]	SGD-BB	[illegible]
AdamNC	[illegible]	Gravity	[illegible]	SGD-G2	[illegible]
AdaMod	[illegible]	HAdam	[illegible]	SGDEM	[illegible]
AdamP/SGDP	[illegible]	HyperAdam	[illegible]	SGDHess	[illegible]
AdamT	[illegible]	K-BFGS/K-BFGS(L)	[illegible]	SGDM	[illegible]
AdamW	[illegible]	KF-QN-CNN	[illegible]	SGDR	[illegible]
AdamX	[illegible]	KFAC	[illegible]	SHAdagrad	[illegible]
ADAS	[illegible]	KFLR/KFRA	[illegible]	Shampoo	[illegible]
AdaS	[illegible]	L4Adam/L4Momentum	[illegible]	SignAdam++	[illegible]
AdaScale	[illegible]	LAMB	[illegible]	SignSGD	[illegible]
AdaSGD	[illegible]	LaProp	[illegible]	SKQN/S4QN	[illegible]
AdaShift	[illegible]	LARS	[illegible]	SM3	[illegible]
AdaSqrt	[illegible]	LHOPT	[illegible]	SMG	[illegible]
Adathm	[illegible]	LookAhead	[illegible]	SNGM	[illegible]
AdaX/AdaX-W	[illegible]	M-SVAG	[illegible]	SoftAdam	[illegible]
AEGD	[illegible]	MADGRAD	[illegible]	SRSGD	[illegible]
ALI-G	[illegible]	MAS	[illegible]	Step-Tuned SGD	[illegible]
AMSBound	[illegible]	MEKA	[illegible]	SWATS	[illegible]
AMSGrad	[illegible]	MTAdam	[illegible]	SWNTS	[illegible]
AngularGrad	[illegible]	MVRC-1/MVRC-2	[illegible]	TAdam	[illegible]
ArmijoLS	[illegible]	Nadam	[illegible]	TEKFAC	[illegible]
ARSG	[illegible]	NAMSB/NAMSG	[illegible]	VAdam	[illegible]
ASAM	[illegible]	ND-Adam	[illegible]	VR-SGD	[illegible]
AutoLRS	[illegible]	Nero	[illegible]	vSGD-b/vSGD-g/vSGD-l	[illegible]
AvaGrad	[illegible]	Nesterov	[illegible]	vSGD-fd	[illegible]
BAdam	[illegible]	Noisy Adam/Noisy K-FAC	[illegible]	WNGrad	[illegible]
BGAdam	[illegible]	NosAdam	[illegible]	YellowFin	[illegible]
BPGrad	[illegible]	Novograd	[illegible]	Yogi	[illegible]
BRMSProp	[illegible]	NT-SGD	[illegible]		
BSGD	[illegible]	Padam	[illegible]		

from "Descending through a Crowded Valley"
(Schmidt, Schneider, Hennig; 2021)

The state of deep learning training methods

A confusingly crowded field of methods & hyperparameters

A huge number of optimizers...

from "Descending through a Crowded Valley" (Schmidt, Schneider, Hennig; 2021)

...and training tricks

- OneCycle scheduler, gradient checkpointing
- Genetic Algorithm for Hyperparameters
- Avoid batches that lead to NaN/inf losses
- One cycle, low fidelity training, SGD with restarts
- Proximal optimization for regularizers
- Line searches for the maximum learning rate
- Normalized updates
- Distributed Shampoo, Normformer, GLU
- Weight averaging
- FreezeOut
- A different epsilon value!
- Check hyperparameter performance over multiple seeds
- Lowering the learning rate!
- Normalizing data works better than batch or layer norm
- Mixed precision training
- Train with a small subset
- Cyclic and one cycle LR
- Label smoothing
- ...

from the NeurIPS "HITY Workshop" (Schneider et al.; 2022)

The AlgoPerf competition

A (very) short summary

ML Commons

The goal of the AlgoPerf: Training Algorithms benchmark & competition is to measure speed-ups in neural network training due to algorithmic improvements.
What are the best algorithms to train neural networks?

The AlgoPerf competition

A (very) short summary

ML Commons

The goal of the AlgoPerf: Training Algorithms benchmark & competition is to measure speed-ups in neural network training due to algorithmic improvements.
What are the best algorithms to train neural networks?

Why?

- There was no established protocol to benchmark deep learning training methods.
- There are lots of subtle pitfalls when evaluating training methods, such as hyperparameter tuning, training horizons, isolating the algorithm, etc. (see Dahl et al. (2023))
- Unreasonably hard task for researchers to perform a convincing, fair, and practically relevant comparison with strong baselines.

The Key Features of AlgoPerf

Training real-world deep learning workloads as fast as possible

Task	Dataset	Model	Loss	Metric	Validation Target	Maximum Runtime
Clickthrough rate prediction	**Criteo 1TB**	**DLRMsmall**	Cross Entropy	Cross Entropy	0.123 735	7703
MRI reconstruction	**fastMRI**	**U-Net**	L1	SSIM	0.7344	8859
Image classification	**ImageNet**	**ResNet-50**	Cross Entropy	Error Rate	0.225 69	63 008
		ViT	Cross Entropy	Error Rate	0.226 91	77 520
Speech recognition	**LibriSpeech**	**Conformer**	CTC	Word Error Rate	0.085 884	61 068
		DeepSpeech	CTC	Word Error Rate	0.119 936	55 506
Molecular property prediction	**OGBG**	**GNN**	Cross Entropy	mAP	0.280 98	18 477
Translation	**WMT**	**Transformer**	Cross Entropy	BLEU	30.8491	48 151

The Key Features of AlgoPerf

Isolating algorithmic improvements

Submissions can only modify the training **algorithm** and must leave all other aspects untouched

- `update_params`: Typically involves optimizers such as SGD, Adam, or custom methods.
- `init_optimizer_state`: Define a method's internal state, e.g. to define learning rate schedules.
- `data_selection`: How to construct batches of data.
- `get_batch_size`: Batch sizes for each workload, e.g. the largest batch size fitting in memory.
- (In the external tuning ruleset) `hyperparameter_search_space`: A *workload-agnostic* tuning space for a method's hyperparameters.

The Key Features of AlgoPerf

Two distinct rulesets simulating different use cases

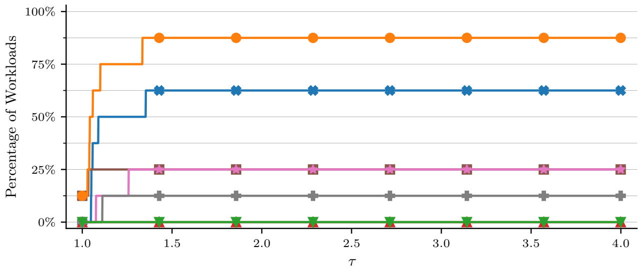
External Tuning Ruleset	**Self-Tuning Ruleset**
Parallel tuning across 5 tuning trials	No additional tuning, i.e. a single trial
Fastest trial counts for scoring	All computations are "on-the-clock"
Submissions must define a workload-agnostic search space	Any required workload-adaptation must be part of the method
Simulates training with parallel ressources, e.g. multiple devices	Simulates (sequential) training using a single device
Examples: Learning rate tuning using a log grid or a list of five hyperparameter configurations	Examples: Adam with default hyperparameters or inner-loop turing during the run

Repeat this process five times across different *studies* (with different random seeds) and take the median for a more robust final score.

The Key Features of AlgoPerf

Aggregate scoring using performance profiles

- **Workload scores** = median wall-clock runtimes to reach the target $t_{s,w}$.
- **Performance ratio** = workload score relative to the fastest workload score, i.e. $r_{s,w} = \frac{t_{s,w}}{\min_{s \in S} t_{s,w}}$.
- **Performance profile** = plot the fraction of workloads where a submission is less than τ away from the fastest submission, i.e. workloads where $r_{s,w} \leq \tau$.
- **Benchmark score** = integrate the performance profile, i.e. $B_s \in [0, 1]$.



Key Takeaways I

The key features of AlgoPerf

- **A competitive time-to-results benchmark**
  $\rightarrow$ Strong baselines.
- **Fixed hardware, workloads, and evaluation protocol**
  $\rightarrow$ Submissions need to innovate on the *training algorithms*.
- **Fully-specified algorithms that need to perform well across multiple workloads**
  $\rightarrow$ No cherry-picking, general-purpose methods with complete training recipes & properly account for hyperparameter tuning.

The Key Results of AlgoPerf

New SOTA in neural network training methods

External Tuning Ruleset

Submission	Authors	Institutions
Distributed Shampoo	Shi, Lee, et al.	Meta Platforms
Schedule Free AdamW	Defazio, Yang, Mishchenko	Meta AI, Samsung AI
Generalized Adam	Dahl, Medapati, et al.	Google
Cyclic LR	Ajroldi, Orvieto, Geiping	MPI-IS, ELLIS Tübingen
NadamP	Dahl, Medapati, et al.	Google
Baseline		
Amos	Tian	Google
CASPR Adaptive	Duvvuri, Dhillon, Hsieh	UT Austin, UCLA, Google
LAWA Queue	Ajroldi, Orvieto, Geiping	MPI-IS, ELLIS Tübingen
LAWA EMA	Ajroldi, Orvieto, Geiping	MPI-IS, ELLIS Tübingen
Schedule Free Prodigy	Defazio, Yang, Mishchenko	Meta AI, Samsung AI

Self-Tuning Ruleset

Submission	Authors	Institutions
Schedule Free AdamW	Defazio, Yang, Mishchenko	Meta AI, Samsung AI
Baseline		
NadamW Sequential	Dahl, Medapati, et al.	Google
SINV6 75	Moudgil	Mila, Concordia University
SINV6	Moudgil	Mila, Concordia University
AdamG	Pang	Michigan State University

The Key Results of AlgoPerf

New SOTA in neural network training methods

External Tuning Ruleset

Submission	Authors	Institutions	Score
Distributed Shampoo	Shi, Lee, et al.	Meta Platforms	0.7784
Schedule Free AdamW	Defazio, Yang, Mishchenko	Meta AI, Samsung AI	0.7077
Generalized Adam	Dahl, Medapati, et al.	Google	0.6383
Cyclic LR	Ajroldi, Orvieto, Geiping	MPI-IS, ELLIS Tübingen	0.6301
NadamP	Dahl, Medapati, et al.	Google	0.5909
Baseline			0.5707
Amos	Tian	Google	0.4918
CASPR Adaptive	Duvvuri, Dhillon, Hsieh	UT Austin, UCLA, Google	0.4722
LAWA Queue	Ajroldi, Orvieto, Geiping	MPI-IS, ELLIS Tübingen	0.3699
LAWA EMA	Ajroldi, Orvieto, Geiping	MPI-IS, ELLIS Tübingen	0.3384
Schedule Free Prodigy	Defazio, Yang, Mishchenko	Meta AI, Samsung AI	0

Self-Tuning Ruleset

Submission	Authors	Institutions	Score
Schedule Free AdamW	Defazio, Yang, Mishchenko	Meta AI, Samsung AI	0.8542
Baseline			0.8194
NadamW Sequential	Dahl, Medapati, et al.	Google	0.3308
SINV6 75	Moudgil	Mila, Concordia University	0.1420
SINV6	Moudgil	Mila, Concordia University	0.0903
AdamG	Pang	Michigan State University	0

The Key Results of AlgoPerf

More intuitive speedup numbers

- Distributed Shampoo is on average 28 % faster than the (external tuning) Baseline.
- Schedule Free AdamW is on average 8 % faster than the (self-tuning) Baseline.
- Comparisons across rulesets:
  - Distributed Shampoo is on average 24 % faster than (self-tuning) Schedule Free AdamW.
  - (self-tuning) Schedule Free AdamW 10 % faster than the (external tuning) Baseline.[1]

[1] Across the seven workloads both methods trained successfully.

The Key Results of AlgoPerf

Robustness is a major aspect of training methods

	Criteo 1TB	fastMRI	ResNet	ViT	Conformer	DeepSpeech	OGBG	WMT
Distributed Shampoo	**0.65**	0.15	inf	**0.43**	0.78	0.62	0.18	0.80
Schedule Free AdamW	0.67	0.13	inf	0.57	0.92	0.78	0.29‡	**0.33**
Generalized Adam	0.83	0.18	**0.97**	0.84	inf	0.68	0.31‡	0.63
Cyclic LR	0.67	0.25	inf	0.81	0.94	0.70	0.38‡	0.49
NadamP	0.80	0.22	inf	0.88	0.94	0.60	0.43‡	0.80
Baseline	0.94	0.23	inf	0.91	0.90	0.65	0.42‡	0.86
Amos	inf	0.33	inf	0.65	**0.71**	**0.57**	0.60*	0.68
CASPR Adaptive	NaN	**0.13**	inf	0.58	inf	0.75	**0.12**	0.67‡
LAWA Queue	inf	0.22	inf	0.66	inf	inf	0.25	0.56
LAWA EMA	0.69	0.29	inf	0.80	inf	inf	0.57*	0.89
Schedule Free Prodigy	NaN	0.21‡	inf	inf	inf	inf	0.61*	inf

Key Takeaways II

The key results of AlgoPerf

- **Significant improvements in neural net training**
  → The winners provide 28 % and 10 % faster training vs. their baseline.
  → Novel methods can improve over Adam (e.g. non-diagonal, second-order, ...).
- **Despite these improvements, there is ample potential left to explore**
  → No submission dominates across workloads.
  → For external tuning, 5 submissions were the fastest across the 8 workloads.
- **AlgoPerf can meaningfully evaluate training algorithms and identify practically useful methods**
  → The results are robust to may of our benchmarking decisions.
  → The benchmark must evolve and improve alongside the methods.

Summary & What This Means for You

Results of the Inaugural AlgoPerf Competition

- With Shampoo & Schedule-Free, we have two new exciting SOTA training algorithms.
  - **Try them out and let us know your results!**
- There is even more potential for future improvement thanks to the clear signal provided by AlgoPerf.
  - **Help us try out SOAP, Muon, AdEMAMix, ...!**
- The benchmark needs to evolve and improve with the submissions.
  - **Help us shape the next iteration of AlgoPerf!**

...and so many more!

Benchmark Code: `github.com/mlcommons/algorithmic-efficiency`

Blog Post: `mlcommons.org/2024/08/mlc-algoperf-benchmark-competition`

Results Paper: Accepted at ICLR 2025! `openreview.net/forum?id=CtM5xjRSfm`

Appendix

Performance profiles: External tuning ruleset

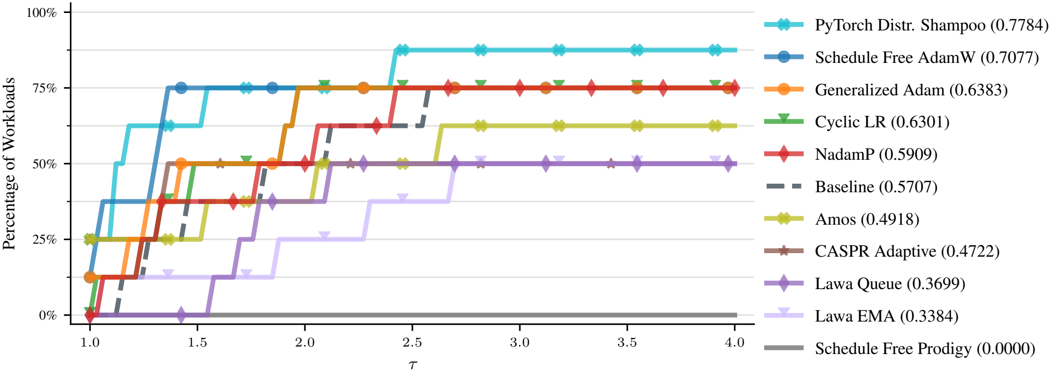


Figure 1: Performance profiles for the external tuning ruleset

Appendix

Performance profiles: Self-tuning ruleset

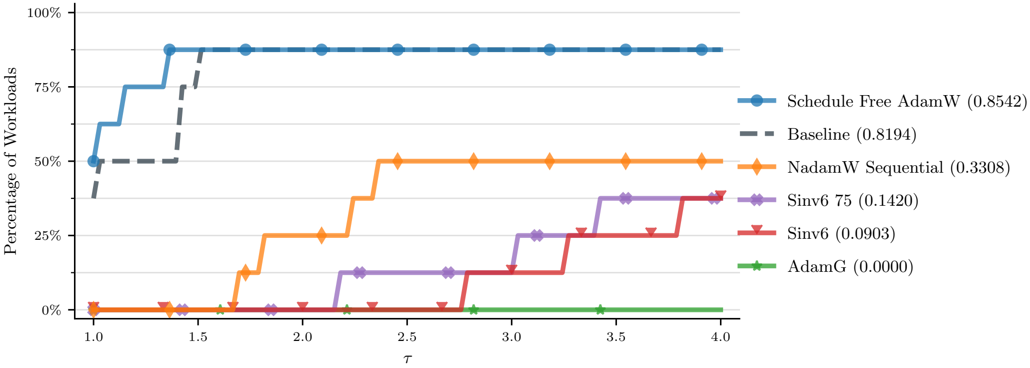


Figure 2: Performance profiles for the self-tuning ruleset

Appendix

Performance profiles: External tuning ruleset (without held-out workloads)

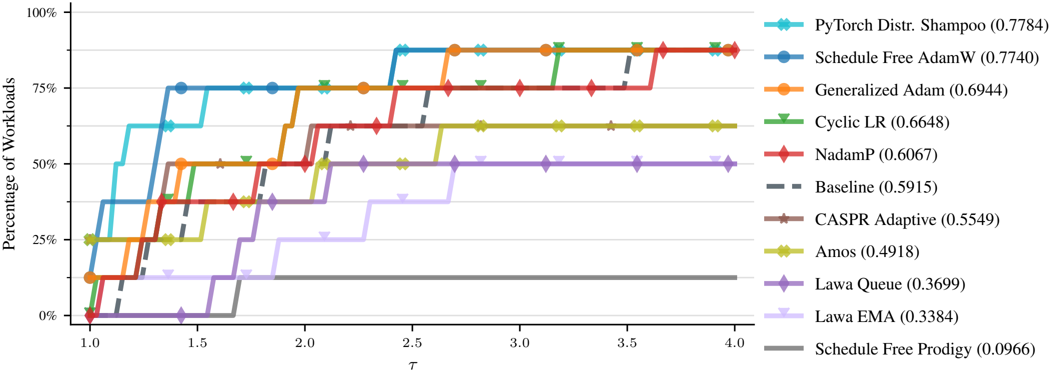


Figure 3: Performance profiles for the external tuning ruleset when ignoring held-out workloads.

Appendix

Performance profiles: Self-tuning ruleset (without held-out workloads)

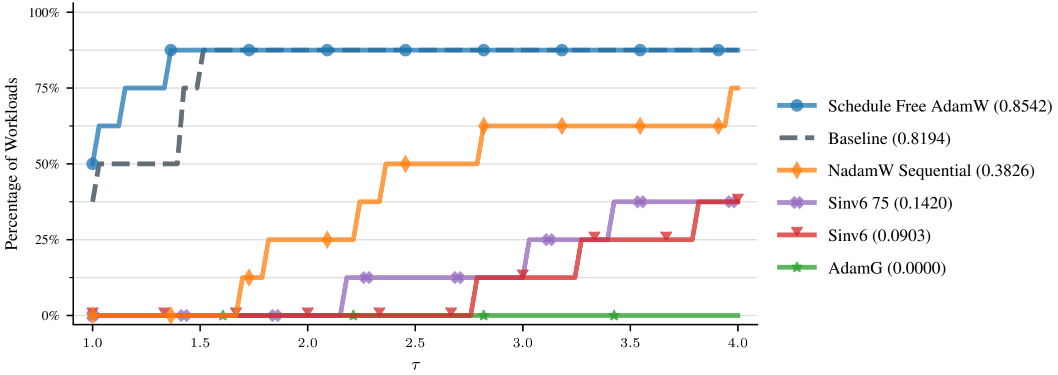


Figure 4: Performance profiles for the self-tuning ruleset when ignoring held-out workloads

Appendix

Performance profiles: External tuning ruleset (qualification set)

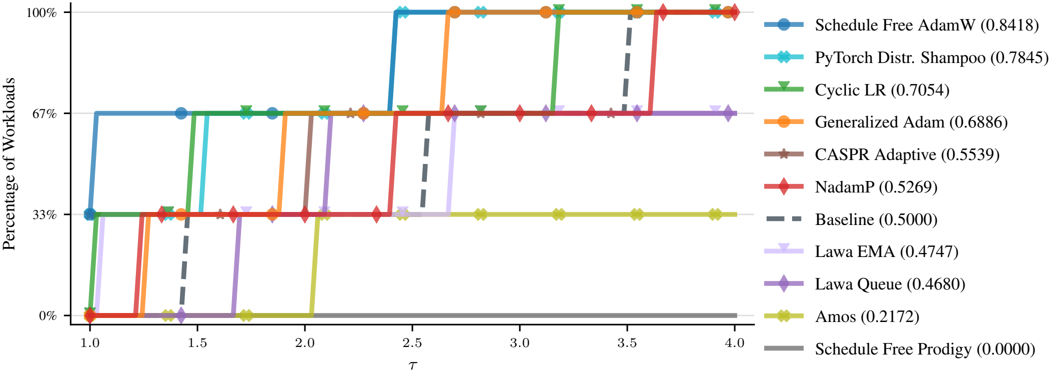


Figure 5: Performance profiles for the external tuning ruleset on the qualification set

Appendix

Performance profiles: Self-tuning ruleset (qualification set)

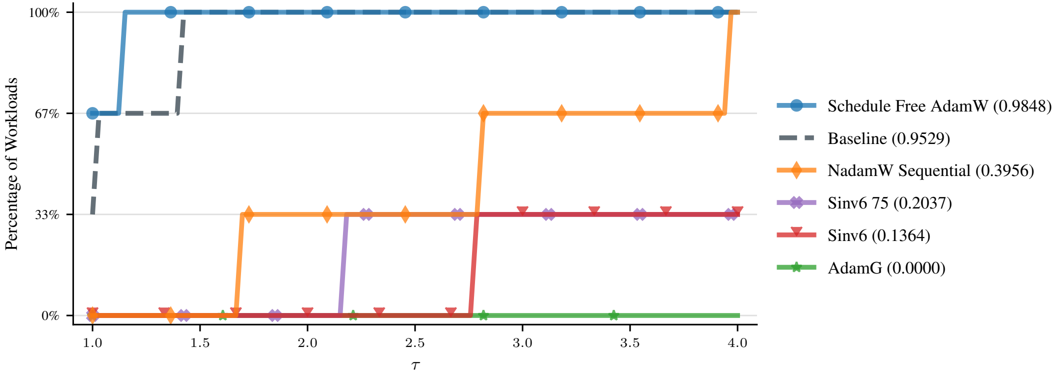


Figure 6: Performance profiles for the self-tuning ruleset on the qualification set

Appendix

Per-workload runtimes: Self-tuning ruleset

	Criteo 1TB	fastMRI	ResNet	ViT	Conformer	DeepSpeech	OGBG	WMT
AdamG	inf	inf	inf	inf	inf	inf	inf	inf
Baseline	0.75	0.22	inf	0.95	**0.94**	**0.65**	0.46	**0.84**
NadamW Sequential	2.96‡	0.27	inf	1.58	inf	1.45	0.55	2.36‡
Schedule Free AdamW	**0.75**	**0.15**	inf	**0.68**	0.97	0.88	**0.32**	0.94
Sinv6	NaN	0.49	inf	inf	inf	2.47	1.35*	2.32
Sinv6 75	NaN	0.45	inf	inf	inf	2.21	1.50*	1.82

Appendix

ResNet near misses: Distributed Shampoo

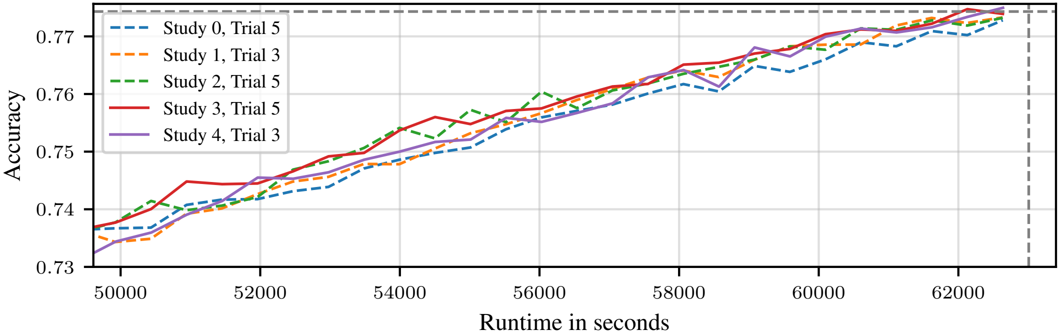


Figure 7: PyTorch Distributed Shampoo

Appendix

ResNet near misses: NadamP

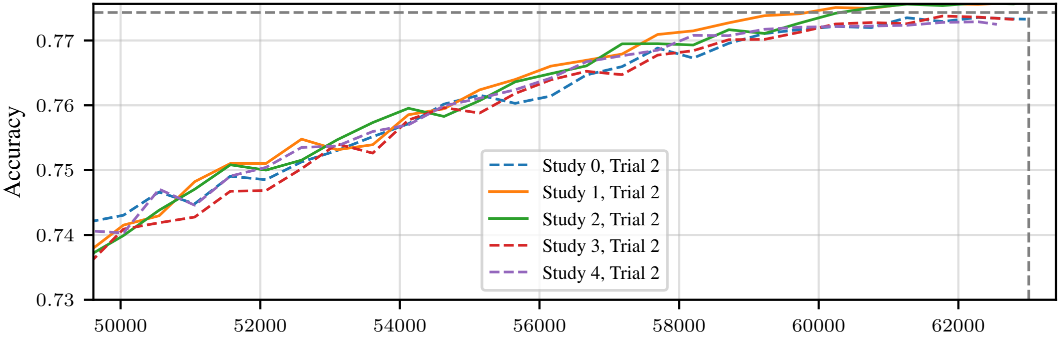


Figure 8: NadamP

Appendix

ResNet near misses: BASELINE

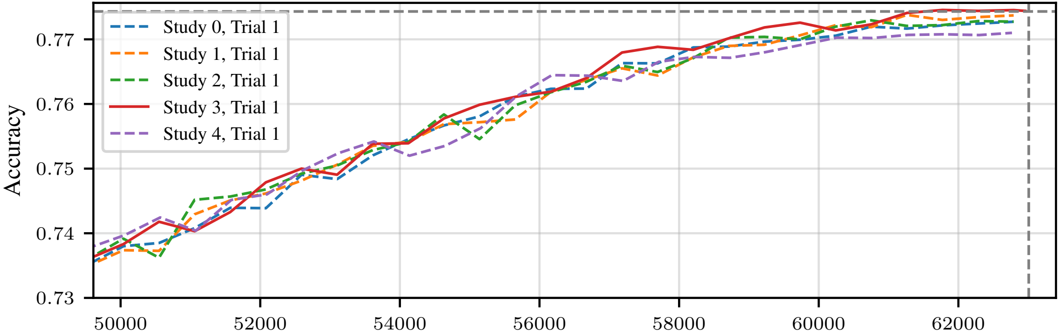


Figure 9: BASELINE

Appendix

Benchmark scores as a function of $\tau_{\max}$: External tuning

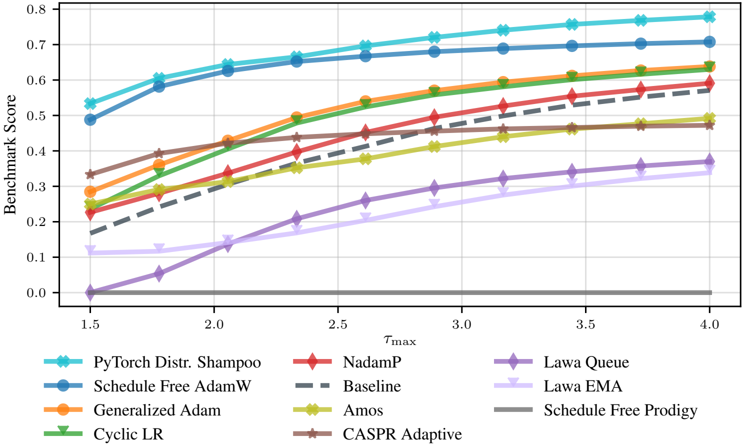


Figure 10: Benchmark scores as a function of $\tau_{\max}$ (external tuning).

Appendix

Benchmark scores as a function of τ_{max}: Self-tuning

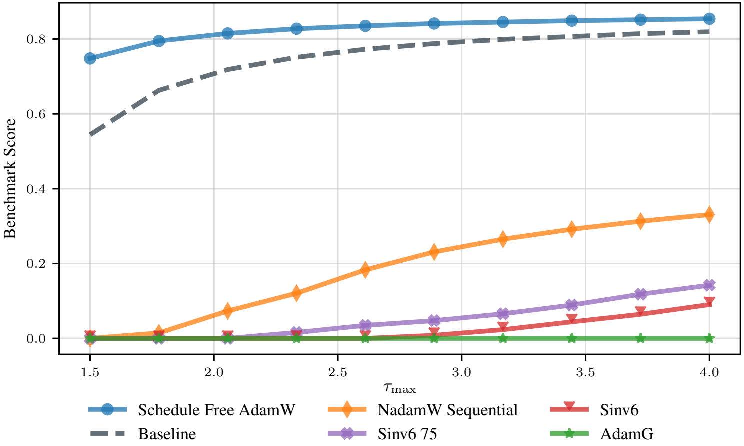


Figure 11: Benchmark scores as a function of τ_{max} (self-tuning).

Appendix

(a) **External tuning ruleset**

	Full		Criteo 1TB		FastMRI		ResNet		ViT		Conformer		DeepSpeech		OGBG		WMT	
	Score	Rank	S.	R.	S.	R.	S.	R.	S.	R.	S.	R.	S.	R.	S.	R.	S.	R.
PyTorch Distr. Shampoo	0.78	1	0.75	1	0.75	1	0.89	1	0.75	1	0.75	1	0.75	1	0.77	2	0.81	1
Schedule Free AdamW	0.71	2	0.67	2	0.67	2	0.81	2	0.68	2	0.68	3	0.68	2	0.81	1	0.67	2
Generalized Adam	0.64	3	0.60	3	0.61	4	0.59	6	0.63	3	0.73	2	0.59	3	0.73	3	0.63	3
Cyclic LR	0.63	4	0.58	4	0.62	3	0.72	3	0.62	4	0.59	4	0.59	4	0.72	4	0.60	4
NadamP	0.59	5	0.54	6	0.57	5	0.68	4	0.58	5	0.55	5	0.53	5	0.68	5	0.60	5
Baseline	0.57	6	0.53	8	0.55	6	0.65	5	0.56	6	0.52	7	0.52	6	0.65	6	0.58	6
Amos	0.49	7	0.56	5	0.50	7	0.56	7	0.44	7	0.42	9	0.42	8	0.56	7	0.47	8
CASPR Adaptive	0.47	8	0.54	7	0.40	8	0.54	8	0.41	8	0.54	6	0.41	9	0.40	8	0.54	7
LAWA Queue	0.37	9	0.42	9	0.32	9	0.42	9	0.31	9	0.42	8	0.42	7	0.33	10	0.31	10
LAWA EMA	0.34	10	0.25	10	0.31	10	0.39	10	0.28	10	0.39	10	0.39	10	0.39	9	0.32	9
Schedule Free Prodigy	0.00	11	0.00	11	0.00	11	0.00	11	0.00	11	0.00	11	0.00	11	0.00	11	0.00	11

Appendix

(a) **Self-tuning ruleset**

	Full		Criteo 1TB		fastMRI		ResNet		ViT		Conformer		DeepSpeech		OGBG		WMT	
	Score	Rank	S.	R.	S.	R.	S.	R.	S.	R.	S.	R.	S.	R.	S.	R.	S.	R.
Schedule Free AdamW	0.85	1	0.83	1	0.83	1	0.98	1	0.83	1	0.83	1	0.85	1	0.83	1	0.84	1
Baseline	0.82	2	0.79	2	0.82	2	0.94	2	0.81	2	0.79	2	0.79	2	0.81	2	0.79	2
NadamW Sequential	0.33	3	0.38	3	0.27	3	0.38	3	0.30	3	0.38	3	0.29	3	0.27	3	0.38	3
SinV6 75	0.14	4	0.16	4	0.12	4	0.16	4	0.16	4	0.16	4	0.13	4	0.16	4	0.08	4
SinV6	0.09	5	0.10	5	0.07	5	0.10	5	0.10	5	0.10	5	0.09	5	0.10	5	0.04	5
AdamG	0.00	6	0.00	6	0.00	6	0.00	6	0.00	6	0.00	6	0.00	6	0.00	6	0.00	6