

# Accelerating Neural Network Training

## An Analysis of the ALGO<sub>PERF</sub> Competition

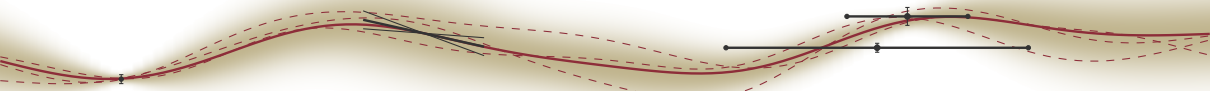
ICLR 2025

**Priya Kasimbeg, Frank Schneider**, R. Eschenhagen, J. Bae, C. Shama Sastry, M. Saroufim, B. Feng,  
L. Wright, E. Z. Yang, Z. Nado, S. Medapati, P. Hennig, M. Rabbat, G. E. Dahl

EBERHARD KARLS  
UNIVERSITÄT  
TÜBINGEN



UNIVERSITY OF  
TORONTO



# The ALGOPERF Competition

A (very) short motivation & summary

► **Neural network training is notoriously tricky.**

Choosing the **optimizer** (SGD, ADAM, NADAMW, K-FAC, LAMB, etc.), the **learning rate** ( $1e - 3$ ?  $3e - 4$ ? Tune it?), the learning rate **schedule** (cosine, cyclic warmup-stable-decay, etc.), etc.

# The ALGOPERF Competition

A (very) short motivation & summary

- **Neural network training is notoriously tricky.**

Choosing the **optimizer** (SGD, ADAM, NADAMW, K-FAC, LAMB, etc.?), the **learning rate** ( $1e - 3$ ?  $3e - 4$ ? Tune it?), the learning rate **schedule** (cosine, cyclic warmup-stable-decay, etc.?), etc.

- **The ALGOPERF benchmark.**

Measure speedups due to **algorithmic improvements**, but fix model, hardware, software, tuning protocols, etc.

# The ALGOERF Competition

A (very) short motivation & summary

- **Neural network training is notoriously tricky.**

Choosing the **optimizer** (SGD, ADAM, NADAMW, K-FAC, LAMB, etc.), the **learning rate** ( $1e - 3$ ?  $3e - 4$ ? Tune it?), the learning rate **schedule** (cosine, cyclic warmup-stable-decay, etc.), etc.

- **The ALGOERF benchmark.**

Measure speedups due to **algorithmic improvements**, but fix model, hardware, software, tuning protocols, etc.

- **Competition results.**

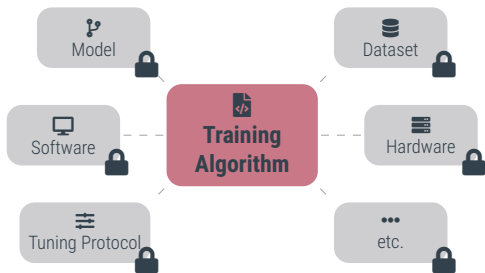
Significant training speedups by the winners **DISTRIBUTED SHAMPOO** (30%) & **SCHEDULE FREE ADAMW** (10%) over a well-tuned baseline.



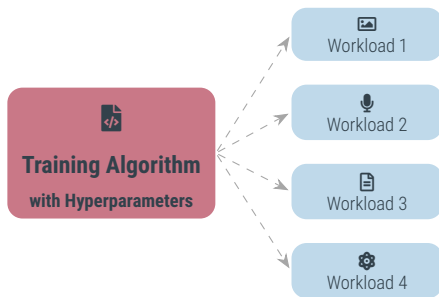
# The Key Features of ALGOPERF

Our two key principles

## Isolation



## Generic Methods



# The Key Features of ALGOPERF

Isolating algorithmic improvements

## Submissions can only modify these functions:

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

# The Key Features of ALGOPERF

Isolating algorithmic improvements

## Submissions can only modify these functions:

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

# The Key Features of ALGOPERF

Isolating algorithmic improvements

## Submissions can only modify these functions:

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

# The Key Features of ALGOPERF

Isolating algorithmic improvements

## Submissions can only modify these functions:

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

# The Key Features of ALGOPERF

Training real-world deep learning workloads as fast as possible

Task	Dataset	Model	Metric	Validation Target	Maximum Runtime
Clickthrough rate prediction	<b>CRITEO 1TB</b>	<b>DLRMSMALL</b>	Cross Entropy	0.123735	7703
MRI reconstruction	<b>FASTMRI</b>	<b>U-NET</b>	SSIM	0.7344	8859
Image classification	<b>IMAGENET</b>	<b>RESNET-50</b>	Error Rate	0.22569	63,008
		<b>ViT</b>	Error Rate	0.22691	77,520
Speech recognition	<b>LIBRISPEECH</b>	<b>CONFORMER</b>	Word Error Rate	0.085884	61,068
		<b>DEEPSPEECH</b>	Word Error Rate	0.119936	55,506
Molecular property prediction	<b>OGBG</b>	<b>GNN</b>	mAP	0.28098	18,477
Translation	<b>WMT</b>	<b>TRANSFORMER</b>	BLEU	30.8491	48,151

# The Key Features of ALGOPERF

Two distinct rulesets simulating different use cases

## External Tuning Ruleset

---

Parallel tuning across **5 tuning trials**

Fastest trial counts for scoring

Submissions must define a workload-agnostic search space

Simulates training with parallel resources, e.g. multiple devices

**Examples:** Learning rate tuning using a log grid or a list of five hyperparameter configurations

## Self-Tuning Ruleset

---

No additional tuning, i.e. a **single trial**

All computations are “on-the-clock”

Any required workload-adaptation must be part of the method

Simulates (sequential) training using a single device

**Examples:** ADAM with default hyperparameters or inner-loop tuning during the run

# The Key Features of ALGOPERF

Two distinct rulesets simulating different use cases

## External Tuning Ruleset

---

Parallel tuning across **5 tuning trials**

Fastest trial counts for scoring

Submissions must define a workload-agnostic search space

Simulates training with parallel resources, e.g. multiple devices

**Examples:** Learning rate tuning using a log grid or a list of five hyperparameter configurations

## Self-Tuning Ruleset

---

No additional tuning, i.e. a **single trial**

All computations are “on-the-clock”

Any required workload-adaptation must be part of the method

Simulates (sequential) training using a single device

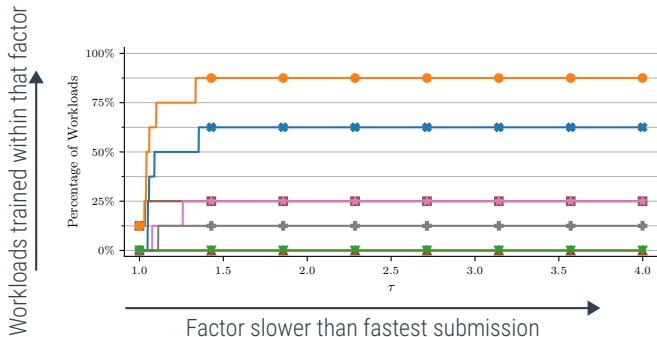
**Examples:** ADAM with default hyperparameters or inner-loop tuning during the run



# The Key Features of ALGOPERF







Aggregate scoring using performance profiles

- ▶ Plot **Performance Profiles** (right).
- ▶ Integrate Performance Profiles for a **Benchmark Score** *relative* to all submissions.
- ▶ Benchmark score is  $[0, 1]$  with 1 meaning fastest submission in each workload.



# The Results of the First ALGOPERF Competition

The Self-Tuning Ruleset

Submission	Team	Line	Score
SCHEDULE FREE ADAMW	Defazio, Yang, Mishchenko		0.8542
BASLINE			0.8194
NADAMW SEQUENTIAL	Dahl, Medapati, et al.		0.3308
SINV6 75	Moudgil		0.1420
SINV6	Moudgil		0.0903
ADAMG	Pang		0

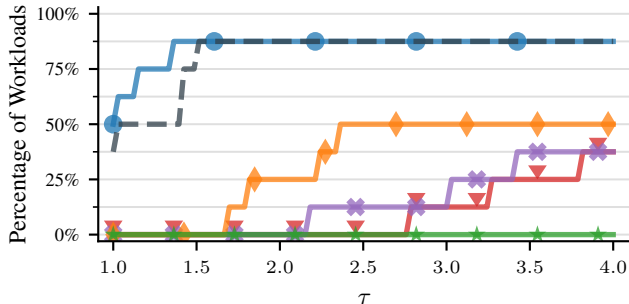
(a) Self-tuning leaderboard

# The Results of the First ALGOPERF Competition

The Self-Tuning Ruleset

Submission	Team	Line	Score
SCHEDULE FREE ADAMW	Defazio, Yang, Mishchenko	—●—	0.8542
BASELINE		- - -	0.8194
NADAMW SEQUENTIAL	Dahl, Medapati, et al.	—◆—	0.3308
SINV6 75	Moudgil	—✱—	0.1420
SINV6	Moudgil	—▼—	0.0903
ADAMG	Pang	—★—	0







(a) Self-tuning leaderboard



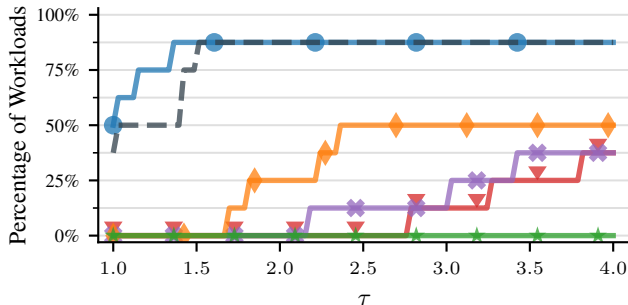
(b) Self-tuning performance profiles

# The Results of the First ALGOPERF Competition

The Self-Tuning Ruleset

Submission	Team	Line	Score
SCHEDULE FREE ADAMW	Defazio, Yang, Mishchenko		0.8542
BASELINE			0.8194
NADAMW SEQUENTIAL	Dahl, Medapati, et al.		0.3308
SINV6 75	Moudgil		0.1420
SINV6	Moudgil		0.0903
ADAMG	Pang		0

(a) Self-tuning leaderboard














(b) Self-tuning performance profiles

**SCHEDULE FREE ADAMW is on average  $\approx 10\%$  faster than the self-tuning BASELINE.**

# The Results of the First ALGOPERF Competition












The External-Tuning Ruleset

Submission	Team	Line	Score
DISTRIBUTED SHAMPOO	Shi, et al.		0.7784
SCHEDULE FREE ADAMW	Defazio, et al.		0.7077
GENERALIZED ADAM	Dahl, et al.		0.6383
CYCLIC LR	Ajroldi, et al.		0.6301
NADAMP	Dahl, et al.		0.5909
BASILINE			0.5707
AMOS	Tian		0.4918
CASPR	Duvvuri, et al.		0.4722
LAWA QUEUE	Ajroldi, et al.		0.3699
LAWA EMA	Ajroldi, et al.		0.3384
S.F. PRODIGY	Defazio, et al.		0

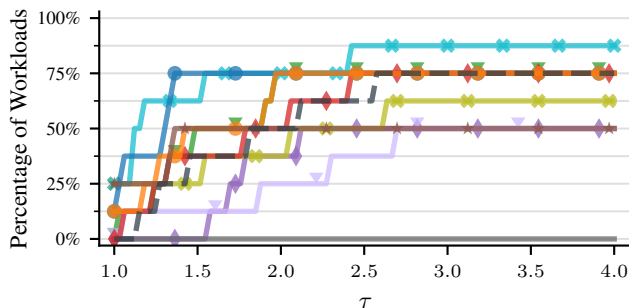
(a) External tuning leaderboard

# The Results of the First ALGOPERF Competition

The External-Tuning Ruleset

Submission	Team	Line	Score
DISTRIBUTED SHAMPOO	Shi, et al.		0.7784
SCHEDULE FREE ADAMW	Defazio, et al.		0.7077
GENERALIZED ADAM	Dahl, et al.		0.6383
CYCLIC LR	Ajroldi, et al.		0.6301
NADAMP	Dahl, et al.		0.5909
<b>BASELINE</b>			0.5707
AMOS	Tian		0.4918
CASPR	Duvvuri, et al.		0.4722
LAWA QUEUE	Ajroldi, et al.		0.3699
LAWA EMA	Ajroldi, et al.		0.3384
S.F. PRODIGY	Defazio, et al.		0












(a) External tuning leaderboard



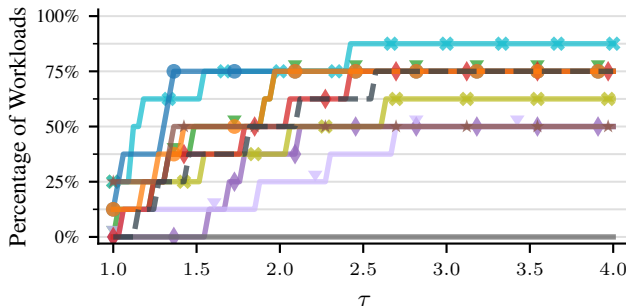
(b) External tuning performance profiles

# The Results of the First ALGOPERF Competition

The External-Tuning Ruleset

Submission	Team	Line	Score
DISTRIBUTED SHAMPOO	Shi, et al.		0.7784
SCHEDULE FREE ADAMW	Defazio, et al.		0.7077
GENERALIZED ADAM	Dahl, et al.		0.6383
CYCLIC LR	Ajroldi, et al.		0.6301
NADAMP	Dahl, et al.		0.5909
BASELINE			0.5707
AMOS	Tian		0.4918
CASPR	Duvvuri, et al.		0.4722
LAWA QUEUE	Ajroldi, et al.		0.3699
LAWA EMA	Ajroldi, et al.		0.3384
S.F. PRODIGY	Defazio, et al.		0

(a) External tuning leaderboard



(b) External tuning performance profiles

**DISTRIBUTED SHAMPOO is on average  $\approx 30\%$  faster than the external tuning BASELINE.**

# The Results of the First ALGOERF Competition

Robustness is a major aspect of training methods

	CRITEO 1TB	FASTMRI	RESNET	VIT	CONFORMER	DEEPSPEECH	OGBG	WMT
DISTRIBUTED SHAMPOO	<b>0.65</b>	0.15	inf	<b>0.43</b>	0.78	0.62	0.18	0.80
SCHEDULE FREE ADAMW	0.67	0.13	inf	0.57	0.92	0.78	0.29 <sup>‡</sup>	<b>0.33</b>
GENERALIZED ADAM	0.83	0.18	<b>0.97</b>	0.84	inf	0.68	0.31 <sup>‡</sup>	0.63
CYCLIC LR	0.67	0.25	inf	0.81	0.94	0.70	0.38 <sup>‡</sup>	0.49
NADAMP	0.80	0.22	inf	0.88	0.94	0.60	0.43 <sup>‡</sup>	0.80
BASLINE	0.94	0.23	inf	0.91	0.90	0.65	0.42 <sup>‡</sup>	0.86
AMOS	inf	0.33	inf	0.65	<b>0.71</b>	<b>0.57</b>	0.60 <sup>*</sup>	0.68
CASPR ADAPTIVE	NaN	<b>0.13</b>	inf	0.58	inf	0.75	<b>0.12</b>	0.67 <sup>‡</sup>
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 <sup>*</sup>	0.89
SCHEDULE FREE PRODIGY	NaN	0.21 <sup>‡</sup>	inf	inf	inf	inf	0.61 <sup>*</sup>	inf

**No single submission dominates across all workloads.**



# Summary

Results of the Inaugural ALGOPERF Competition

- ▶ SHAMPOO & SCHEDULE-FREE are new SOTA training algorithms.
  - ▶ **30% and 10% faster training vs. the baseline!**
- ▶ Even more potential for future improvements.
  - ▶ **Help us try out SOAP, MUON, ADEMAMix, ...!**
- ▶ The benchmark needs to evolve alongside the submissions.
  - ▶ **Help us shape the next iteration of ALGOPERF!**



...and so many more!

**Paper:** [openreview.net/forum?id=CtM5xjRSfm](https://openreview.net/forum?id=CtM5xjRSfm)

**Blog Post:** [mlcommons.org/2024/08/mlc-algoperf-benchmark-competition](https://mlcommons.org/2024/08/mlc-algoperf-benchmark-competition)

**Benchmark Code:** [github.com/mlcommons/algorithmic-efficiency](https://github.com/mlcommons/algorithmic-efficiency)

**Leaderboard Updates:** Bluesky or X (@algoperf)