

# OPTIMIZER BENCHMARKING NEEDS TO ACCOUNT FOR HYPERPARAMETER TUNING

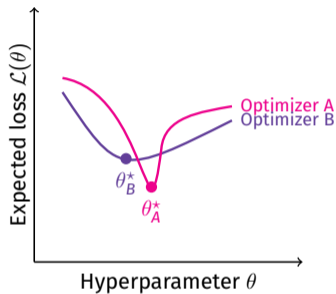
PRABHU TEJA S<sup>\*1,2</sup> FLORIAN MAI<sup>\*1,2</sup> THIJS VOGELS<sup>2</sup>  
MARTIN JAGGI<sup>2</sup> FRANÇOIS FLEURET<sup>1,2</sup>

<sup>1</sup>IDIAP RESEARCH INSTITUTE, <sup>2</sup>EPFL, SWITZERLAND

\*EQUAL CONTRIBUTION

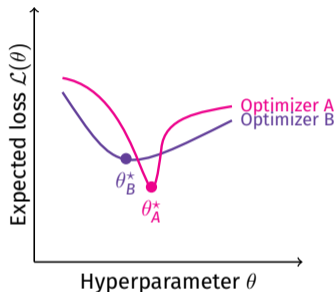


# THE PROBLEM OF OPTIMIZER EVALUATION



**Figure:** Two optimizers A & B with hyperparameter  $\theta$ . Which one do we prefer in practice?

# THE PROBLEM OF OPTIMIZER EVALUATION

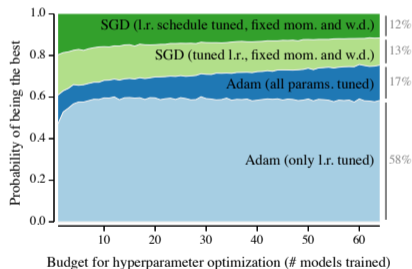


**Figure:** Two optimizers A & B with hyperparameter  $\theta$ . Which one do we prefer in practice?

1. The absolute performance of the optimizer  $\rightarrow \mathcal{L}(\theta_A^*), \mathcal{L}(\theta_B^*)$
2. Difficulty of finding good hyperparameter configuration  $\approx \theta_A^*, \theta_B^*$ .

## THE PROBLEM OF OPTIMIZER EVALUATION: SGD VS ADAM

1. SGD often achieves better peak performance than Adam in previous literature
2. We take into cognizance the cost of automatic Hyperparameter Optimization (HPO), and find:



Our method eliminates human biases arising from manual hyperparameter tuning.

## REVISITING THE NOTION OF AN OPTIMIZER

### Definition

An optimizer is a pair  $\mathcal{M} = (\mathcal{U}_\Theta, p_\Theta)$ , which applies its update rule  $\mathcal{U}(S_t; \Theta)$  at each step  $t$  depending on its current state  $S_t$ .

Its hyperparameters  $\Theta = (\theta_1, \dots, \theta_N)$  have a prior probability distribution  $p_\Theta : (\Theta \rightarrow \mathbb{R})$  defined.

$p_\Theta$  should be specified by the optimizer designer,  
e.g., Adam's  $\epsilon > 0$  and close to 0  $\implies \epsilon \sim \text{Log-uniform}(-8, 0)$

# HPO AWARE OPTIMIZER BENCHMARKING

---

**Algorithm 1** Benchmark with ‘expected quality at budget’

---

**input:** optimizer  $O$ , cross-task hyperparameter prior  $p_{\Theta}$ , task  $T$ , tuning budget  $B$

**Initialize**  $list \leftarrow []$ .

**for**  $R$  repetitions **do**

    Perform random search with budget  $B$ :

- $S \leftarrow$  sample  $B$  elements from  $p_{\Theta}$ .
- $list \leftarrow [\text{BEST}(S), \dots list]$ .

**return**  $\text{MEAN}(list)$ ,  $\text{VAR}(list)$ , or other statistics

---

## CALIBRATED TASK INDEPENDENT PRIORS $p_{\Theta}$

| Optimizer | Tunable parameters  | Cross-task prior |
|-----------|---|------------------|
| SGD       | Learning rate<br>Momentum<br>Weight decay<br>Poly decay ( $p$ ) | ??               |
| Adagrad   | Learning rate   |                  |
| Adam      | Learning rate<br>$\beta_1, \beta_2$<br>$\epsilon$               |                  |

## CALIBRATED TASK INDEPENDENT PRIORS $p_{\Theta}$

| Optimizer | Tunable parameters  | Cross-task prior |
|-----------|---|------------------|
| SGD       | Learning rate<br>Momentum<br>Weight decay<br>Poly decay ( $p$ ) | ??               |
| Adagrad   | Learning rate   |                  |
| Adam      | Learning rate<br>$\beta_1, \beta_2$<br>$\epsilon$               |                  |

- Sample a large number of points and their performance from a large range of admissible values
- Maximum Likelihood Estimate (MLE) of the prior's parameters using the top 20% performant values from the previous step.



## CALIBRATED TASK INDEPENDENT PRIORS $p_{\Theta}$

| Optimizer | Tunable parameters | Cross-task prior         |
|-----------|--------------------|--------------------------|
| SGD       | Learning rate      | Log-normal(-2.09, 1.312) |
|           | Momentum           | $\mathcal{U}[0, 1]$      |
|           | Weight decay       | Log-uniform(-5, -1)      |
|           | Poly decay ( $p$ ) | $\mathcal{U}[0.5, 5]$    |
| Adagrad   | Learning rate      | Log-normal(-2.004, 1.20) |
| Adam      | Learning rate      | Log-normal(-2.69, 1.42)  |
|           | $\beta_1, \beta_2$ | 1- Log-uniform(-5, -1)   |
|           | $\epsilon$         | Log-uniform(-8, 0)       |

- Sample a large number of points and their performance from a large range of admissible values
- Maximum Likelihood Estimate (MLE) of the prior's parameters using the top 20% performant values from the previous step.

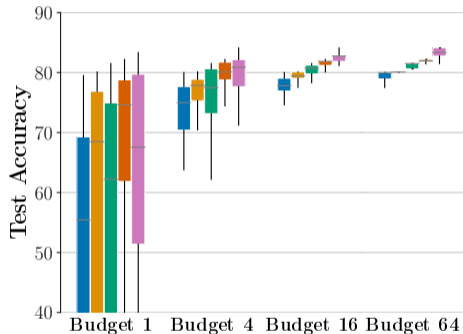
## THE IMPORTANCE OF RECIPES

| Optimizer label                   | Tunable parameters   |
|-----------------------------------|--|
| SGD-M <sup>C</sup> W <sup>C</sup> | SGD( $\gamma, \mu=0.9, \lambda=10^{-5}$ )                      |
| SGD-M <sup>C</sup> D              | SGD( $\gamma, \mu=0.9, \lambda=10^{-5}$ ) + Poly Decay( $p$ )  |
| SGD-MW                            | SGD( $\gamma, \mu, \lambda$ )                                  |
| Adam-LR                           | Adam( $\gamma, \beta_1=0.9, \beta_2=0.999, \epsilon=10^{-8}$ ) |
| Adam                              | Adam( $\gamma, \beta_1, \beta_2, \epsilon$ )                   |

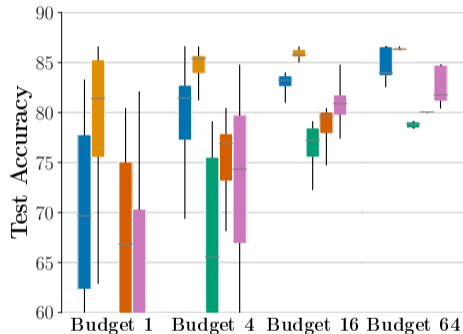
SGD( $\gamma, \mu, \lambda$ ) is SGD with  $\gamma$  learning rate,  $\mu$  momentum,  $\lambda$  weight decay coefficient.  
Adagrad( $\gamma$ ) is Adagrad with  $\gamma$  learning rate, Adam( $\gamma, \beta_1, \beta_2, \epsilon$ ) is Adam with learning rate  $\gamma$ , momentum parameters  $\beta_1, \beta_2$ , and normalization parameter  $\epsilon$

# PERFORMANCE AT A BUDGET

## CIFAR 10

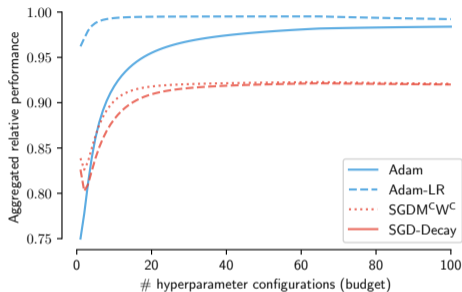


## IMDb LSTM



Performance of **Adam-LR**, **Adam**, **SGD-M<sup>C</sup>W<sup>C</sup>**, **SGD-MW**, **SGD-M<sup>C</sup>D** at various hyperparameter search budgets

## SUMMARIZING OUR FINDINGS



Summary statistics:

$$S(o, k) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \frac{o(k, p)}{\max_{o' \in \mathcal{O}} o'(k, p)},$$

where  $o(k, p)$  denotes the expected performance of optimizer  $o \in \mathcal{O}$  on test problem  $p \in \mathcal{P}$  after  $k$  iterations of hyperparameter search.

## OUR FINDINGS

1. Support the hypothesis that adaptive gradient methods are easier to tune than non-adaptive methods
  - The substantial value of the adaptive gradient methods, specifically Adam, is its amenability to hyperparameter search.

## OUR FINDINGS

1. Support the hypothesis that adaptive gradient methods are easier to tune than non-adaptive methods
  - The substantial value of the adaptive gradient methods, specifically Adam, is its amenability to hyperparameter search.
2. Tuning optimizers' hyperparameters apart from the learning rate becomes more useful as the available tuning budget goes up.
  - Even with relatively large tuning budget, tuning only the learning rate of Adam is the safer choice, as it achieves good results with high probability.

*THANK YOU*