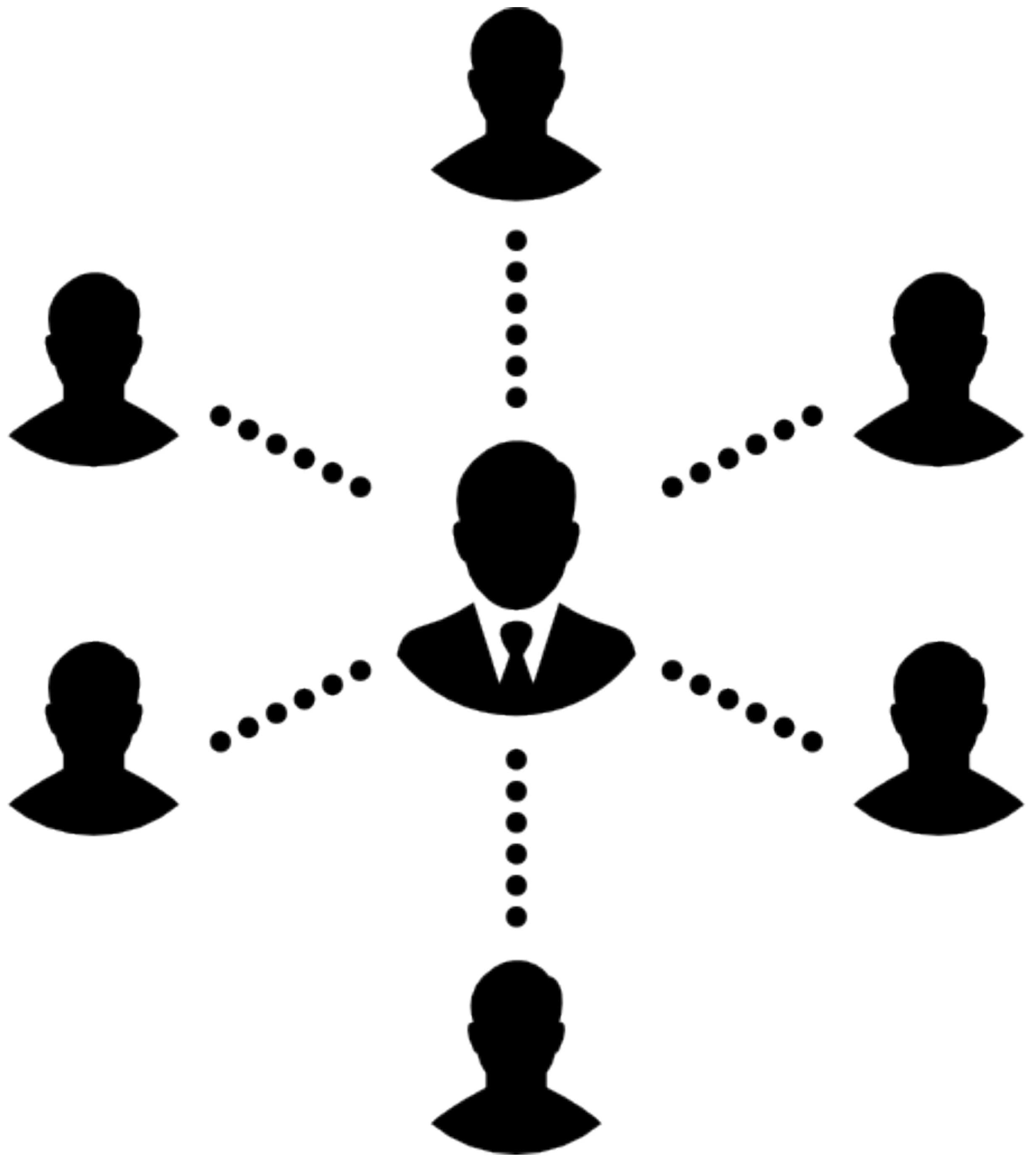
The Role of Adaptive Optimizers for Honest Private Hyperparameter Selection

Shubhankar Mohapatra[+], Sajin Sasy[+], Xi He[+], Gautam Kamath[+], Om Thakkar[*]
University of Waterloo[+], Google[*]

ML models use private data



Models may leak unintended information

<u>Re-identification (NS'06)</u>

Identifying individuals by extrapolating to publicly available dataset.

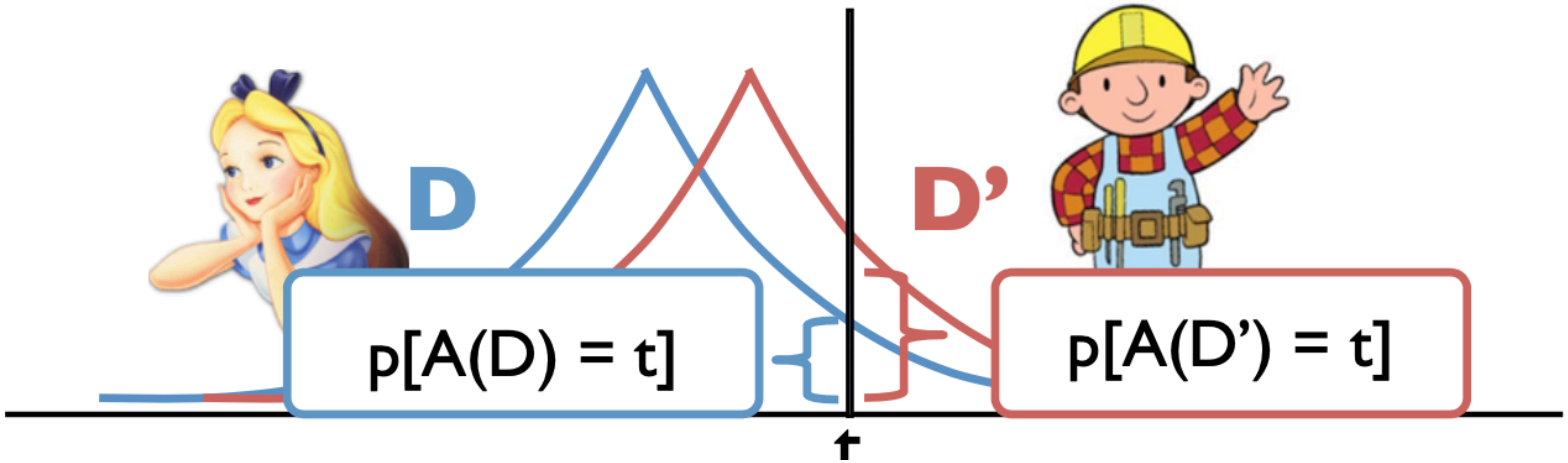
<u>Membership Inference (SSSS'17)</u>

Determining whether a sample was part of the training set.

<u>Model Inversion (FJR'15)</u>

Reconstruct training samples

Differential Privacy (DMNS'06)



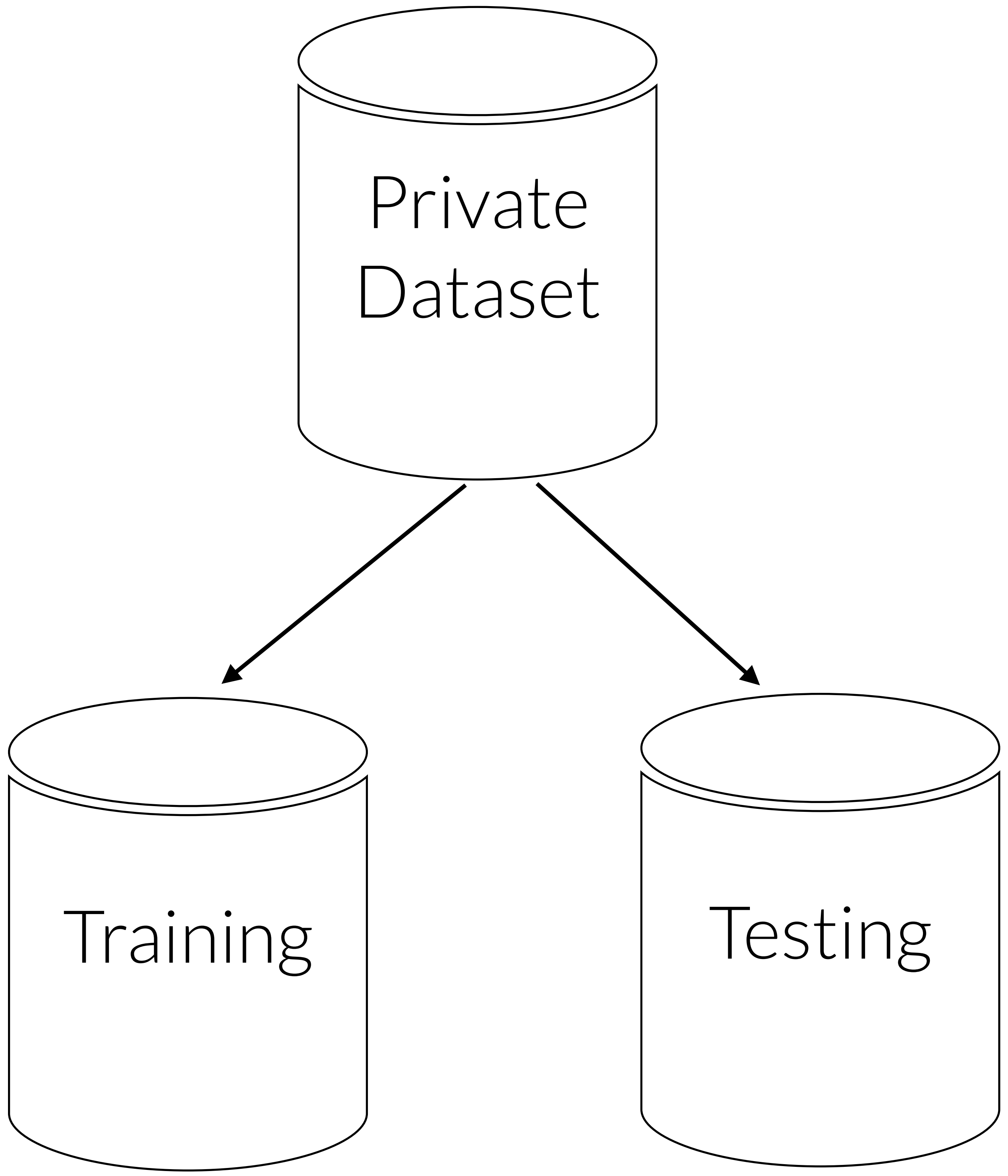
A randomized algorithm $\mathscr{A}: \mathscr{D} \rightarrow \mathscr{R}$ satisfies (ϵ, δ)-differential privacy (DP) if for any two adjacent inputs $\mathscr{D}, \mathscr{D}' \in \mathscr{D}$ that differ in an entry and for any subset of outputs $t \subseteq \mathscr{R}$ it holds that :

$$\Pr[\mathscr{A}(D) \in t] \leq e^{\epsilon} \Pr[\mathscr{A}(D') \in t] + \delta$$

Quantifies information leakage

Allows for small probability of failure

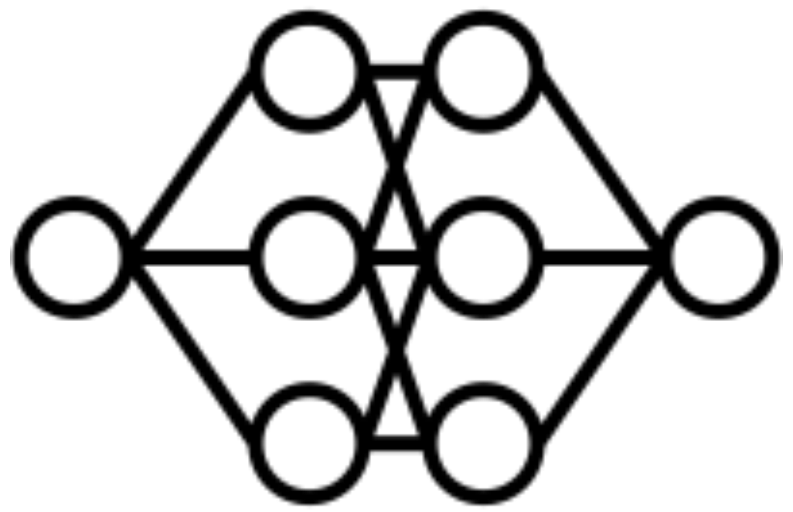
Problem Setup



Known attributes:
- Size
- Schema

Privacy Budget
(ϵ_f, δ_f)

Privacy Firewall



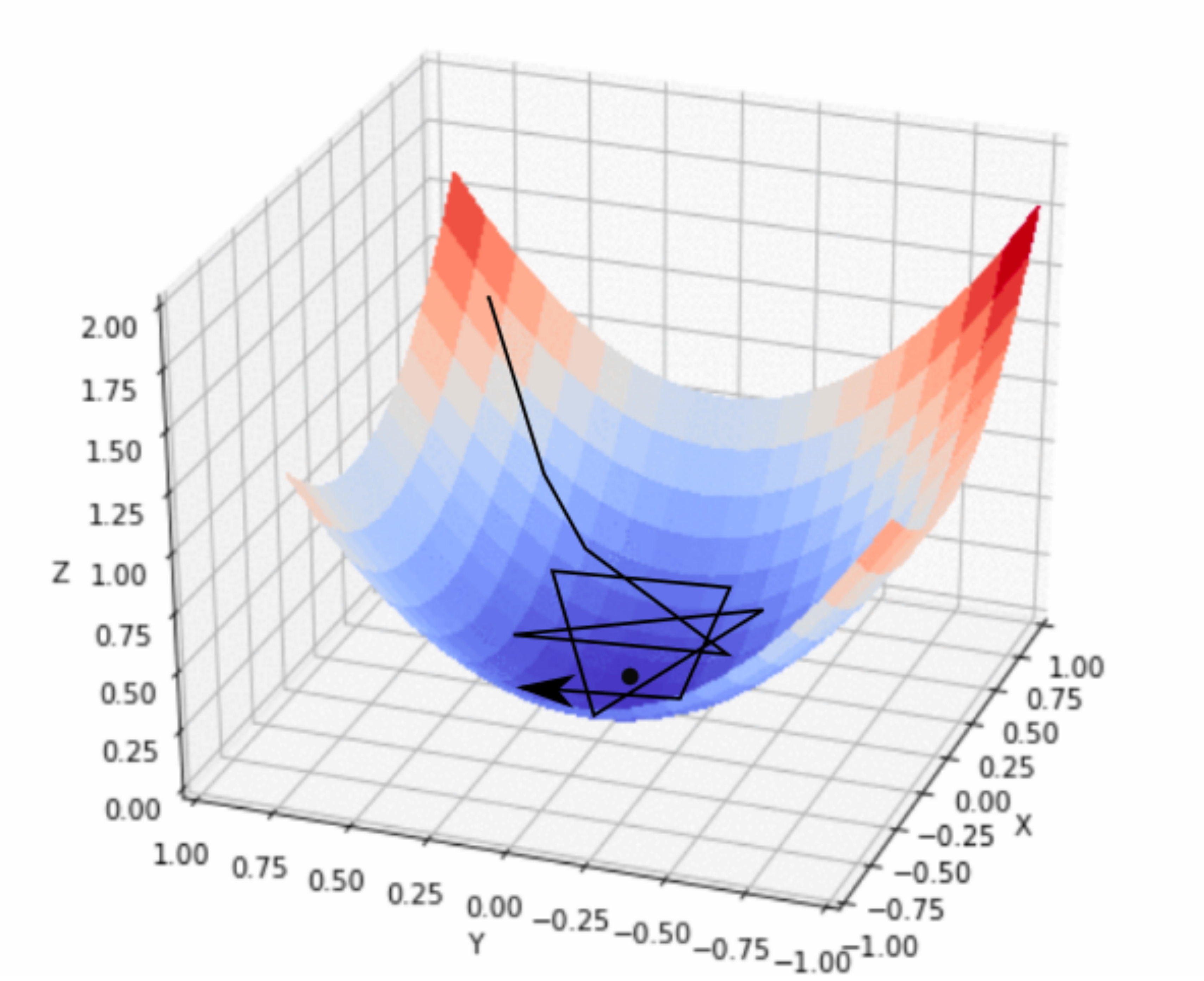
Target : build a ML model s.t best accuracy on the test set.

Task : decide model and its hyperparameters

Constraint : End to end privacy budget (ϵ_f, δ_f)

Note : Test set is also private and queries on it should also be privatised. We will assume separate budget for such queries on test set.

DP Stochastic Gradient Descent



Moments Accountant (ACG+'16) is used to compose noise added in each iteration

Hyperparameter tuning:

1. Model architecture
2. Noise multiplier (σ)
3. Batch size
4. Iterations
5. Learning rate
6. Clipping threshold (C)

Sample lot of size L from training set with probability L/n

↓

Compute gradients w.r.t weights

↓

Clip gradients to norm bound C and add noise $\mathcal{N}(0, C^2\sigma^2)$ before step

6D tuning is hard

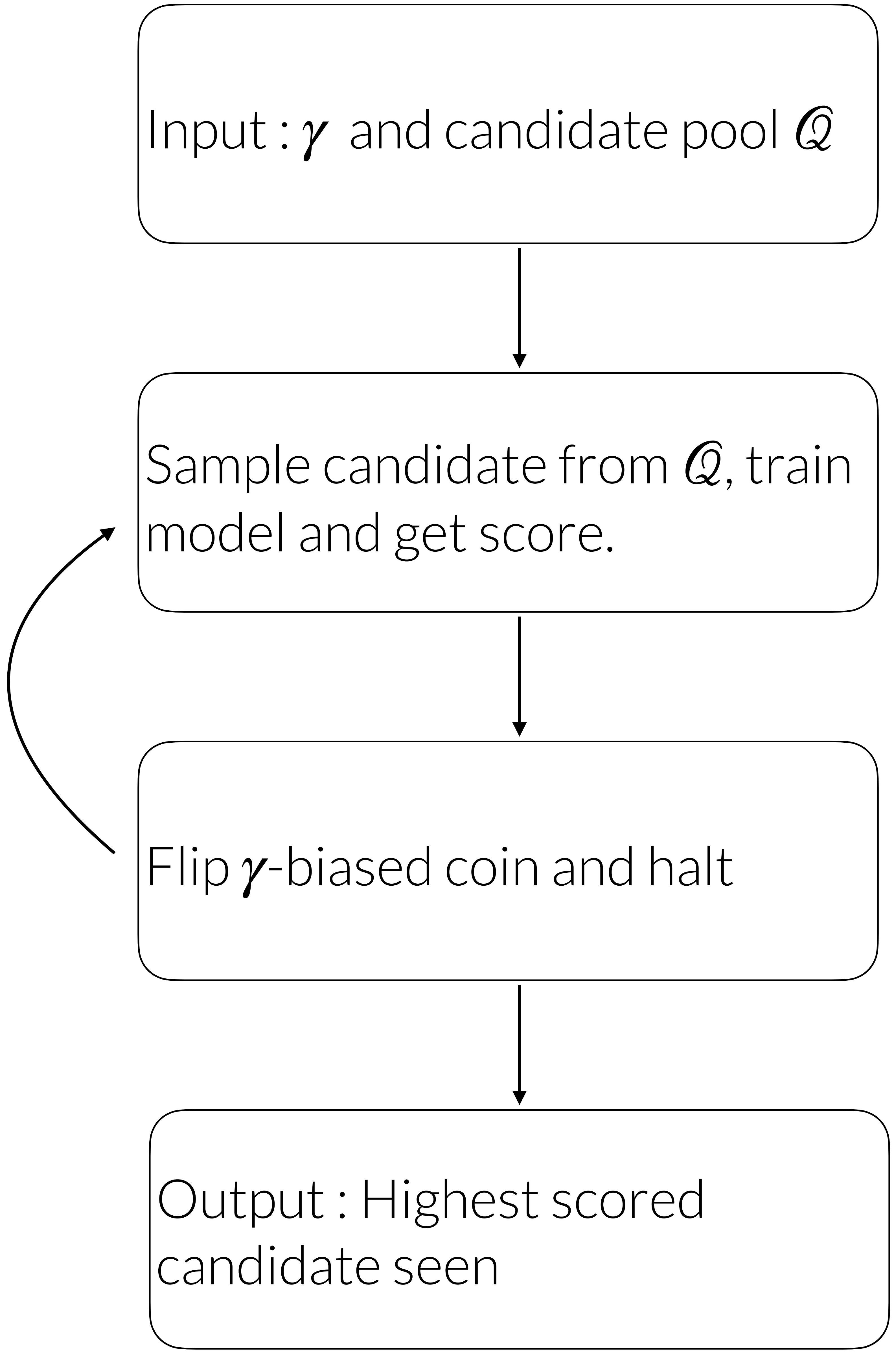
Training multiple times incurs privacy cost

Focus on learning rate and clipping threshold

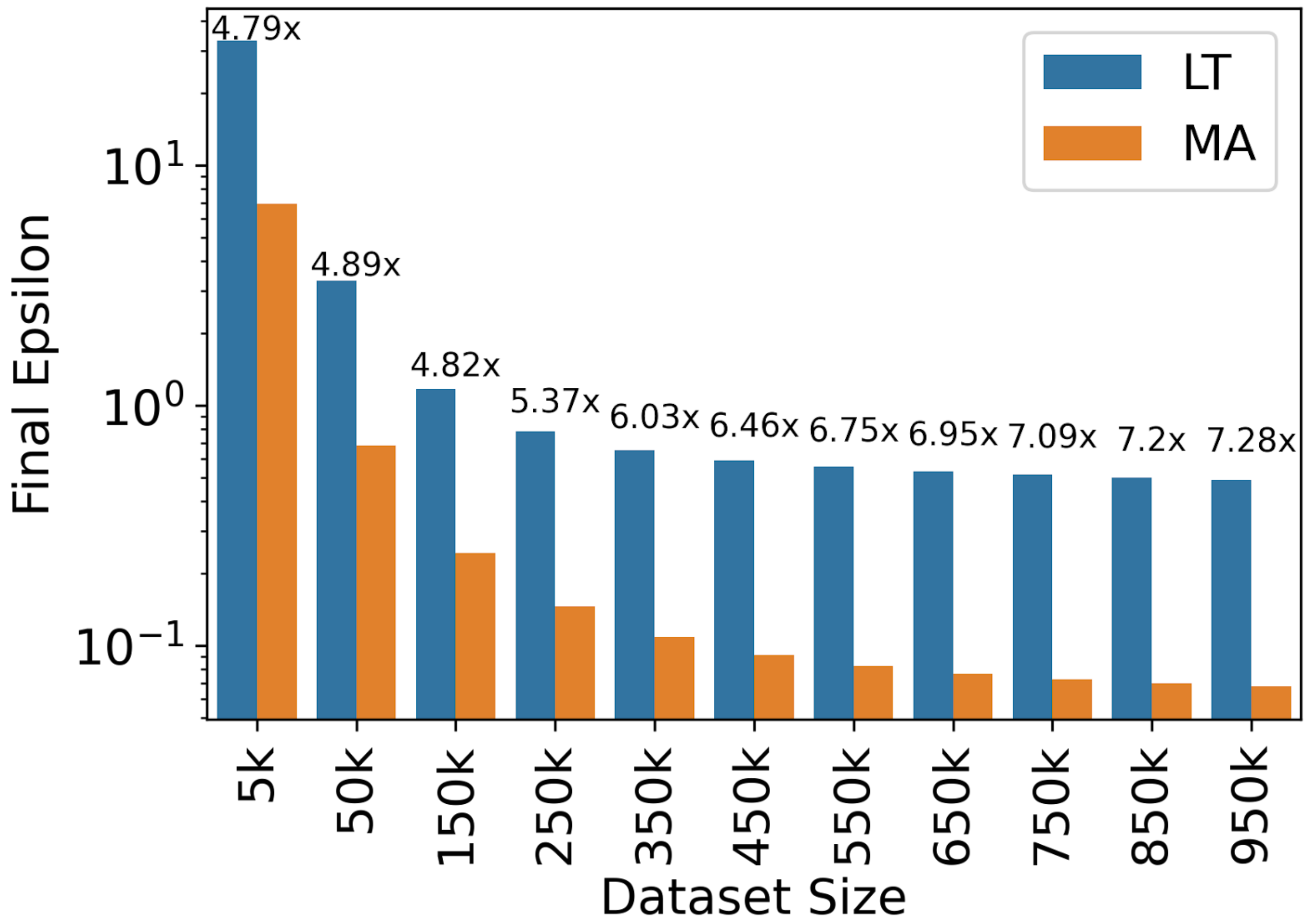
Tuning procedures

1. DP Composition using Moments Accountant (MA)
2. Liu and Talwar'19 (LT)

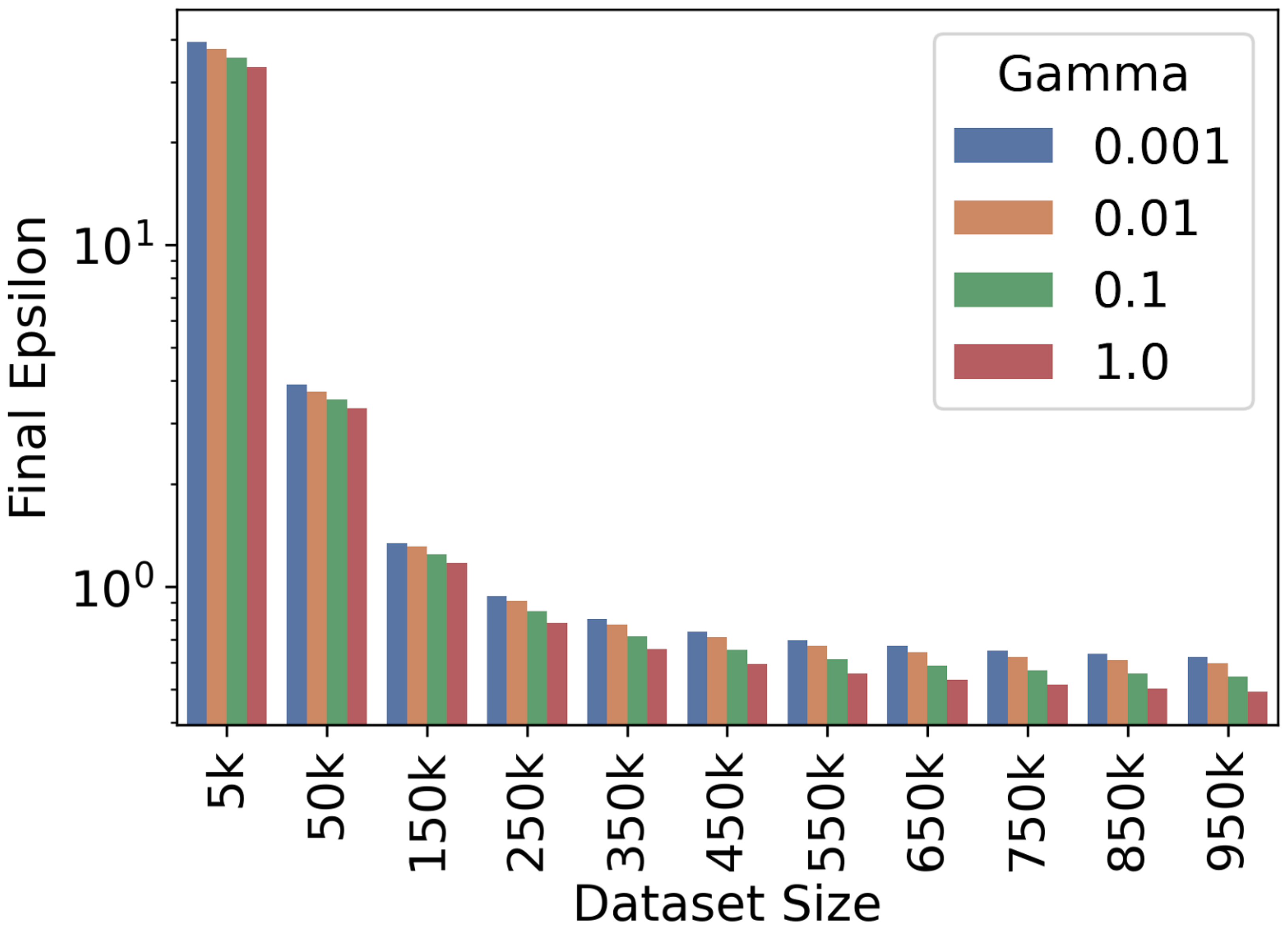
- Privacy Cost $\epsilon_f = 3\epsilon_1 + 3\sqrt{2\delta_1}$, $\delta_f = \sqrt{2\delta_1}\gamma + \delta_2$
- Choice of γ affects δ_1 which causes blowup of ϵ_1
- This blowup is ~5x of cost for 1 model train



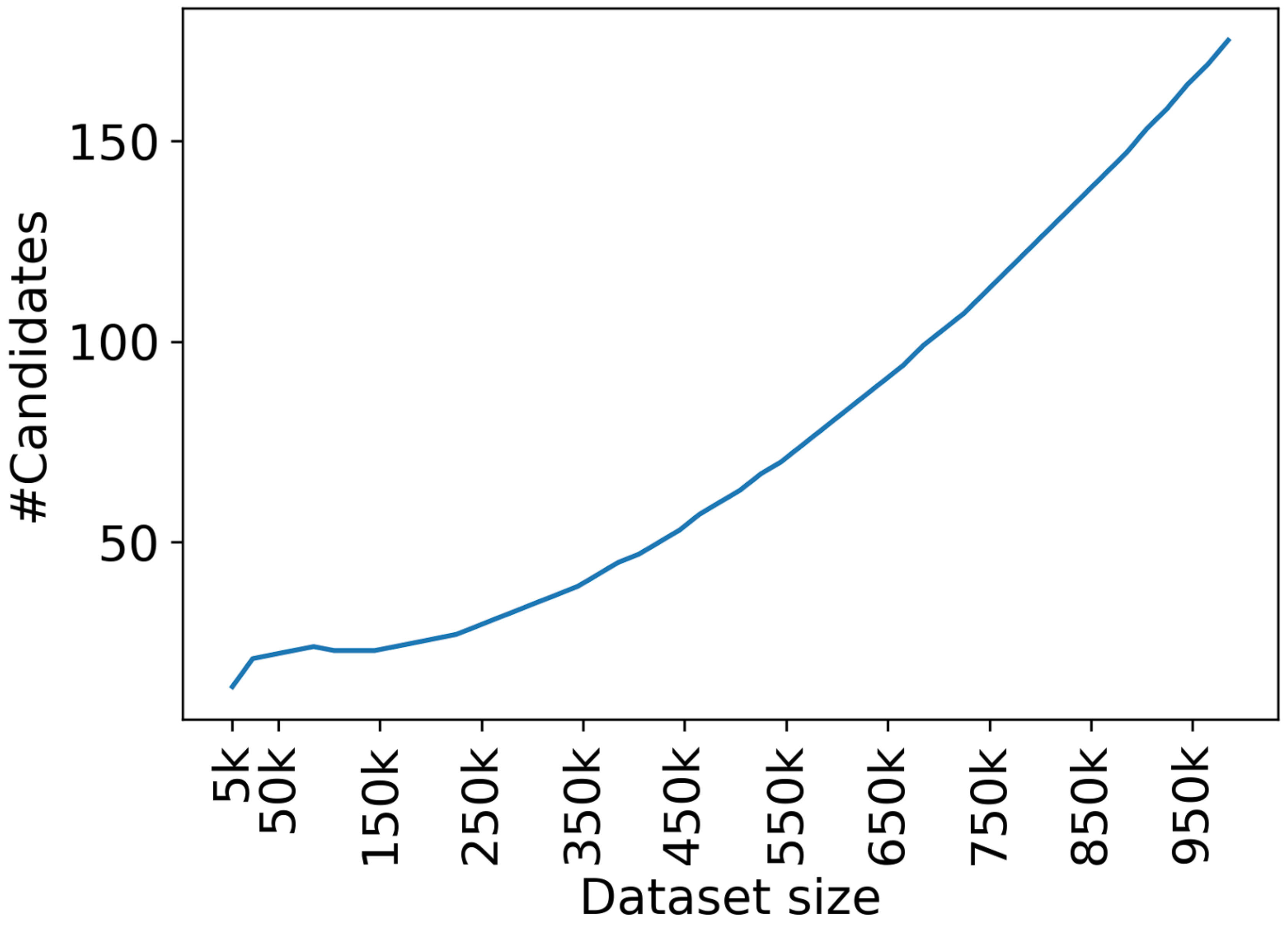
Cost of tuning LT vs MA



Privacy Blowup



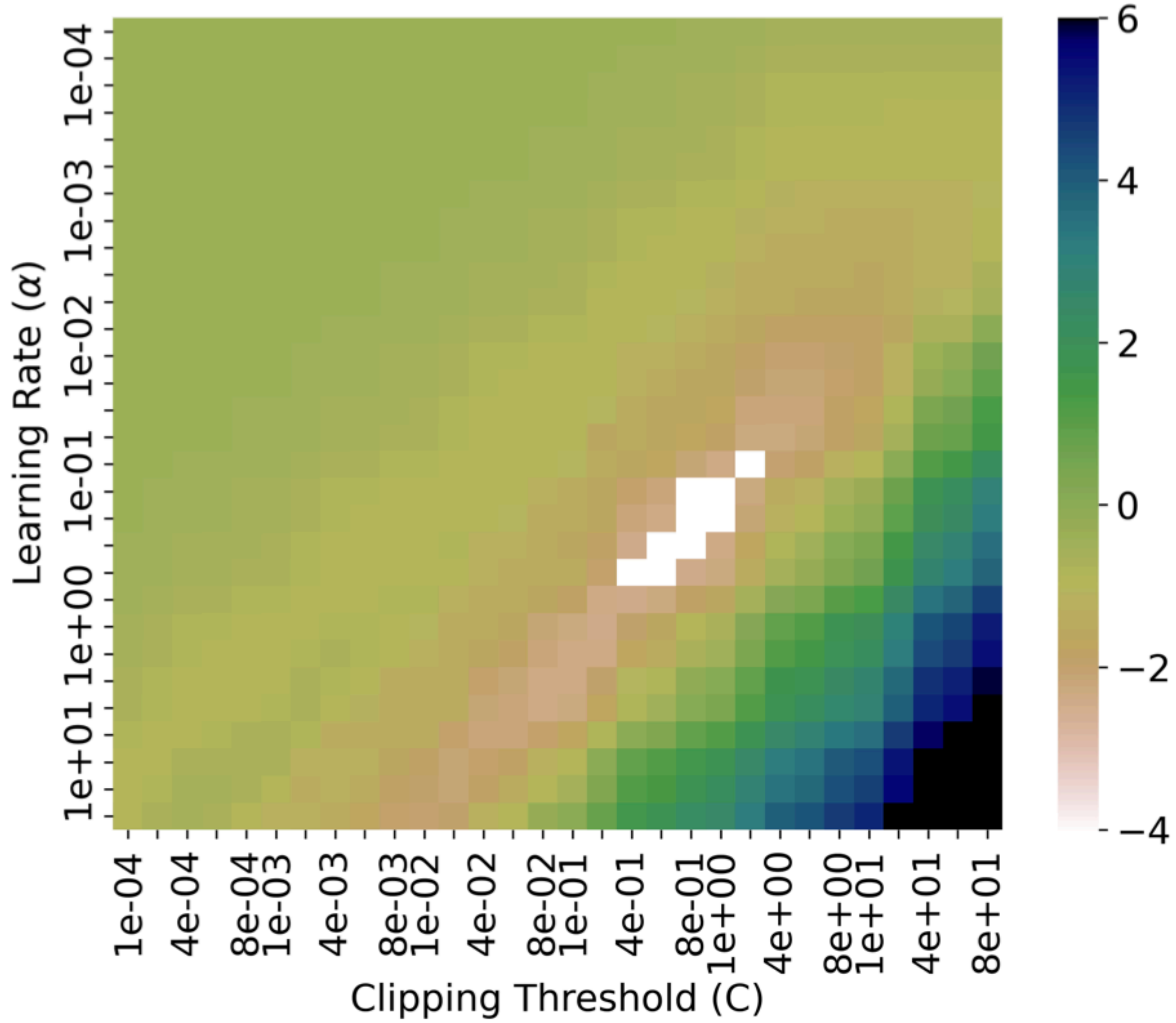
Effect of gamma



LT vs MA

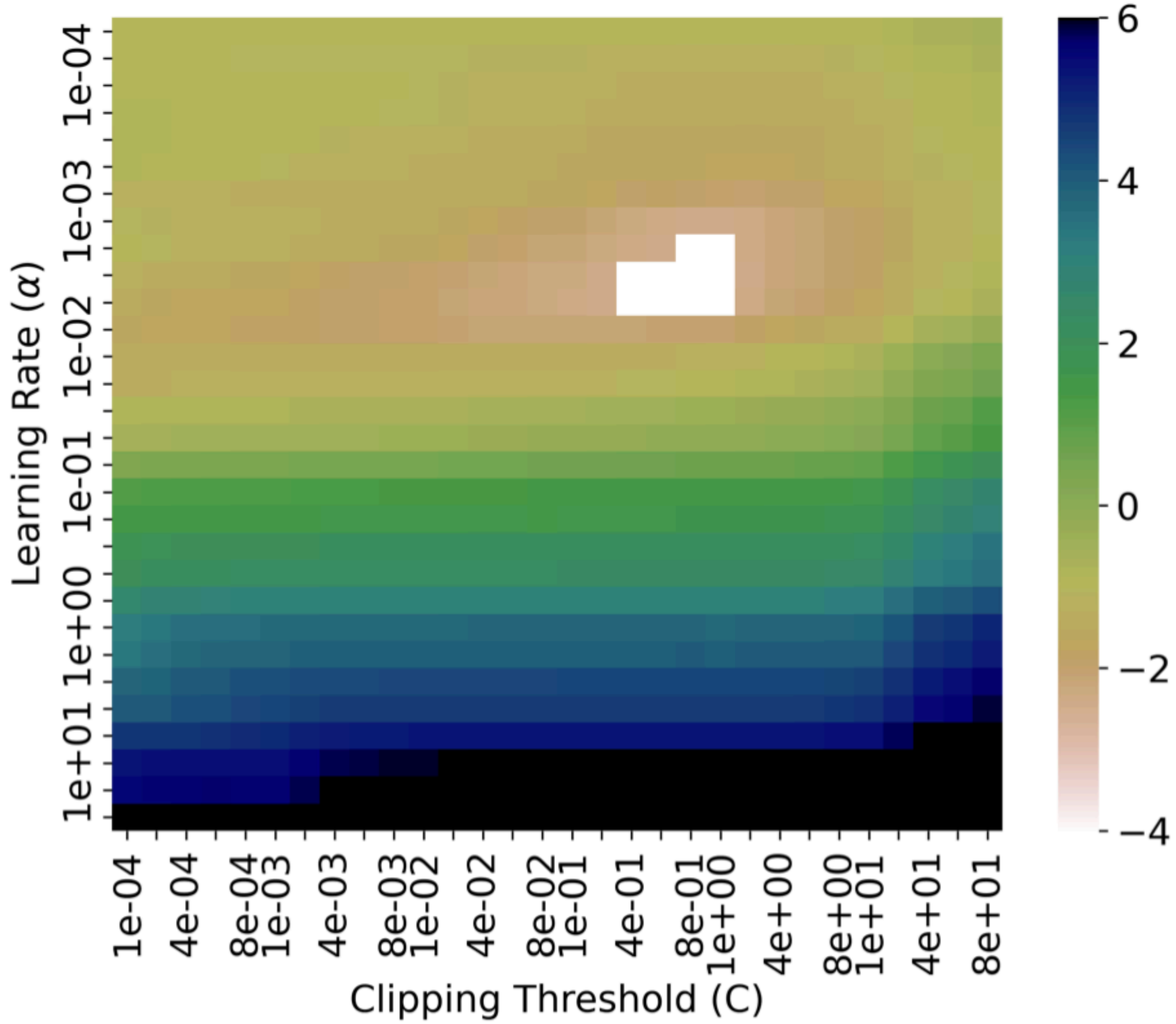
Tuning problem is still hard. Which are the best candidates to choose?

Relation between LR and C



DPSGD

- LR and C have inverse relation
- Tune both to get best candidate



DPAdam

- LR is adaptively adjusted
- Tune only C to get best candidate

Experimental setup

Dataset	Type	Samples	Dimensions	Classes
MNIST	Image	70000	784	10
Gisette	Image	6000	5000	2
Adult	Structured	45222	202	2
ENRON	Textual	5172	5512	2

Experimental datasets

Parameter	Values
Learning rate	0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1
Clipping norm	0.1, 0.2, 0.5, 1

Parameter Grid

- For each dataset, we split train = 80% and test = 20%
- Train two layer NN (TLNN) and logistic regression (LR) models for each
- Run each model 3 times and report average

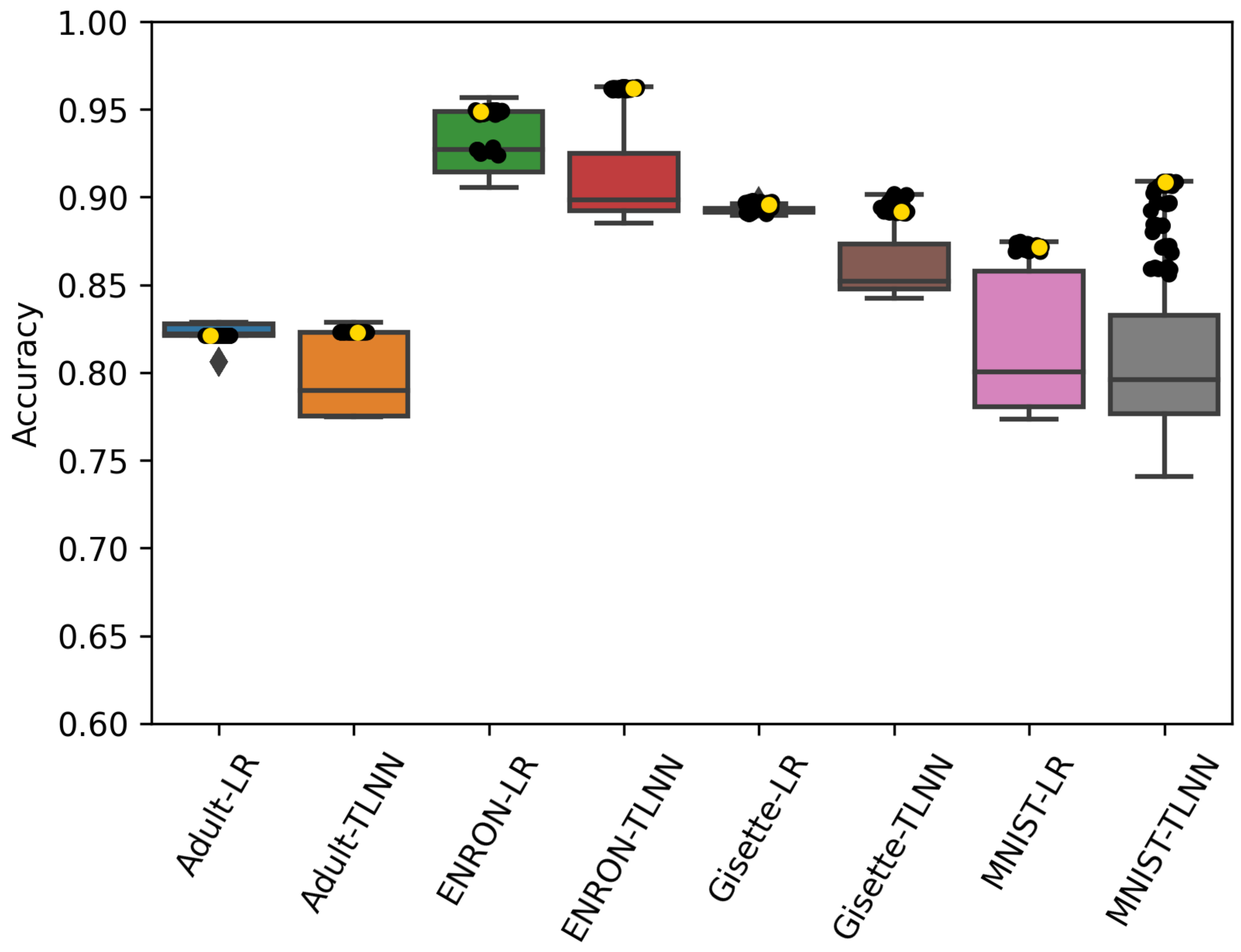
Tuning DPAdam

DPAdam inherits 3 hyperparameters from Adam:

1. Initial learning rate (α)
2. First moment decay rate (β_1)
3. Second moment decay rate (β_2)

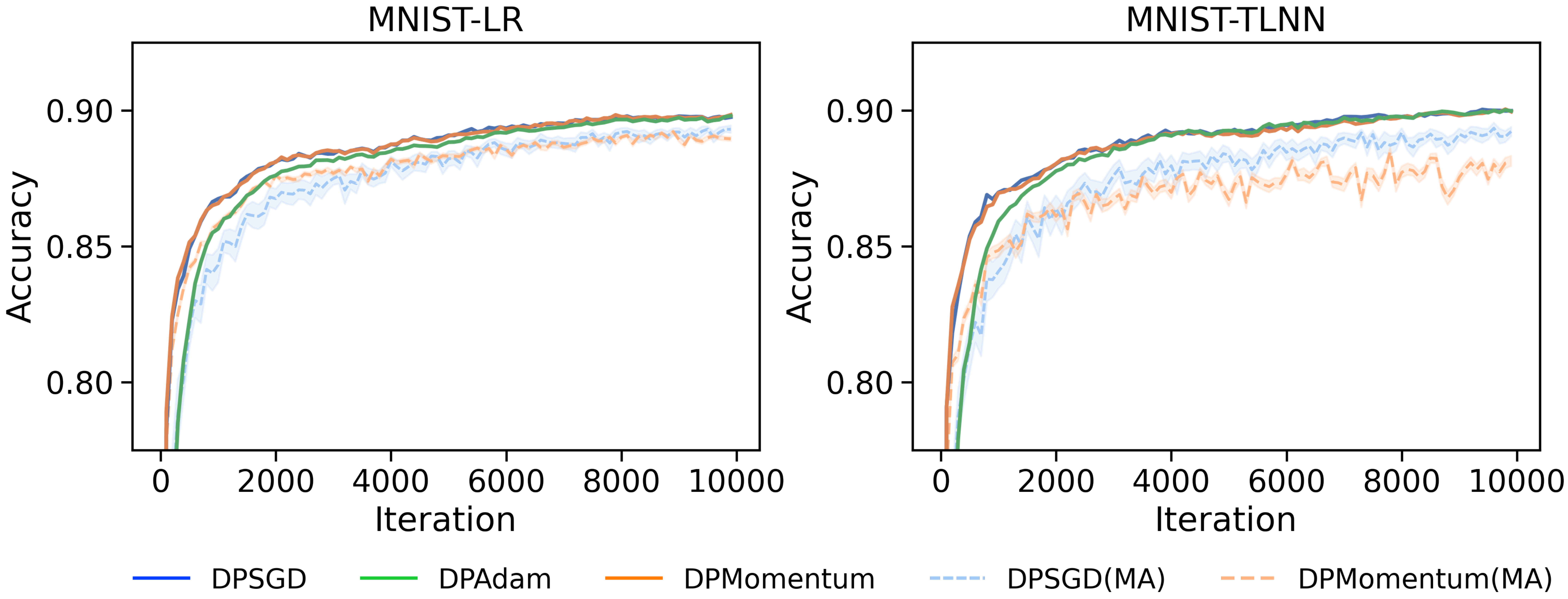
Suggested default values are α = 0.001, β_1 = 0.9, β_2 = 0.999

These values translate to DP setting



Black dots (α = 0.001) and Gold dots (default)

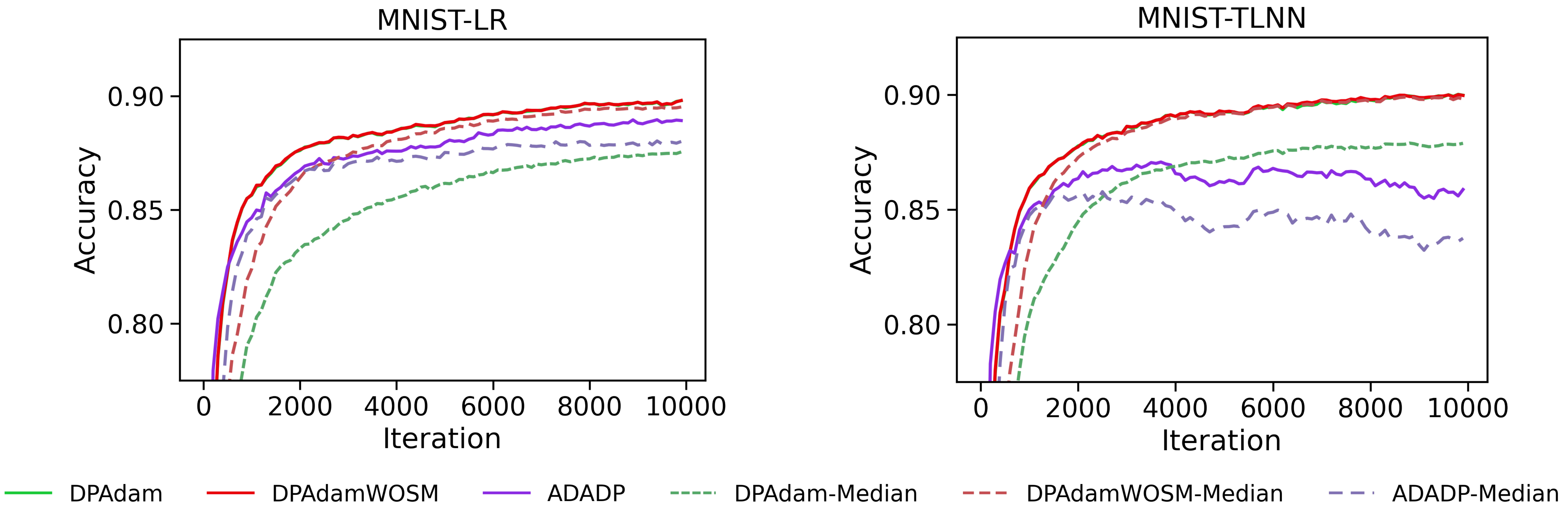
Adaptive vs Non-adaptive optimizers



Tuning parameters — DPMomentum: LR, C and M (280) DPSGD: LR and C (40) DPAdam: C (4)

- DPAdam performs at par with DPSGD and DPMomentum
- Due to #candidates, DPSGD and DPMomentum tune using LT algorithm with more privacy budget
- DPSGD and DPMomentum have subpar performance if randomly 4 candidates are chosen

DPAdamWOSM



The learning rate of DPAdam converges to a static value, effective step size (ESS):

$$ESS = \frac{\alpha}{(\sigma C/L) + \xi}$$

DPAdamWOSM saves the second moment computation and sets LR = ESS

Conclusion

1. Investigated honest hyperparameter tuning for DP optimizers
2. Compared LT vs MA as tuning procedures.
3. LT is better when large candidates while MA when candidates are less.
4. Explored that LR and C show inverse relationship for DPSGD.
5. Compared non-adaptive and adaptive DP optimizers
6. Proposed DPAdamWOSM, which avoids second moment computation and has better performance during earlier iterations.

Thank you for listening!