# Data-Driven Performance Guarantees for Classical and Learned Optimizers

## Neurips 2025:
## Journal-to-Conference track

**Data-Driven Performance Guarantees for Classical and Learned Optimizers**
R. Sambharya, B. Stellato
*Journal of Machine Learning Research, 2025*

Rajiv Sambharya

Penn
University of Pennsylvania

Bartolomeo Stellato

Princeton University

# Data-Driven Performance Guarantees for Classical and Learned Optimizers

## Neurips 2025:
## Journal-to-Conference track

**Data-Driven Performance Guarantees for Classical and Learned Optimizers**
R. Sambharya, B. Stellato
*Journal of Machine Learning Research, 2025*

Rajiv Sambharya

Penn
University of Pennsylvania

Bartolomeo Stellato

PRINCETON
UNIVERSITY

# Real-world optimization is parametric

parameter $x$ → 

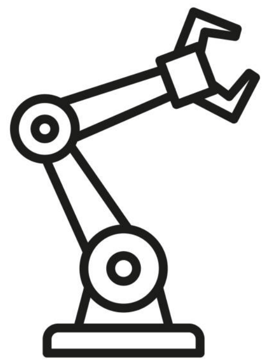
$$\begin{array}{ll} \text{minimize} & f(z, x) \\ \text{subject to} & z \in \Omega(x) \end{array}$$

convex problem in $z$

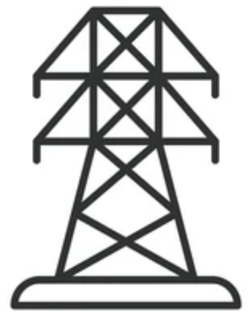
→ optimal solution $z^\star(x)$

**applications**

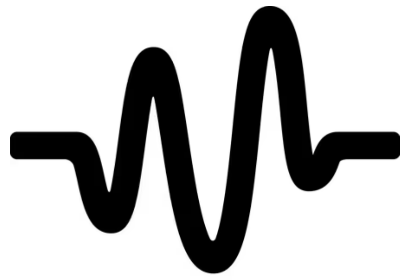
robotics



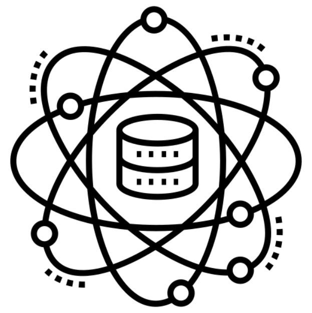
power grids



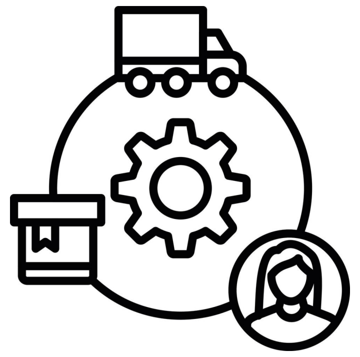
signal processing



data science



operations research



first-order methods are popular…

…but classical worst-case convergence bounds can be very loose!

fixed-point iterations

$$z^{k+1}(x) = T(z^k(x), x)$$

example: projected gradient descent

$$z^{k+1}(x) = \Pi_{\Omega(x)} \underbrace{\left(z^k - \theta^k \nabla f(z^k(x), x)\right)}_{\text{gradient step}}$$

projection

# Building probabilistic guarantees for classical optimizers

can we exploit the parametric structure to get tighter guarantees?

$\{x_i\}_{i=1}^N$

assume access to a **dataset** of parameters

algorithm steps

tolerance

$$e(x) = \mathbf{1}(\ell^k(x) > \epsilon)$$

any metric

e.g., the fixed-point residual

$$\ell^k(x) = \|z^k(x) - z^{k-1}(x)\|_2$$

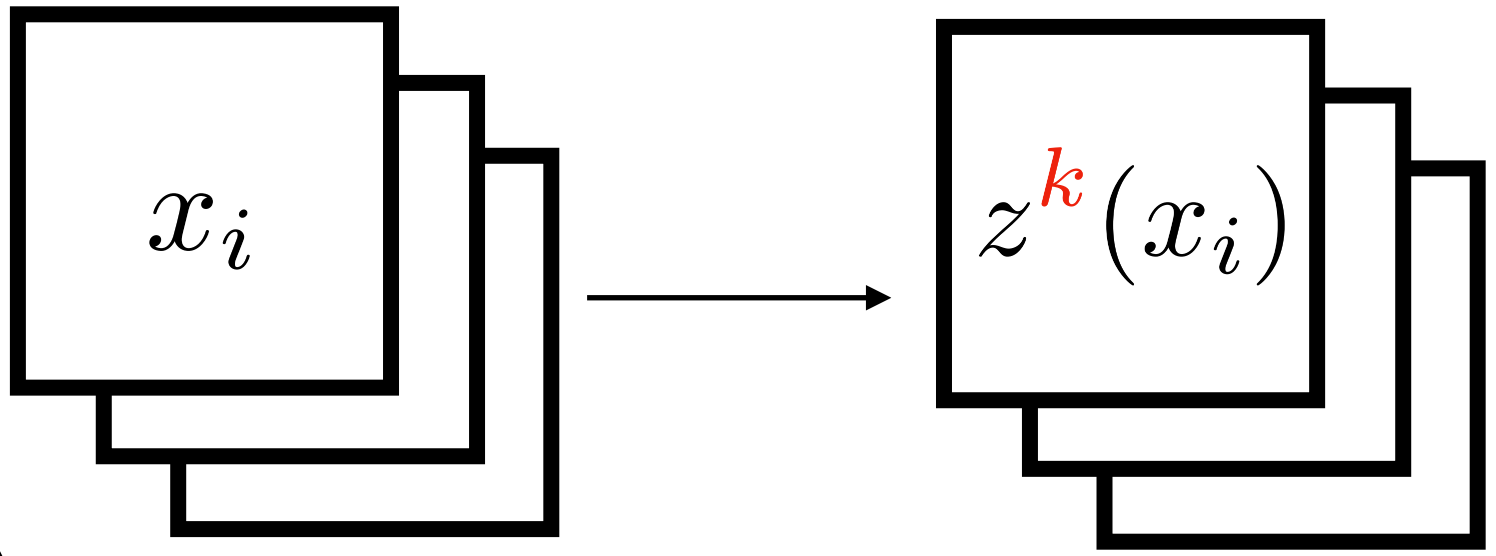
**step 1**

run $k$ steps

for each problem

parameters

candidate solutions




**step 2**

compute the empirical risk

$$\frac{1}{N}\sum_{i=1}^N e(x_i)$$

**step 3**

bound the risk

$$\mathop{\mathbf{E}}_{x \sim \mathcal{X}} e(x) \leq \mathrm{kl}^{-1}\left(\frac{1}{N}\sum_{i=1}^N e(x_i) \,\middle|\, \frac{\log(2/\delta)}{N}\right)$$

risk

inverse kl divergence (*1D convex problem*)

empirical risk

regularizer

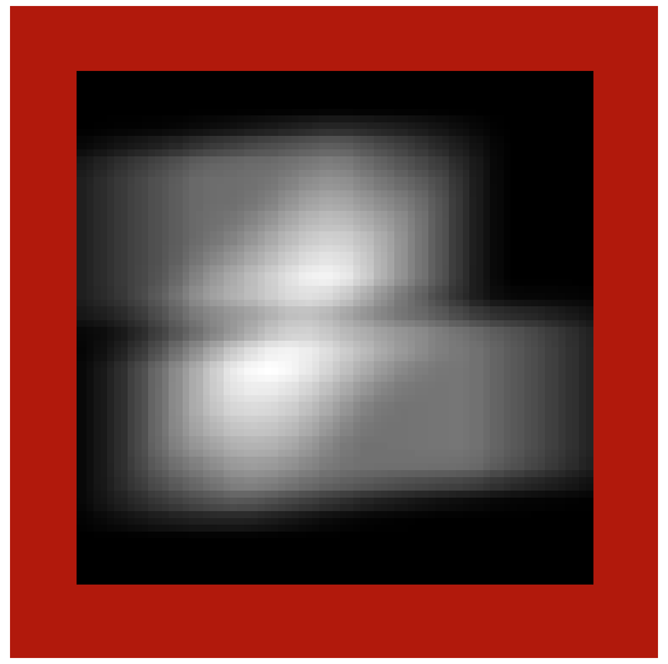
# Tight guarantees for image deblurring

blurry image $x$ →

**image deblurring**

$$\begin{array}{ll}\text{minimize} & (1/2)\|Az - x\|_2^2 + \lambda\|z\|_1 \\ \text{subject to} & 0 \leq z \leq 1\end{array}$$

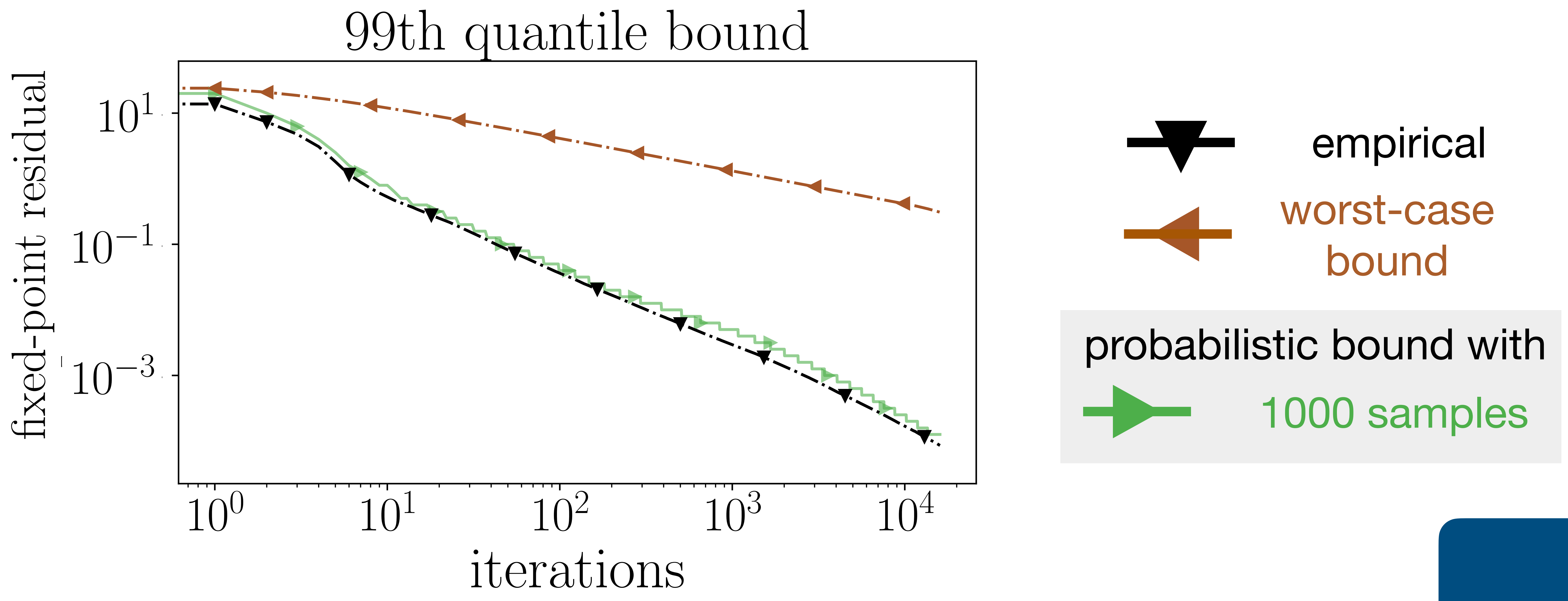
→ deblurred image $z^\star$



OSQP solver →

Stellato et al. 2020

**quantile bounds**




our probabilistic bounds are much tighter than classical worst-case guarantees!

# Data-Driven Performance Guarantees for Classical and Learned Optimizers

**Neurips 2025:**
**Journal-to-Conference track**

**Data-Driven Performance Guarantees for Classical and Learned Optimizers**
R. Sambharya, B. Stellato
*Journal of Machine Learning Research, 2025*


Rajiv Sambharya

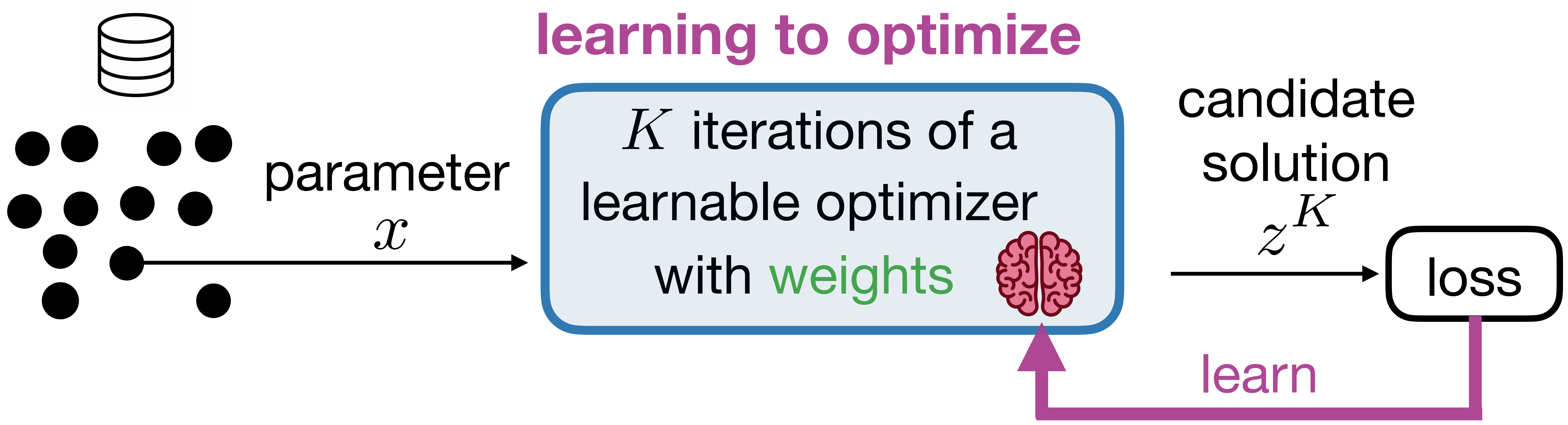
University of Pennsylvania



Bartolomeo Stellato

Princeton University

# The learning to optimize paradigm




learned optimizers have seen lots of empirical success…

…however, they lack guarantees on unseen data

# Optimizing PAC-Bayes guarantees for learned optimizers

**learning task**

$$\min_{\Theta} \mathop{\mathbf{E}}_{\theta\sim\Theta} \mathop{\mathbf{E}}_{x\sim\mathcal{X}} e_\theta(x)$$

distribution of algorithm weights (e.g., step sizes, warm starts)

1. pick a prior $\Theta_0$ before observing data
2. observe data $\{x_i\}_{i=1}^N$
3. learn the posterior $\Theta : \theta \sim \Theta$
4. bound the performance

$$\mathbf{P}\left(\mathop{\mathbf{E}}_{\theta\sim\Theta} \mathop{\mathbf{E}}_{x\sim\mathcal{X}} e_\theta(x) \leq \hat{t}_N\right) \geq 1-\delta$$

McAllester 1999 Maurer 2004

**data-driven bound**

$$\hat{t}_N = \mathrm{kl}^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} \mathop{\mathbf{E}}_{\theta\sim\Theta} e_\theta(x_i) \,\middle|\, \frac{\mathrm{KL}(\Theta||\Theta_0) + \log(2\sqrt{N}/\delta)}{N}\right)$$

empirical risk regularizer

minimize the data-driven bound itself!

Dziugaite et al. 2017
Majumdar et al. 2021

# Learned ISTA results for sparse coding

noisy measurement
$x$

**signal reconstruction**

$$\text{minimize} \quad (1/2)\|Az - x\|_2^2 + \lambda\|z\|_1$$

ground truth sparse signal
$z^\star$

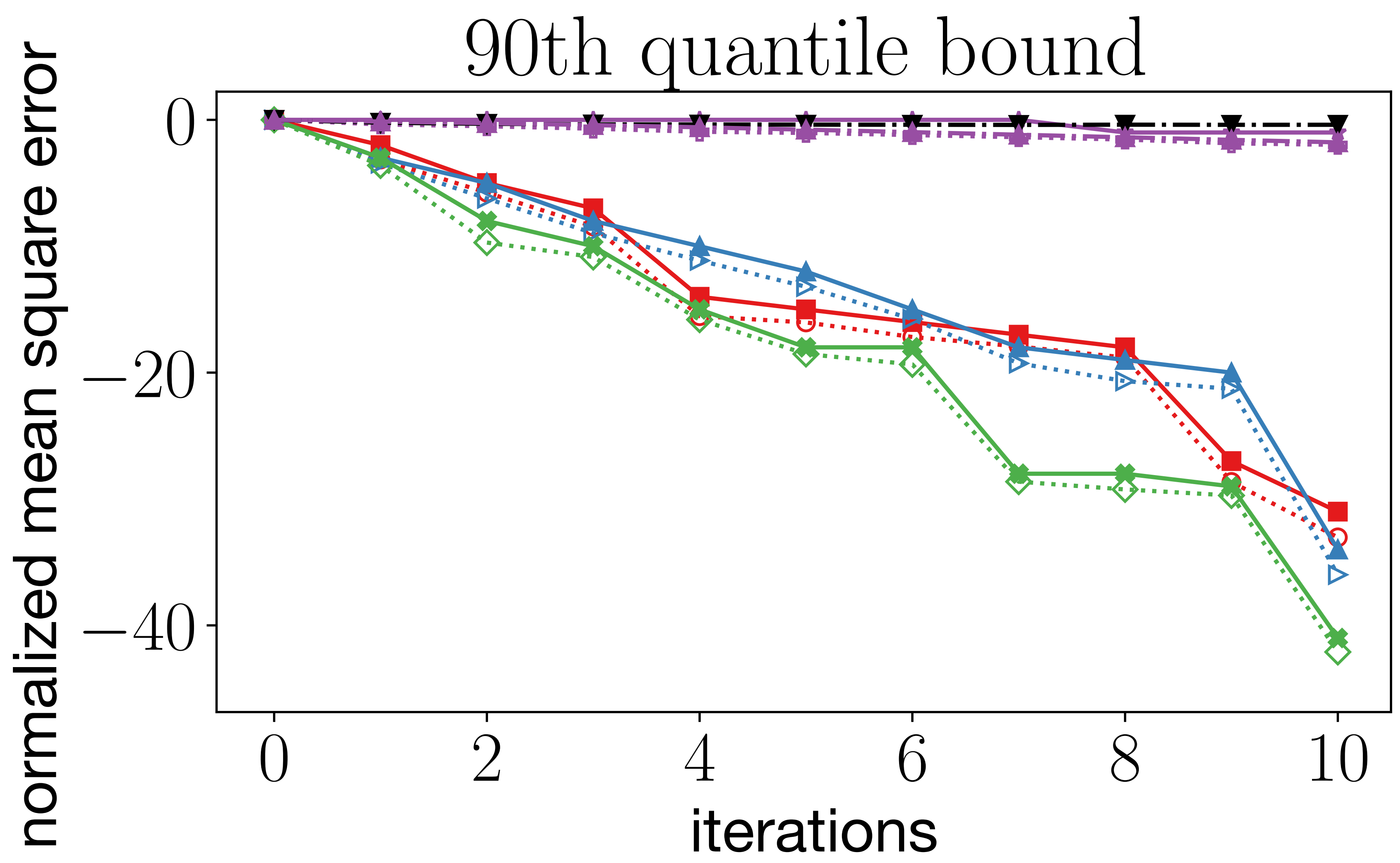
ISTA (iterative shrinkage thresholding algorithm)

$$z^{j+1} = \text{soft threshold}_{\frac{\lambda}{L}}\left(z^j - \frac{1}{L}A^T(Az^j - b)\right)$$

Learned ISTA

$$z^{j+1} = \text{soft threshold}_{\psi^j}\left(W_1^j z^j + W_2^j b\right)$$

| | Not learned | Learned | | | |
|---|---|---|---|---|---|
| | ISTA | LISTA | ALISTA | TiLISTA | GLISTA |
| Bound | | | | | |
| Empirical | | | | | |
| | | Gregor et al. 2010 | Liu et al. 2019 | Liu et al. 2019 | Wu et al. 2020 |




our bounds are close to empirical performance

learned optimizers provably perform well in just 10 steps