Data-Driven Performance Guarantees for Classical and Learned Optimizers

IOS Talk 2024
Rajiv Sambharya

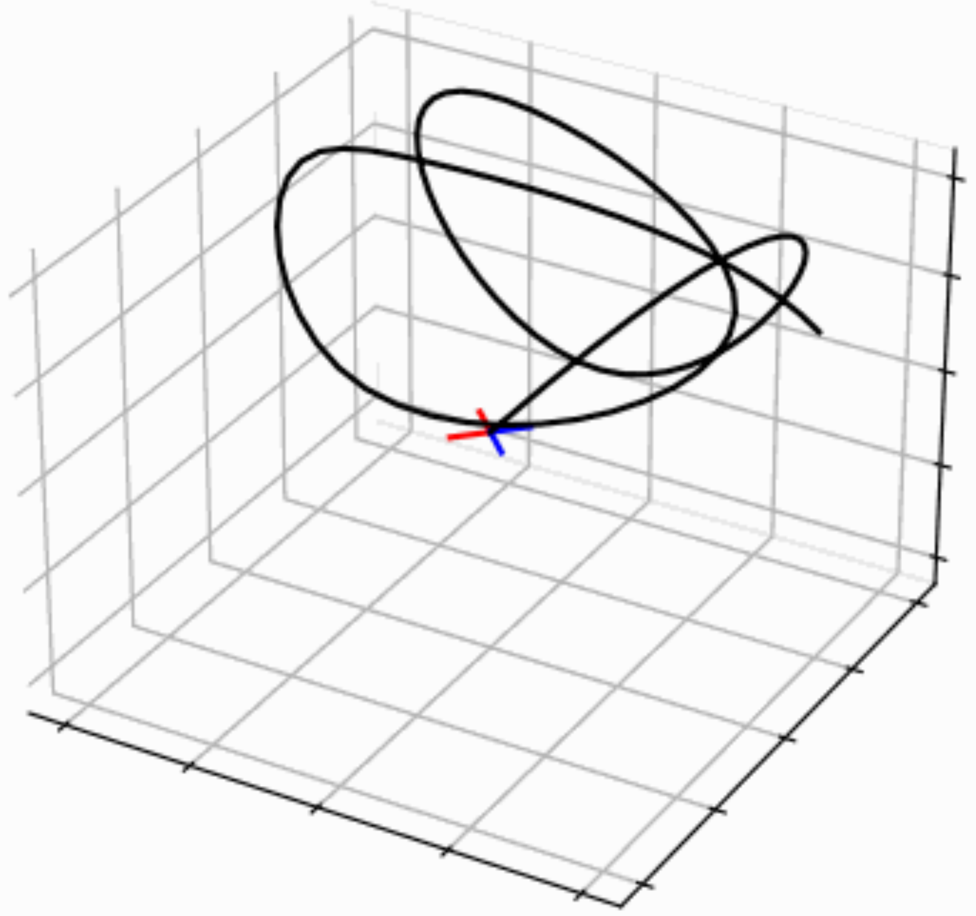
PRINCETON UNIVERSITY

ORFE

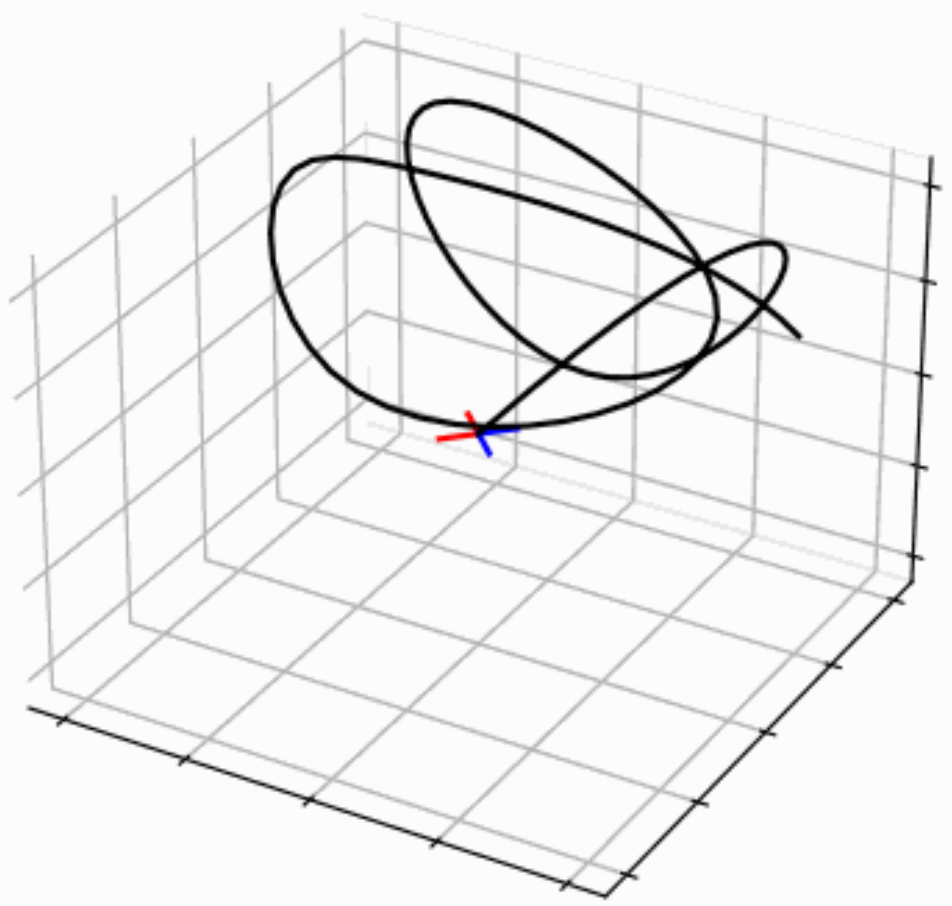
Collaborators

Bartolomeo
Stellato

Tracking a reference trajectory with a quadcopter



Success!
(If given enough time)



Failure: not enough time to solve

Model predictive control
optimize over a smaller horizon (T steps),
implement first control,
repeat

Current state,
reference trajectory

Model predictive controller

$$\begin{array}{ll} \text{minimize} & \sum_{t=1}^{T} \|x_t - x_t^{\text{ref}}\|_2^2 \\ \text{subject to} & x_{t+1} = A x_t + B u_t \\ & x_t \in \mathcal{X}, \quad u_t \in \mathcal{U} \\ & x_0 = x_{\text{init}} \end{array}$$

Control
inputs

Challenge: we need faster methods for optimization

Empirically | **Guarantees**

Claim: real-world optimization is parametric

Parameter θ $\longrightarrow$

minimize	$f_\theta(z)$
subject to	$g_\theta(z) \leq 0$

$\longrightarrow$ Optimal solution $z^\star(\theta)$

Robotics and control

Energy

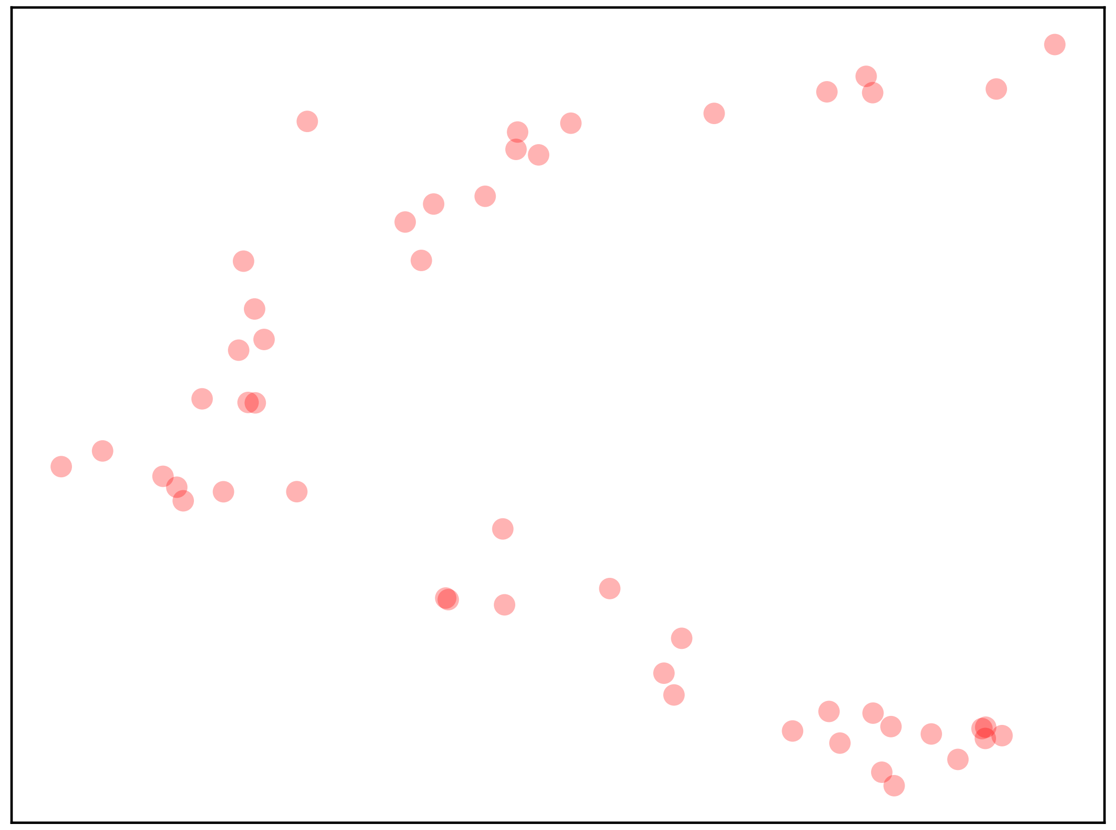
Data-Driven Performance Guarantees for Classical and Learned Optimizers

Parametric setting ✓

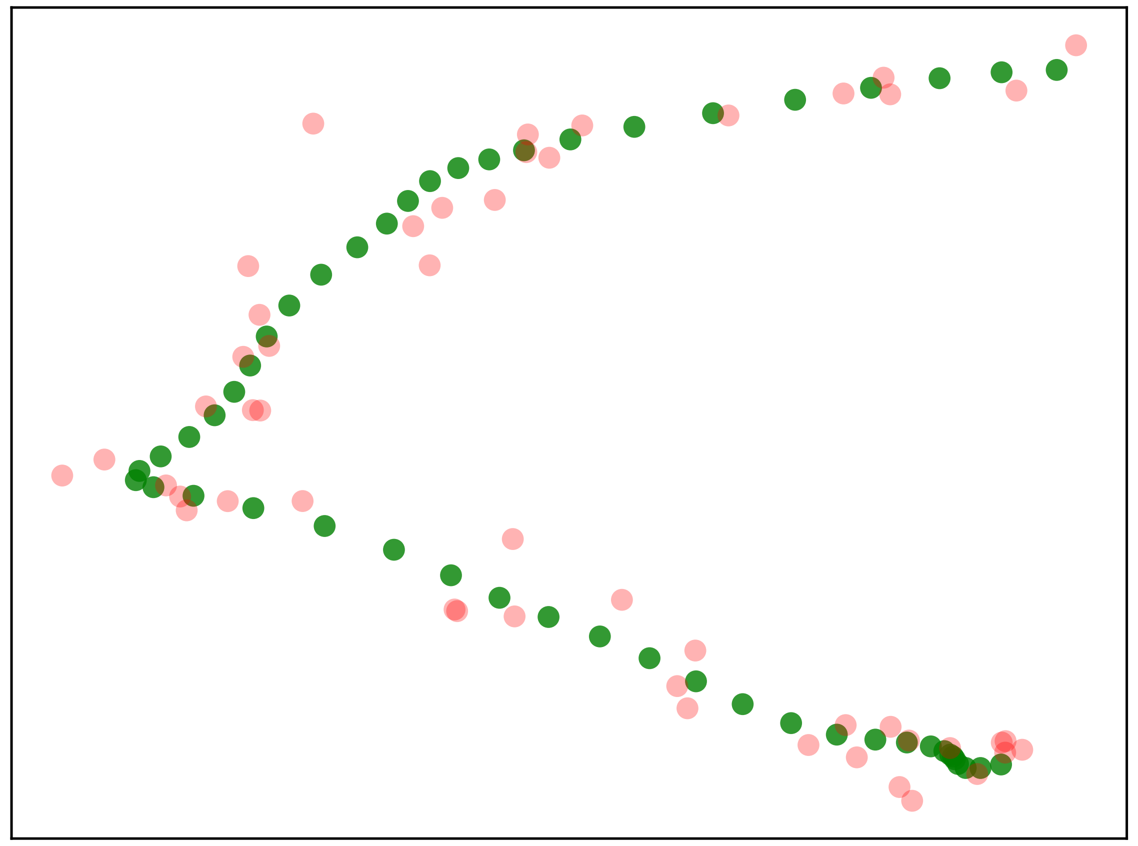
Faster optimization methods

~~Empirical~~ Guarantees

A running example: Robust Kalman filtering



Robust Kalman filtering



$\theta = \{y_t\}_{t=0}^{T-1}$

Noisy trajectory

Second-order cone program

$$\begin{array}{ll} \text{minimize} & \sum_{t=0}^{T-1} \|w_t\|_2^2 + \mu\psi_\rho(v_t) \\ \text{subject to} & x_{t+1} = Ax_t + Bw_t \quad \forall t \\ & y_t = Cx_t + v_t \quad \forall t \end{array}$$

$\{x_t^\star, w_t^\star, v_t^\star\}_{t=0}^{T-1}$

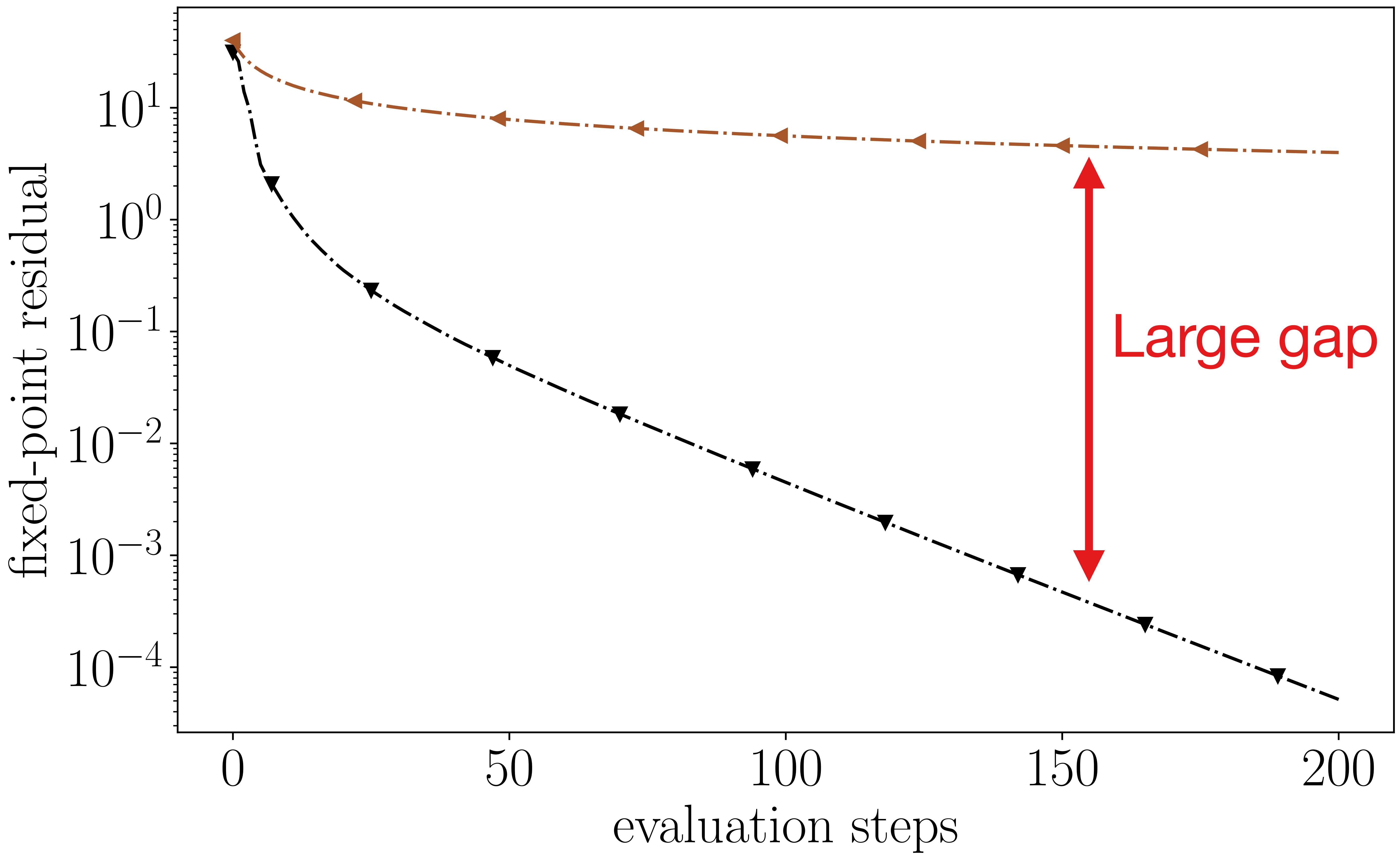
Recovered trajectory

Dynamics matrices: A, B

Observation matrix: C

Huber loss: ψ_ρ

Worst-case bounds can be very loose



Example: robust Kalman filtering

Second-order cone program

$$
\begin{array}{ll}
\text{minimize} & \sum_{t=0}^{T-1} \|w_t\|_2^2 + \mu \psi_\rho(v_t) \\
\text{subject to} & x_{t+1} = A x_t + B w_t \quad \forall t \\
 & y_t = C x_t + v_t \quad \forall t
\end{array}
$$

SCS empirical average performance over 1000 parametric problems

Worst-case bound

In practice: **linear** convergence over the parametric family

Worst-case analysis: **sublinear** convergence

Worst-case bounds do not consider the **parametric** structure

Our goal: fill this gap with data-driven methods

Our recipe for guarantees for classical optimizers

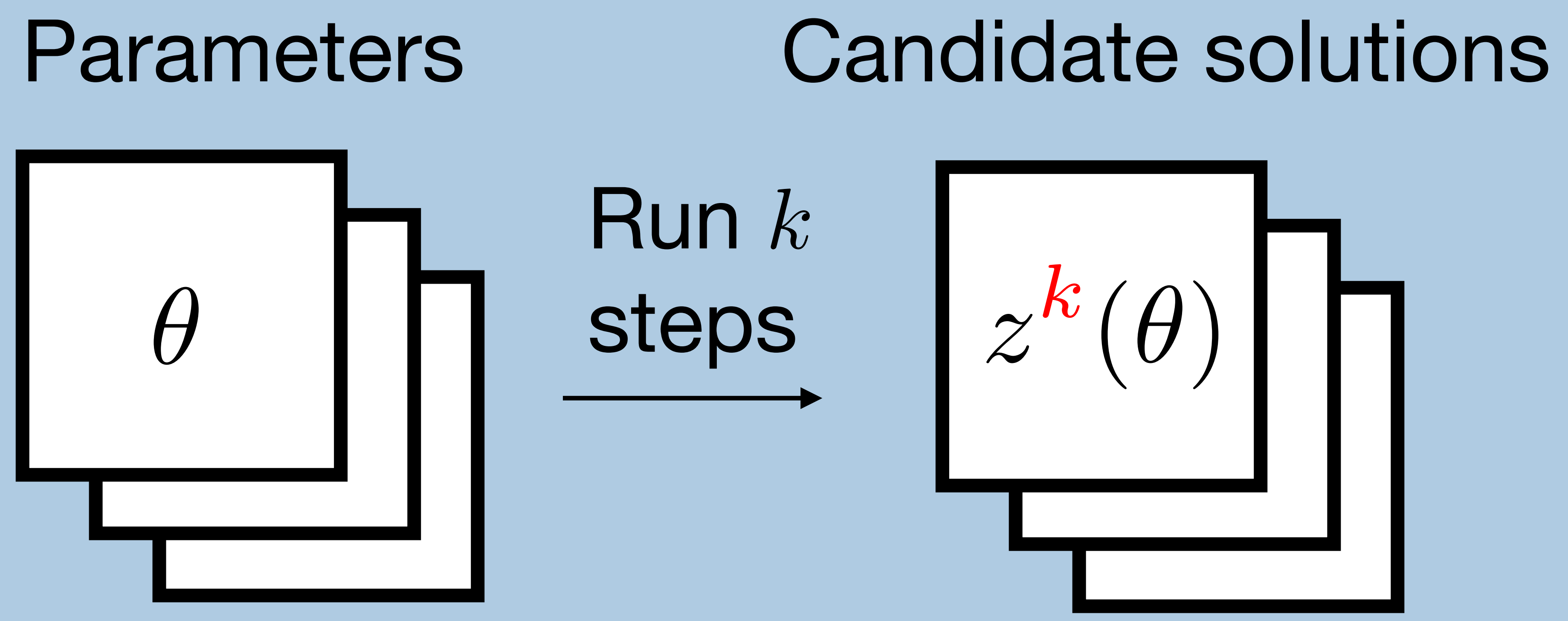
algorithm steps

tolerance

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

Any metric
(*e.g.*, fixed-point residual)

Step 1
Run k steps
for N parametric problems



Step 2
Evaluate the empirical risk

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

Step 3
Bound the risk
(Next slide)

risk $= \mathbf{E}_{\theta \sim \mathcal{X}}\, e(\theta) \leq$ bound

Statistical learning theory can bound the risk

algorithm steps

tolerance

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

Any metric
(*e.g.*, fixed-point residual)

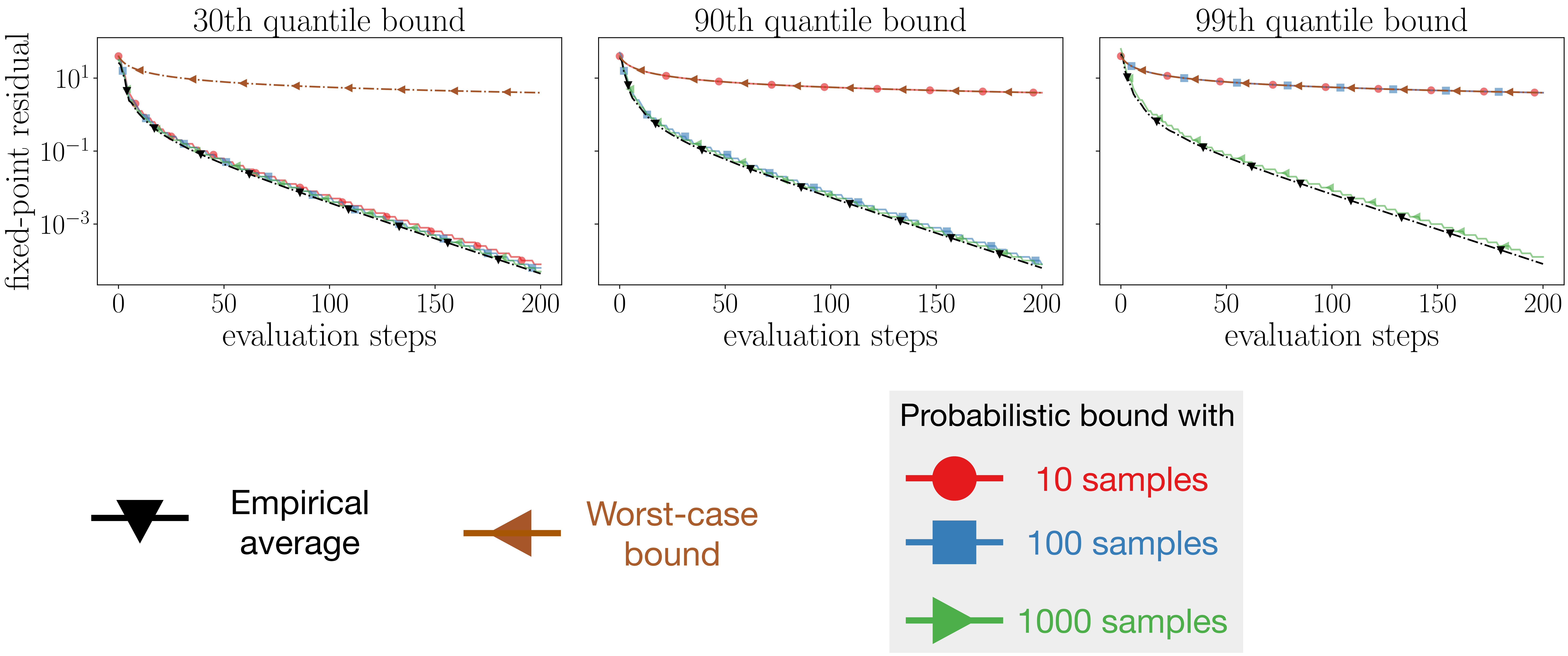
Sample convergence bound: with probability $1-\delta$ [Langford et. al 2001]

$$\mathbf{E}_{\theta\sim\mathcal{X}} e(\theta) \leq \mathrm{KL}^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} e(\theta_i) \middle| \frac{\log(2/\delta)}{N}\right)$$

$\mathbf{P}(\ell^k(\theta) > \epsilon) =$ risk $\leq \mathrm{KL}^{-1}$ (empirical risk | regularizer)

"With probability $1-\delta$, 90% of the time the fixed-point residual is below $\epsilon = 0.01$ after $k = 20$ steps"

Robust Kalman filtering guarantees



With 1000 samples, we provide strong probabilistic guarantees on the 99th quantile

Data-Driven Performance Guarantees for Classical and Learned Optimizers

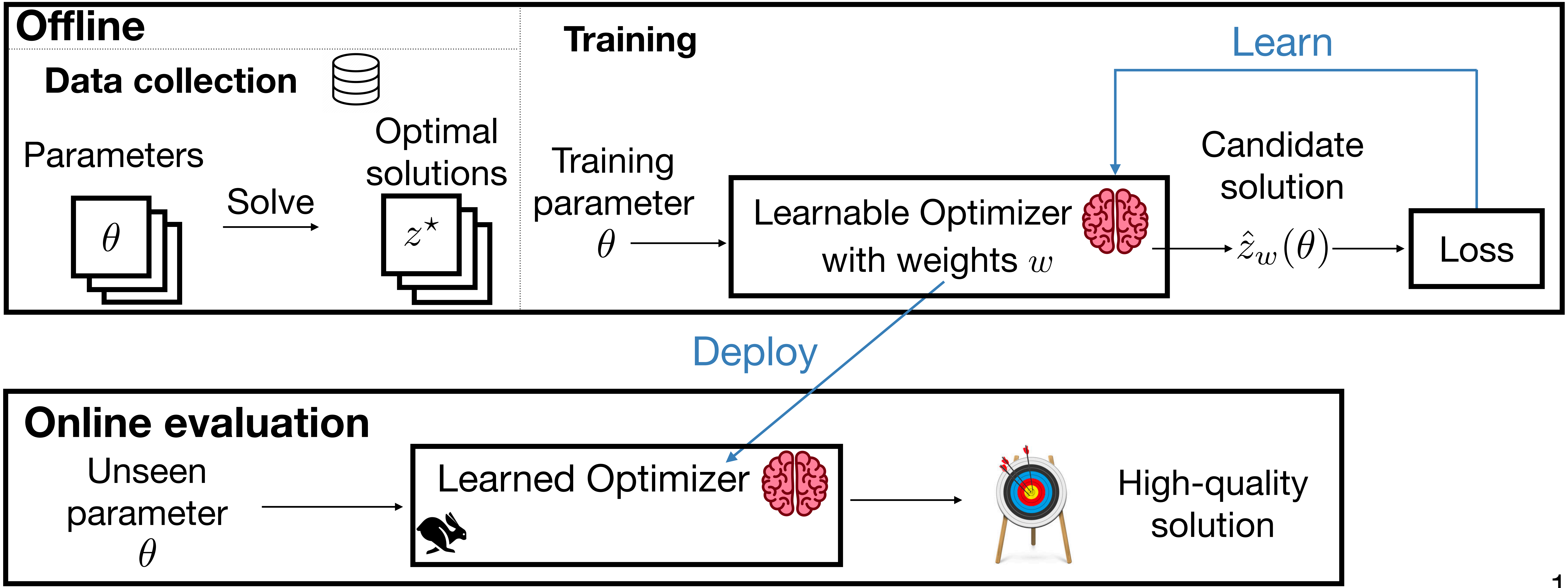
Parametric setting ✓

Uses machine learning to accelerate the optimizer

The learning to optimize paradigm

Goal: solve the parametric optimization problem fast

$$\begin{array}{ll} \text{minimize} & f_\theta(z) \\ \text{subject to} & g_\theta(z) \leq 0 \end{array}$$



Data-Driven Performance Guarantees for Classical and Learned Optimizers

Parametric setting ✓

Faster optimization methods

Empirical

Guarantees

Goal: endow learned optimizers with generalization guarantees

PAC-Bayes guarantees for learned optimizers

algorithm steps

tolerance

$$e_w(\theta) = \mathbf{1}(\ell_w^k(\theta) > \epsilon)$$

learnable weights

McAllester bound: given posterior and prior distributions [McAllester et. al 2003]
P and P_0, with probability $1-\delta$

$$\mathbf{E}_{\theta\sim\mathcal{X}}\mathbf{E}_{w\sim P}e_w(\theta) \leq \mathrm{KL}^{-1}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{E}_{w\sim P}e_w(\theta_i)\middle|\frac{1}{N}\left(\mathrm{KL}(\mathrm{P}\parallel\mathrm{P}_0)+\log(\mathrm{N}/\delta)\right)\right)$$

risk $\leq \mathrm{KL}^{-1}$ (empirical risk | regularizer)

Optimize the bounds directly

Learned algorithms for sparse coding

Noisy measurements
$\theta = b$

→

Sparse coding

Recover sparse $z^\star$ from $b = Dz^\star + \sigma$

→

Ground truth sparse signal
$z^\star$

D: dictionary, σ: noise

Standard technique

minimize $\|Dz - b\|_2^2 + \lambda \|z\|_1$

ISTA (iterative shrinkage thresholding algorithm)
(Classical optimizer)

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

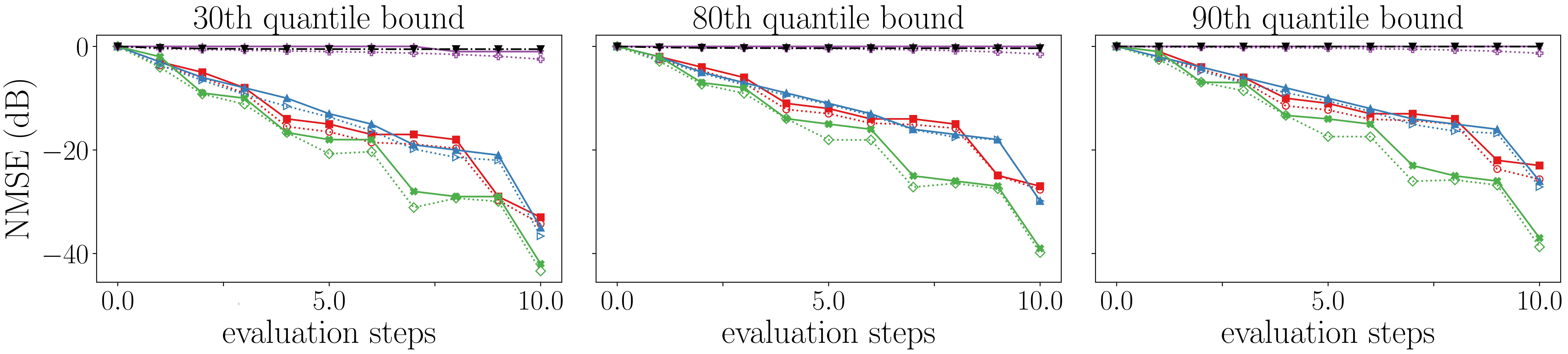
Learned ISTA
(Learned optimizer)

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

+ variants [Gregor and LeCun 2010, Liu et. al 2019]

$\text{soft threshold}_\psi(z) = \textbf{sign}(z)\max(0, |z| - \psi)$

Learned ISTA results for sparse coding



Conclusions

Real-world optimization is **parametric**

Data-driven methods can provide **guarantees** for classical and learned optimizers

Classical optimizers: apply a sample convergence bound

Learned optimizers: minimize the generalization bound directly

Data-Driven Performance Guarantees
for Classical and Learned Optimizers

To be on Arxiv soon!

rajivs@princeton.edu

rajivsambharya.github.io