Bayesian Optimization in O(N)

Speeding Automated Design and Discovery

David Sweet, Siddhant anand Jadhav // 20250514
Katz Faculty Research Initiative

Bayesian Optimization

- Engineering design
- Recommenders, ads
- Aerospace
- Pharmaceuticals
- Chemistry
- Materials

- Low-boom supersonic aircraft wing
- TikTok, Instagram, Spotify, etc.
- New catalyst, hydrogen fuel
- Better Skin Absorption
- CO_2 capture process
- Fast charging EV batteries
- Self-driving labs: BO + Robotics

Timescales

- Days-weeks: Website A/B testing
- Hours - day: Material synthesis
- 10's of seconds - minutes: Robotics simulation, industrial process simulation

Many Observations

- Observation = Design + Evaluation
- Short time scales + computer clusters ==> **Many** observations
  - Example:
    - 100 observations / [10 minute round] on a cluster
    - x 100 rounds = 10,000 observations

Dimensionality

- Parameters: weights, thresholds, timeouts, lengths, materials, chemicals
  - “parameter” == “dimension” in Bayesian optimization
- More parameters ==> More designs to evaluate
- High dimensionality?
  - ==> Need many observations
- Have capacity for many observations?
  - ==> Can investigate high dimensions

Chicken, Egg

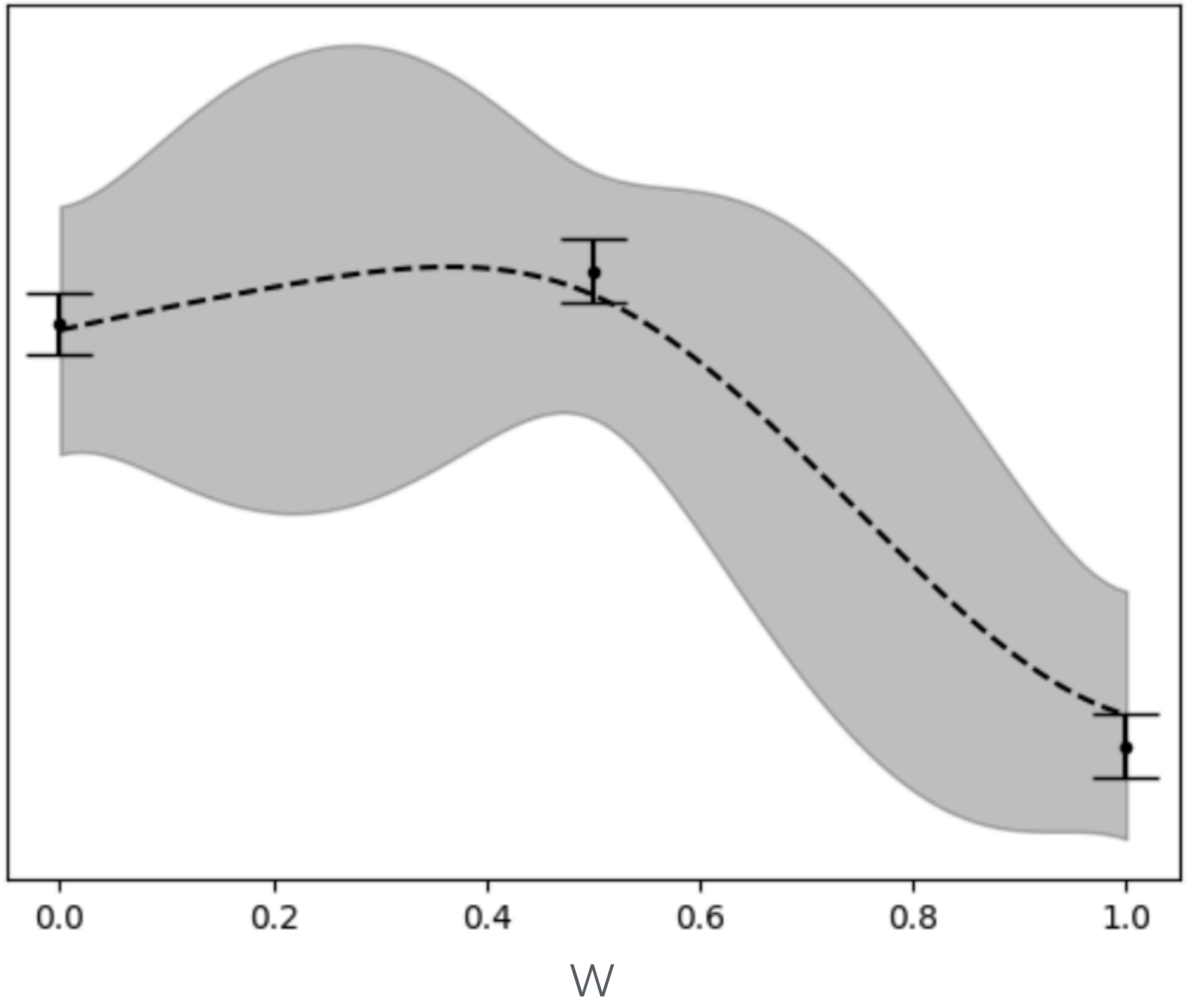
Bayesian Optimization Algorithm

- Observations: $\mathscr{D} = (x_i, y_i)$
- Fit a surrogate to $\mathscr{D}$: $y(x) \sim \mathcal{N}(\mu(x), \sigma^2(x))$
- Optimize surrogate: $x_{arm} = \arg\max \alpha(\mu(x), \sigma(x))$
  - $\alpha(\cdot\,,\cdot)$ is *acquisition function*
- Evaluate x_{arm} via experiment or simulation

Repeat until satisfied with evaluation

Poor Scaling

- Gaussian process (GP), usual surrogate
- Epistemic uncertainty: Knows what it knows
- Excellent when few observations, N
    - Query takes $O(N^2)$
- Slow when N is large, many observations

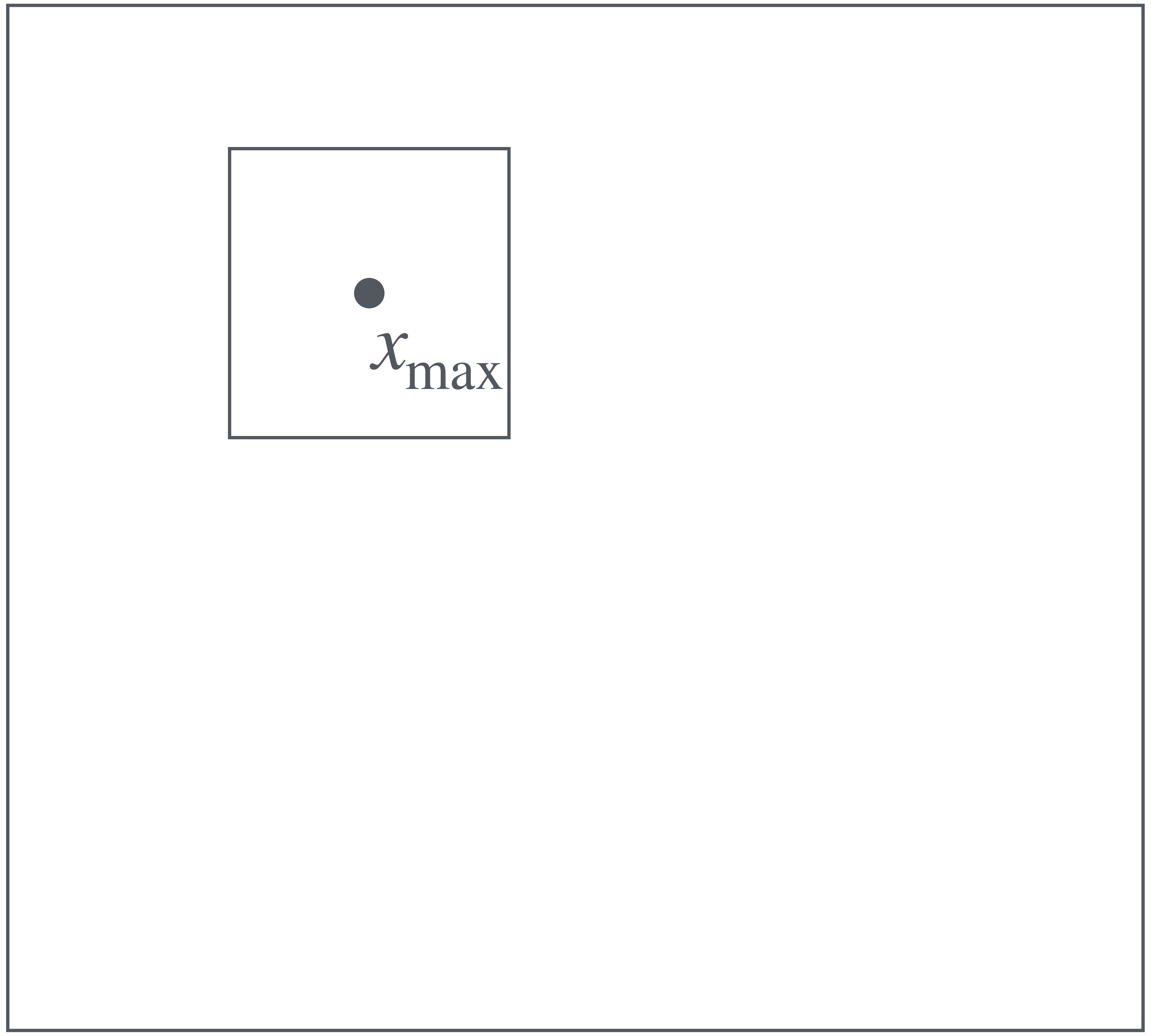


Scaling to Many Observations

- BOMO: Bayesian Optimization with Many Observations
- Alternative surrogates:
  - Random forest, neural networks: $O(N)$, but fit is complex
  - Parzen estimator (Optuna): $O(N)$

Scaling to Many Observations

- TuRBO: Restrict to trust region
  - Need fewer queries for $\mathbf{arg\,max}$
  - Maybe fewer queries for fitting as well
  - Still $O(N^2)$



Epistemic Nearest Neighbors (ENN)

- Find K nearest-neighbor observations to x, (x_i, y_i), set

$$\mu(c \mid x_i) = y_i, \quad \sigma^2(x \mid x_i) = \|(x - x_i\|^2$$

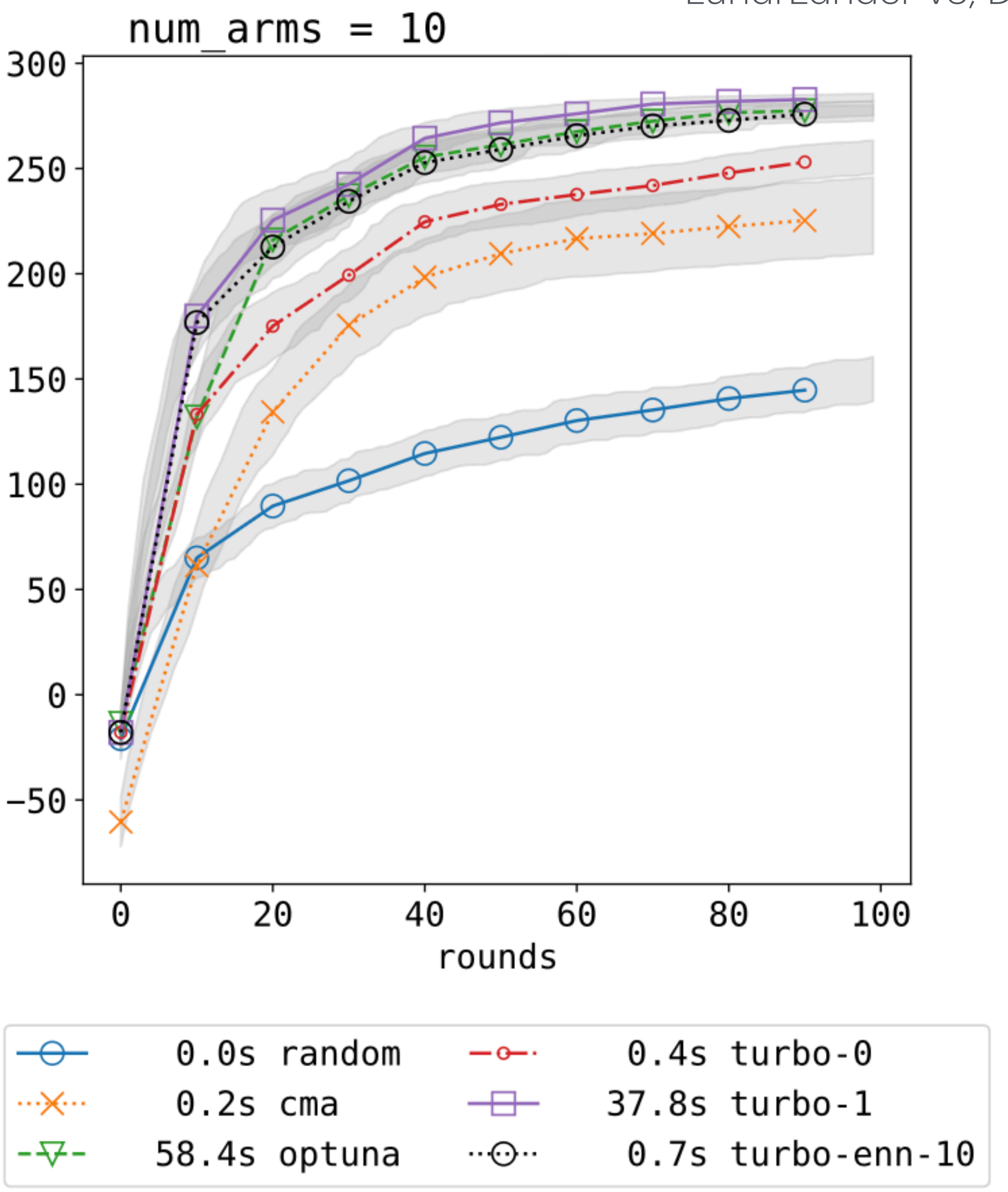
- MVUE, assuming (asserting) independence of observations

$$\boxed{\mu(x) = \frac{\sum_i^K \sigma^{-2}(x \mid x_i)\mu(x \mid x_i, y_i)}{\sum_i^K \sigma^{-2}(x \mid x_i)} \qquad \sigma^2(x) = \frac{1}{\sum_i^K \sigma^{-2}(x \mid x_i)}}$$

TuRBO-ENN

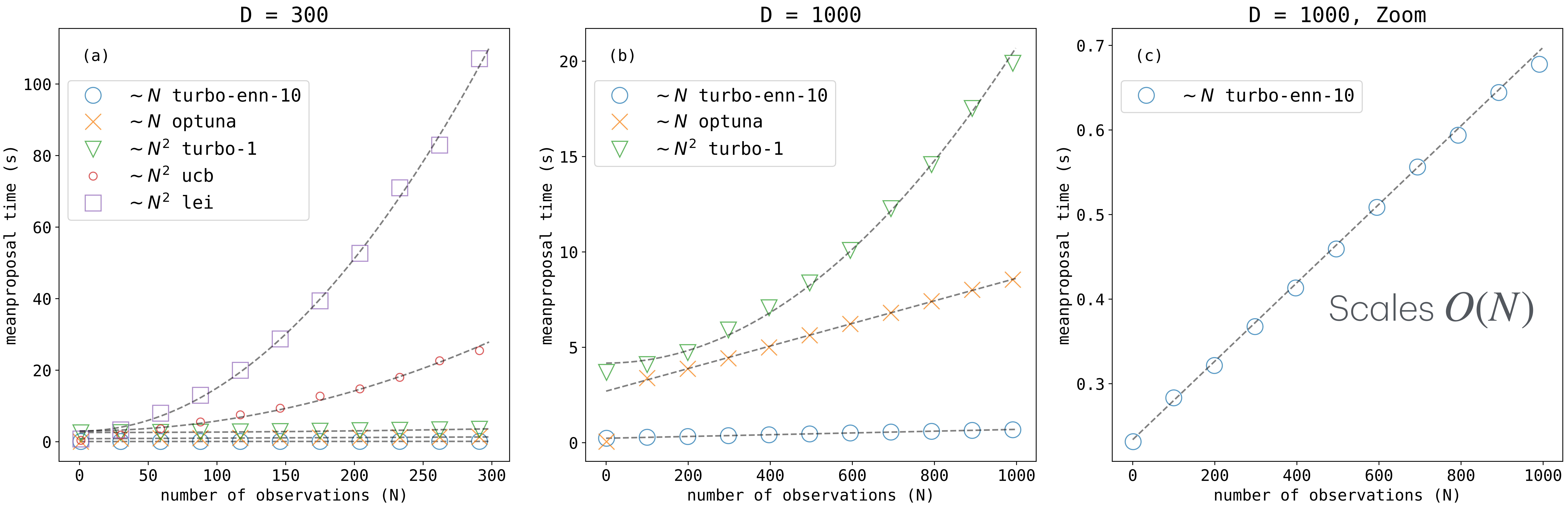
- Compared to TuRBO
  - Speedup 50x (= 37.8s / 0.7s)
  - Comparable quality designs
- $N = 1000$

LunarLander-v3, D= 12



TuRBO-ENN

Speedup increases w/N



TuRBO-ENN

Method	Surrogate	Acquisition	Cost / proposal	RL Score	RL Time
`random`	none	Uniform sample	$O(1)$	$0.079 \pm .018$	0.00023
CMA–ES	none	Gaussian sample	$O(1)$	—	—
`optuna`	Parzen KDE	Modified EI	$O(N)$	—	–
`lei`	GP	Log-EI	$O(N^2)$	—	—
`ucb`	GP	UCB	$O(N^2)$	—	—
`turbo-1`	GP	Thompson sample trust region	$O(N^2)$	0.35 ± 0.017	1.0
`turbo-0`	none	Uniform sample trust region	$O(1)$	$0.25 \pm .008$	0.008
`turbo-enn-10`	ENN ($K = 10$)	Pareto(μ,σ) front	$O(N)$	$0.32 \pm .005$	0.014

Nine simulators

Works

Fast

Summary

- TuRBO-ENN
- Designs 10 - 100X faster without loss of quality
- $O(N)$ queries
- No fitting step

Taking the GP Out of the Loop

David Sweet
Department of Computer Science
Yeshiva University
New York, NY 10033
david.sweet@yu.edu

Siddhant anand Jadhav
Department of Data Science and Visualization
Yeshiva University
New York, NY 10033
sjadhav1@yu.edu