Learning for Decision-Making under Uncertainty

Bartolomeo Stellato

Department of Operations Research and Financial Engineering

Department of Electrical and Computer Engineering

Department of Computer Science

PRINCETON UNIVERSITY

Robust Optimization Webinar — May 31 2024

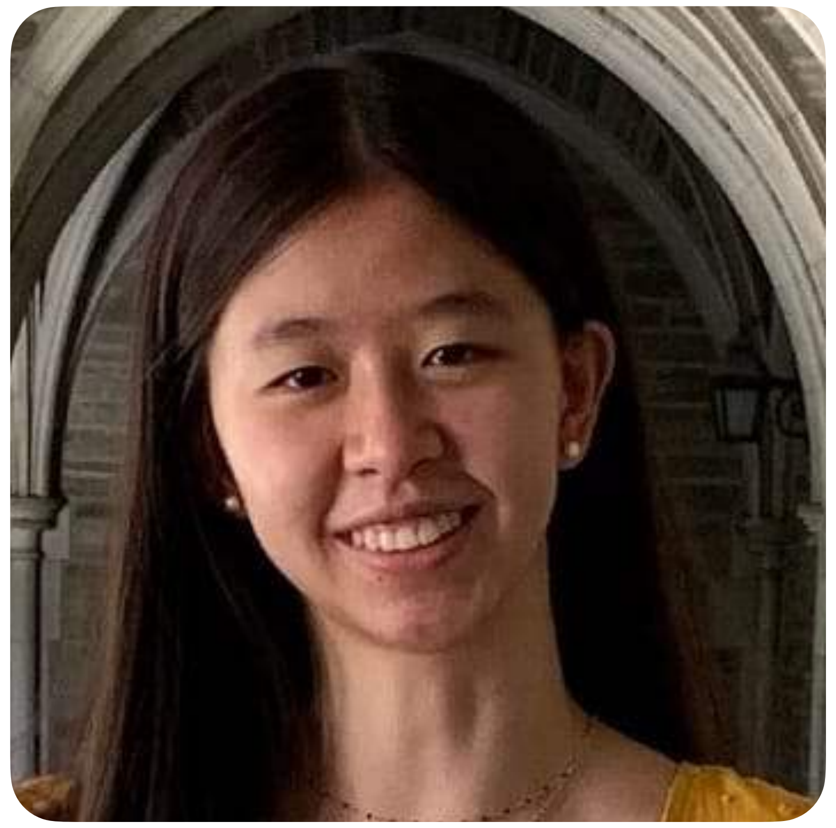
Joint work with

Irina Wang
Princeton ORFE

Bart Van Parys
MIT

Amit Solomon
Princeton OIT

Cole Becker
MIT

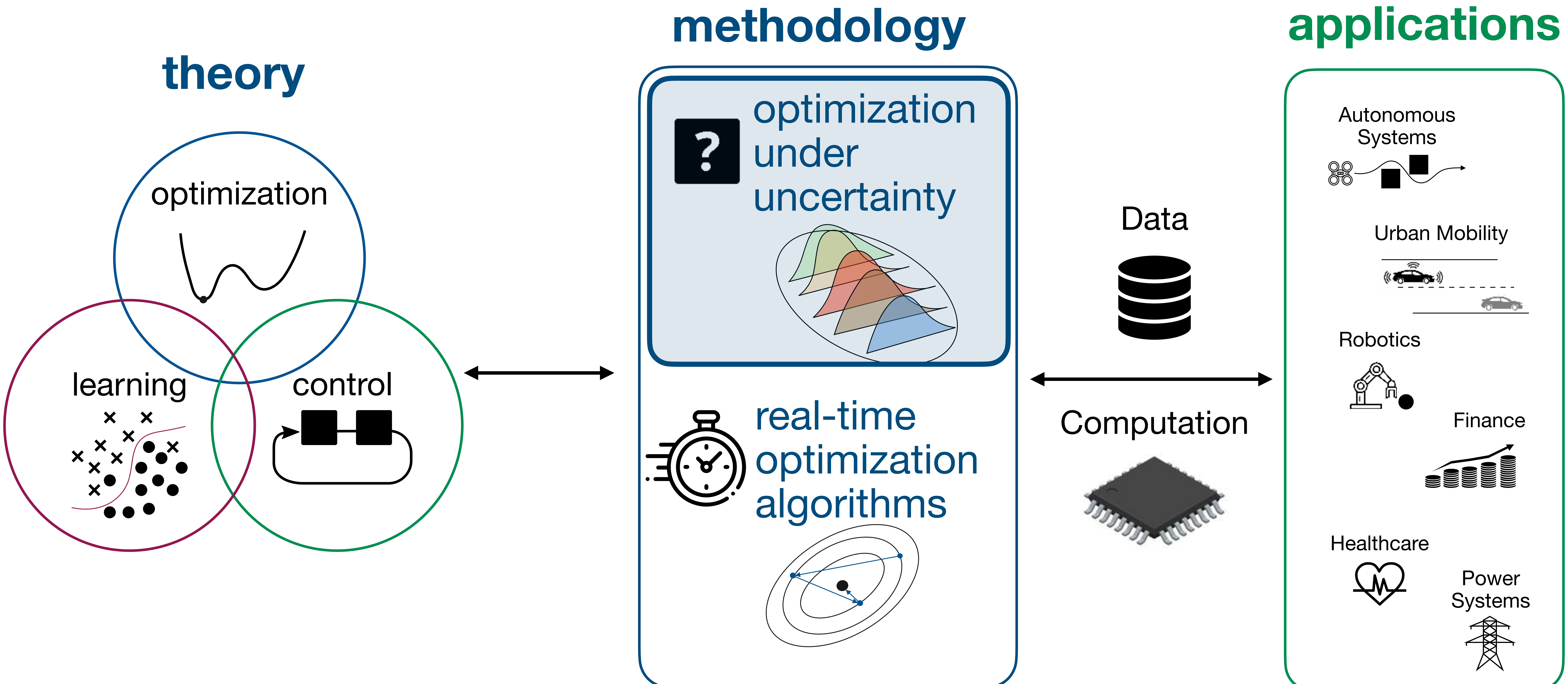


Annie Liang
Princeton ORFE




Most applications require fast and safe decisions in presence of uncertainty

My research in a 🥜shell



Problem setup with uncertain constraints

optimization
variable

$$\begin{array}{ll} \text{minimize} & f(\underbrace{x}_{}) \\ \text{subject to} & g(x, u) \leq 0 \end{array}$$

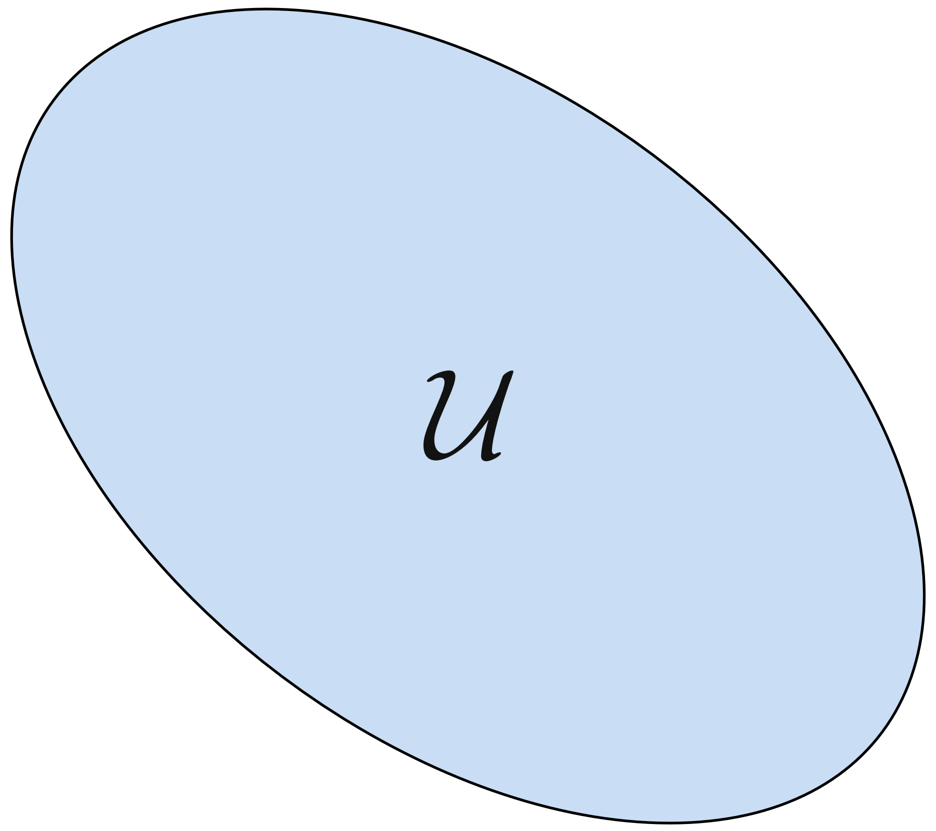
uncertain
parameter

How do we guarantee constraint satisfaction?

Robust optimization recipe

Recipe

1. Pick uncertainty set $\mathcal{U}$
2. Ensure constraint satisfaction $\forall u \in \mathcal{U}$

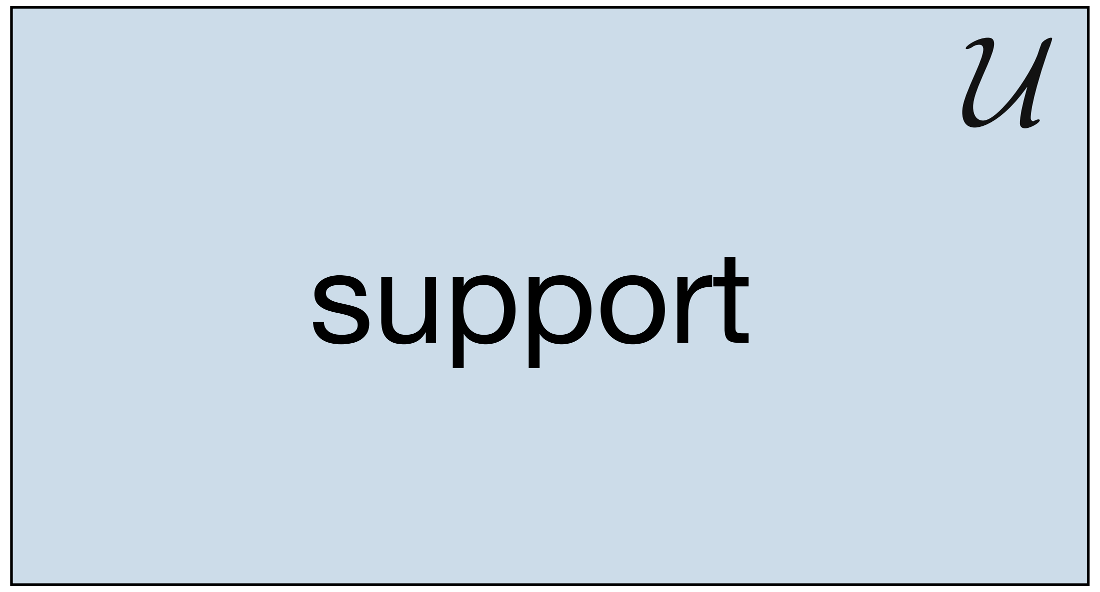


$$
\begin{array}{ll}
\text{minimize} & f(x) \\
\text{subject to} & g(x,u) \leq 0, \quad \forall u \in \mathcal{U}
\end{array}
$$

How do we pick the uncertainty set?

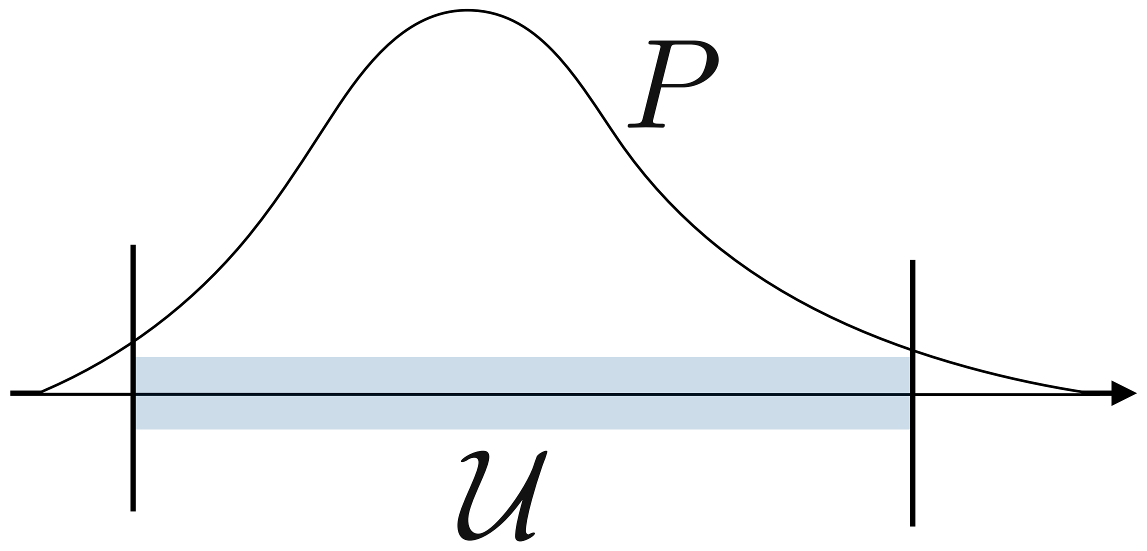
Picking the uncertainty set is difficult

Worst-case approach



❌ Very conservative

Probabilistic approach



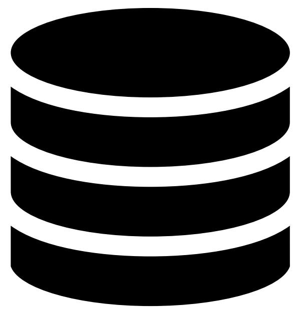
❌ nobody knows P

Data-driven approach

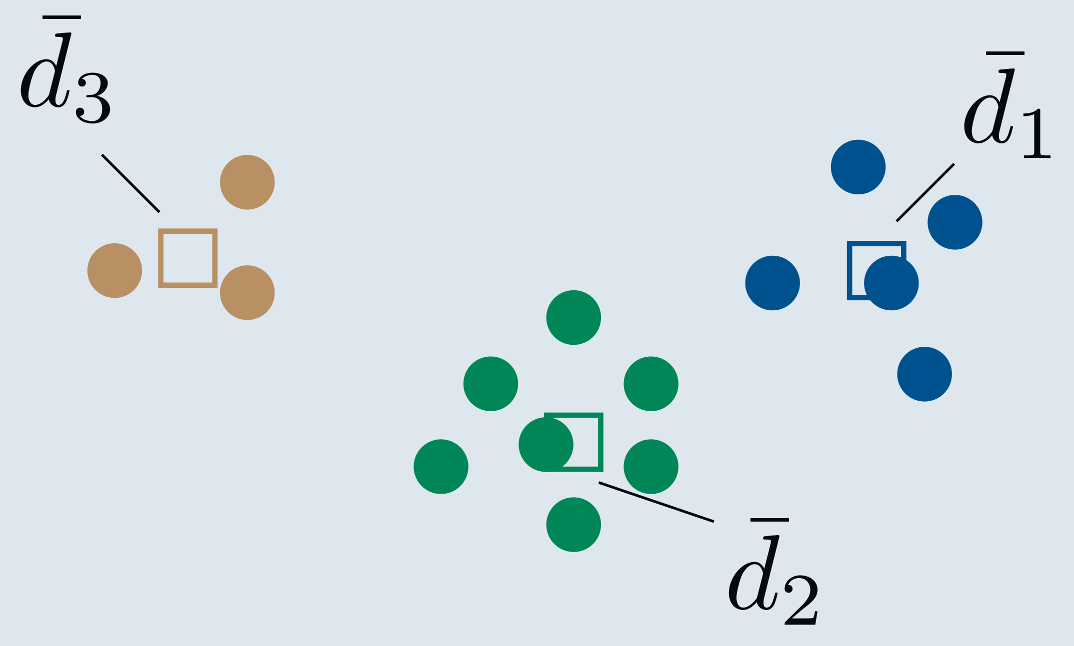
Can we use data to construct uncertainty sets?

Learning for Decision-Making under Uncertainty

How can we use data to build
tractable and **high-performance**
uncertainty sets?



Clustering



Differentiable Optimization

optimization layer

$y \longrightarrow$

argmin	$f(x, y)$
subject to	$g(x, u, y) \leq 0 \quad \forall u \in \mathcal{U}(\theta)$

$\longrightarrow x(\theta, y)$

Finite sample probabilistic guarantees in expectation

$$\mathbf{E}(g(x,u)) \leq 0$$

$$u \sim P$$

(but we never know P!)

Data

$$\mathcal{D} = \{d_i\}_{i=1}^{N}$$

Data-driven probabilistic guarantees

$$\boxed{\mathbf{P}^N} \left(\mathbf{E}(g(\boxed{\hat{x}_N}, u)) \leq 0\right) \geq \boxed{1-\beta}$$

Product Distribution

data-driven solution

probability of constraint satisfaction

The empirical distribution can help us

Data-Driven Distributionally Robust Optimization

Data

$$\mathcal{D} = \{d_i\}_{i=1}^{N}$$

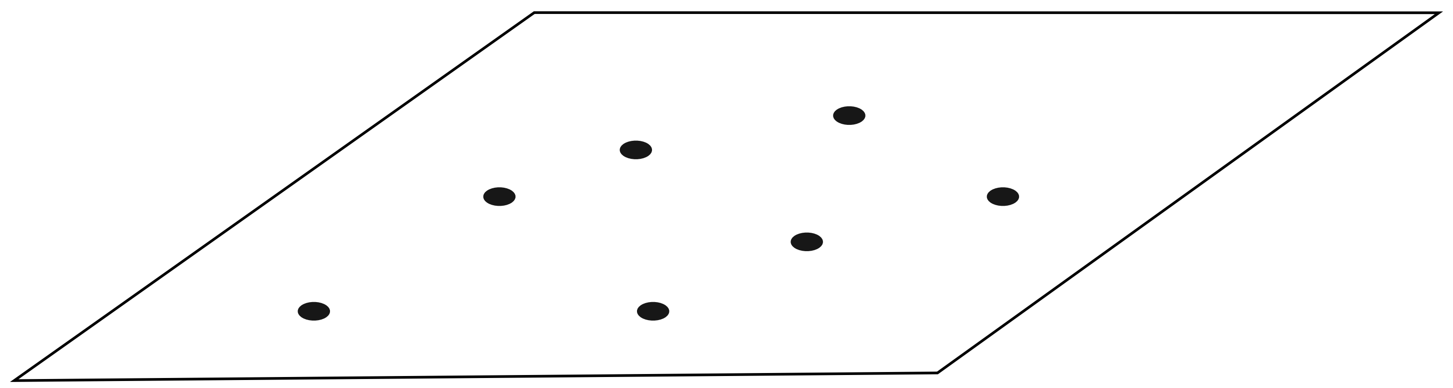
Empirical Distribution

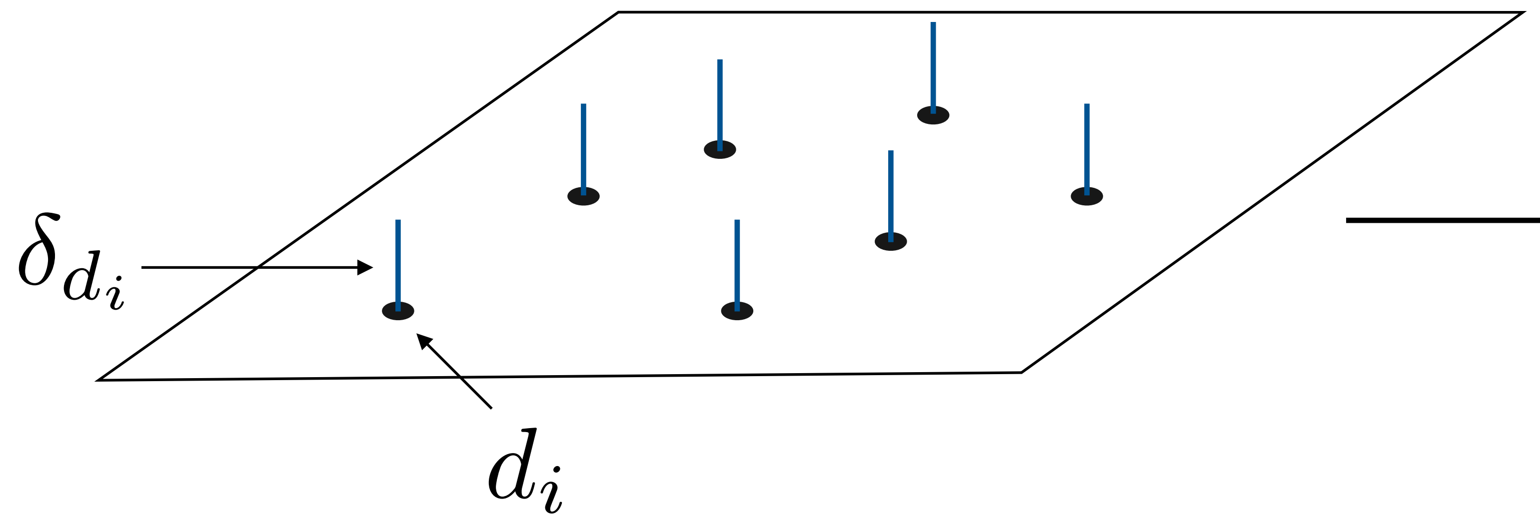
$$\hat{\mathbf{P}}^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{d_i}$$

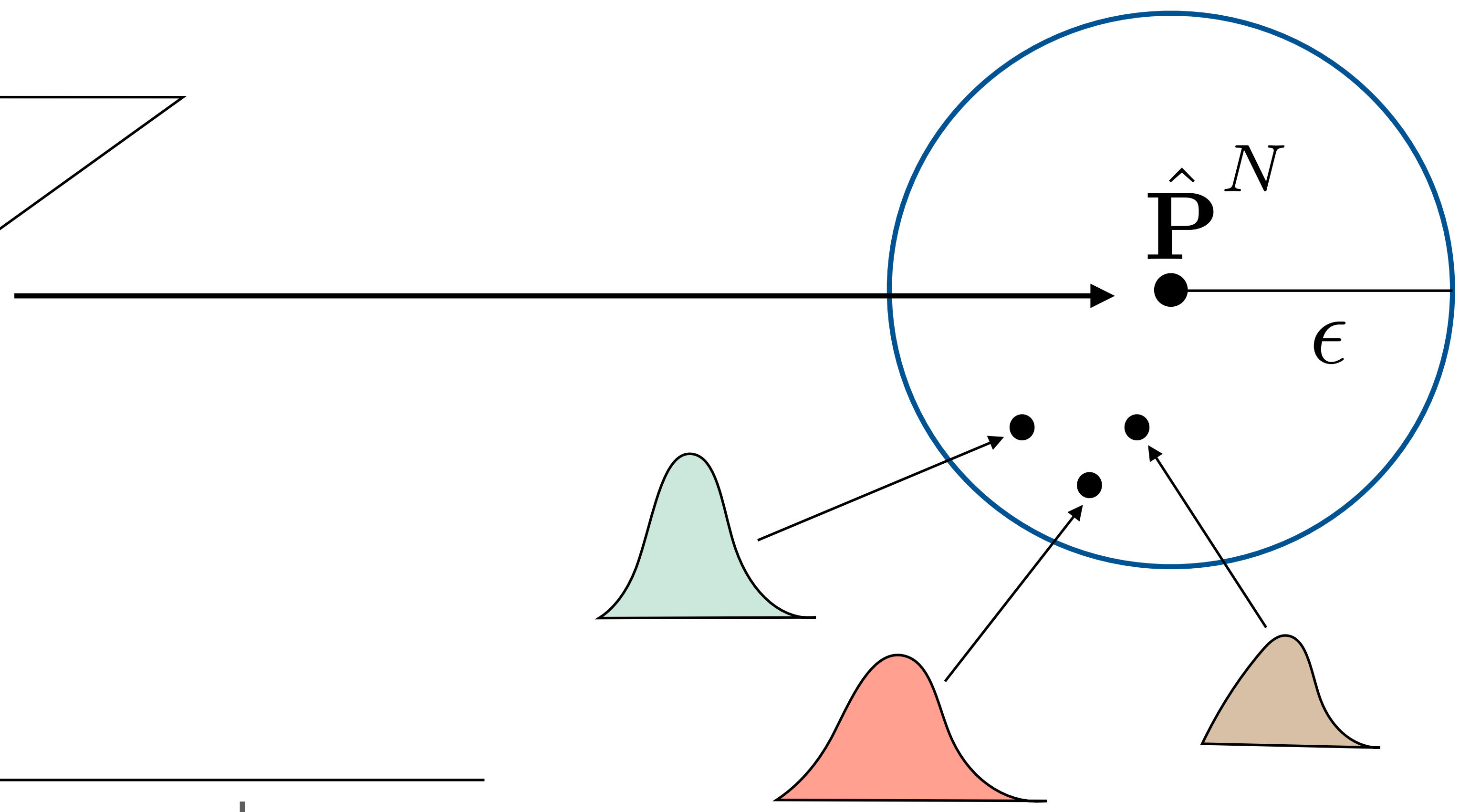
Ambiguity Set

$$\mathcal{P}_N = \left\{ W_p(\hat{\mathbf{P}}^N, \mathbf{P}) \leq \epsilon \right\}$$

Wasserstein distance

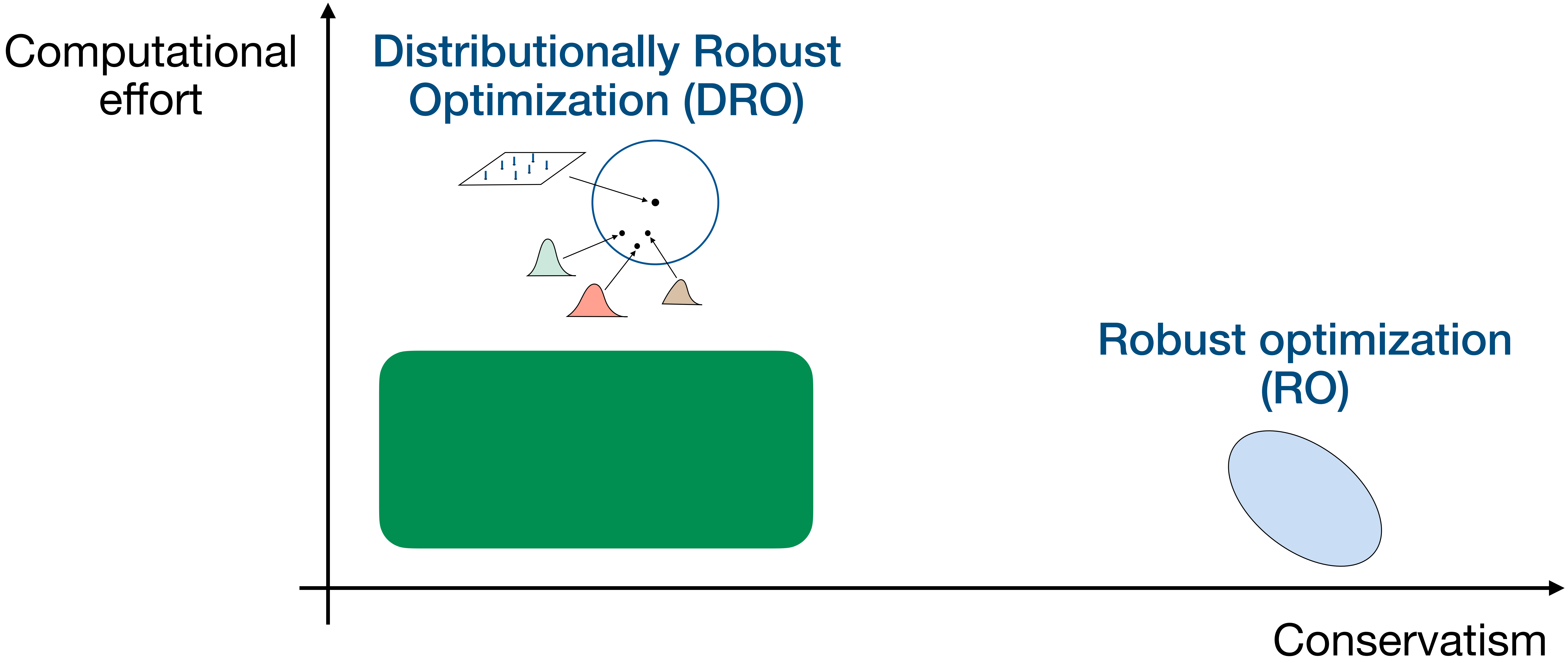






Data-driven Wasserstein DRO. D. Kuhn, P. Mohajerin Esfahani, S. Shafieezadeh Abadeh, W. Wiesemann, and many more
Optimal Transport. G. Monge, L. Kantorovic, L. Wasserstein, C. Villani, …and many more

Robust vs Distributionally Robust Optimization

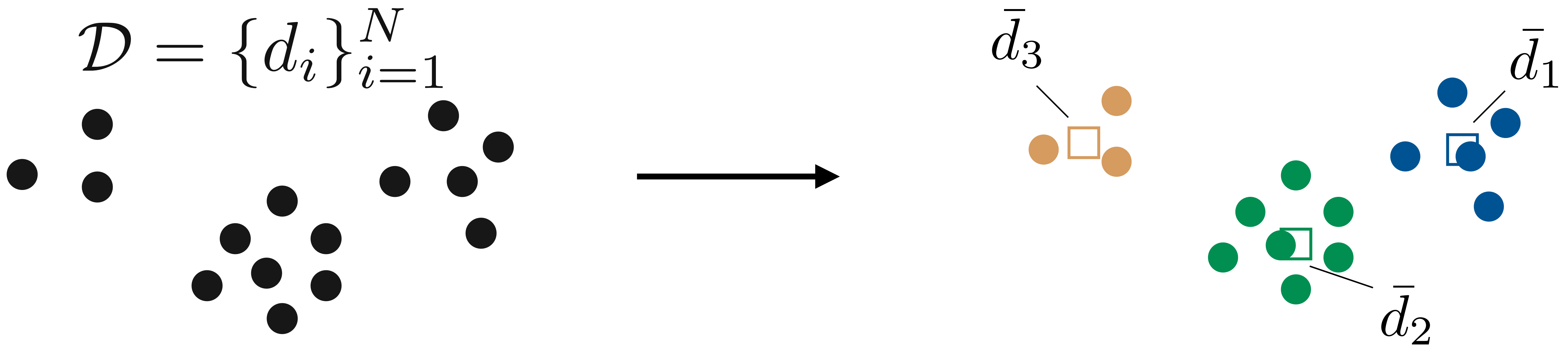


Can we get the best of both worlds?

Clustering reduces dimensionality and computation time

Data

$\mathcal{D} = \{d_i\}_{i=1}^N$



$$\text{minimize} \quad \sum_{k=1}^{K} \sum_{i \in C_k} \|d_i - \bar{d}_k\|^2$$

cluster centers

Main idea

Use cluster centers instead of original data

Our approach: Mean Robust Optimization (MRO)

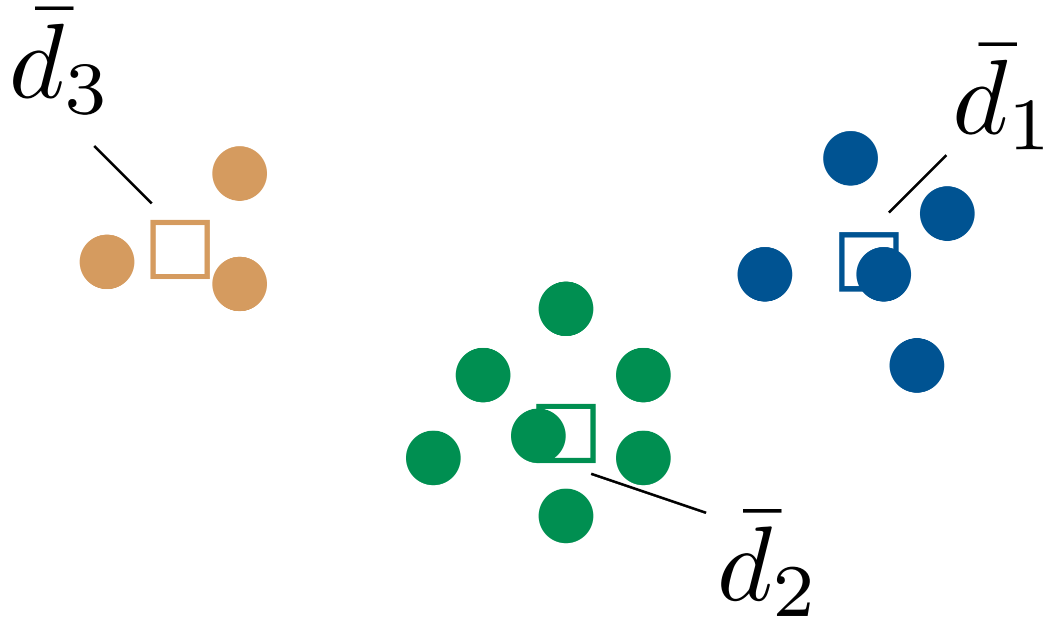
Data

$\mathcal{D}_N \longrightarrow$ **Clustering** $\longrightarrow$ **Mean Robust Optimization Problem** $\longrightarrow \hat{x}_N$

Uncertainty Set

$\mathcal{U}(K, \epsilon)$

Data-driven solution



$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & \bar{g}(x,u) \leq 0 \quad \forall u \in \mathcal{U}(K,\epsilon) \end{array}$$

constraint function

uncertainty set

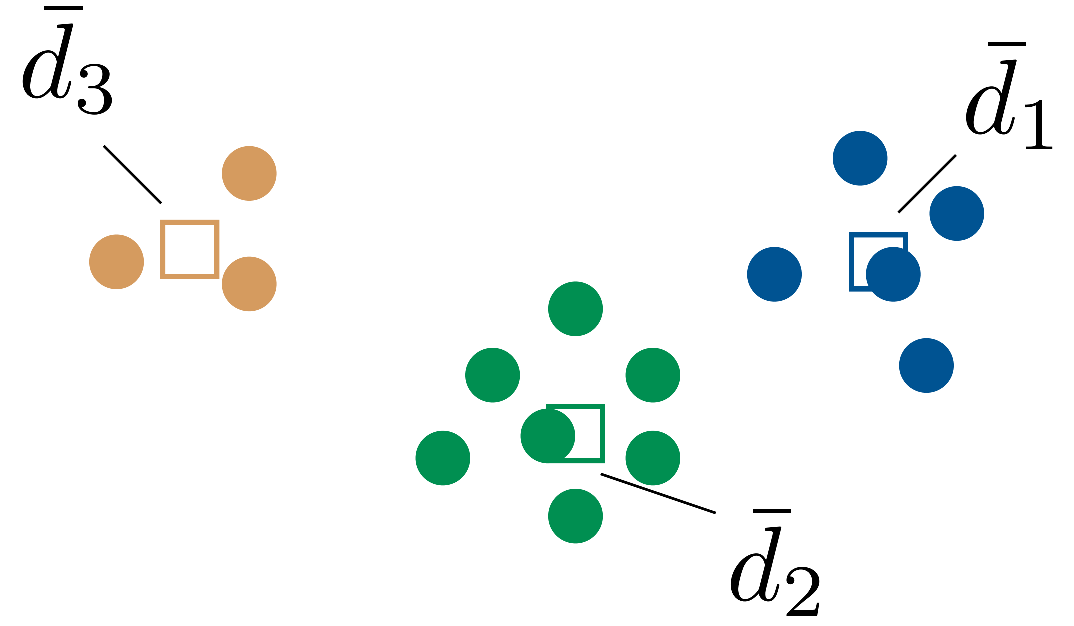
Uncertainty set

cluster weights

order

$$\mathcal{U}(K,\epsilon) = \left\{ u = (v_1, \dots, v_K) \;\middle|\; \sum_{k=1}^{K} w_k \|v_k - \bar{d}_k\|^p \leq \epsilon^p \right\}$$

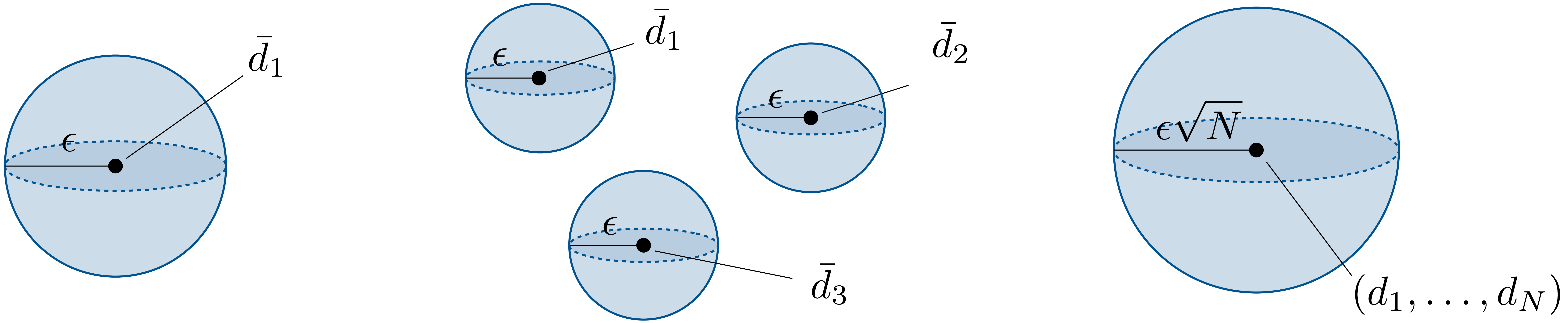
cluster centers



Examples

K

$K = 1$ | $K = 3, p = \infty$ | $K = N, p = 2$



Back to the Mean Robust Optimization problem

uncertainty set

$$\left\{ \sum_{k=1}^{K} w_k \|v_k - \bar{d}_k\|^p \leq \epsilon^p \right\}$$

Uncertain variable lifting

$u = (v_1, \ldots, v_K)$

$\longrightarrow$

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & \bar{g}(x,u) \leq 0 \quad \forall u \in \mathcal{U}(K,\epsilon) \end{array}$$

constraint function

$$\sum_{k=1}^{K} w_k g(x, v_k)$$

Solving the MRO problem

dualize constraint

$$\bar{g}(x,u) \le 0, \ \forall u \in \mathcal{U}(K,\epsilon)$$

minimize $\quad f(x)$

subject to $\quad \sum_{k=1}^{K} w_k s_k \le 0$

$$[-g]^*(x, z_k) - z_k^T \bar{d}_k + \phi(p)\lambda \|z_k/\lambda\|_*^{p/(p-1)} + \lambda \epsilon^p \le s_k, \quad k = 1, \dots, K$$

$$\lambda \ge 0$$

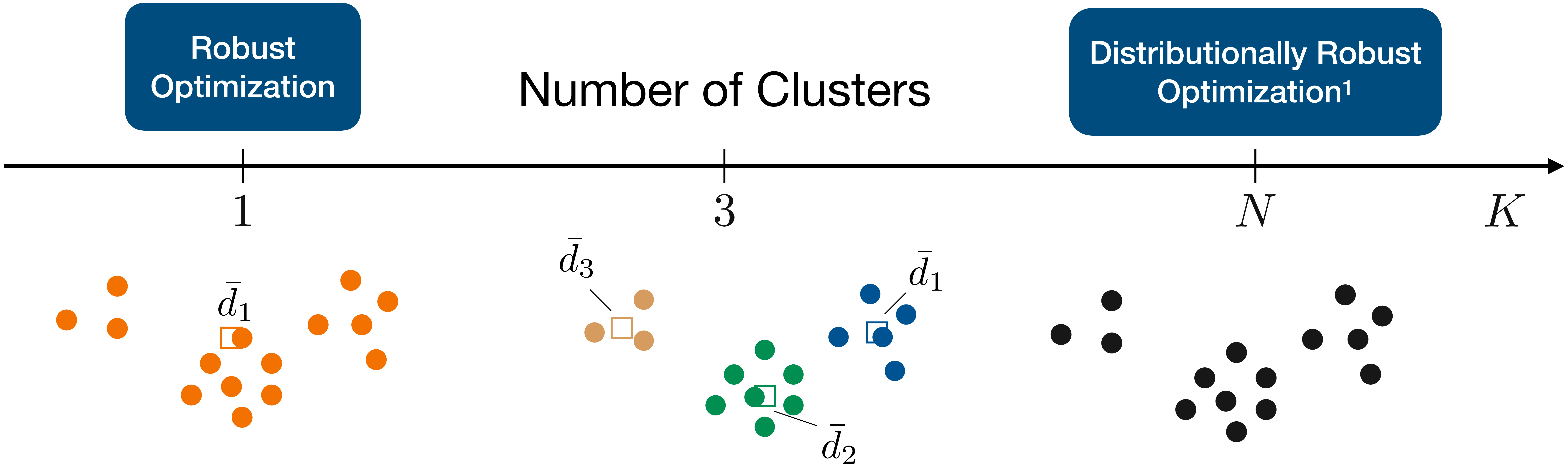
conjugate function

cluster centers

function of $p \ge 1$
$\phi(p) \to 1$ as $p \to \infty$
$\phi(1) = 0$

It can be very expensive when K is large (e.g., $K = N$)

MRO bridges RO and DRO



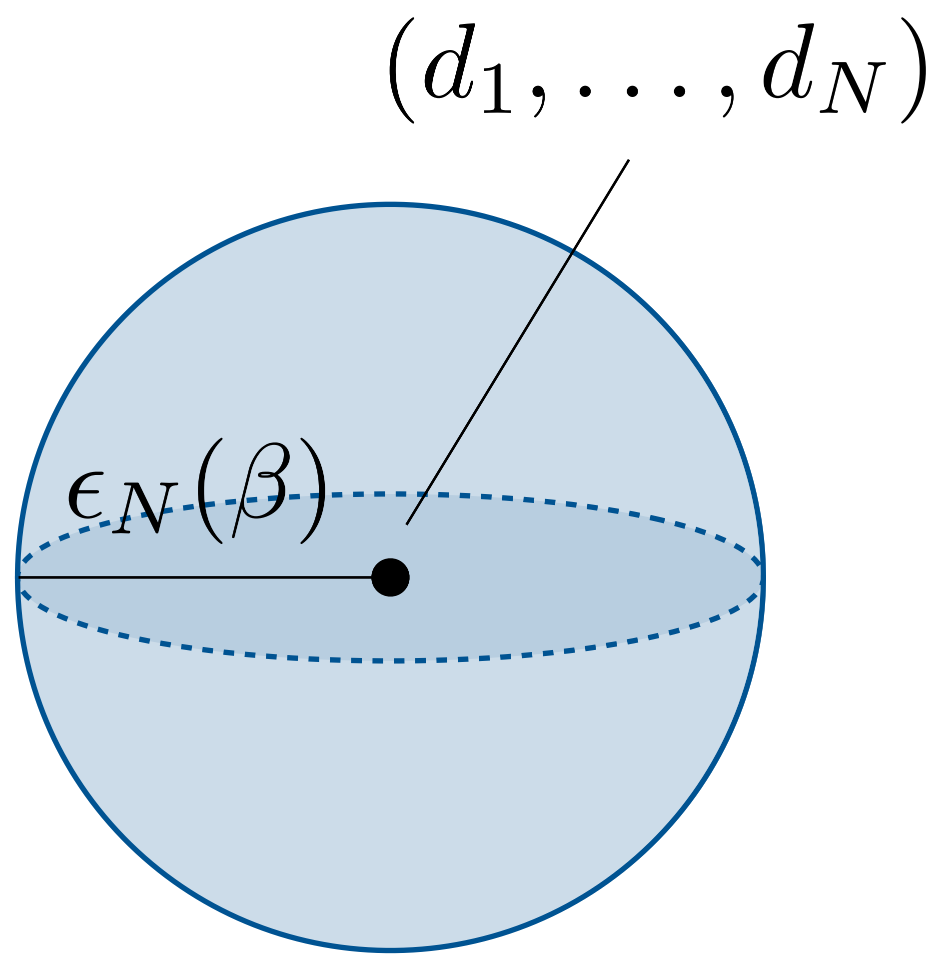
1. D Kuhn, P M Esfahani, V A Nguyen, and S Shafieezadeh-Abadeh, "Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning"

Probabilistic guarantees in MRO

probability of constraint satisfaction

uncertainty set radius

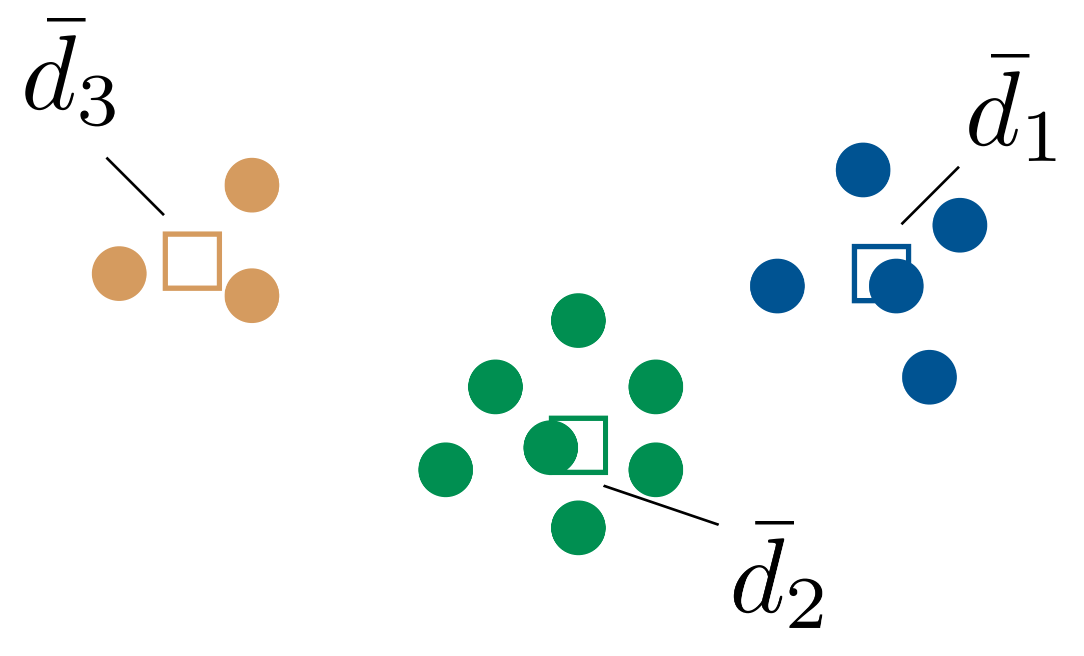
$$\mathbf{P}^N \left(\mathbf{E}(g(\hat{x}_N, u)) \leq 0\right) \geq 1 - \beta \xrightarrow{\text{light-tailed}} \mathcal{U}(N, \epsilon_N(\beta))$$



MRO clustering

$$\mathcal{U}(K, \epsilon_N(\beta) + \eta_N(K))$$

$$\frac{1}{N} \sum_{k=1}^{K} \sum_{d_i \in C_k} \|d_i - \bar{d}_k\|^p$$



Quite conservative bounds… can we do better?

Bounding the conservatism

Worst-case values

$$\bar{g}^{N}(x) = \underset{u \in \mathcal{U}(N,\epsilon)}{\text{maximize}} \quad \bar{g}(x,u)$$

$$\bar{g}^{K}(x) = \underset{u \in \mathcal{U}(K,\epsilon)}{\text{maximize}} \quad \bar{g}(x,u)$$

MRO constraint

$$\bar{g}(x,u) \le 0 \quad \forall u \in \mathcal{U}(K,\epsilon)$$

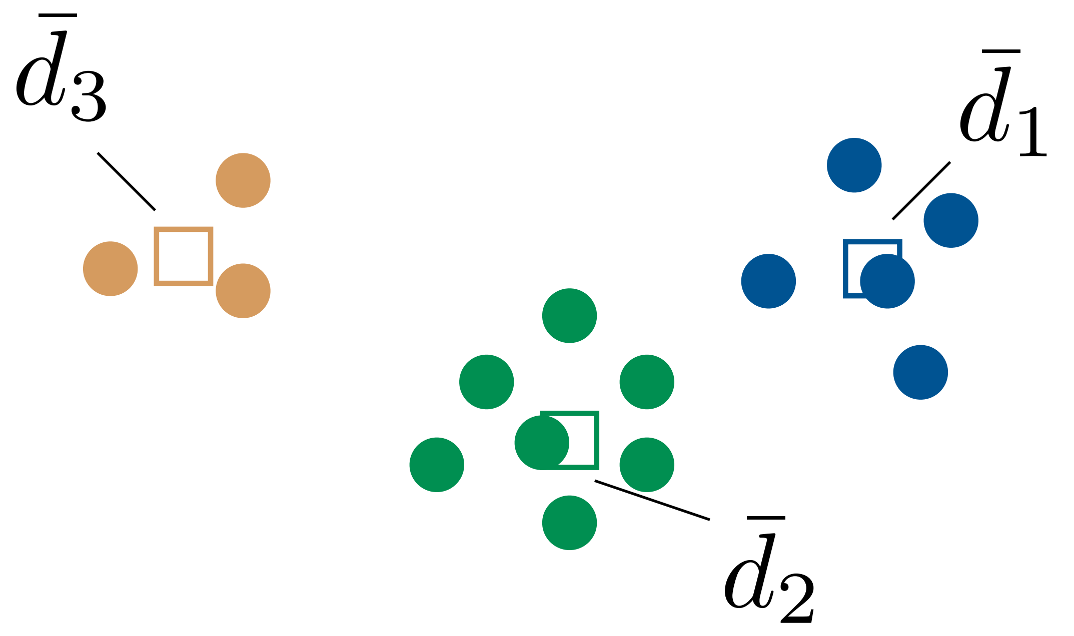
Theorem

If $-g$ is L-smooth in u, we have

$$\bar{g}^{N}(x) \le \bar{g}^{K}(x) \le \bar{g}^{N}(x) + \frac{L}{2} D(K)$$

clustering objective

$$\min \ \frac{1}{N} \sum_{k=1}^{K} \sum_{d_i \in C_k} \|d_i - \bar{d}_k\|^2$$



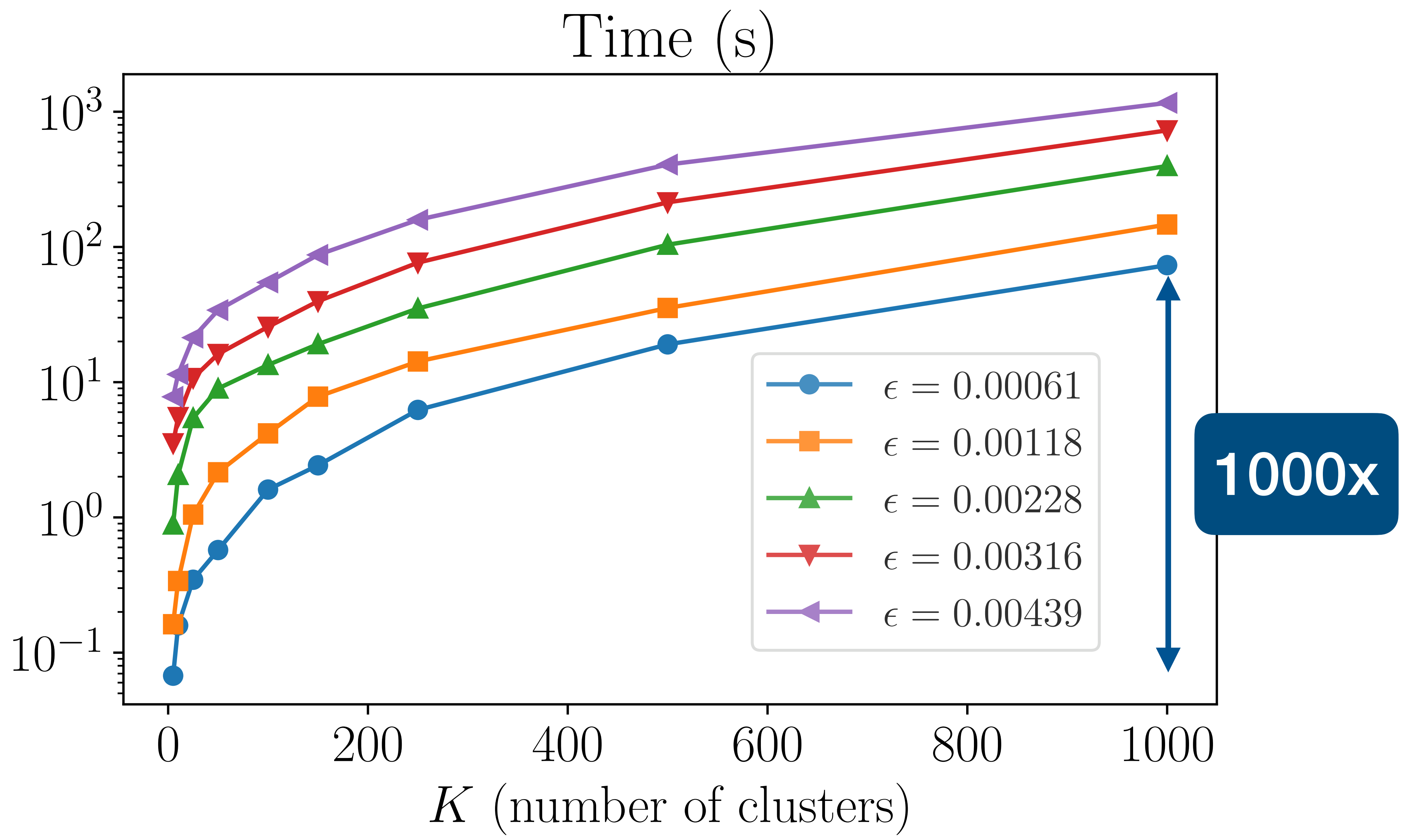
When g is affine in u ($L = 0$), clustering makes no difference to the optimal value or optimal solution

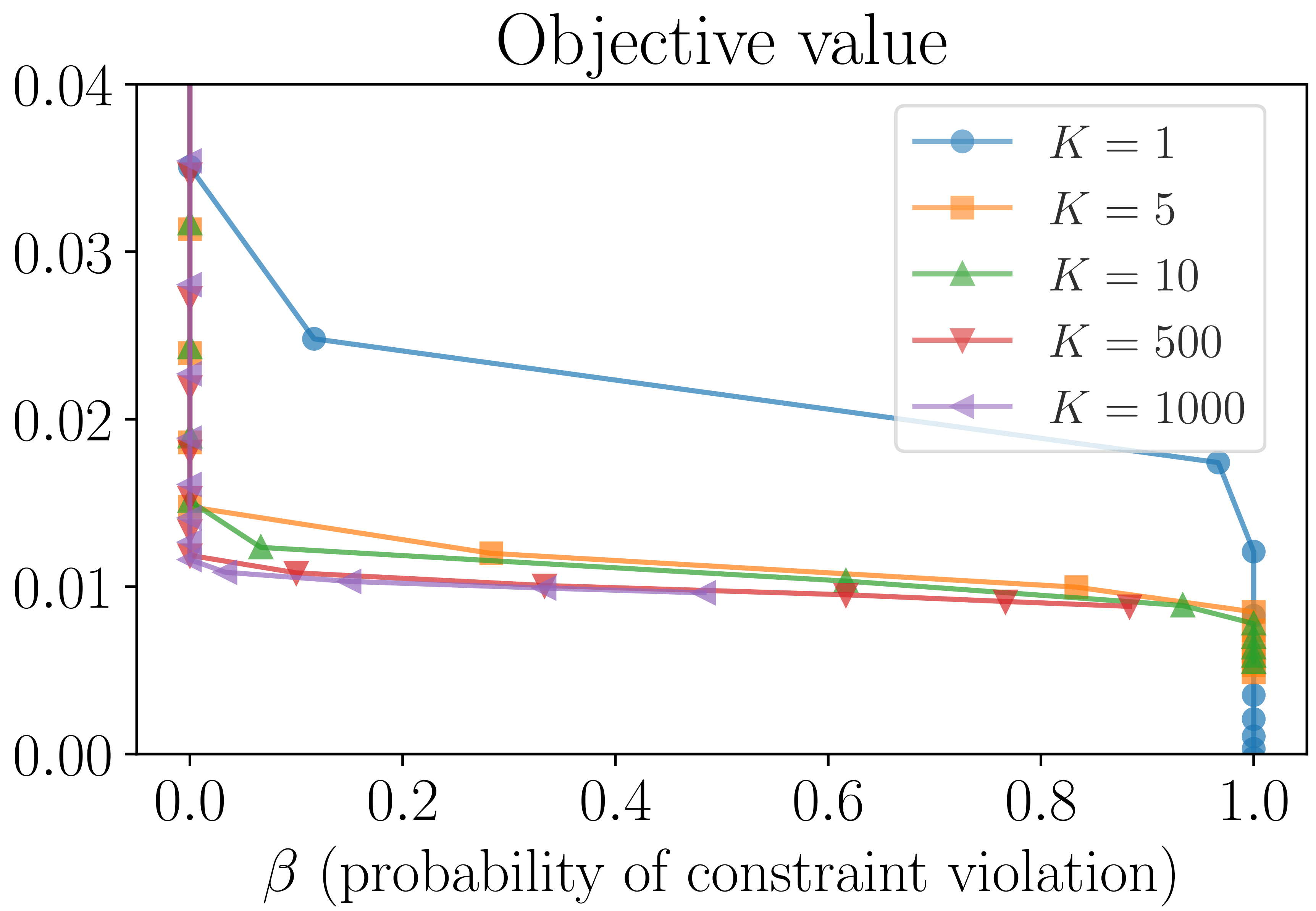
Computational speedups on sparse portfolio optimization

$$\begin{array}{ll}\text{minimize} & \mathbf{CVaR}(-u^T x, \eta) \\ \text{subject to} & \mathbf{1}^T x = 1, \quad x \geq 0 \\ & \mathbf{card}(x) \leq C\end{array}$$

conditional value-at-risk

cardinality constraint





near-optimal performance with 5 clusters

Mean Robust Optimization

- Bridge RO and DRO

- Clustering effect
  - g affine in u ⟶ zero clustering effect!
  - g concave in u ⟶ performance bound

- Multiple orders of magnitude speedups

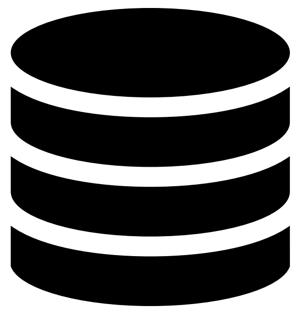
https://github.com/stellatogrp/mro_experiments

Mean Robust Optimization
I. Wang, C. Becker, B. Van Parys, and B. Stellato
arxiv.org: 2207.10820, 2023

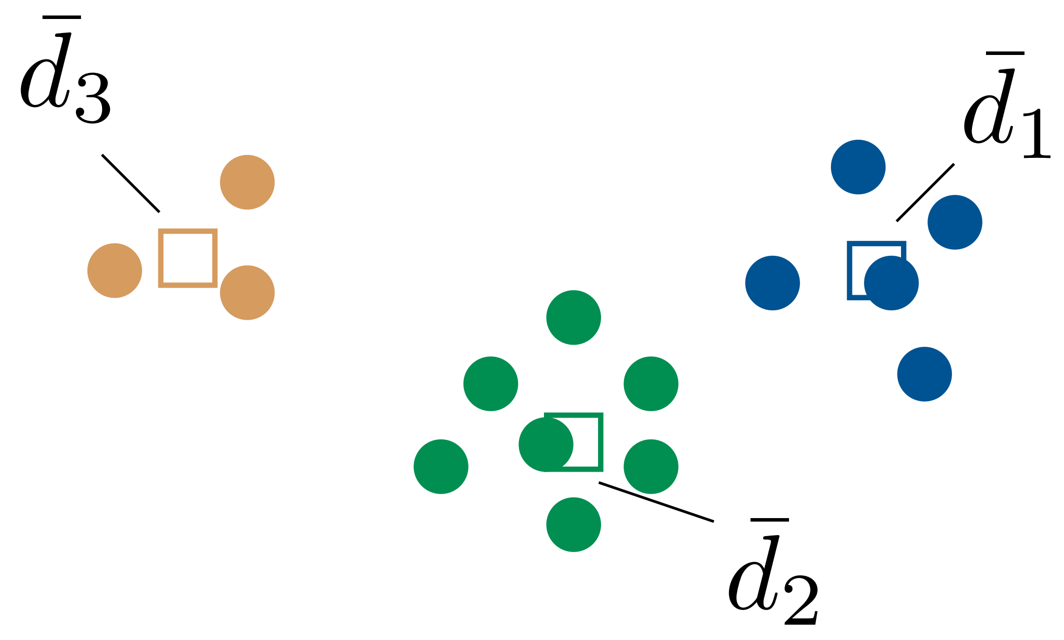
INFORMS Computing Society Student Paper Award

Learning for Decision-Making under Uncertainty

How can we use data to build
tractable and **high-performance**
uncertainty sets?



Clustering



Differentiable Optimization

optimization layer

$$y \longrightarrow \begin{array}{ll} \text{argmin} & f(x,y) \\ \text{subject to} & g(x,u,y) \leq 0 \quad \forall u \in \mathcal{U}(\theta) \end{array} \longrightarrow x(\theta, y)$$

Newsvendor problem

uncertain demand

$$\log(u) \sim \mathcal{N}(\mu, \Sigma)$$

$$\mu = \begin{bmatrix} 1.1 \\ 1.7 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 0.6 & -0.3 \\ -0.3 & 0.1 \end{bmatrix}$$

sell prices

objective

order prices

$$k^T x - p^T \min\{x, u\}$$

order decisions

uncertainty set

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & k^T x - p^T \min\{x, u\} \le t \quad \forall u \in \mathcal{U}(\theta) \\ & x \ge 0 \end{array}$$

how do we pick the uncertainty set?

Mean-variance vs reshaped uncertainty sets

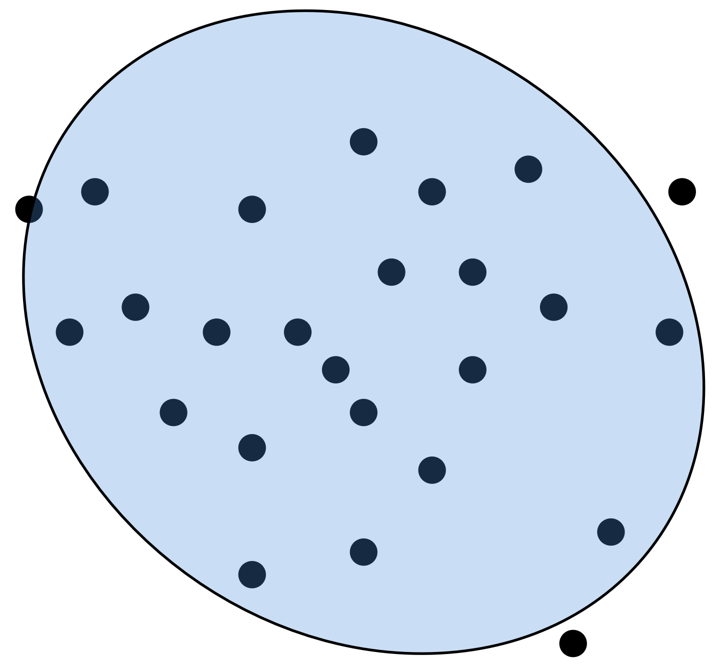
parameters

$$\theta = (A, b)$$

mean-variance set

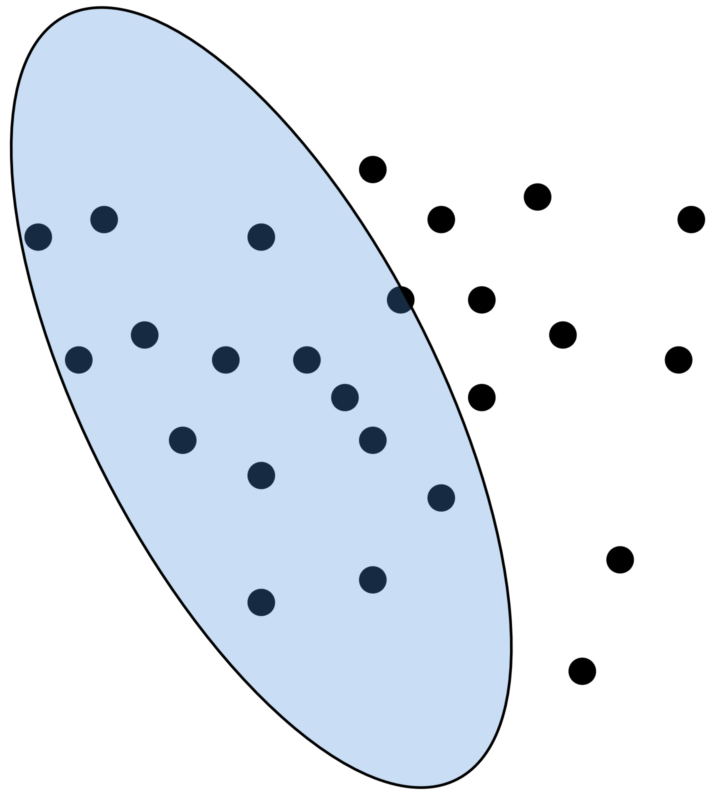
$$\mathcal{U}^{\mathrm{mv}}(\theta) = \{u = \hat{\mu} + \hat{\Sigma}^{1/2} z \mid \|z\|_2 \leq \rho\} = \{b^{\mathrm{mv}} + A^{\mathrm{mv}} z \mid \|z\|_2 \leq \rho\}$$

empirical
mean and covariance



reshaped uncertainty set

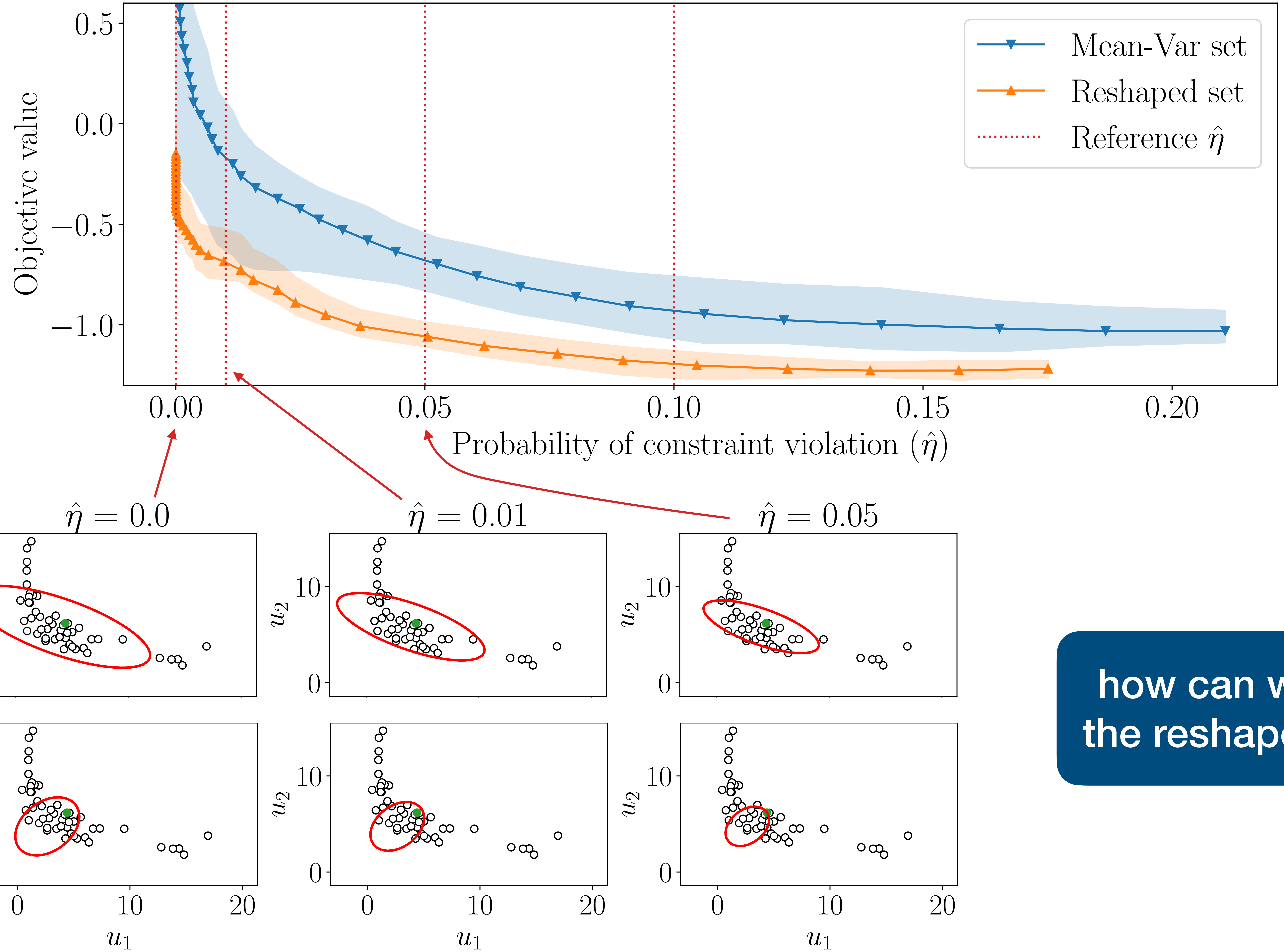
$$\mathcal{U}^{\mathrm{re}}(\theta) = \{u = b^{\mathrm{re}} + A^{\mathrm{re}} z \mid \|z\|_2 \leq \rho\}$$



can the reshaped set do better?

Reshaped set performs better

pareto curves for varying size ρ



mean-variance $\mathcal{U}^{\mathrm{mv}}(\theta)$

reshaped $\mathcal{U}^{\mathrm{re}}(\theta)$

how can we find the reshaped set?

Data-driven methods for robust optimization

Hypothesis testing
D. Bertsimas, V. Gupta, and N. Kallus (2014)

Quantile estimation
L. Jeff Hong, Z. Huang, and H. Lam (2021)

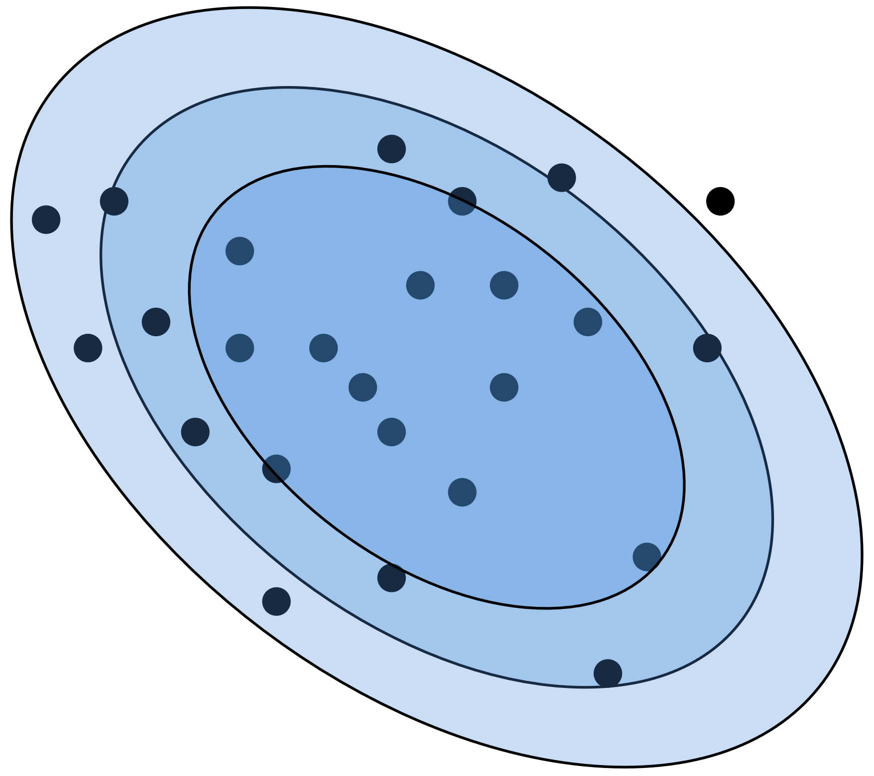
Wasserstein Distributionally Robust Optimization
P. M. Esfahani and D. Kuhn. (2018).
D. Bertsimas, S. Shtern, B. Sturt (2022)
I. Wang, C. Becker, B. Van Parys, and B. Stellato (2023)

Deep Learning
M. Goerigk, J. Kurtz (2023)

Differentiable Optimization
A. Chenreddy, E. Delage (2024)

Issues

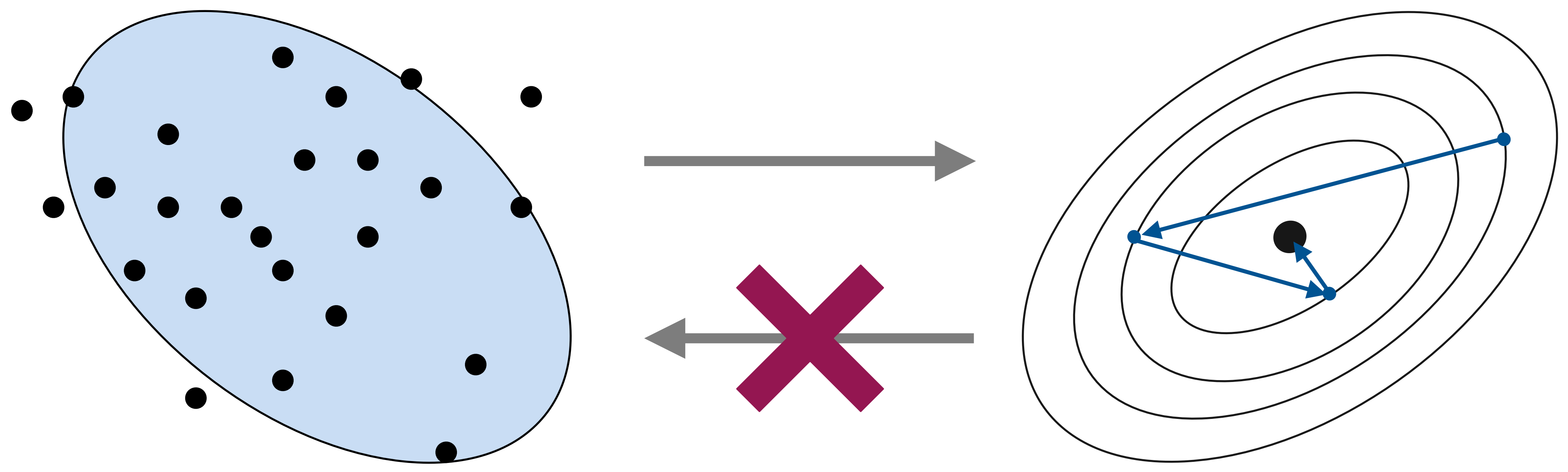
only size tuned



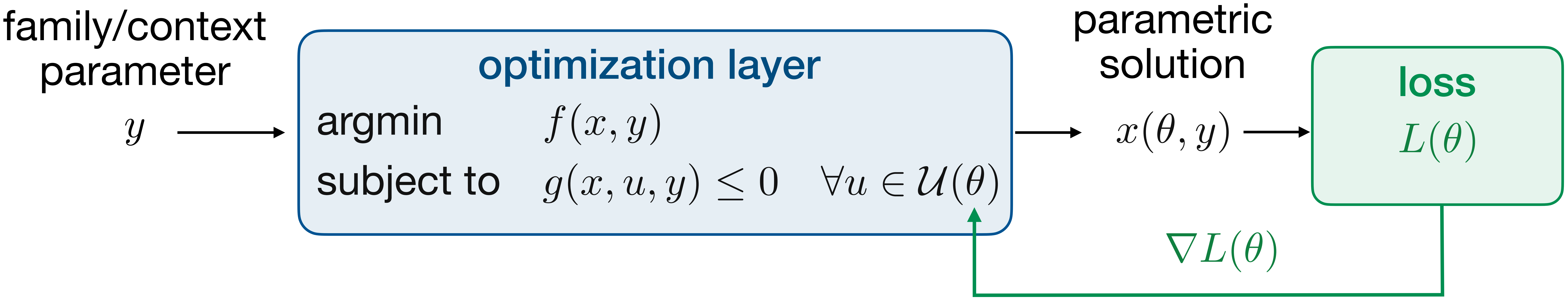
decoupling

Uncertainty set construction

Downstream optimization task



Leveraging the solution to tune the uncertainty sets



Main idea

Use differentiable optimization
to automatically learn
shape and *size*

Decision-making with uncertain constraints

parametric robust optimization

$$
\begin{aligned}
x(\theta, y) \in \ & \text{argmin} && f(x) \\
& \text{subject to} && g(x, u, y) \leq 0 \quad \forall u \in \mathcal{U}(\theta)
\end{aligned}
$$

family/context parameter

decisions

uncertain parameter

Which probabilistic guarantees can we ensure?

Probabilistic guarantees over distribution of instances

conditional probabilistic guarantees

$$\mathbf{P}_{(u|y)}\big(g(x(\theta,y),u,y)\leq 0 \mid y\big)\geq 1-\eta$$

strong condition

aggregate probabilistic guarantees

$$\mathbf{P}_{(u,y)}\big(g(x(\theta,y),u,y)\leq 0\big)\geq 1-\eta$$

Constraint satisfaction over a *distribution* of problem instances

Enforcing probabilistic guarantees with CVaR

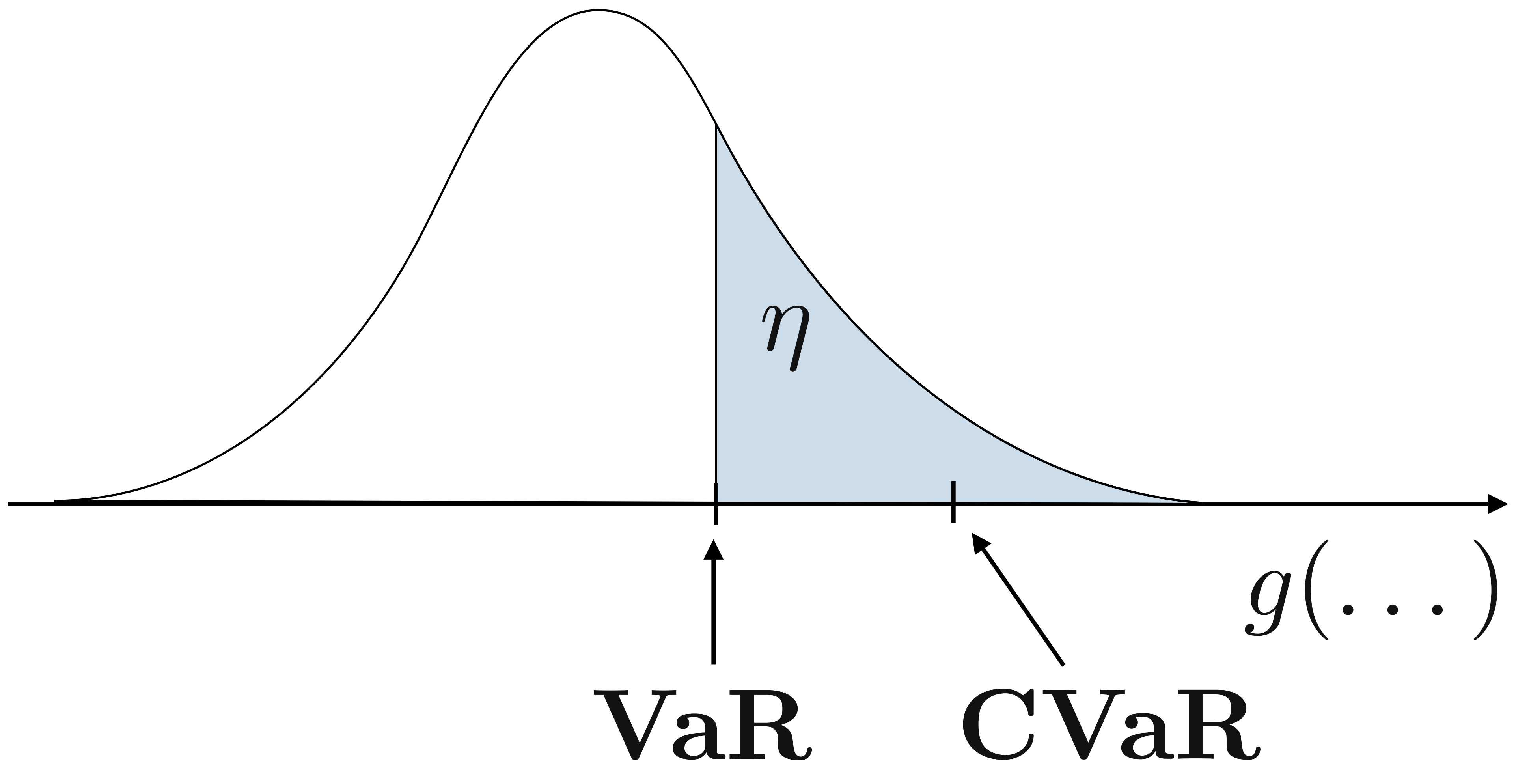
probabilistic guarantees

$$\mathbf{P}_{(u,y)}\big(g(x(\theta, y), u, y) \leq 0\big) \geq 1 - \eta$$

tractable approximation

$$\mathbf{CVaR}(g(x(\theta, y), u, y), \eta) \leq 0$$

same as $\mathbf{VaR}(g(\dots), \eta) \leq 0$



turn into constraint

threshold

$$\mathbf{E}_{(u,y)}\left(\frac{(g(x(\theta, y), u, y) - \alpha)_+}{\eta} + \alpha\right) \leq \kappa$$

Stochastic bilevel optimization to learn the uncertainty set

Loss

training problem

$$\begin{array}{ll} \text{minimize} & \mathbf{E}_y f(x(\theta, y)) \\ \text{subject to} & \mathbf{E}_{(u,y)} \left((g(x(\theta, y), u, y) - \alpha)_+ / \eta + \alpha \right) \leq \kappa \end{array}$$

CVaR constraint

inner robust problem

$$x(\theta, y) \in \begin{array}{ll} \text{argmin} & f(x) \\ \text{subject to} & g(x, u, y) \leq 0 \quad \forall u \in \mathcal{U}(\theta) \end{array}$$

we must reformulate the infinite dimensional constraints

Uncertainty set parameters enter nicely in the reformulation

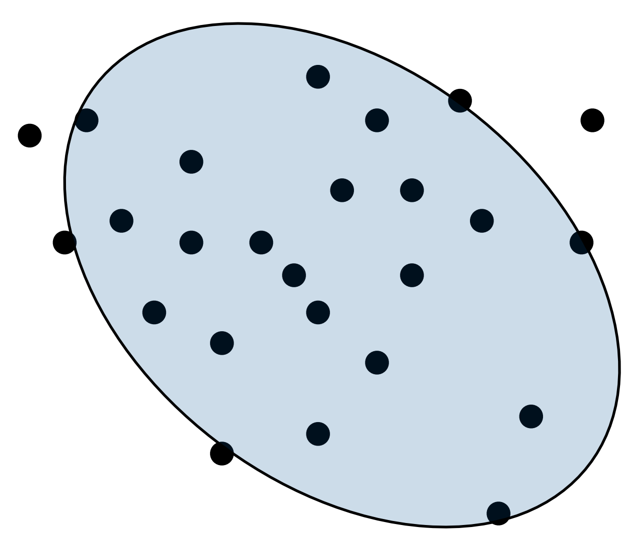
$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x,u,y) \leq 0 \quad \forall u \in \mathcal{U}(\boxed{\theta}) \end{array} \longleftarrow$$

learned parameters

Example: ellipsoidal set

$$\mathcal{U}(\boxed{\theta}) = \{u = \boxed{b} + \boxed{A}z \mid \|z\|_2 \leq 1\}$$

$$\theta = (A, b)$$



linear constraint

$$g(x,u,y) = (y + Pu)^T x \leq 0, \quad \forall u \in \mathcal{U}(\theta)$$

$\longrightarrow$

robust counterpart

$$y^T x + \boxed{b}^T Px + \|\boxed{A}^T P^T x\|_2 \leq 0$$

Learning using Stochastic Augmented Lagrangian Algorithm

$$\begin{array}{ll} \text{minimize} & \underbrace{\mathbf{E}_y f(x(\theta, y))}_{F(\theta)} \\ \text{subject to} & \underbrace{\mathbf{E}_{(u,y)} \left((g(x(\theta, y), u, y) - \alpha)_+ / \eta + \alpha \right)}_{H(\alpha, \theta)} \leq \kappa \end{array}$$

augmented Lagrangian

penalty

$$L(\alpha, \theta, s, \lambda, \mu) = F(\theta) + \underbrace{\lambda}_{\text{multiplier}}(H(\alpha, \theta) + s - \kappa) + \frac{\overbrace{\mu}^{\text{penalty}}}{2} \|H(\alpha, \theta) + s - \kappa\|^2$$

multiplier

stochastic gradient computation

$\hat{\nabla}_\theta L$ *depends on jacobian* $J_x(\theta)$

differentiate through the KKT optimality conditions

$$y \rightarrow \boxed{\text{inner robust problem}} \rightarrow x(\theta, y)$$

"Differentiable Convex Optimization Layers", A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Zico Kolter (2019)
"Differentiating Through a Conic Program", A. Agrawal, S. Barratt, S. Boyd, E. Busseti, W. M. Moursi (2019)

Finite-sample probabilistic guarantees via threshold

optimality tolerance

target confidence

dimension of x

number of samples of u

threshold training constraint

$$\widehat{\mathbf{CVaR}}(g(x(\theta, y), u, y), \eta) \leq \kappa$$

Implies

Finite-sample probabilistic guarantee

$$\mathbf{P}^{N \times J}\left(\mathbf{P}_{(u,y)}(g(x, u, y) \leq 0) \geq 1 - \eta \quad \forall x \in \mathcal{X}\right) \geq 1 - \beta$$

$\Rightarrow$ it holds also for $x(\theta^\star, y)$

Ingredients

- Tail bounds
- $\mathbf{CVaR} \geq \mathbf{VaR}$

LROPT software package (WIP)

It can be *hard to dualize* robust optimization problems

…not to mention *finding the right uncertainty set!*

LROPT package

github.com/stellatogrp/lropt

1. Easily formulate and dualize robust optimization problems

2. Automatically tune uncertainty sets (using cvxpylayers)

$$\begin{array}{ll} \text{minimize} & x^T P x + y^T x \\ \text{subject to} & (a + Bu)^T x \leq d, \quad \forall u \in \mathcal{U} \end{array}$$

$$\mathcal{U} = \{u = b + Az \mid \|z\|_2 \leq 1\}$$

```
unc_set = lropt.Ellipsoidal(u_data)
u = lropt.UncertainParameter(n,
                             uncertainty_set=unc_set)
x = cp.Variable(n)
y = cp.Parameter(n)
constraints = [(a + B@u) @ x <= d]
objective = cp.Minimize(cp.quad_form(P, x) + y @ x)
problem = lropt.RobustProblem(objective,
                              constraints)
problem.train()
```

Portfolio optimization with reference allocations

uncertain returns

$$u \sim \mathcal{N}(\mu, \Sigma)$$

$$\mu = \begin{bmatrix} 0.3 \\ 0.3 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 0.5 & -0.3 \\ -0.3 & 0.4 \end{bmatrix}$$

objective

$$-u^T x + \lambda \|x - x^{\text{ref}}\|_1$$

investments decisions

reference holdings

$$x^{\text{prev}} \sim \mathrm{Dir}(\alpha)$$
$$\alpha = (2.5, 1)$$

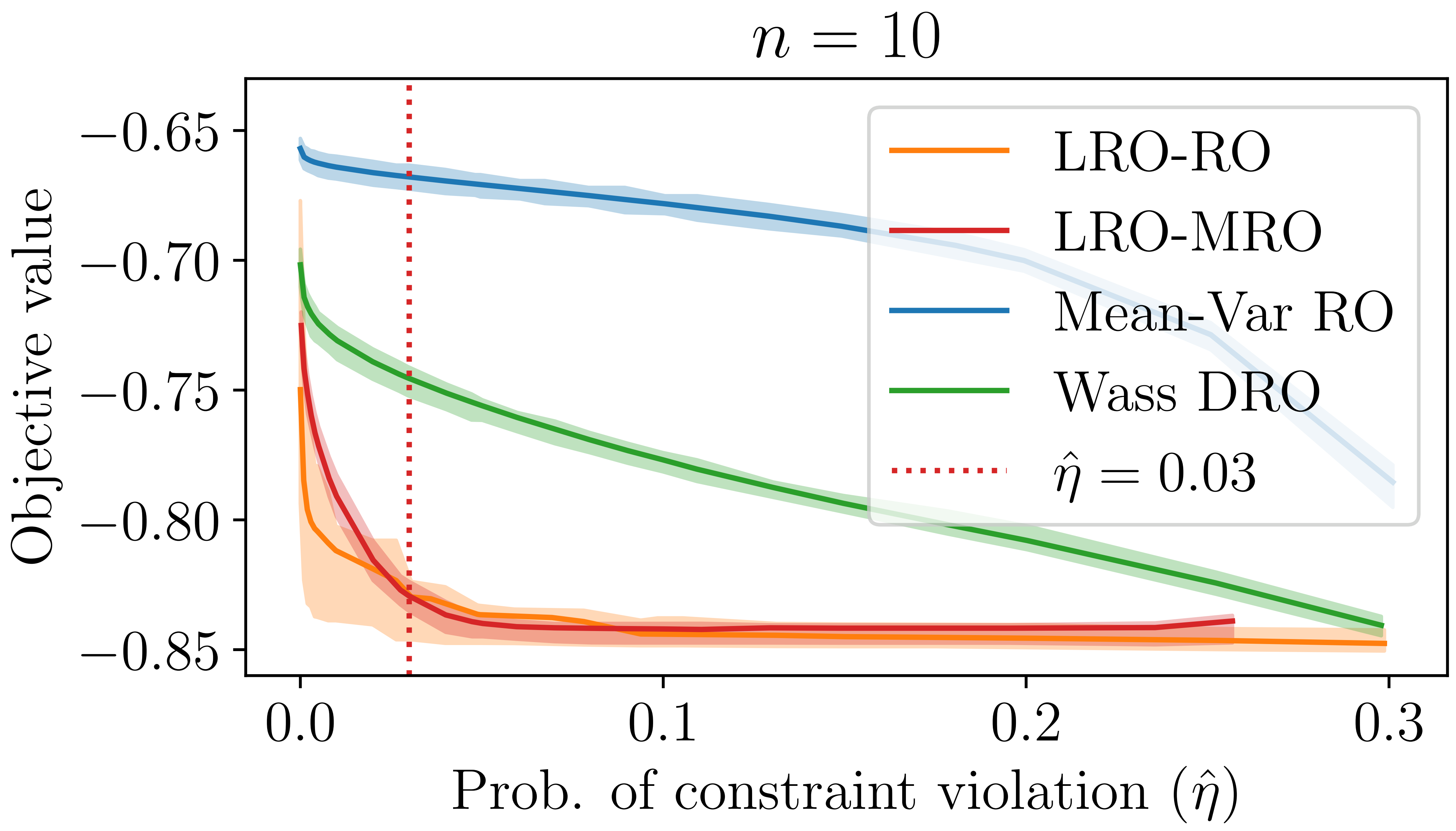
robust problem reformulation

$$\begin{array}{ll} \text{minimize} & t + \lambda \|x - x^{\text{ref}}\|_1 \\ \text{subject to} & -u^T x \le t \quad \forall u \in \mathcal{U}(\theta) \\ & \mathbf{1}^T x = 1, \quad x \ge 0 \end{array}$$

uncertainty set

LRO outperforms original sets in larger portfolio example

$n = 10$ (dimension of u)						
Method	LRO_{RO}	$\text{LRO-T}_{\text{RO},0.03}$	LRO_{MRO}	$\text{LRO-T}_{\text{MRO},0.03}$	$\text{MV-RO}_{0.03}$	$\text{W-DRO}_{0.03}$
Obj.	-0.816	-0.823	-0.819	-0.827	-0.666	-0.744
$\hat{\eta}$	0.02	0.0264	0.0199	0.0276	0.0264	0.0272
$\hat{\beta}$	0	0.2	0	0	0.1	0
time	0.000598	0.000599	0.00219	0.00225	0.000602	0.110

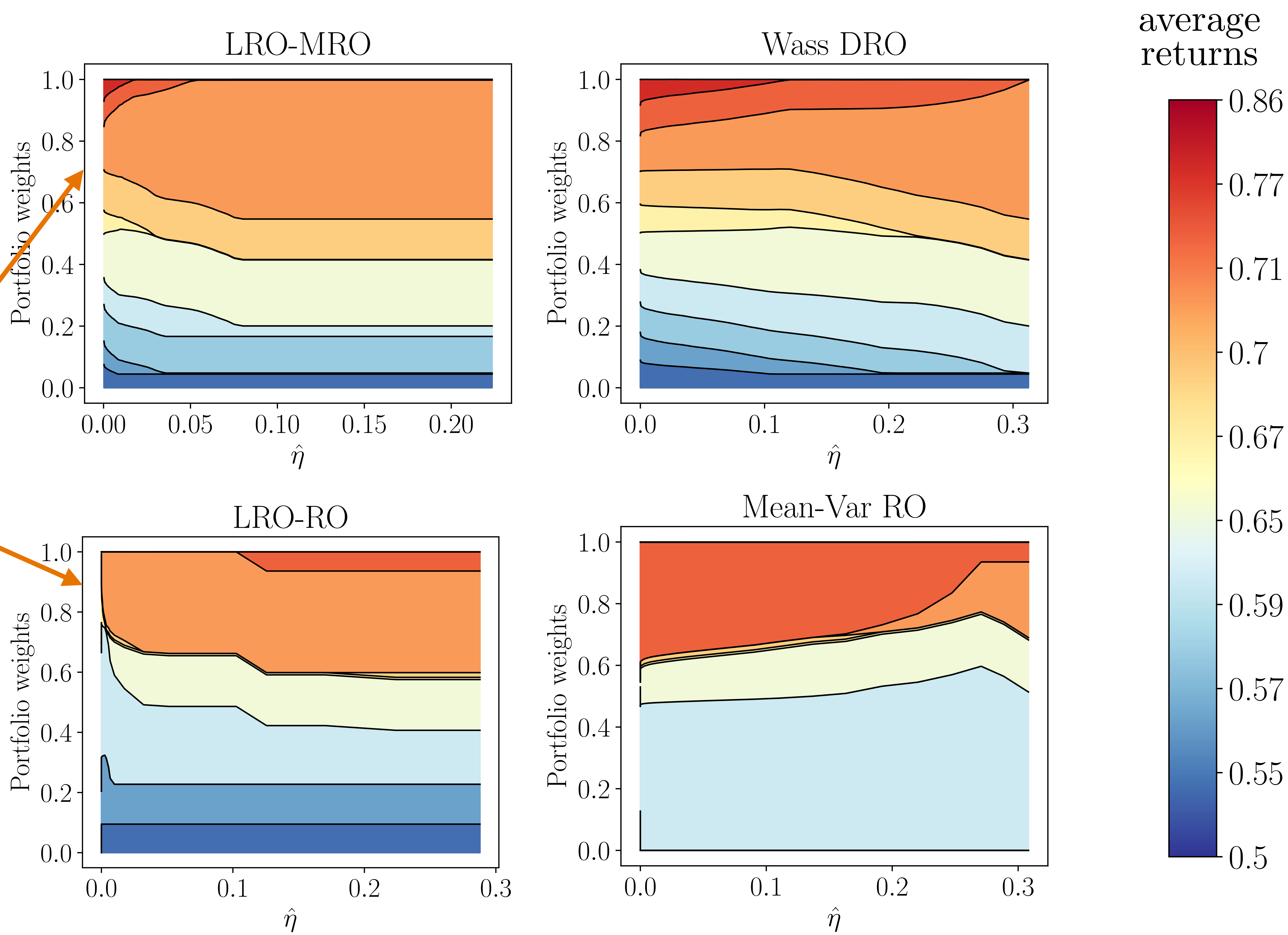


better trade-off between *objective* and *probability of constraint violation*

faster computation times than Wassertstein DRO

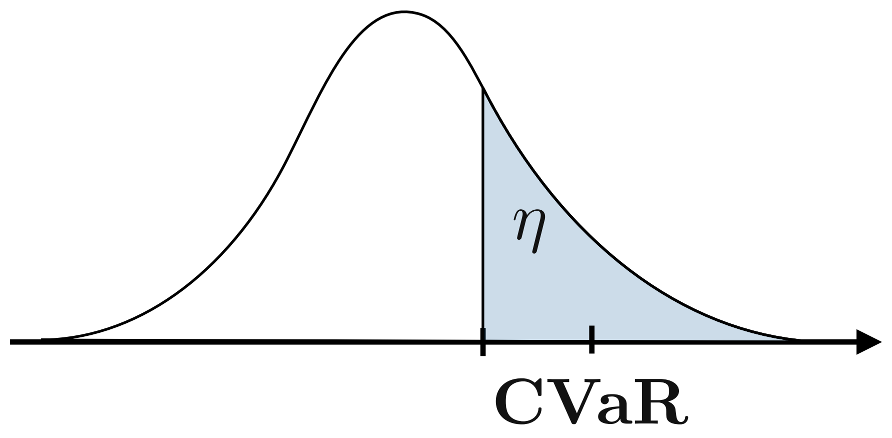
LRO allocations with are less sensitive to target constraint violation

reshaped sets identify high-risk high-return



Learning decision-focused uncertainty sets for robust optimization

- Optimize **shape and size** of uncertainty sets
- **Bi-level optimization** formulation
  - CVaR constraint
  - Differentiable optimization to compute derivatives
  - Probabilistic guarantees



optimization layer

$$y \longrightarrow \begin{array}{ll} \text{argmin} & f(x, y) \\ \text{subject to} & g(x, u, y) \leq 0 \quad \forall u \in \mathcal{U}(\theta) \end{array} \longrightarrow x(\theta, y)$$

- **Improvements over RO and DRO formulations**

https://github.com/stellatogrp/lropt_experiments

arxiv **Learning Decision-Focused Uncertainty Sets for Robust Optimization**
I. Wang, C. Becker, B. Van Parys, and B. Stellato
arxiv.org: 2305.19225, 2023

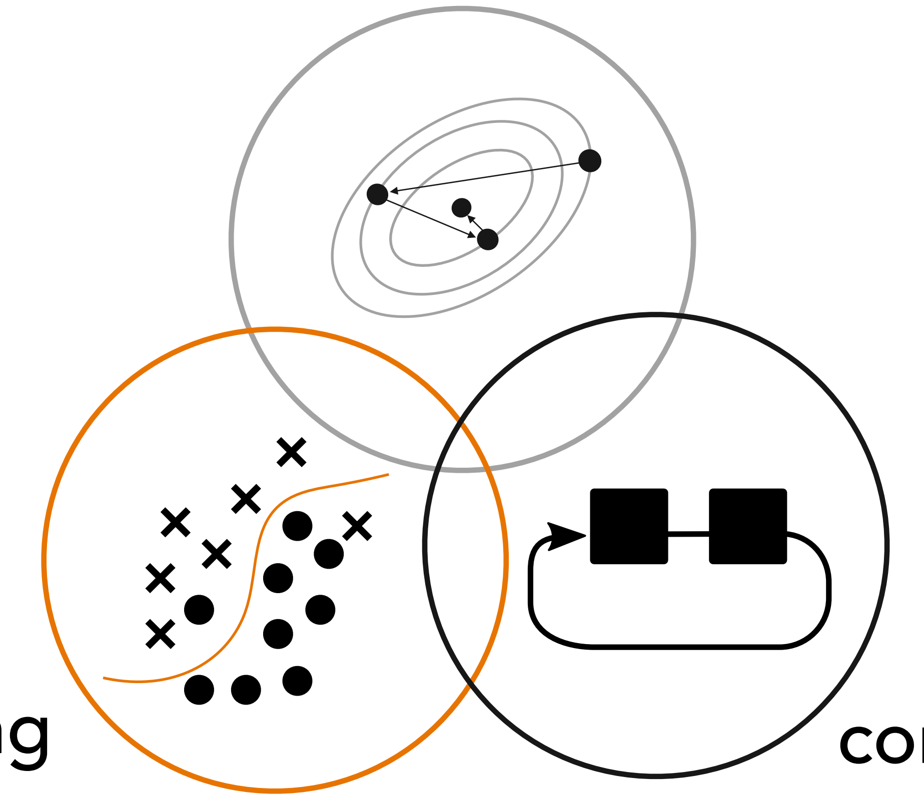
Jun 27–28, 2024, Princeton University

PRINCETON UNIVERSITY

Princeton Workshop on
Optimization, Learning, and Control

The Princeton Workshop on Optimization, Learning, and Control is a single-track workshop highlighting the latest research advances across these disciplines. Its main goal is to foster new interactions and lay the groundwork for new collaborations. The workshop will include a poster session for junior researchers.

optimization



learning

control

Contacts

- **Website**: stellato.io/olc
- **Organizer**: Bartolomeo Stellato, Princeton University — stellato.io
- **Questions**: olc24@princeton.edu

Supported by

National Science Foundation

School of Engineering and Applied Science PRINCETON

ORFE

IBM

Center for Statistics and Machine Learning

GUROBI OPTIMIZATION

Confirmed speakers

Anuradha Annaswamy
MIT

Francesco Borrelli
UC Berkeley

Sanjeeb Dash
IBM

Sarah Dean
Cornell University

Paul Goulart
University of Oxford

Elad Hazan
Princeton University

Andrea Lodi
Cornell Tech

Robert Luce
Gurobi Optimization

Anirudha Majumdar
Princeton University

Melanie Zeilinger
ETH Zurich

Conclusion

Machine Learning tools
can help us
formulate optimization problems

↓

We should think
building robust optimization models
as an (automated)
training/validation procedure

⚙️ tune *train* → ☑️ validate *certify* → 🚀 deploy

@b_stellato bstellato@princeton.edu stellato.io