A Unified Machine Learning Framework for Optimization Under Uncertainty

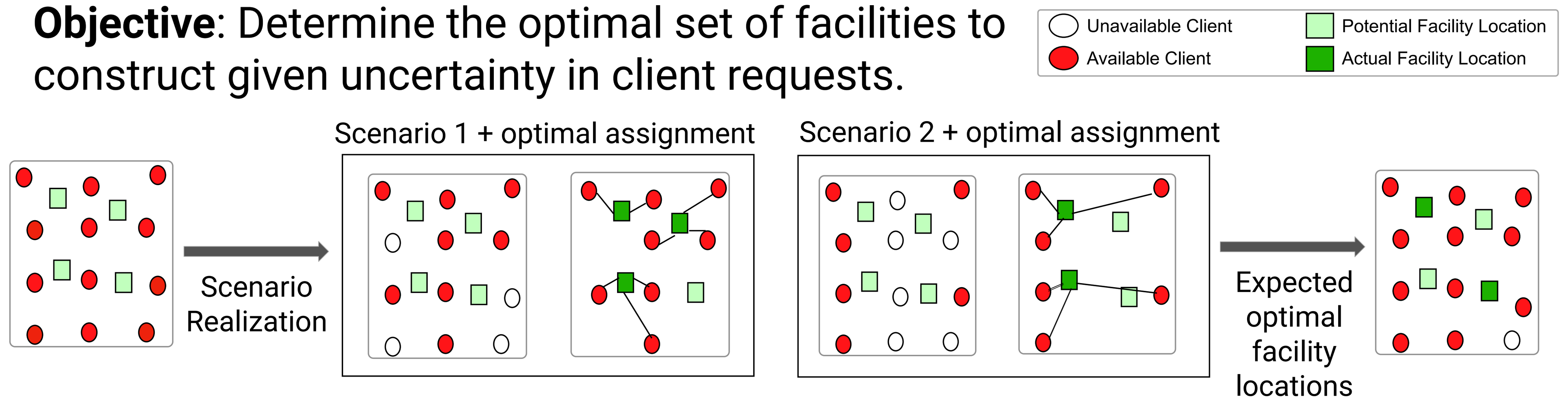
Justin Dumouchelle

Learning for **Two-Stage Stochastic Programming (2SP)**

Published at NeurIPS 2022. Joint work with Rahul Patel, Elias Khalil, and Merve Bodur

Stochastic Facility Location Problem

Objective: Determine the optimal set of facilities to construct given uncertainty in client requests.

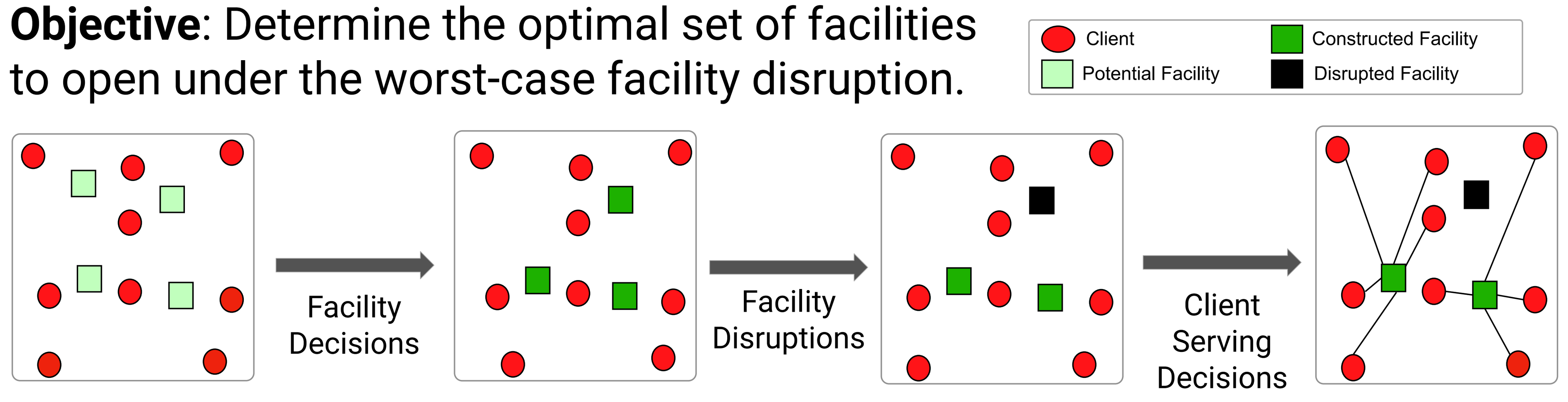


Learning for **Two-Stage Robust Optimization (2RO)**

Under review. Joint work with Esther Julien, Jannis Kurtz, and Elias Khalil

Robust Facility Location Problem

Objective: Determine the optimal set of facilities to open under the worst-case facility disruption.



Optimization Formulation

Objective: Determine optimal first-stage decisions that minimize sum of the first-stage cost and expected second-stage cost.

Challenge: Exact optimization becomes exponentially harder with the number of observed scenarios, K. Integer decision further aggravates intractability.

First-Stage Problem

$$\min_{\mathbf{x}} \left\{ \underbrace{\mathbf{c}^\intercal \mathbf{x}}_{\text{First-stage cost}} + \underbrace{\sum_{k=1}^{K} p_k Q(\mathbf{x}, \boldsymbol{\xi}_k)}_{\text{Expected second-stage cost}} : \mathbf{x} \in \mathcal{X} \right\}$$

First-stage decisions: $\mathbf{x}$

Second-Stage Problem

$$Q(\mathbf{x}, \boldsymbol{\xi}) := \min_{\mathbf{y}} \left\{ \underbrace{F(\mathbf{y}, \boldsymbol{\xi})}_{\text{Second-stage cost}} : \mathbf{y} \in \mathcal{Y}(\mathbf{x}, \boldsymbol{\xi}) \right\}$$

Second-stage decisions: $\mathbf{y}$

Objective: Determine optimal first-stage decisions that minimize the sum of the first-stage cost and worst-case cost.

Challenge: Solving the nested optimization problem is intractable and specialized algorithms only exists for a limited classes of problems.

First-Stage Problem

$$\min_{\mathbf{x}} \left\{ \underbrace{\mathbf{c}^\intercal \mathbf{x}}_{\text{First-stage cost}} + \underbrace{Q(\mathbf{x})}_{\text{Worst-case cost}} : \mathbf{x} \in \mathcal{X} \right\}$$

First-stage decisions: $\mathbf{x}$

Second-Stage Problem

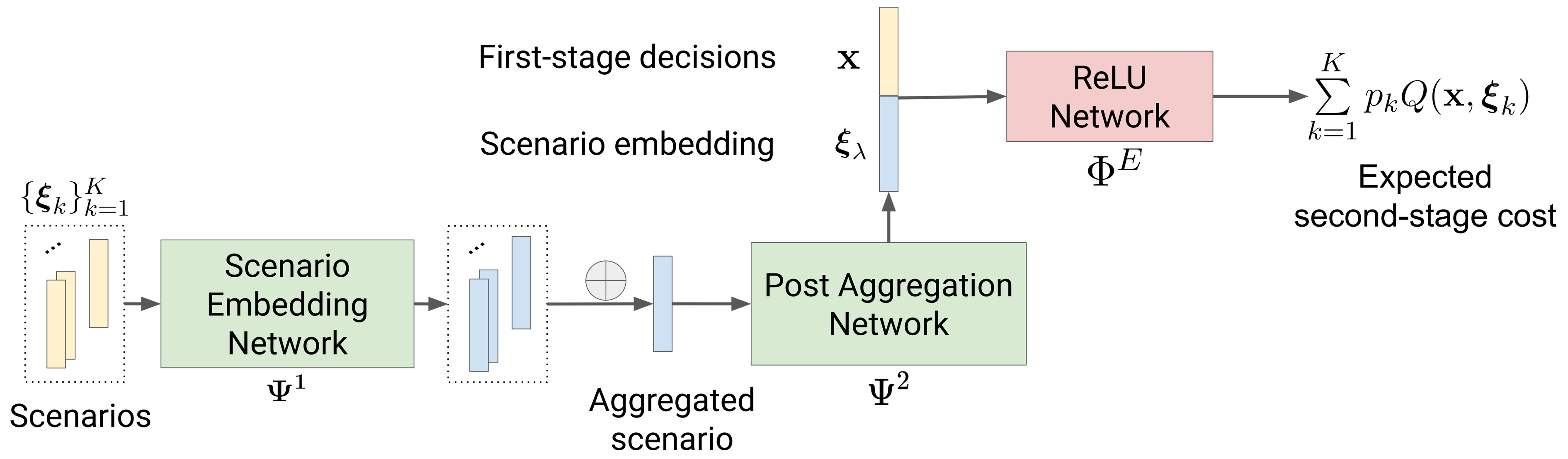
$$Q(\mathbf{x}) := \underbrace{\max_{\boldsymbol{\xi} \in \Xi}}_{\text{Worst-case uncertainty}} \min_{\mathbf{y}} \left\{ \underbrace{\mathbf{d}(\boldsymbol{\xi})^\intercal \mathbf{y}}_{\text{Second-stage cost}} : \mathbf{y} \in \mathcal{Y}(\mathbf{x}, \boldsymbol{\xi}) \right\}$$

Second-stage decisions: $\mathbf{y}$

Machine Learning Methodology

ML Solution: Replace the expected stage-cost value function with a neural network approximation.

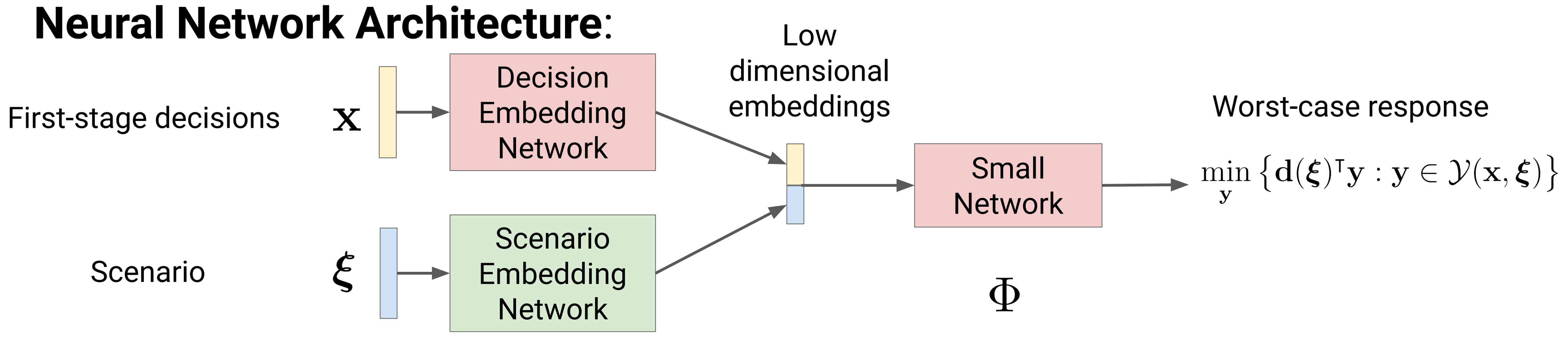
Neural Network Architecture:



Surrogate Optimization Model: $\min_{\mathbf{x}} \left\{ \mathbf{c}^\intercal \mathbf{x} + \mathrm{NN}(\mathbf{x}, \{\boldsymbol{\xi}_k\}_{k=1}^K) : \mathbf{x} \in \mathcal{X} \right\}$

ML Solution: Replace the worst-case response optimization problem with a neural network approximation.

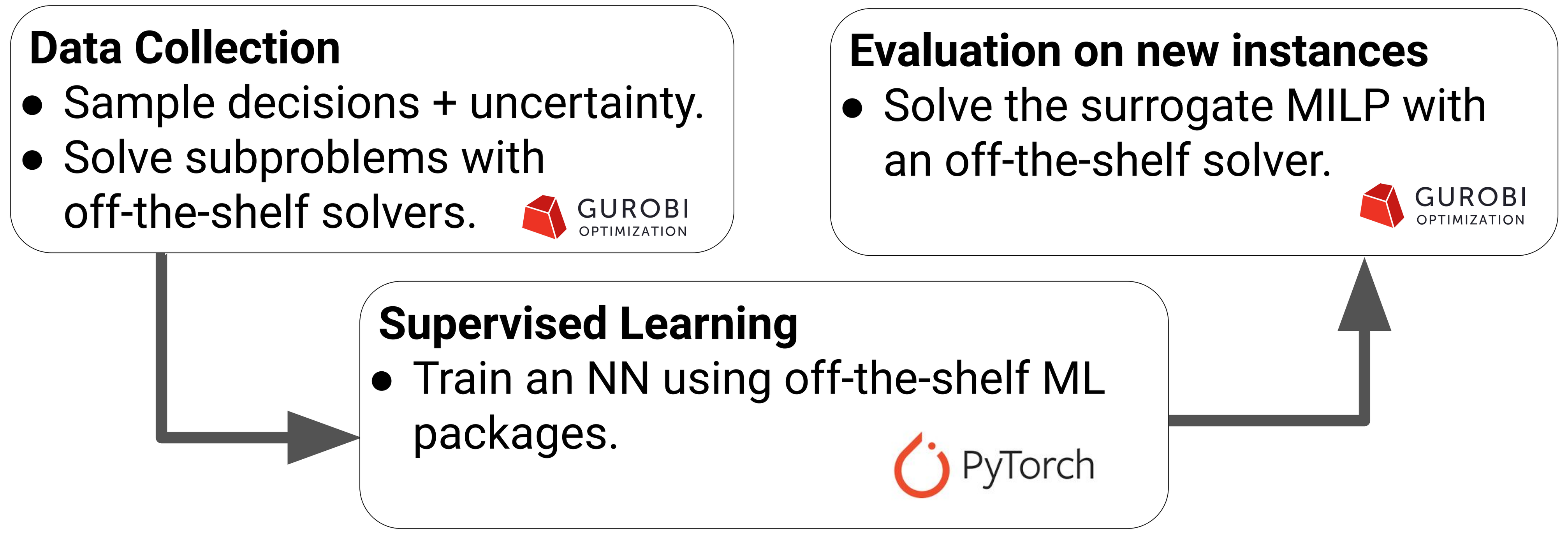
Neural Network Architecture:



Surrogate Optimization Model: $\min_{\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}_a} \left\{ \mathbf{c}^\intercal \mathbf{x} + \mathbf{d}(\boldsymbol{\xi}_a) : \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}(\mathbf{x}, \boldsymbol{\xi}_a), \boldsymbol{\xi}_a \in \arg\max_{\boldsymbol{\xi} \in \Xi'} \{\mathrm{NN}(\mathbf{x}, \boldsymbol{\xi})\} \right\}$

- Optimization problem solved via constraint generation of small networks.
- Worst-case scenarios determined via adversarial problem ($\max_{\boldsymbol{\xi} \in \Xi} \mathrm{NN}(\hat{\mathbf{x}}, \boldsymbol{\xi})$).

Overall framework

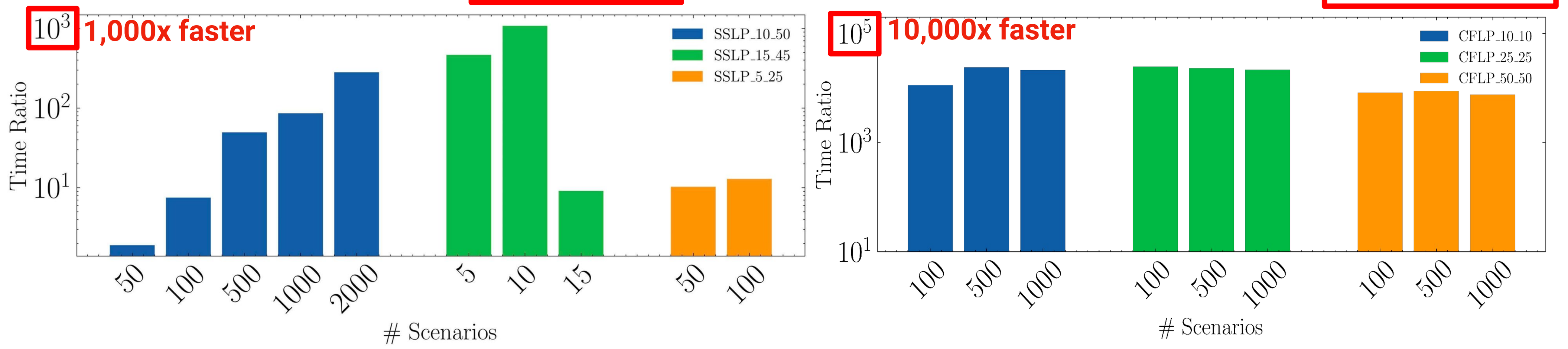


Experimental Results

Stochastic Programming

Reduction factor in solving time over baseline (Integer L-shaped/EF) (higher is better).

Stochastic Server Location (Gap: 0.87%) **Capacitated Facility Location** (Gap: -2.93%)

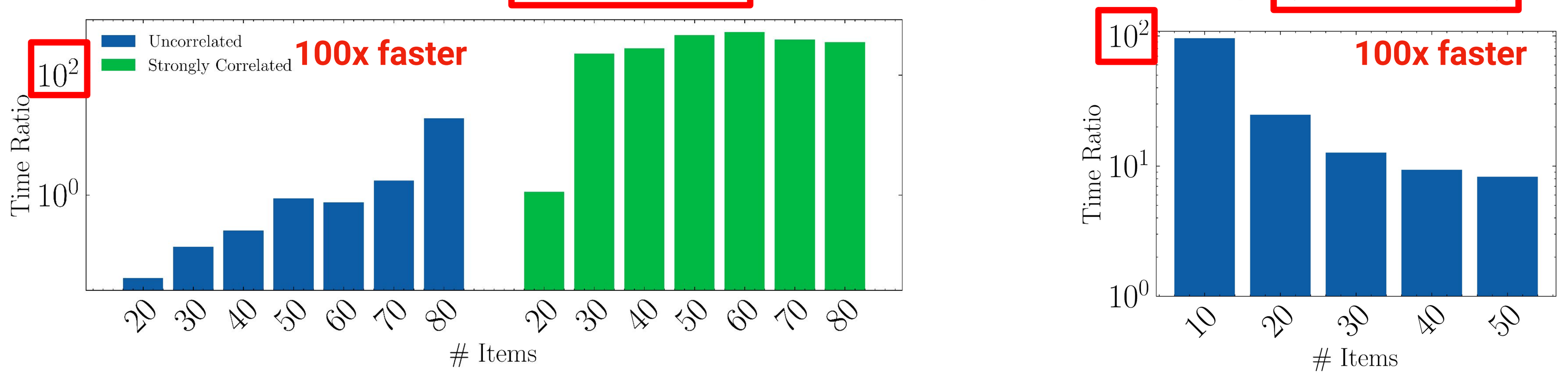


- ML finds OPT on most SSLP instances and best known solutions on most CFLP instances.

Robust Optimization

Reduction factor in solving time over baseline (branch-and-price/k-adaptability) (higher is better).

Robust Knapsack Problem (Gap: 1.30%) **Capital Budgeting** (Gap: 0.33%)



- ML finds best solutions on difficult instances; near OPT on easy instances.

2SP **2RO**

Mechanical & Industrial Engineering
UNIVERSITY OF TORONTO