Robust Decision via Generative Wasserstein Distributionally Robust Optimization

Han Xu, Christopher Yeh, California Institute of Technology

1. BACKGROUND & MOTIVATIONS

Background:

- A lot of decision problems include the stages of both **prediction** and **optimization**.
- Example: Effectively scheduling energy storage (i.e., when to charge/discharge a battery for the most profit) is crucial for battery operators, but it's difficult due to uncertainty in future electricity prices.

Existing Approaches:

- **Predict-then-Optimize:** Standard methods first predict future prices and then optimize decisions. However, prediction errors can lead to very poor decisions, a problem known as the **"optimizer's curse".**
- **End-to-End Robust Optimization (E2E-RO):** These methods learn an "uncertainty set" to make robust decisions but are often limited to simple shapes (e.g., convex sets) and can be **overly conservative**.

Our Goal:

- Develop a framework that learns a flexible representation of uncertainty and makes robust decisions, especially when the real-world data distribution shifts

2. Problem Formulation

Original Problem:

$$z^*(x) := \arg\min_z \mathrm{E}^P[\,f(x,y,z) \mid x\,] \text{ s.t. } g(x,z) \le 0$$

- Notice Conditional Distribution of $P(y|x)$ is **unknown**.
- But we can learn it which might be **inaccurate**.

Distributionally robust optimization (DRO) formulation:

$$z^*(x) := \arg\min_z \max_{Q\in\mathcal{B}_\rho\left(\widehat{P_\theta}(\cdot|x)\right)} \mathrm{E}^Q[f(x,y,z)] \quad \text{s.t. } g(x,z) \le 0$$

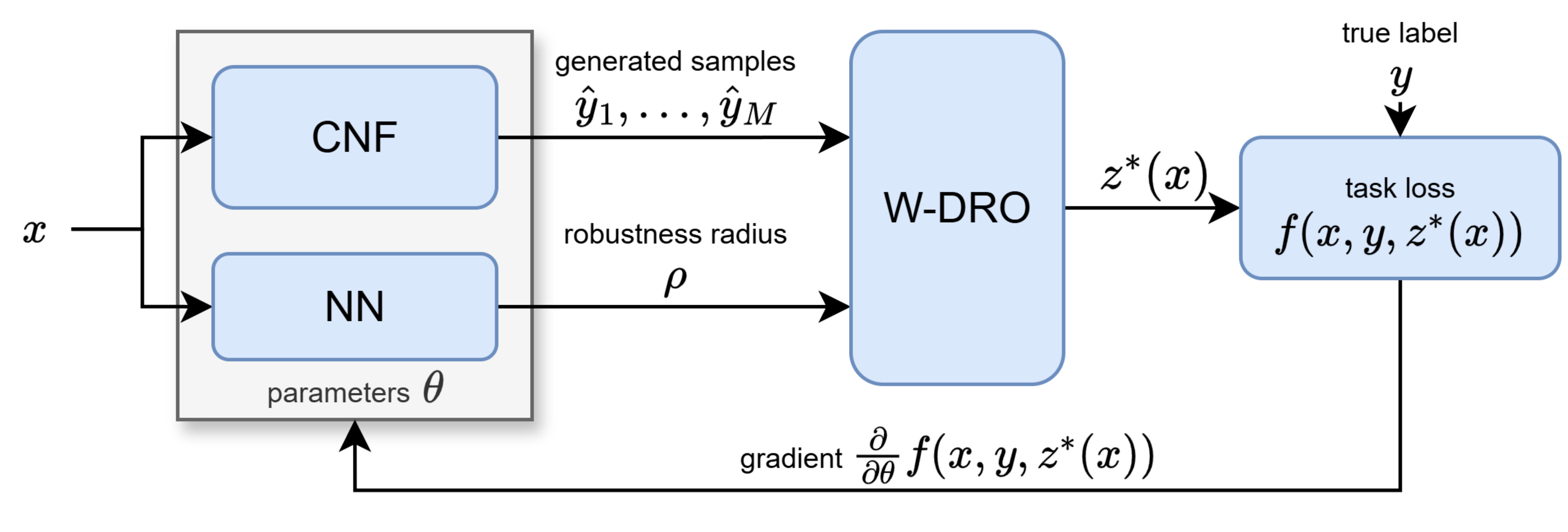
- Even the learned distribution is inaccurate, the decision has robustness
- **Q1**: How to learn the distribution? **Q2**: How to solve DRO?

3. Method: The Gen-WDRO Framework

Our method combines a **generative model (i.e., conditional normalizing flow (CNF))** with **Wasserstein distributionally robust optimization (WDRO)** for robust end-to-end learning.

How it works?

- **Generate Price scenarios:** A **Conditional Normalizing Flow (CNF)** generative model takes input features (like weather forecasts) and generates M possible future electricity price scenarios $\{\hat{y}_1, \dots, \hat{y}_M\}$.
- **Quantify Uncertainty:** A separate **Neural Network (NN)** learns an adaptive **robustness radius** (ρ), which determines how much we distrust the generated price distribution. This radius changes based on the input features.
- **Make a Robust Decision:** The **Wasserstein DRO (W-DRO)** layer finds the optimal charging schedule $z^*(x)$ that performs best under the worst-case price distribution.
- **End-to-End Training:** The entire model is trained to minimize the final decision loss, with gradients flowing back from the task loss through the W-DRO layer to update both the CNF and NN parameters.

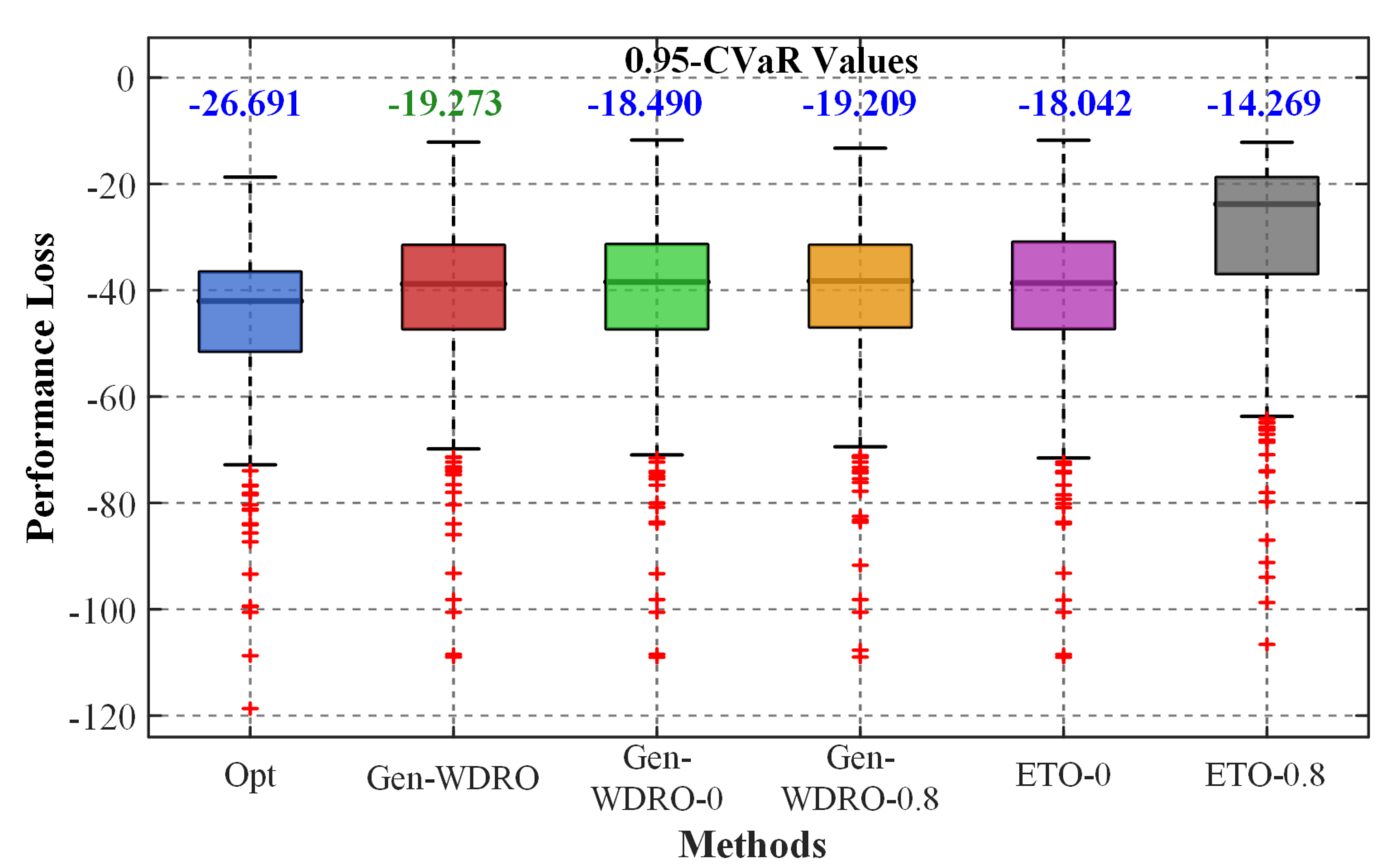


Gen-WDRO Framework

4. EXPERIMENT

Experimental Settings:

- **Task:** Schedule a grid-scale battery's charging/discharging over a 24-hour horizon to maximize profit from price arbitrage.
- **Distribution Shift:** To simulate the increased price volatility from future renewable energy penetration, we tested the models on data where Gaussian noise was added to the electricity prices and the electricity price pattern shifts.
- **Evaluation Metric:** We use the **0.95-Conditional Value-at-Risk (CVaR)** to measure robustness. CVaR quantifies the average loss in the worst 5% of cases. **Lower is better**.



Performance Comparison

- **Among the compared methods:** Our Gen-WDRO method achieves the lowest (best) CVaR, indicating better robustness.
- **Takeaways**:
  - **Adaptive Radius is necessary:** Our method with a learnable radius outperforms fixed-radius versions (Gen-WDRO-0 and Gen-WDRO-0.8), showing that adapting the uncertainty quantification is critical.
  - End-to-end training is necessary: directly improving the decision performance via doing back propagation on generative model can indeed improve the performance.