CATO: End-to-End Optimization of ML-Based Traffic Analysis Pipelines

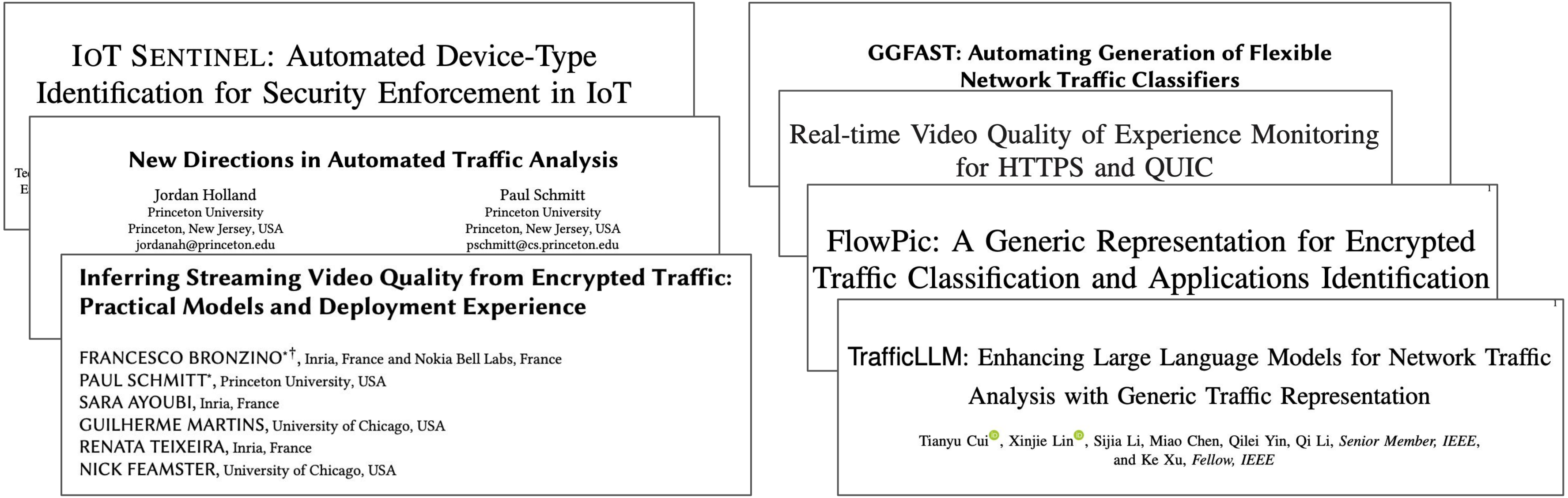
Gerry Wan, Shinan Liu, Francesco Bronzino,
Nick Feamster, Zakir Durumeric

Increasing popularity of ML for network traffic analysis

- Models learn effectively from large volumes of network data
- Complex, evolving traffic patterns
- Encryption undermines traditional rule-based heuristics

Increasing popularity of ML for network traffic analysis

- Models learn effectively from large volumes of network data
- Complex, evolving traffic patterns
- Encryption undermines traditional rule-based heuristics



IoT Sentinel: Automated Device-Type Identification for Security Enforcement in IoT

New Directions in Automated Traffic Analysis

Jordan Holland, Princeton University, Princeton, New Jersey, USA, jordanah@princeton.edu

Paul Schmitt, Princeton University, Princeton, New Jersey, USA, pschmitt@cs.princeton.edu

Inferring Streaming Video Quality from Encrypted Traffic: Practical Models and Deployment Experience

FRANCESCO BRONZINO*†, Inria, France and Nokia Bell Labs, France
PAUL SCHMITT*, Princeton University, USA
SARA AYOUBI, Inria, France
GUILHERME MARTINS, University of Chicago, USA
RENATA TEIXEIRA, Inria, France
NICK FEAMSTER, University of Chicago, USA

GGFAST: Automating Generation of Flexible Network Traffic Classifiers

Real-time Video Quality of Experience Monitoring for HTTPS and QUIC

FlowPic: A Generic Representation for Encrypted Traffic Classification and Applications Identification

TrafficLLM: Enhancing Large Language Models for Network Traffic Analysis with Generic Traffic Representation

Tianyu Cui, Xinjie Lin, Sijia Li, Miao Chen, Qilei Yin, Qi Li, *Senior Member, IEEE*, and Ke Xu, *Fellow, IEEE*

Practical deployment: more than just accuracy

- Systems costs of serving is just as important as predictive performance

<u>Predictive Performance</u>

- Accuracy
- F1 score
- RMSE
- etc.

<u>Systems Costs</u>

- Throughput
- Latency
- Memory usage
- etc.

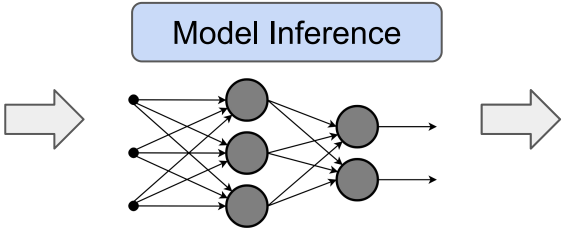
Serving efficiency is especially critical for traffic analysis

- Network traffic analysis is a **real-time** task
- Balancing systems costs are essential for the **validity** of ML-based predictions

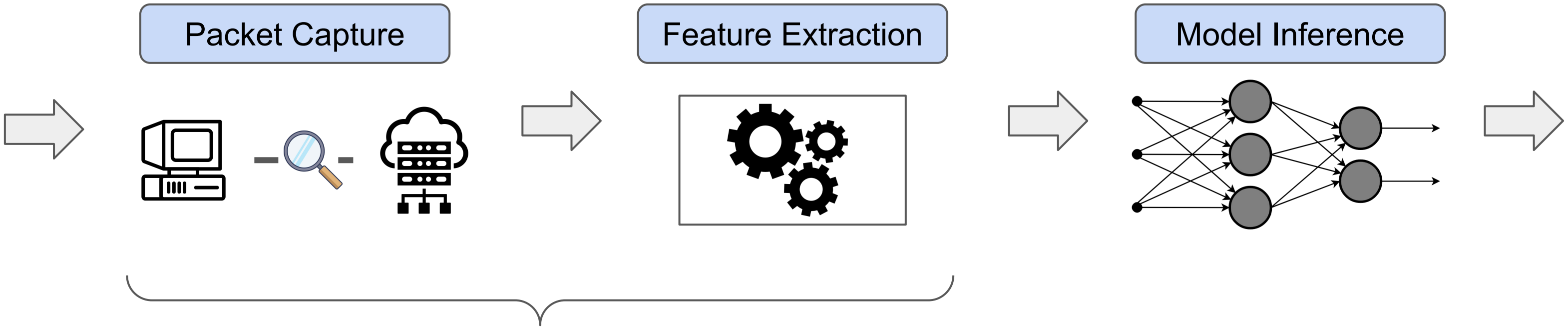
Not keeping up with traffic → stale results

Not keeping up with traffic → packet drop → **invalid results**

Model inference is only one component of the end-to-end pipeline



Model inference is only one component of the end-to-end pipeline



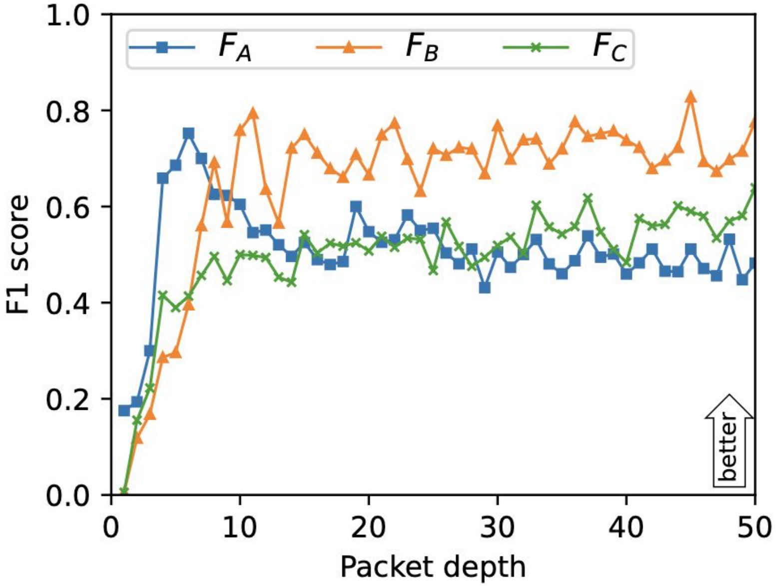
- Filtering
- Header parsing
- Waiting for flow context
- Statistical feature computation

Systems costs vs. predictive performance: not always a tradeoff

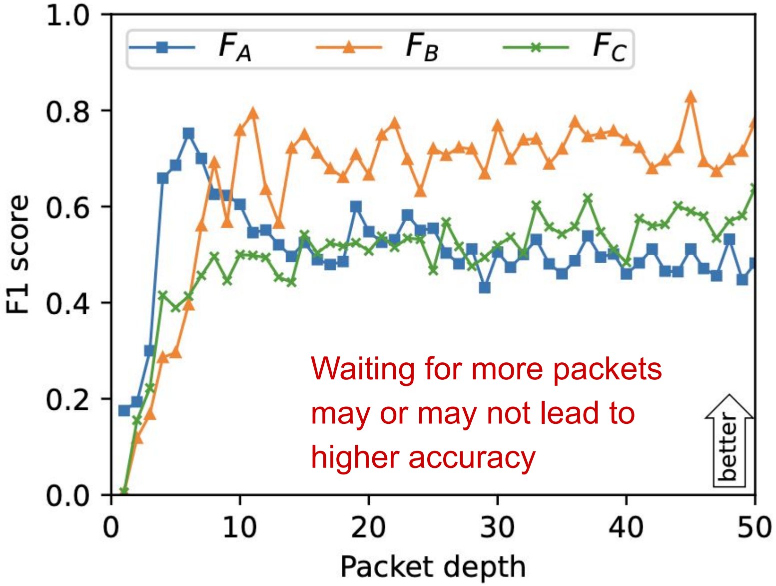
- **Traffic representation** (features and flow context length) strongly impacts both systems costs and predictive performance
- More packets and richer features don’t always guarantee better results
- Relationship is **nonlinear** and **non-obvious**

Tradeoff → Search space

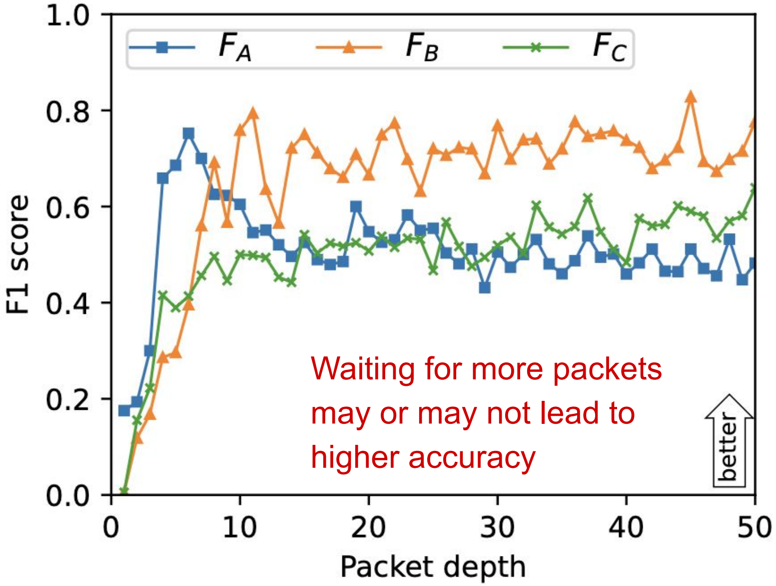
Systems costs vs. predictive performance: not always a tradeoff



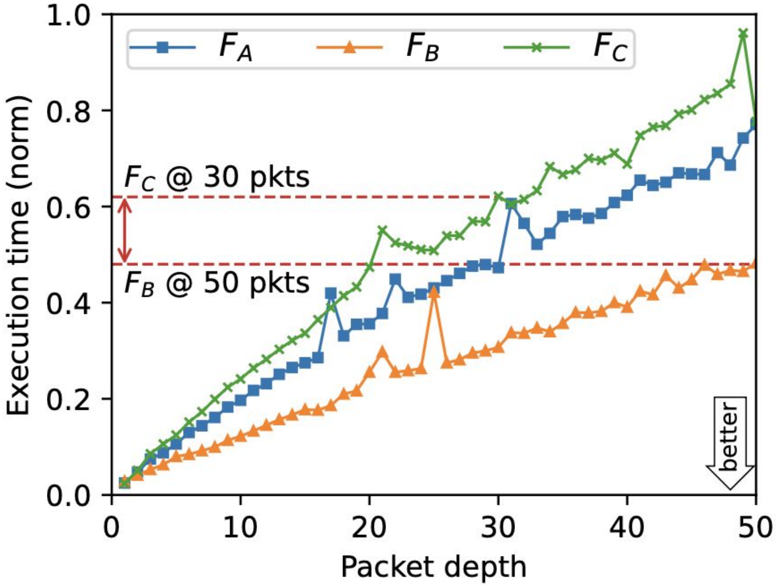
Systems costs vs. predictive performance: not always a tradeoff



Systems costs vs. predictive performance: not always a tradeoff



Waiting for more packets may or may not lead to higher accuracy



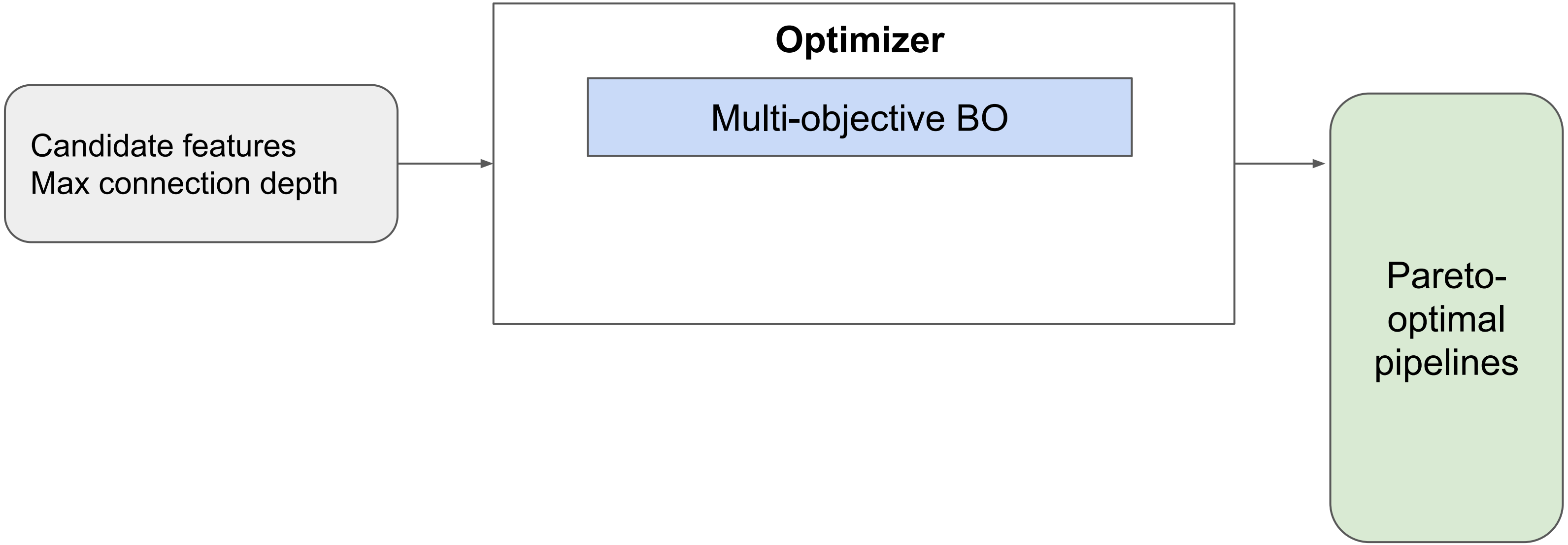
Problem summary

- ML-based traffic analysis must balance system costs and predictive performance
- Traffic representation choices have a large and non-obvious impact
- Multiple objectives and a huge search space makes end-to-end optimization challenging

CATO: Cost-Aware Traffic analysis Optimization

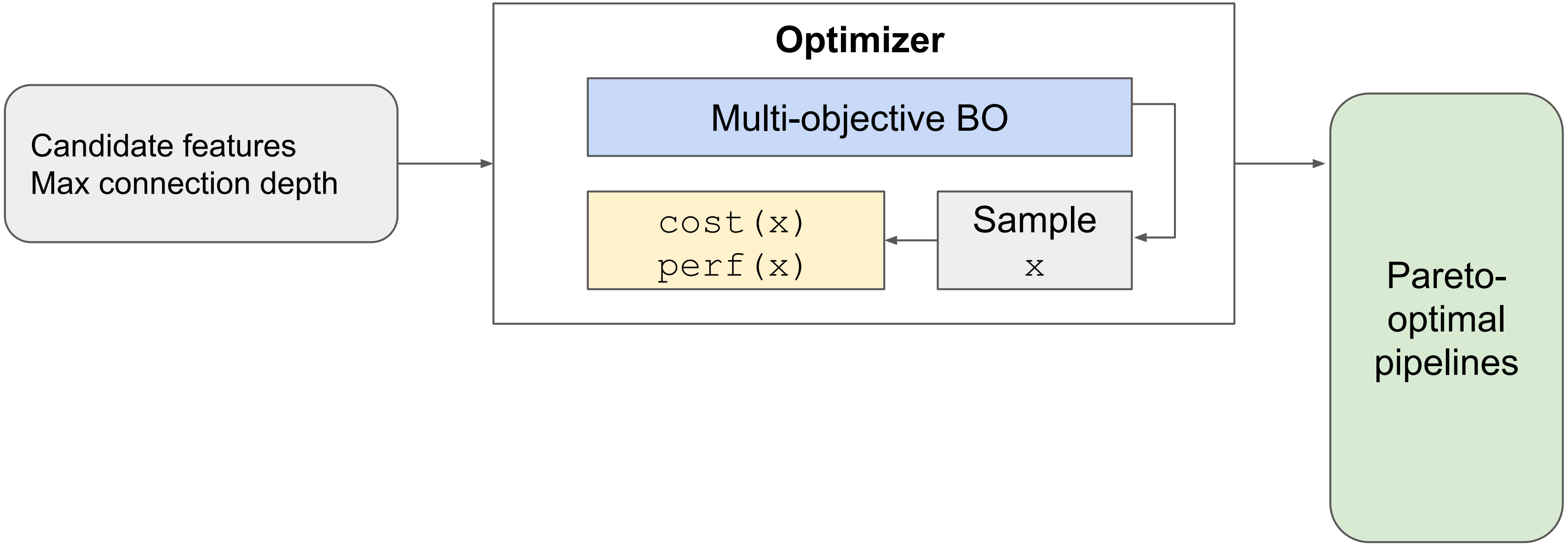
- Goal: automatically construct traffic analysis pipelines that jointly minimize end-to-end systems costs while maximizing predictive performance

CATO: Cost-Aware Traffic analysis Optimization



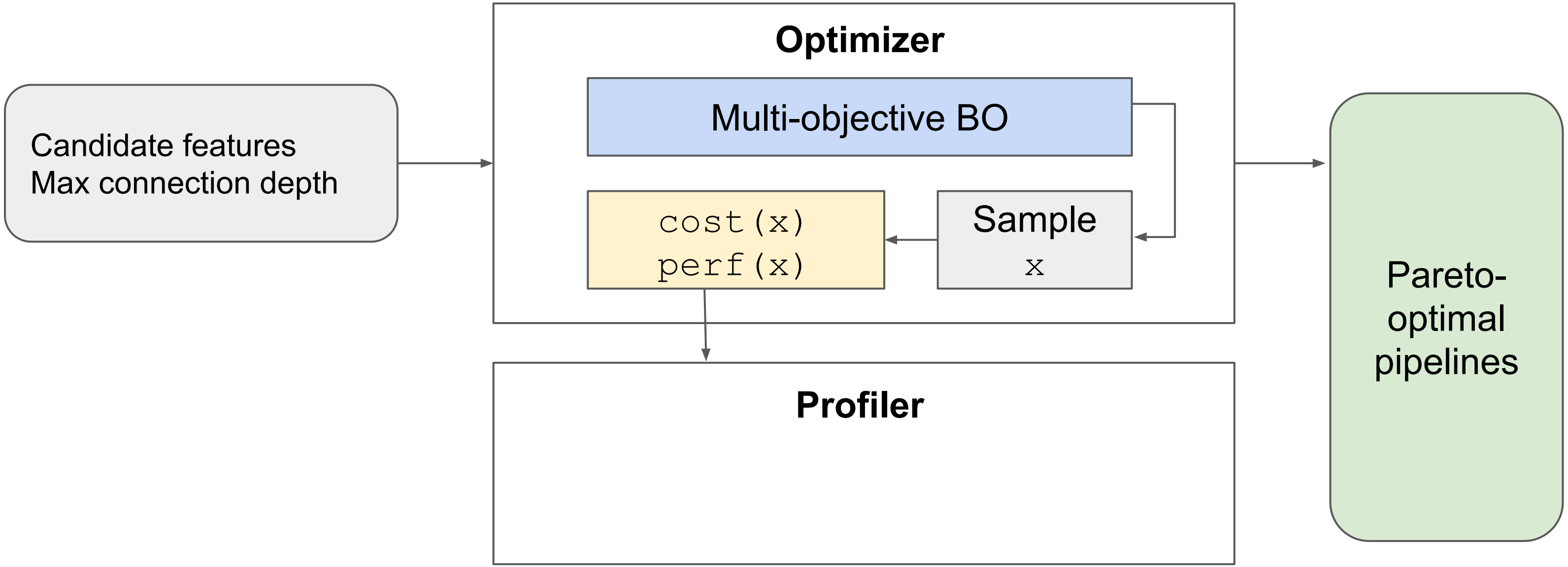
- Optimizer: BO-guided **search** for Pareto-optimal traffic representations
- Profiler: **Guides** optimizer towards Pareto-optimal solutions and **validates** in-network performance of traffic analysis pipelines

CATO: Cost-Aware Traffic analysis Optimization



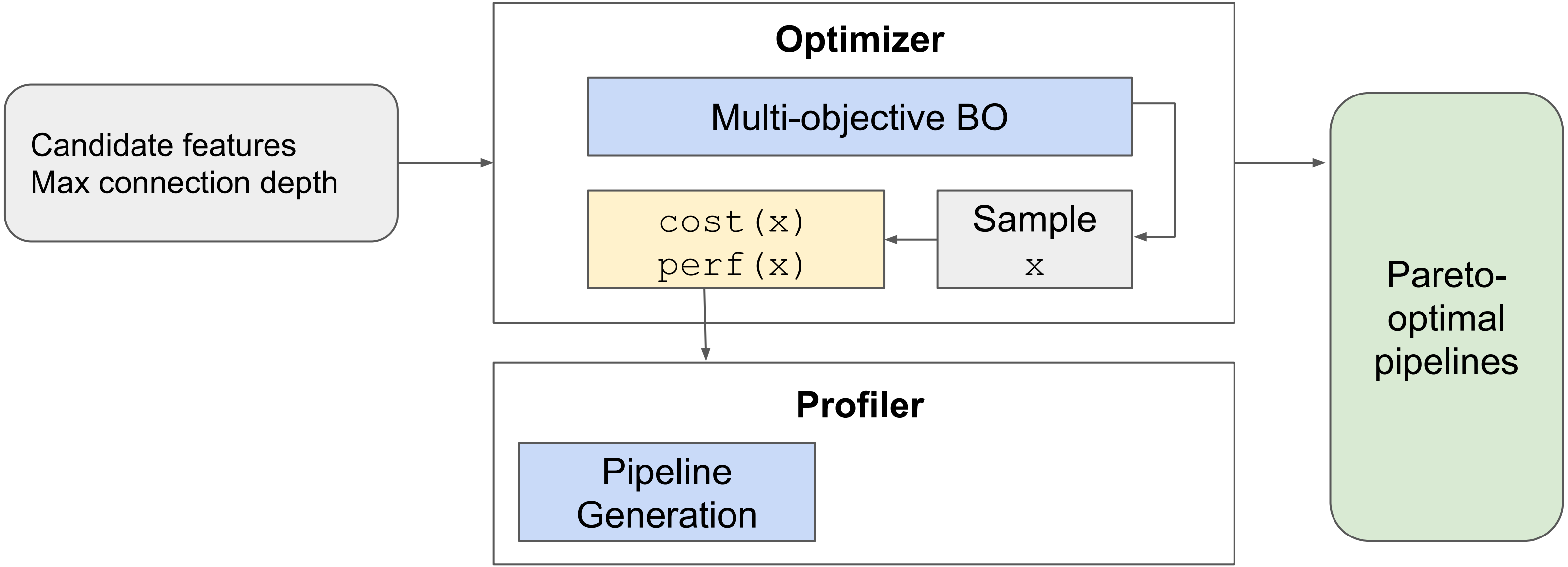
- Optimizer: BO-guided **search** for Pareto-optimal traffic representations
- Profiler: **Guides** optimizer towards Pareto-optimal solutions and **validates** in-network performance of traffic analysis pipelines

CATO: Cost-Aware Traffic analysis Optimization



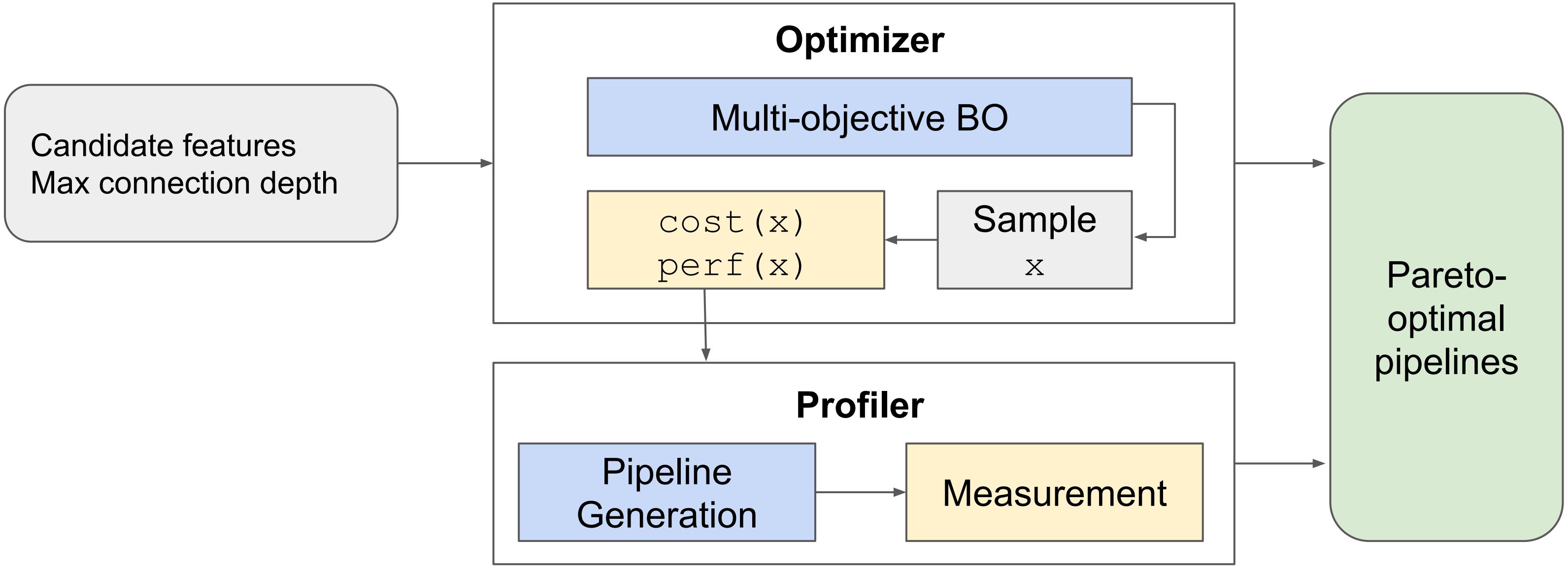
- Optimizer: BO-guided **search** for Pareto-optimal traffic representations
- Profiler: **Guides** optimizer towards Pareto-optimal solutions and **validates** in-network performance of traffic analysis pipelines

CATO: Cost-Aware Traffic analysis Optimization



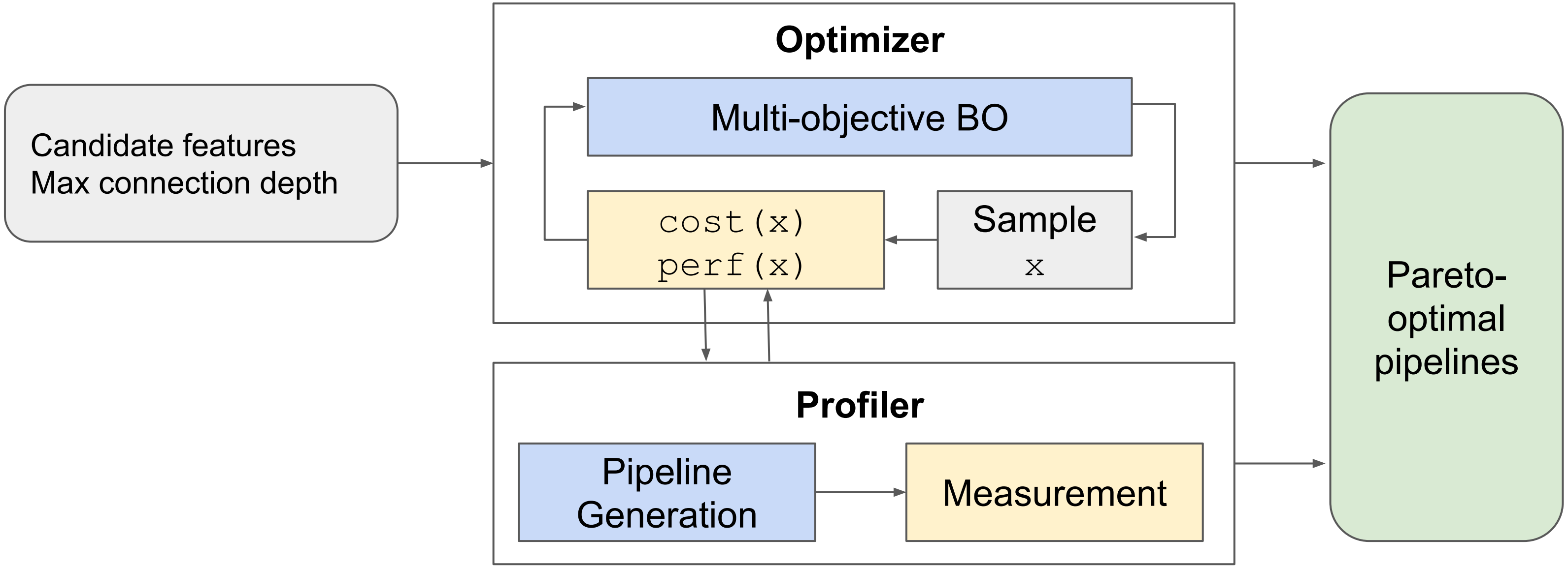
- Optimizer: BO-guided **search** for Pareto-optimal traffic representations
- Profiler: **Guides** optimizer towards Pareto-optimal solutions and **validates** in-network performance of traffic analysis pipelines

CATO: Cost-Aware Traffic analysis Optimization



- Optimizer: BO-guided **search** for Pareto-optimal traffic representations
- Profiler: **Guides** optimizer towards Pareto-optimal solutions and **validates** in-network performance of traffic analysis pipelines

CATO: Cost-Aware Traffic analysis Optimization



- Optimizer: BO-guided **search** for Pareto-optimal traffic representations
- Profiler: **Guides** optimizer towards Pareto-optimal solutions and **validates** in-network performance of traffic analysis pipelines

Direct end-to-end measurement with the profiler

Estimating end-to-end systems costs is difficult → Measure directly

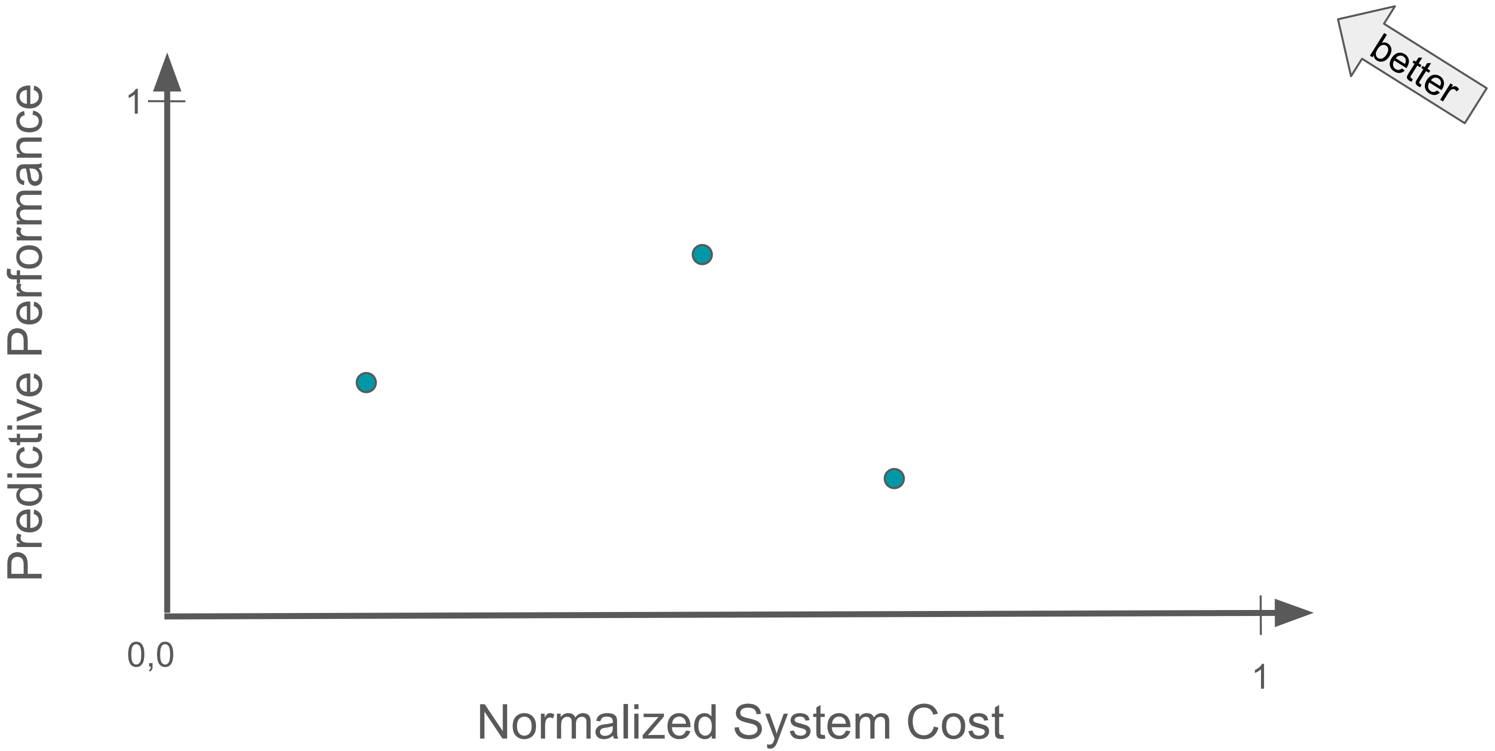
- Train the model, construct the pipeline, deploy, and measure.
- Expensive, but:
  - Helps the Optimizer **make better decisions**
  - **Validates** each traffic analysis pipeline

Multi-objective Bayesian Optimization

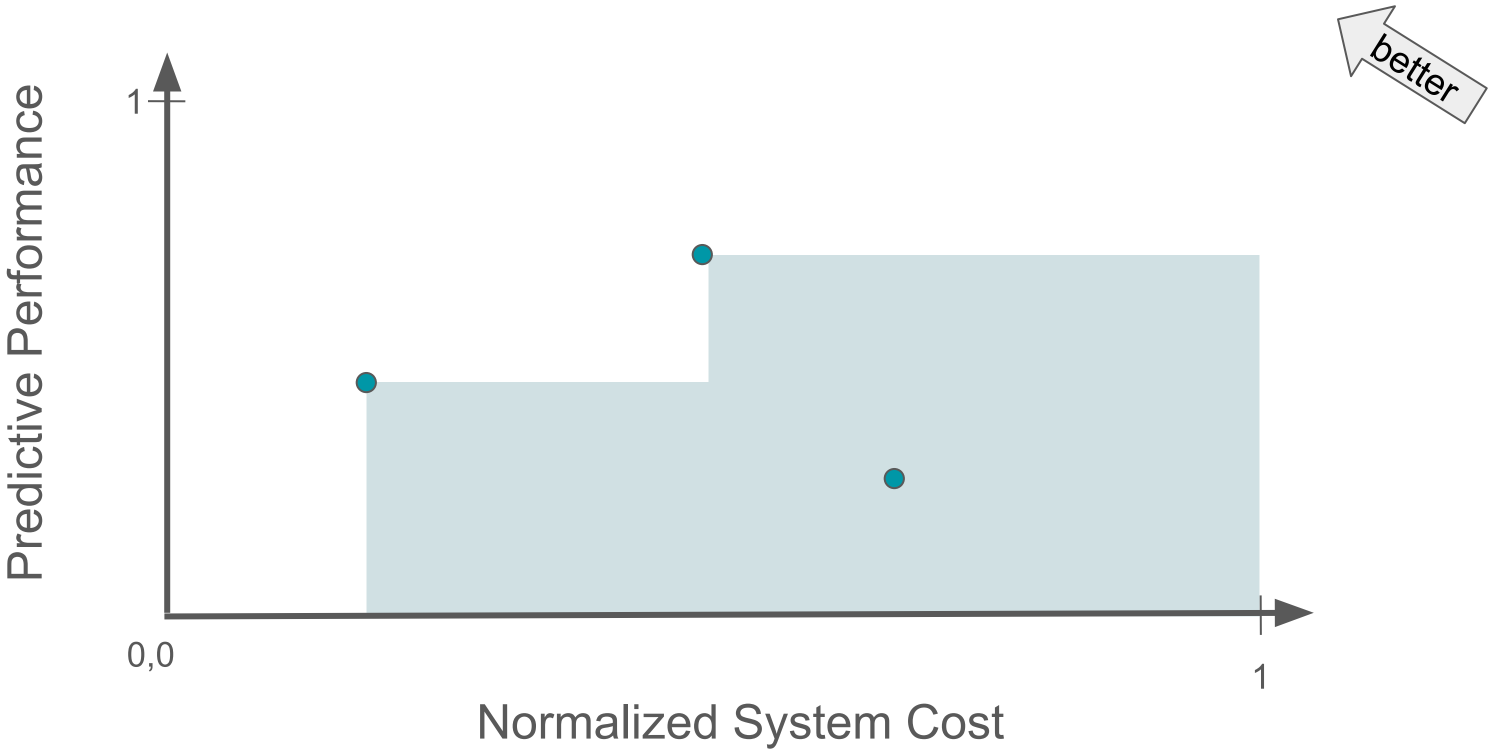
- Measuring systems costs and predictive performance is **expensive**
- **Sample-efficient** compared to other search methods

BO works well for expensive-to-evaluate, black box objectives

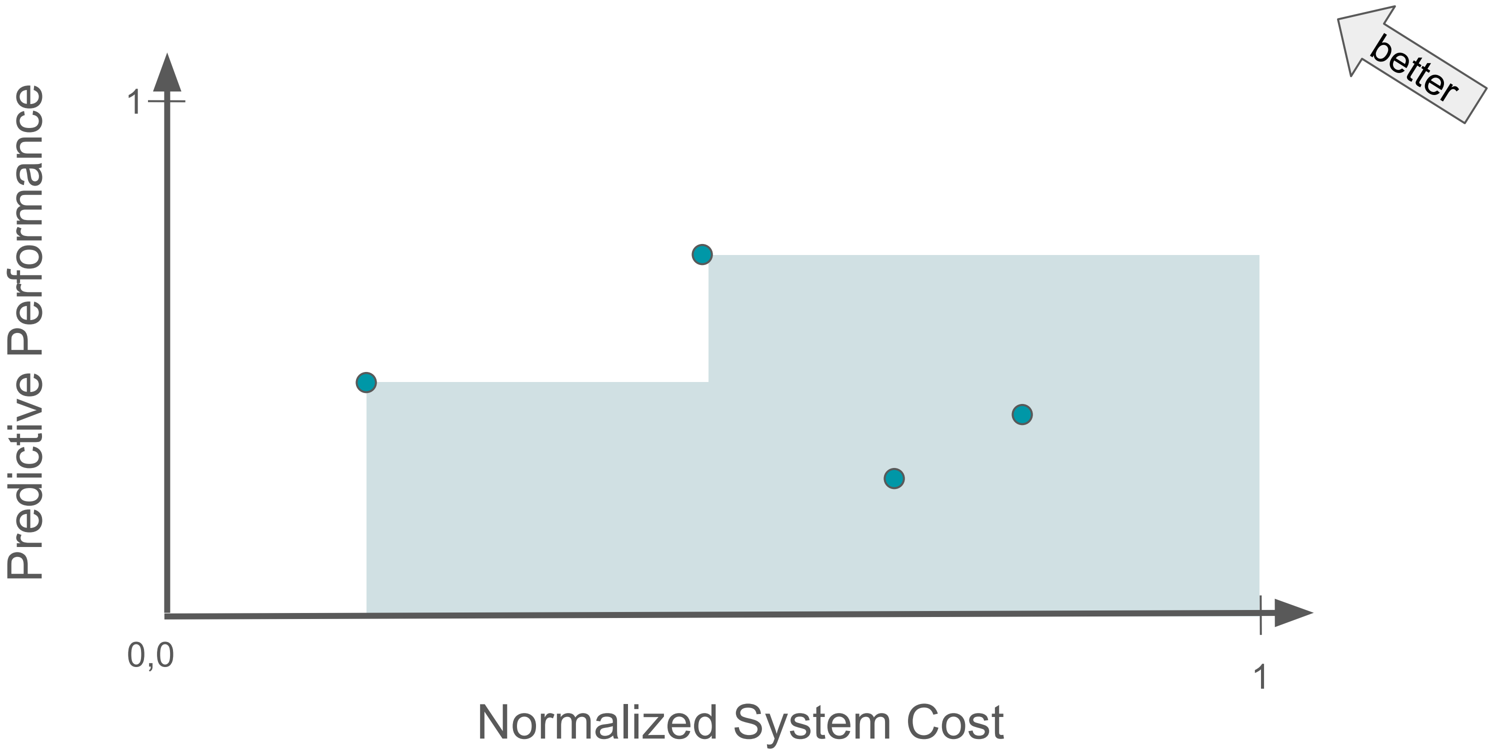
How CATO builds the Pareto front



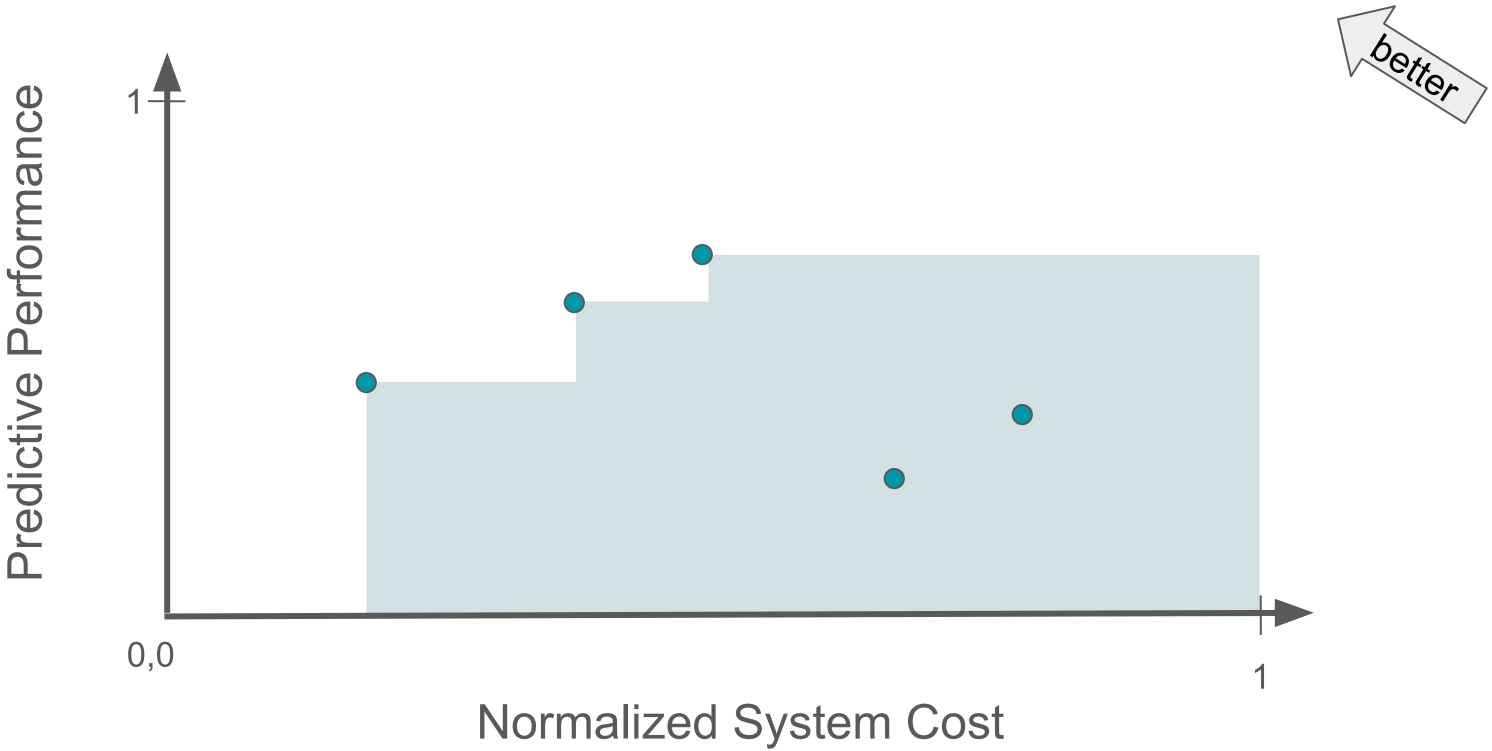
How CATO builds the Pareto front



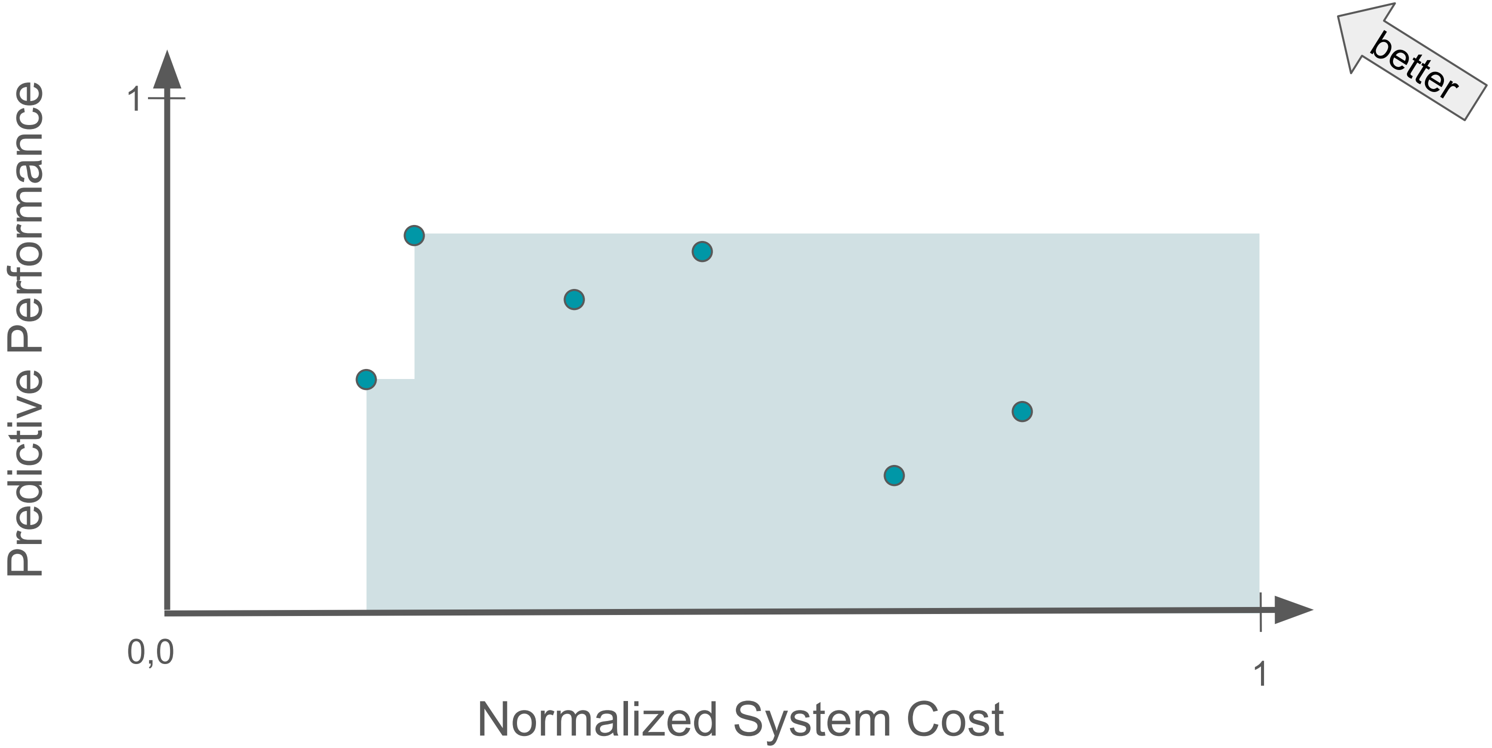
How CATO builds the Pareto front



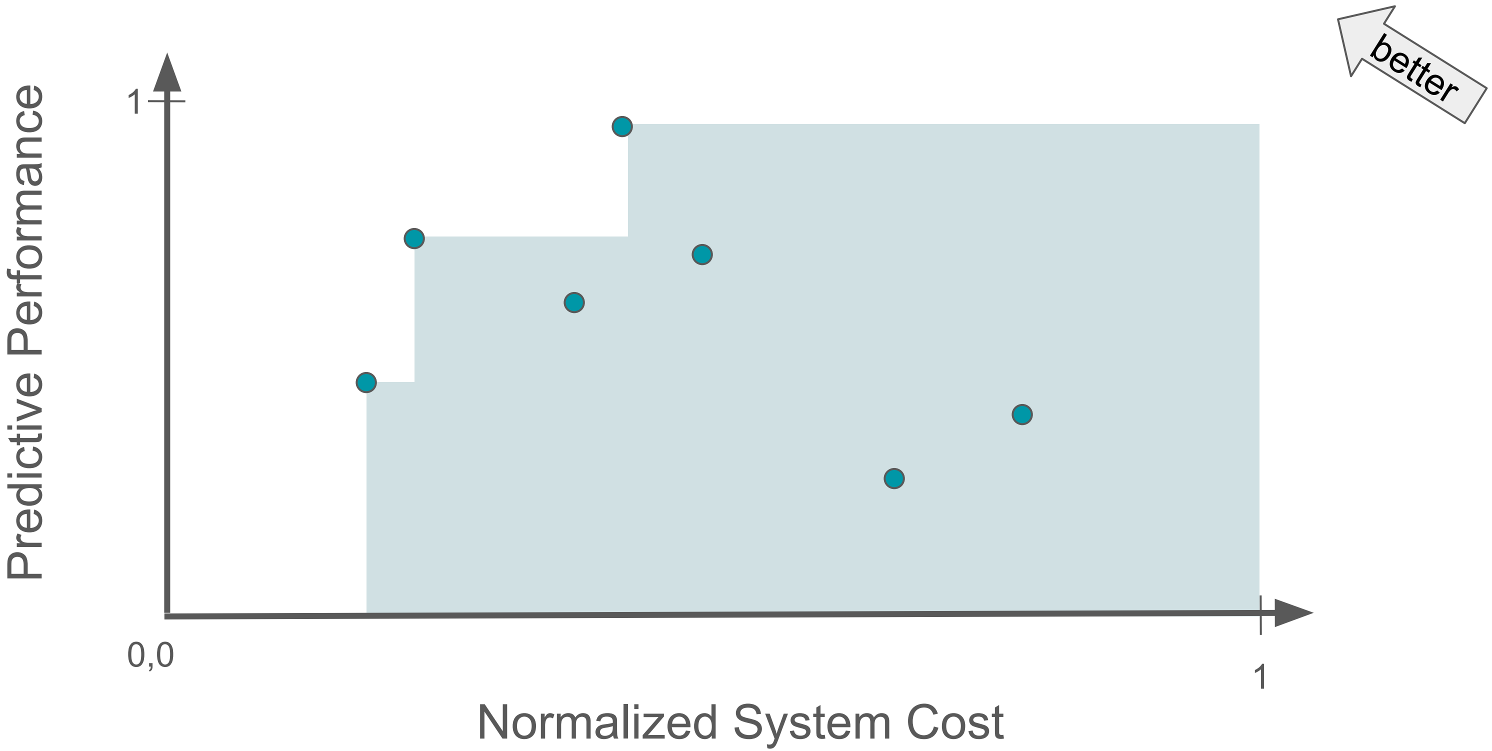
How CATO builds the Pareto front



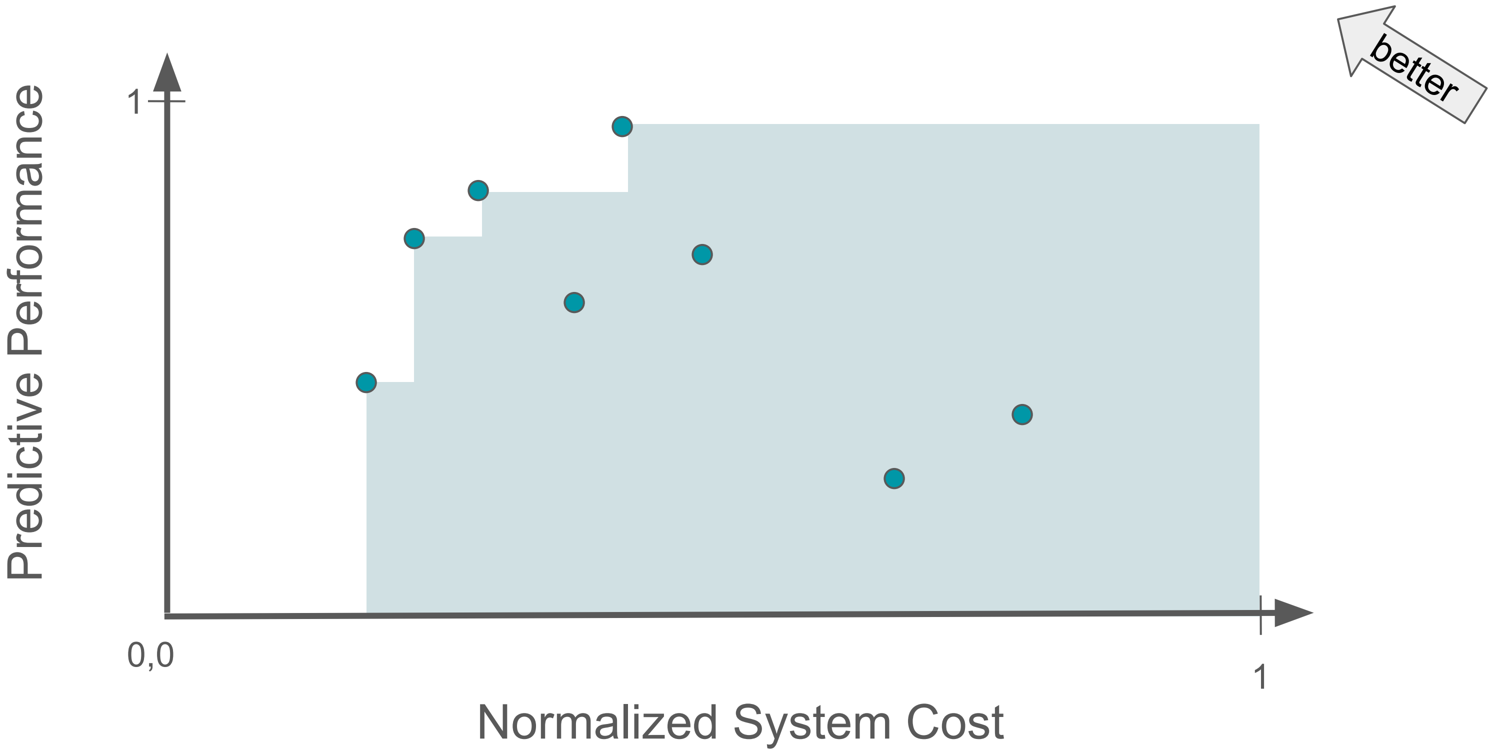
How CATO builds the Pareto front



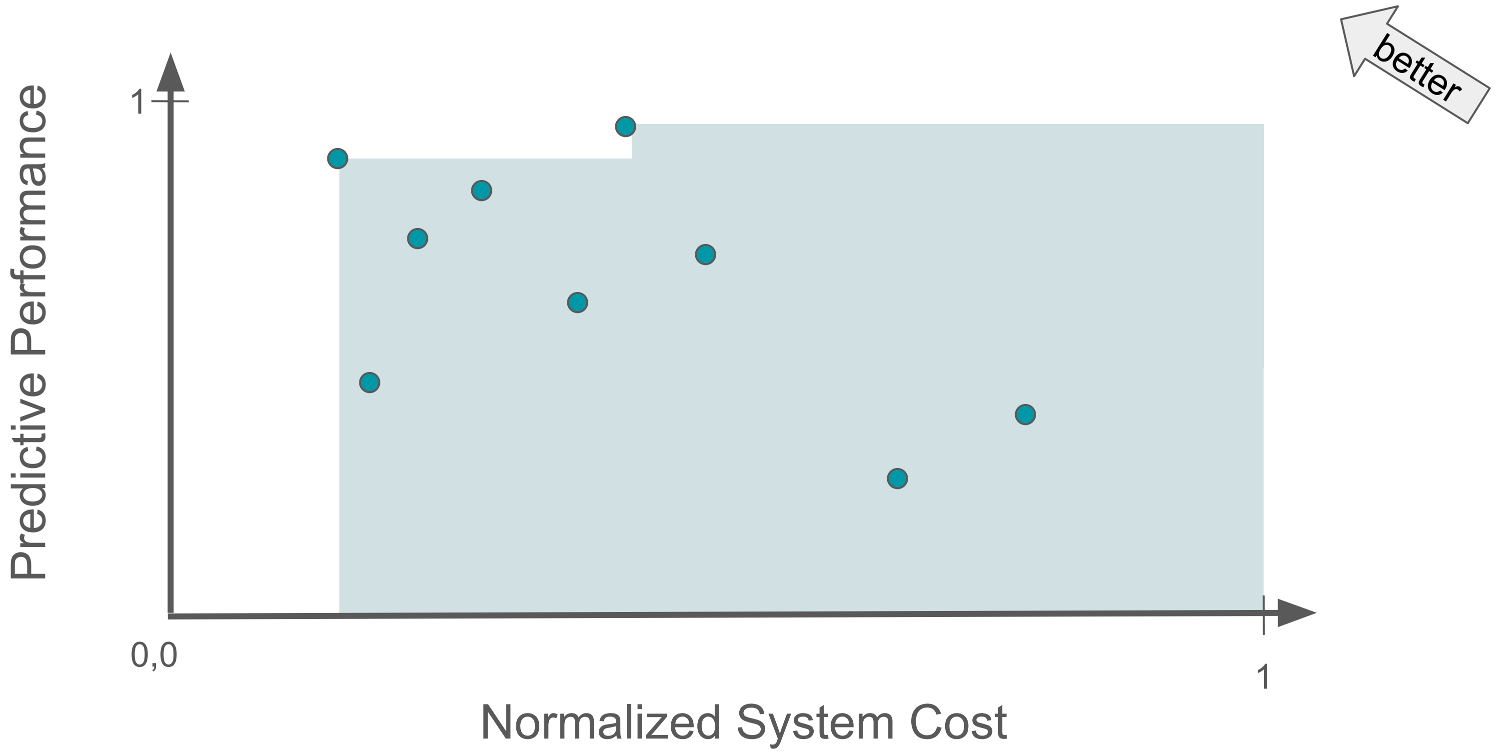
How CATO builds the Pareto front



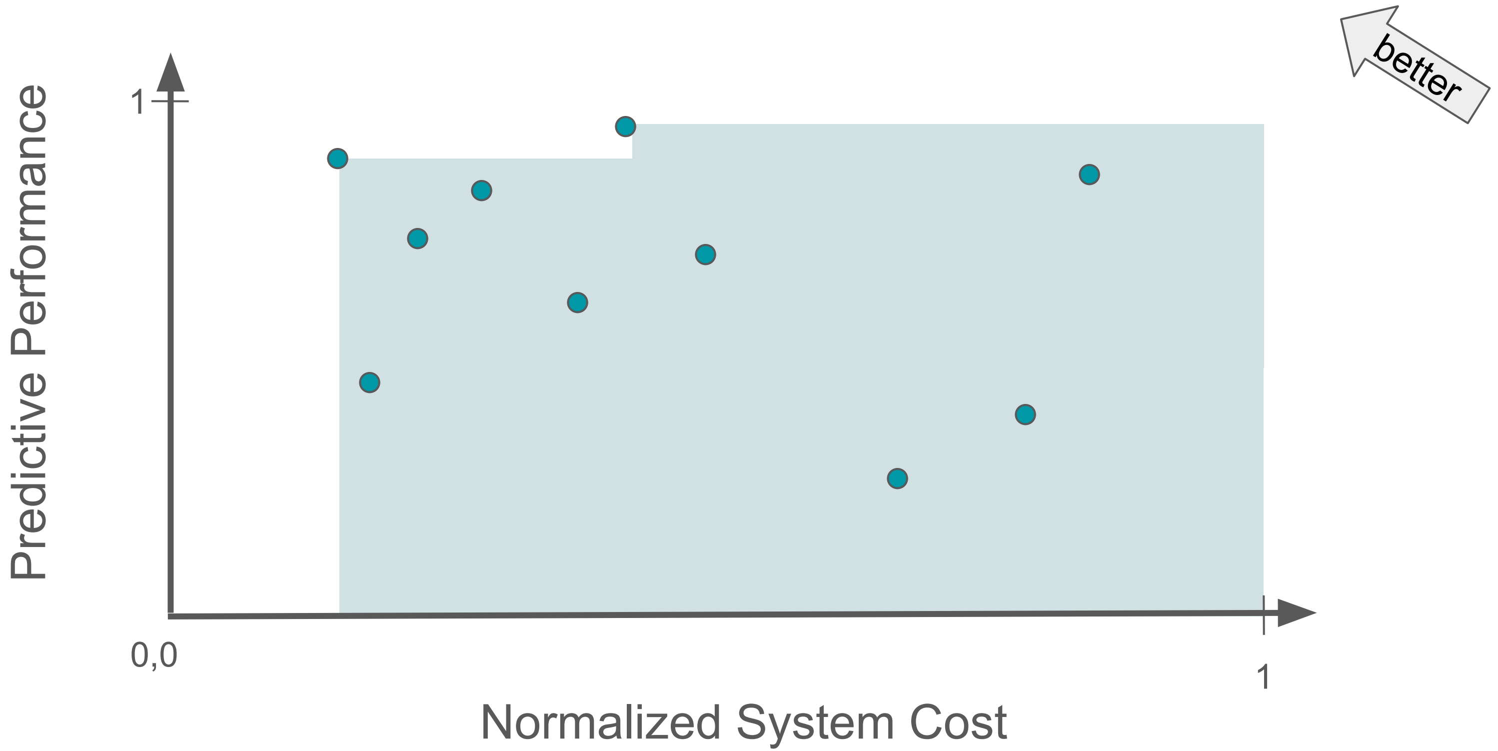
How CATO builds the Pareto front



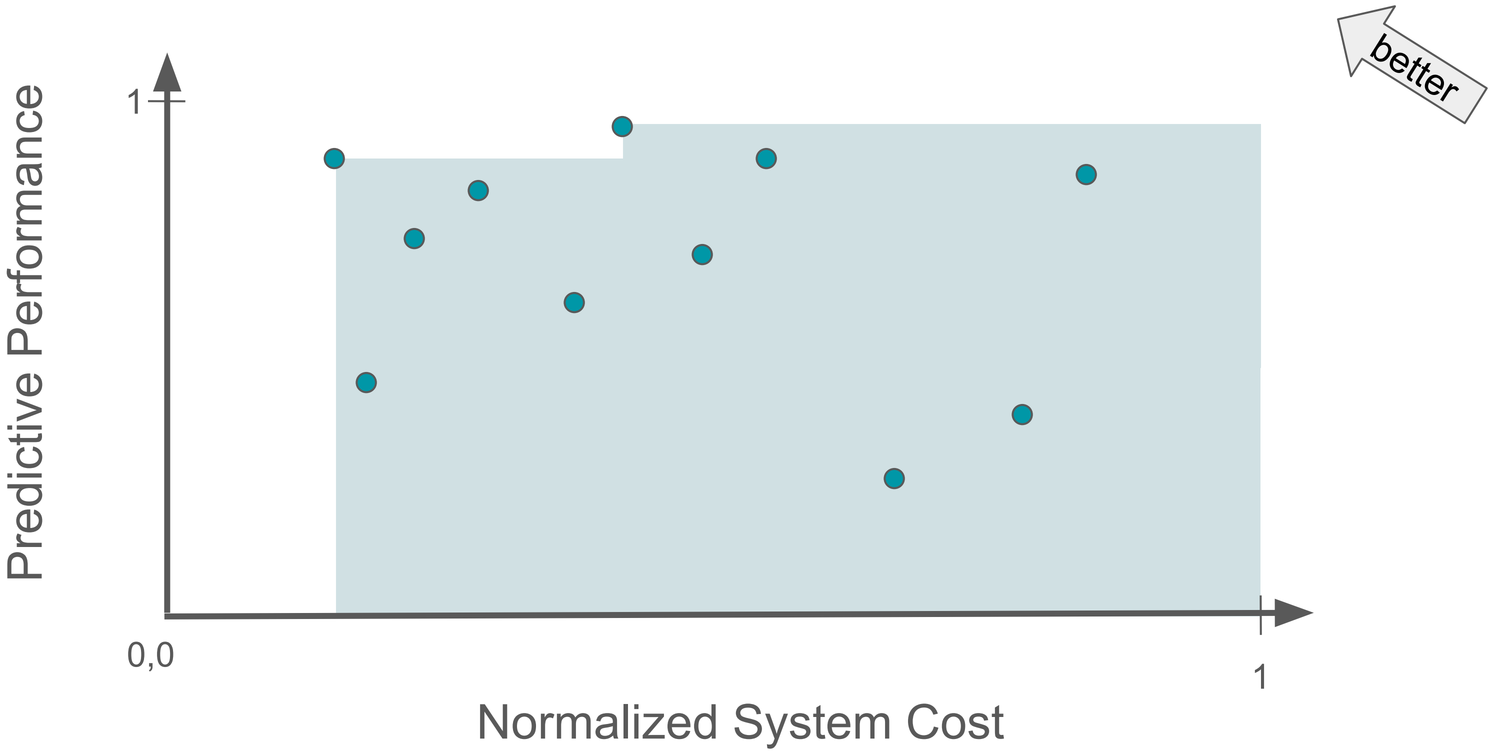
How CATO builds the Pareto front



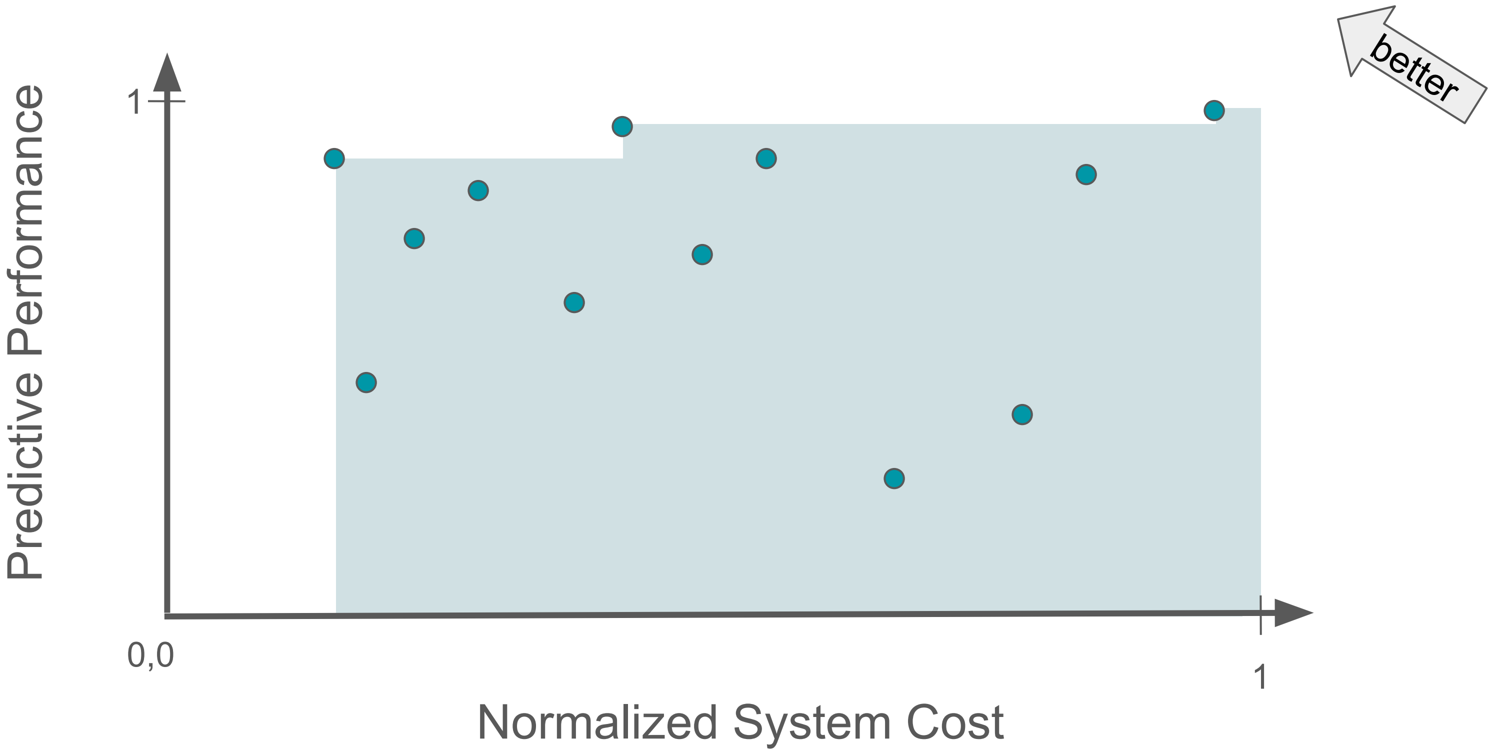
How CATO builds the Pareto front



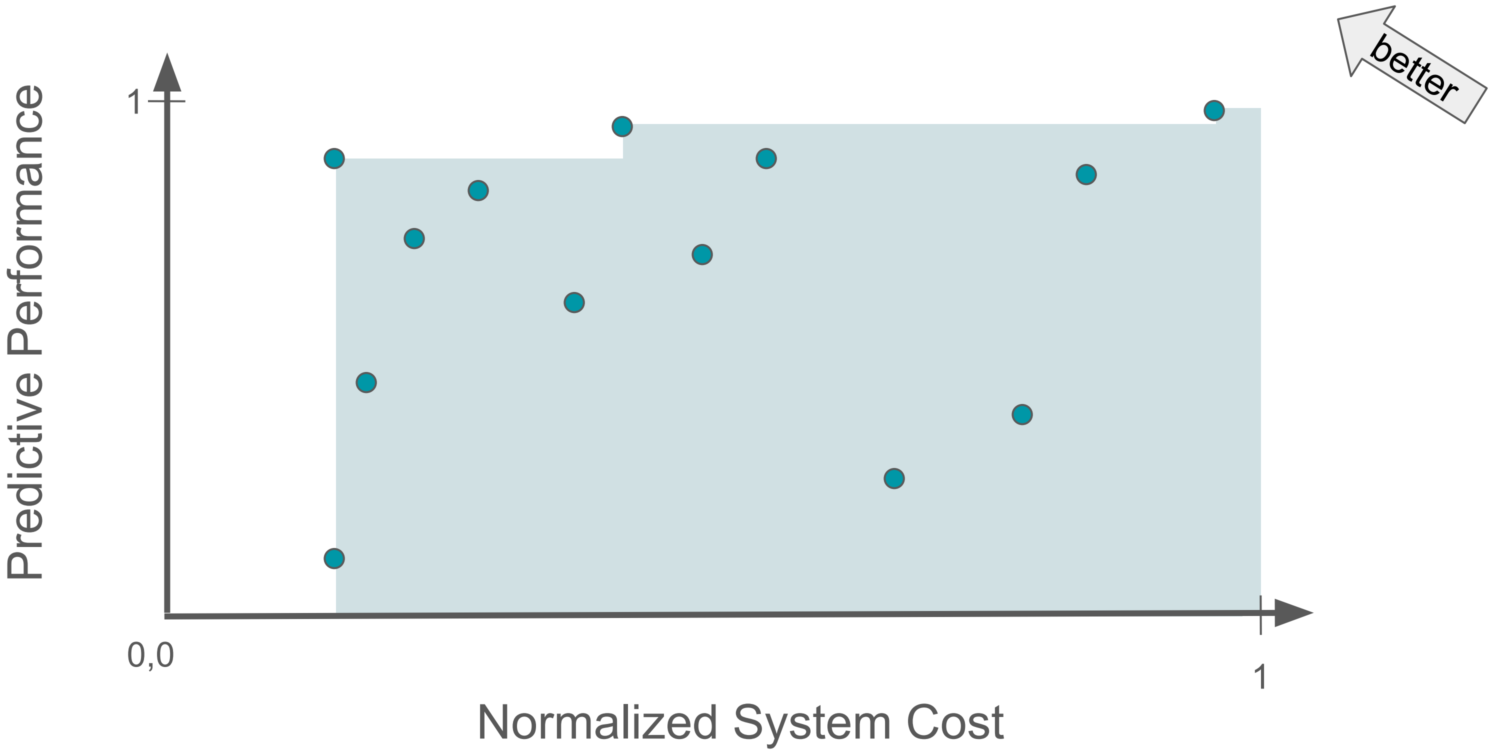
How CATO builds the Pareto front



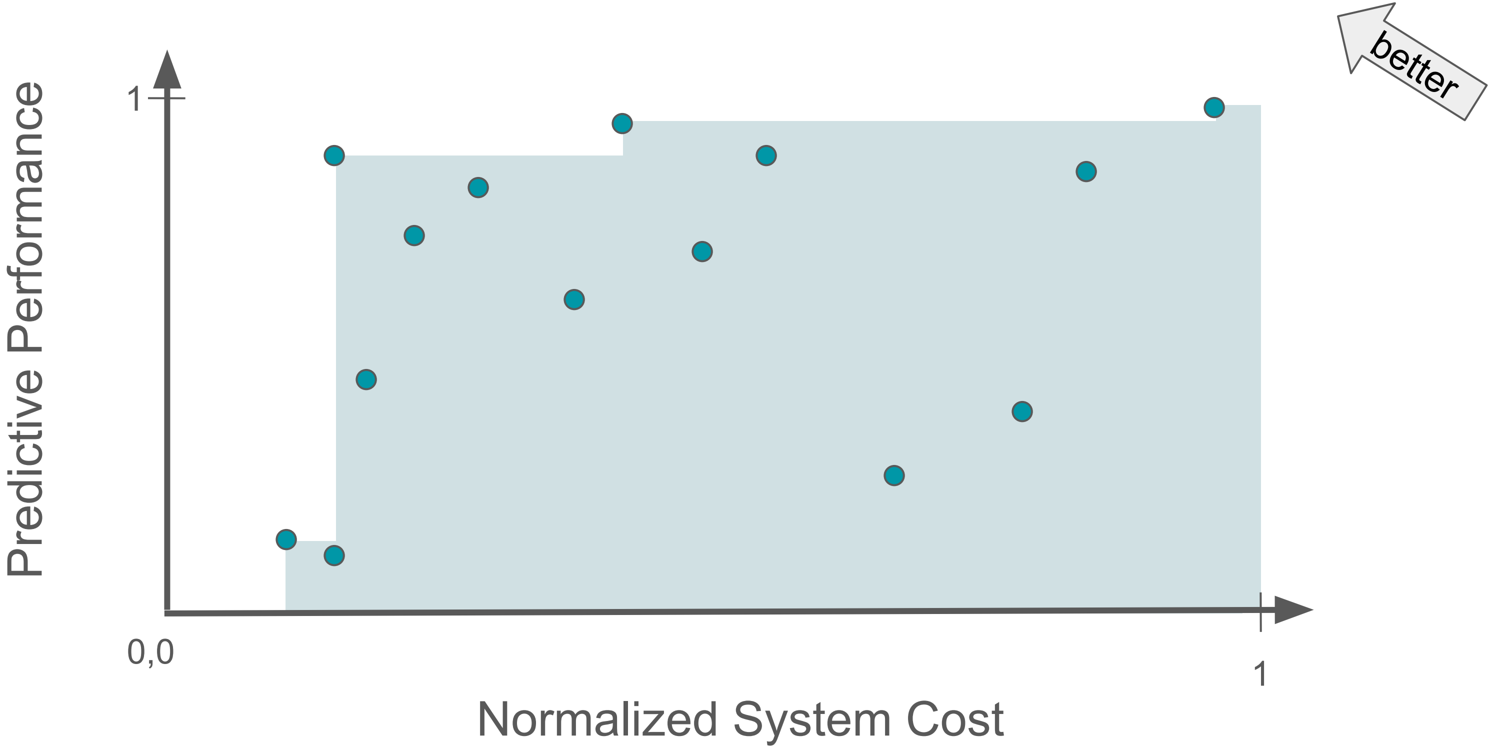
How CATO builds the Pareto front



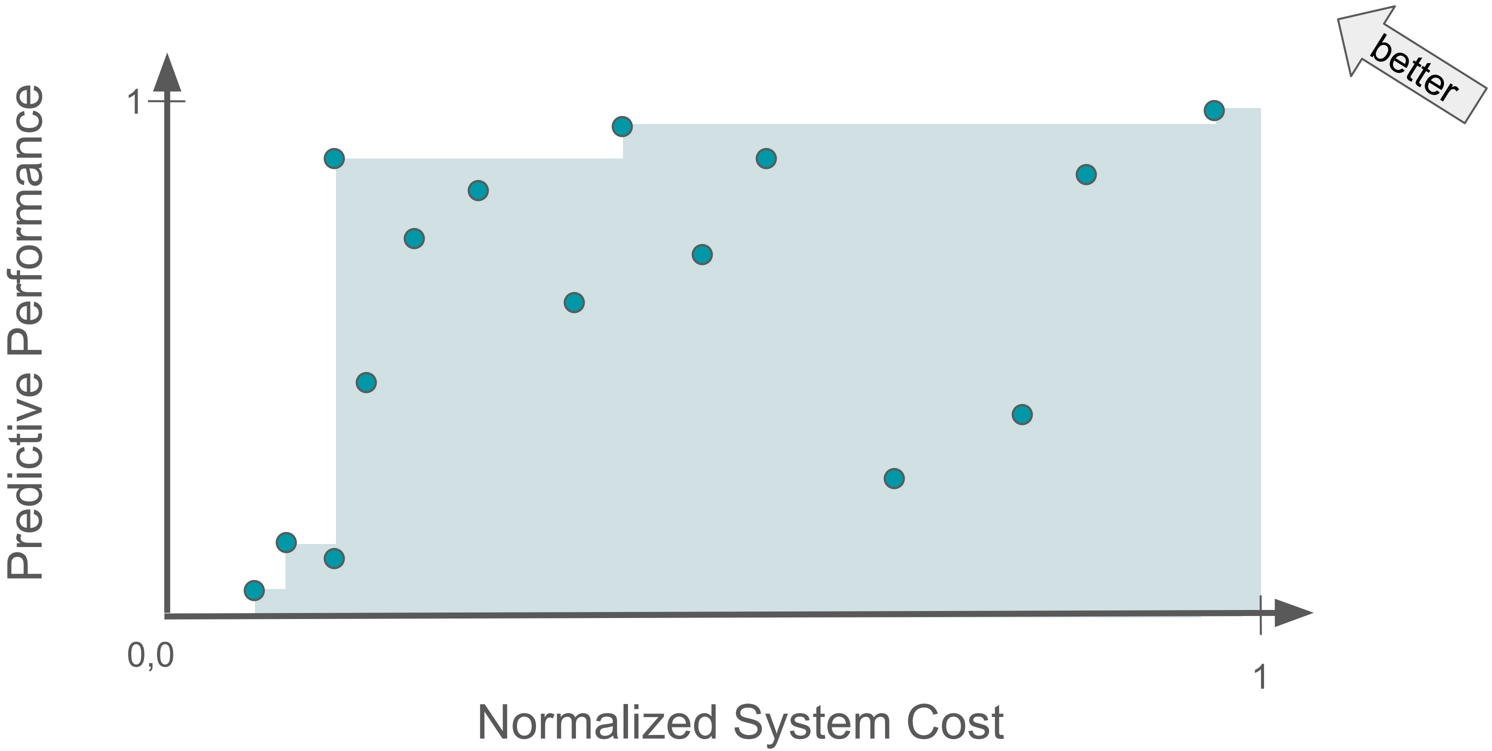
How CATO builds the Pareto front



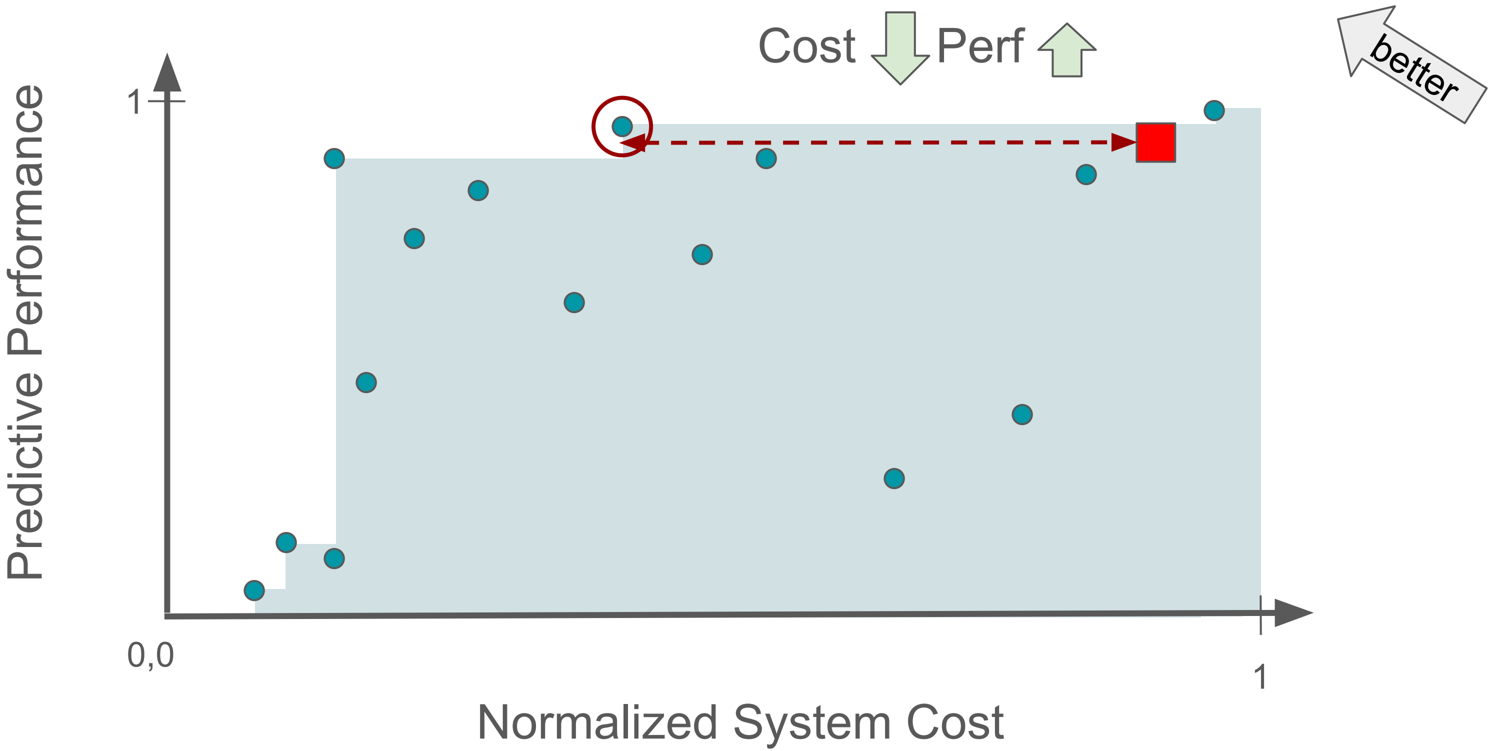
How CATO builds the Pareto front



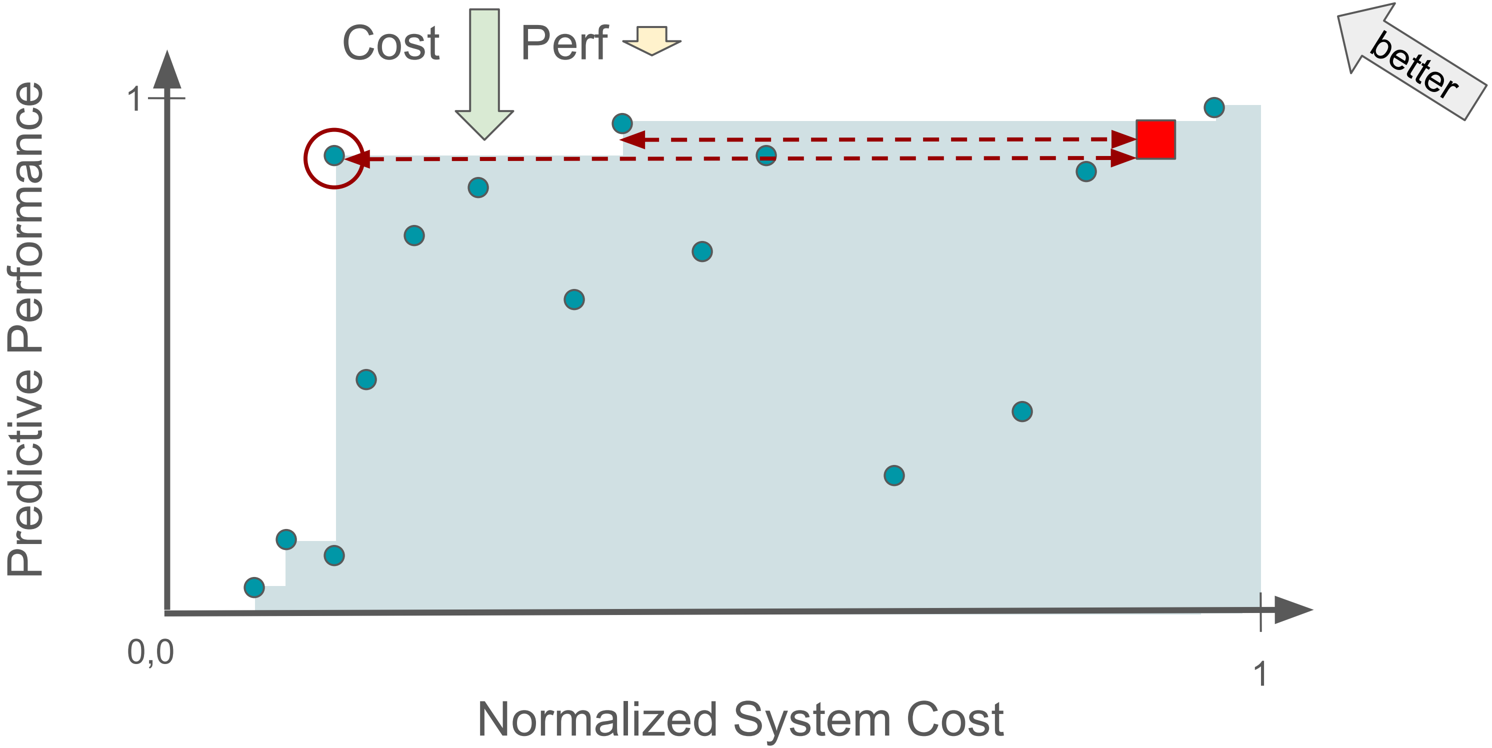
How CATO builds the Pareto front



CATO can achieve lower costs while increasing predictive performance

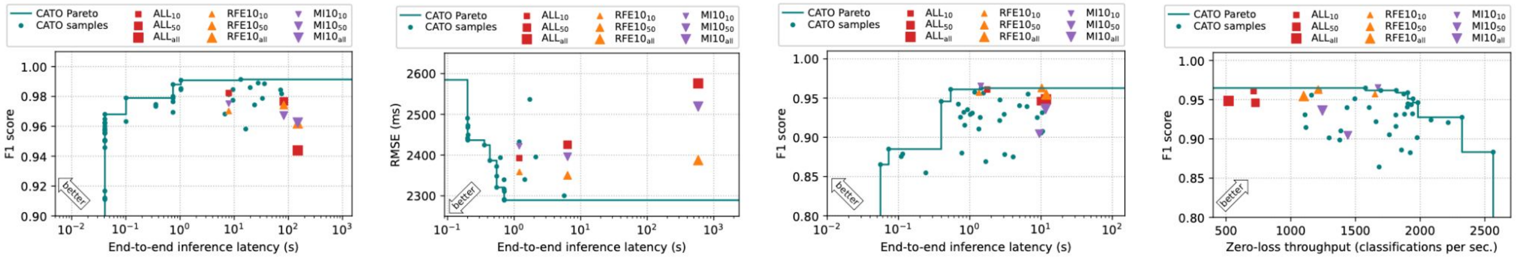


CATO can achieve much lower costs for similar predictive performance



Selected results

- E2E latency: 2.6–19x reduction vs. using first 10 packets
- Classification throughput: 1.6–3.7x speedup vs. using first 10 packets
- Convergence rate: 14.9–16.9x speedup over simulated annealing and random search



Summary

- End-to-end pipeline efficiency is critical for ML-based traffic analysis
- Traffic representation choices have an outsized impact on both predictive performance and systems costs

- **CATO**:
  - Jointly optimizes across **both** predictive performance and systems costs
  - Uses **multi-objective BO** with **direct end-to-end measurement** to construct and validate readily deployable traffic analysis pipelines