Stochastic L-Risk Minimization

Ronak Mehta

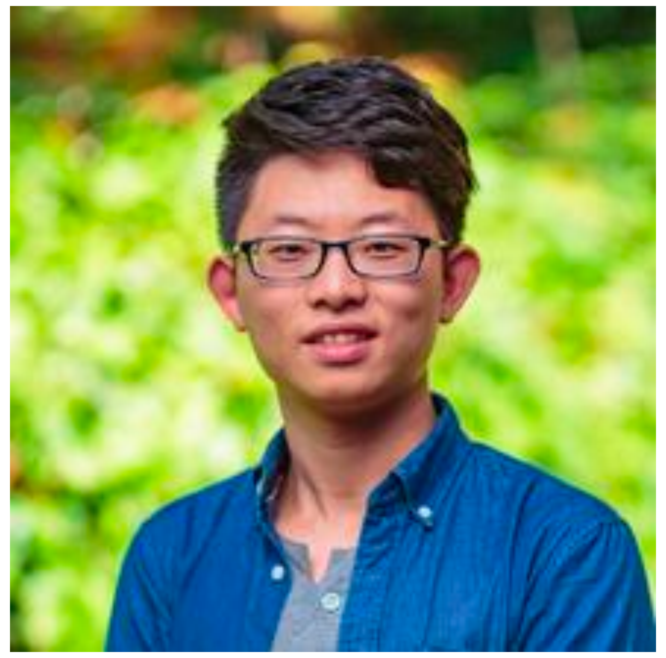
February 17, 2023

Team

Ronak Mehta
University of Washington

Vincent Roulet
Google Research

Krishna Pillutla
Google Research



Lang Liu
University of Washington

Zaid Harchaoui
University of Washington

IFDS
Institute for Foundations of Data Science

NSF

W
UNIVERSITY of WASHINGTON

Motivation: Average-Case ⟶ Worst-Case

- **Current learning paradigm:** optimize average performance of a model across all training examples.
- Averages are simple to analyze and admit efficient optimization algorithms.

Motivation: Average-Case → Worst-Case

- **Current learning paradigm:** optimize average performance of a model across all training examples.
- Averages are simple to analyze and admit efficient optimization algorithms.
- Worst-case performance can be relevant in practical applications.

'I'm the Operator': The Aftermath of a Self-Driving Tragedy

In 2018, an Uber autonomous vehicle fatally struck a pedestrian. In a WIRED exclusive, the human behind the wheel finally speaks.

2 Killed in Driverless Tesla Car Crash, Officials Say

"No one was driving the vehicle" when the car crashed and burst into flames, killing two men, a constable said.

A Tesla driver is charged in a crash involving Autopilot that killed 2 people

January 18, 2022 · 3:00 PM ET

Usual Setting

- $\ell_i(w)$ = loss on example i with parameters/weights $w \in \mathbb{R}^d$.

Empirical Risk Minimization (ERM):

$$\min_{w \in \mathbb{R}^d} \left[\mathcal{R}(w) := \frac{1}{n} \sum_{i=1}^{n} \ell_i(w) \right]$$

Our Setting

- $\ell_i(w)$ = loss on example i with parameters/weights $w \in \mathbb{R}^d$.
- $\ell_{(i)}(w)$ = i th order statistic of $\ell(w) = (\ell_1(w), \ldots, \ell_n(w))$.
- Constants $0 \leq \sigma_1 \leq \ldots \sigma_n, \sum_{i=1}^n \sigma_i = 1$ called **spectrum**.

L-Risk Minimization (LRM):

$$\min_{w \in \mathbb{R}^d} \left[\mathcal{R}_\sigma(w) := \sum_{i=1}^n \sigma_i \ell_{(i)}(w) \right]$$

Related Work and Challenges

- Alternative risk measures (functionals of a loss distribution) are well-established in quantitative finance (He, 2018; Rockafellar 2007; Cotter, 2006; Acerbi, 2002).
- Linear combinations of order statistics comprise a large class of “robust” statistical estimators (Huber, 2009), called L-statistics.
- Examples in machine learning include distributionally robust optimization (Chen, 2020), particularly using the superquantile L-risk (Laguel, 2021).

Related Work and Challenges

- Alternative risk measures (functionals of a loss distribution) are well-established in quantitative finance (He, 2018; Rockafellar 2007; Cotter, 2006; Acerbi, 2002).
- Linear combinations of order statistics comprise a large class of "robust" statistical estimators (Huber, 2009), called L-statistics.
- Examples in machine learning include distributionally robust optimization (Chen, 2020), particularly using the superquantile L-risk (Laguel, 2021).
- Previous optimization approaches are either full-batch (require $O(n)$ gradient evaluations per iterate) or are biased (do not converge to the minimum L-risk) (Levy, 2020; Kawaguchi 2020).
- **Open question:** does there exist a stochastic ($O(1)$ gradient calls per iteration) optimization algorithm that converges to the minimum L-risk?

Contributions

In this work, we:

1. Characterize the subdifferential and continuity properties of the objective.
2. Prove statistical consistency of L-risks for their population counterpart.
3. Quantify the bias of current stochastic approaches.
4. Propose a linearly convergent stochastic algorithm for L-risks.
5. Demonstrate superior convergence of the method on numerical evaluations.

Outline

- **Statistical properties of L-risks.**
- Optimization properties of the L-risks.
- Stochastic optimization algorithms.
- Experimental evaluations.

Consistency

$$\min_{w\in\mathbb{R}^d}\left[\mathcal{R}(w) := \frac{1}{n}\sum_{i=1}^{n}\ell_i(w)\right] \longrightarrow \min_{w\in\mathbb{R}^d}\left[\mathcal{R}_\sigma(w) := \sum_{i=1}^{n}\sigma_i\ell_{(i)}(w)\right]$$

- In ERM, the quantity $\mathcal{R}(w)$ estimates the expected loss in on unseen test example.
- What does $\mathcal{R}_\sigma(w)$ estimate, and with what efficiency?

Statistical Setting

$$Z_1, \ldots, Z_n \sim F$$ i.i.d. sample

$$F_n(x) = (1/n) \sum_{i=1}^{n} \mathbb{1}\left(Z_i \leq x\right)$$ empirical CDF

$$Z_{(1)} \leq \ldots Z_{(n)}$$ order statistics

$$\sum_{i=1}^{n} \sigma_i Z_{(i)}$$ L-estimator (*)

Statistical Setting

$$Z_1, \ldots, Z_n \sim F$$ i.i.d. sample

$$F_n(x) = (1/n) \sum_{i=1}^{n} \mathbb{1}\left(Z_i \leq x\right)$$ empirical CDF

$$Z_{(1)} \leq \ldots Z_{(n)}$$ order statistics

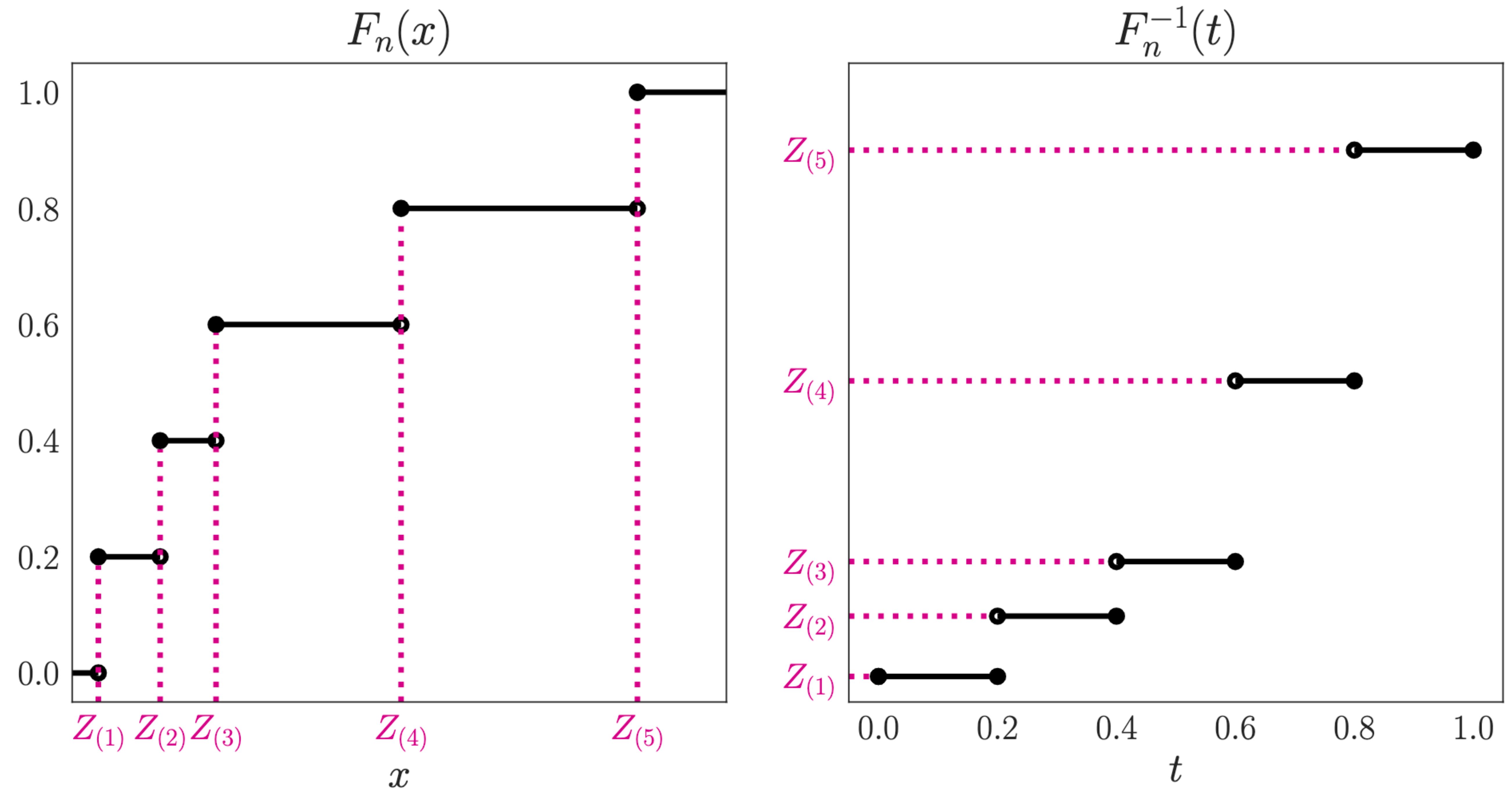
$$\sum_{i=1}^{n} \sigma_i Z_{(i)}$$ L-estimator (*)

- **Goal:** show (*) = $\mathbb{L}_s[F_n]$ for some functional $\mathbb{L}_s$, and that, in probability,

$$\mathbb{L}_s[F_n] \to \mathbb{L}_s[F]$$

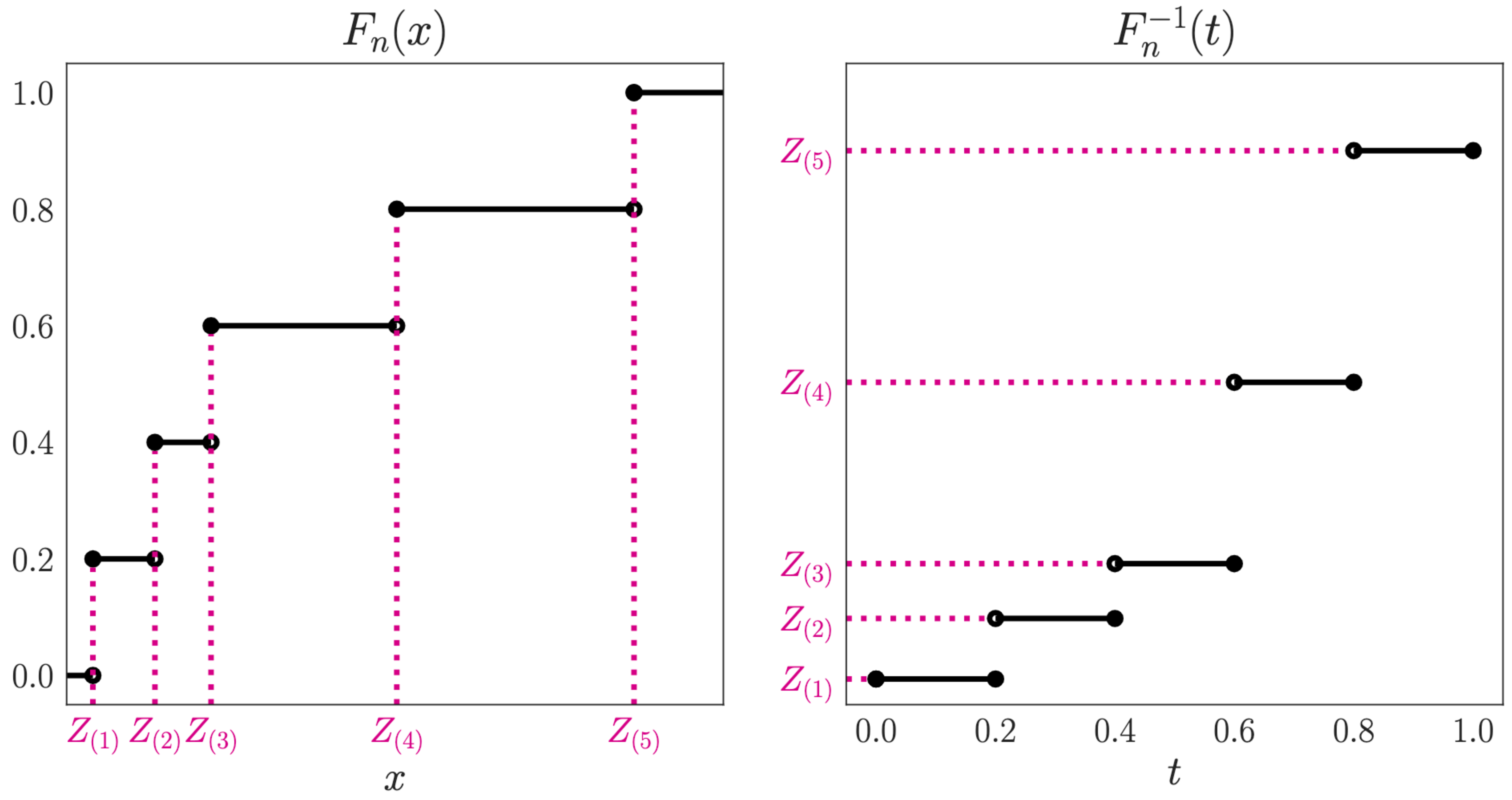
Step 1: Quantile Function

- $F^{-1}(t) = \inf\{x : F(x) \geq t\}$ and $F_n^{-1}(t) = \inf\{x : F_n(x) \geq t\}$ are **quantile** functions.



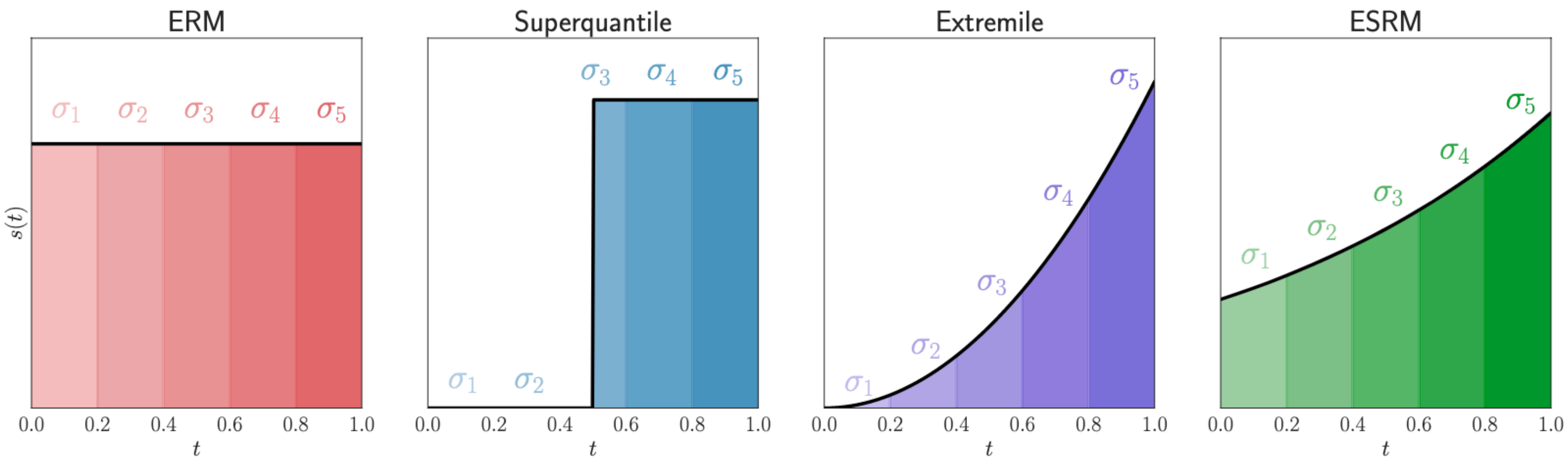
Step 1: Quantile Function

- $F^{-1}(t) = \inf\{x : F(x) \geq t\}$ and $F_n^{-1}(t) = \inf\{x : F_n(x) \geq t\}$ are **quantile** functions.
- Note that $F_n^{-1}(t) = Z_{(i)}$ when $t \in \left(\frac{i-1}{n}, \frac{i}{n}\right)$.



Step 2: Spectrum

- The spectrum $\sigma_1 \leq \ldots \leq \sigma_n$ is assumed to be the discretization of a probability distribution s on $(0,1)$, i.e. $\sigma_i = \int_{(i-1)/n}^{i/n} s(t)dt$.



Spectral Risk Measures

- Let $\mathbb{L}_s[F] = \int_0^1 s(t) \cdot F^{-1}(t)$. Then,

$$\begin{aligned}
\sum_{i=1}^{n} \sigma_i Z_{(i)} &= \sum_{i=1}^{n} \left(\int_{(i-1)/n}^{i/n} s(t)\, \mathrm{d}t \right) \cdot Z_{(i)} \\
&= \sum_{i=1}^{n} \int_{(i-1)/n}^{i/n} s(t) F_n^{-1}(t)\, \mathrm{d}t \\
&= \int_0^1 s(t) \cdot F_n^{-1}(t)\, \mathrm{d}t
\end{aligned}$$

Spectral Risk Measures

- Let $\mathbb{L}_s[F] = \int_0^1 s(t) \cdot F^{-1}(t)$. Then,

$$\begin{aligned}\sum_{i=1}^{n} \sigma_i Z_{(i)} &= \sum_{i=1}^{n} \left(\int_{(i-1)/n}^{i/n} s(t)\, \mathrm{d}t \right) \cdot Z_{(i)} \\ &= \sum_{i=1}^{n} \int_{(i-1)/n}^{i/n} s(t) F_n^{-1}(t)\, \mathrm{d}t \\ &= \int_0^1 s(t) \cdot F_n^{-1}(t)\, \mathrm{d}t\end{aligned}$$

- The functional $\mathbb{L}_s$ is called a **spectral risk measure** with **spectrum** s.

Consistency

Proposition 1. *Assume that* $\mathbb{E}\left|Z\right|^p < \infty$ *for some* $p > 2$ *and that* $\|s\|_\infty := \sup_{t\in(0,1)} |s(t)| < \infty$. *Then,*

$$\mathbb{E}\left|\mathbb{L}_s\left[F_n\right] - \mathbb{L}_s\left[F\right]\right|^2 = O\left(\frac{1}{n}\right).$$

Consistency

Proposition 1. *Assume that* $\mathbb{E}\,|Z|^p < \infty$ *for some* $p > 2$ *and that* $\|s\|_\infty := \sup_{t\in(0,1)} |s(t)| < \infty$. *Then,*

$$\mathbb{E}\left|\mathbb{L}_s\left[F_n\right] - \mathbb{L}_s\left[F\right]\right|^2 = O\left(\frac{1}{n}\right).$$

- The above only requires boundedness of s and moment condition on Z.
- Related results require either boundedness of Z, Lipschitz continuity of s, or trimming of s ($s(t) = 0$ for $t \in [0,\alpha) \cup (\alpha,1]$).

Outline

- Statistical properties of L-risks.
- **Optimization properties of the L-risks.**
- Stochastic optimization algorithms.
- Experimental evaluations.

Optimization Setting

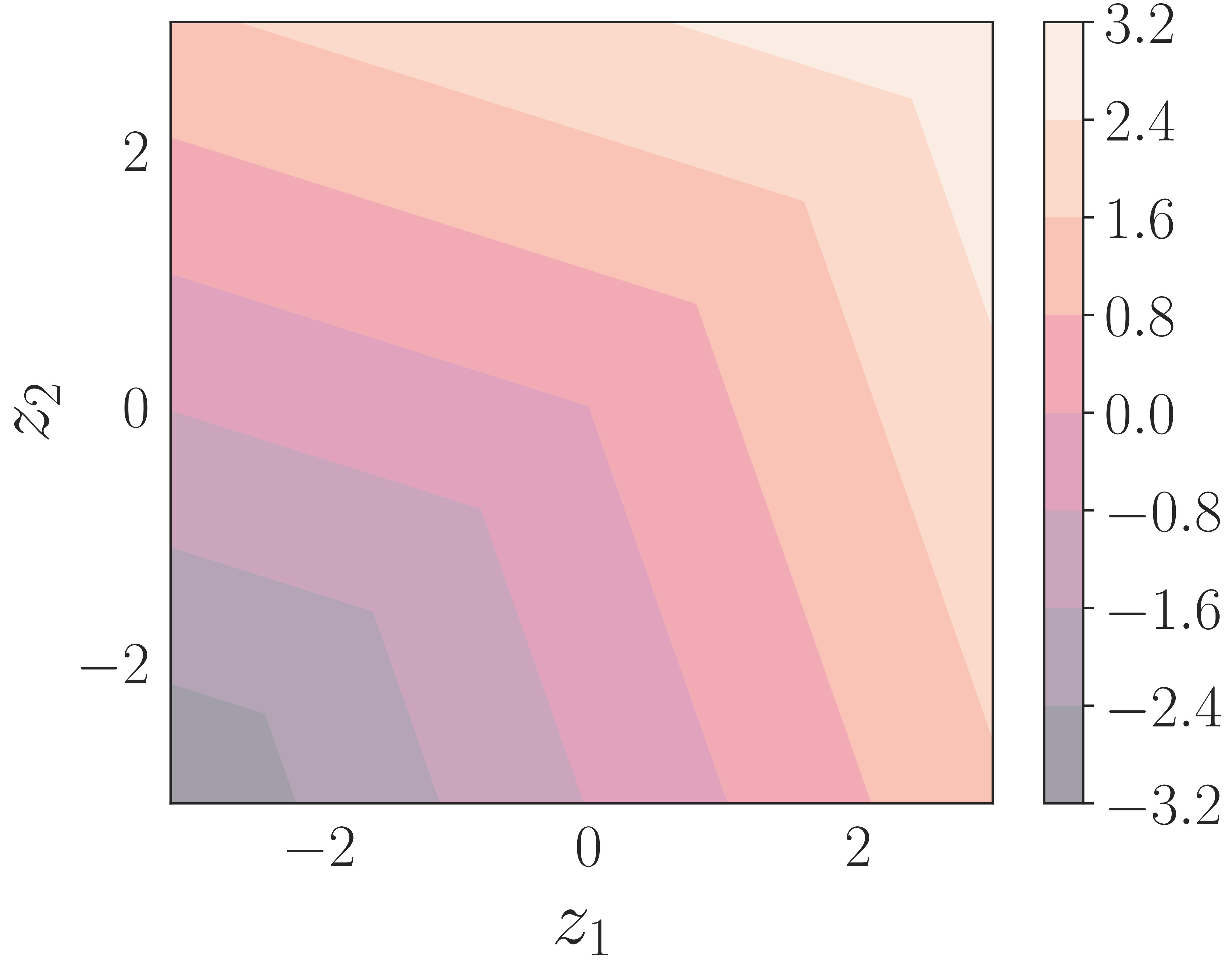
- Recall the original problem:

$$\min_{w \in \mathbb{R}^d} \left[\mathcal{R}_\sigma(w) := \sum_{i=1}^{n} \sigma_i \ell_{(i)}(w) \right]$$

- Is the objective convex?
- Is the objective smooth?
- How to compute (sub)gradients?

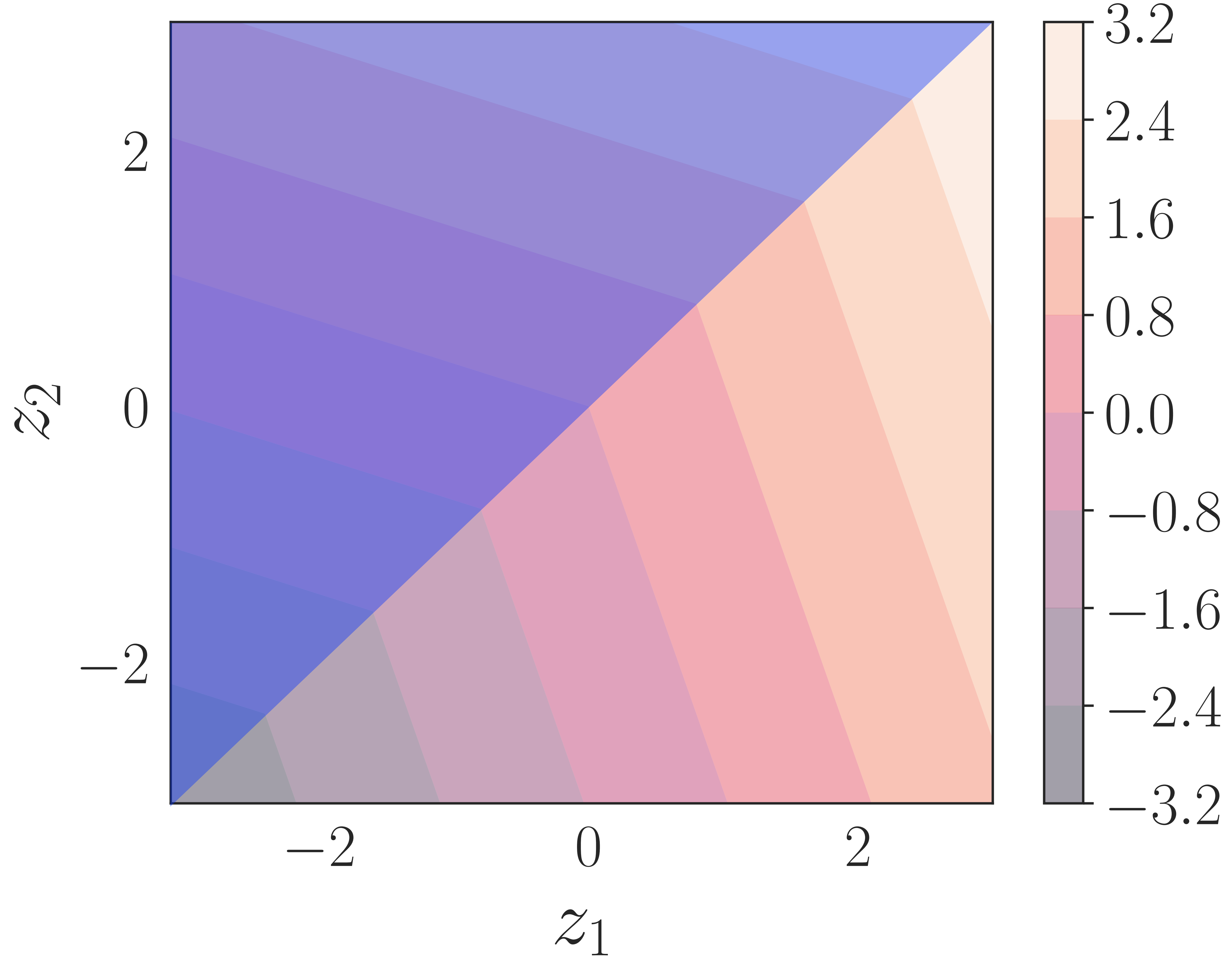
Objective is Piecewise Linear

$$f(z_1, z_2) = 0.3z_{(1)} + 0.7z_{(2)}$$



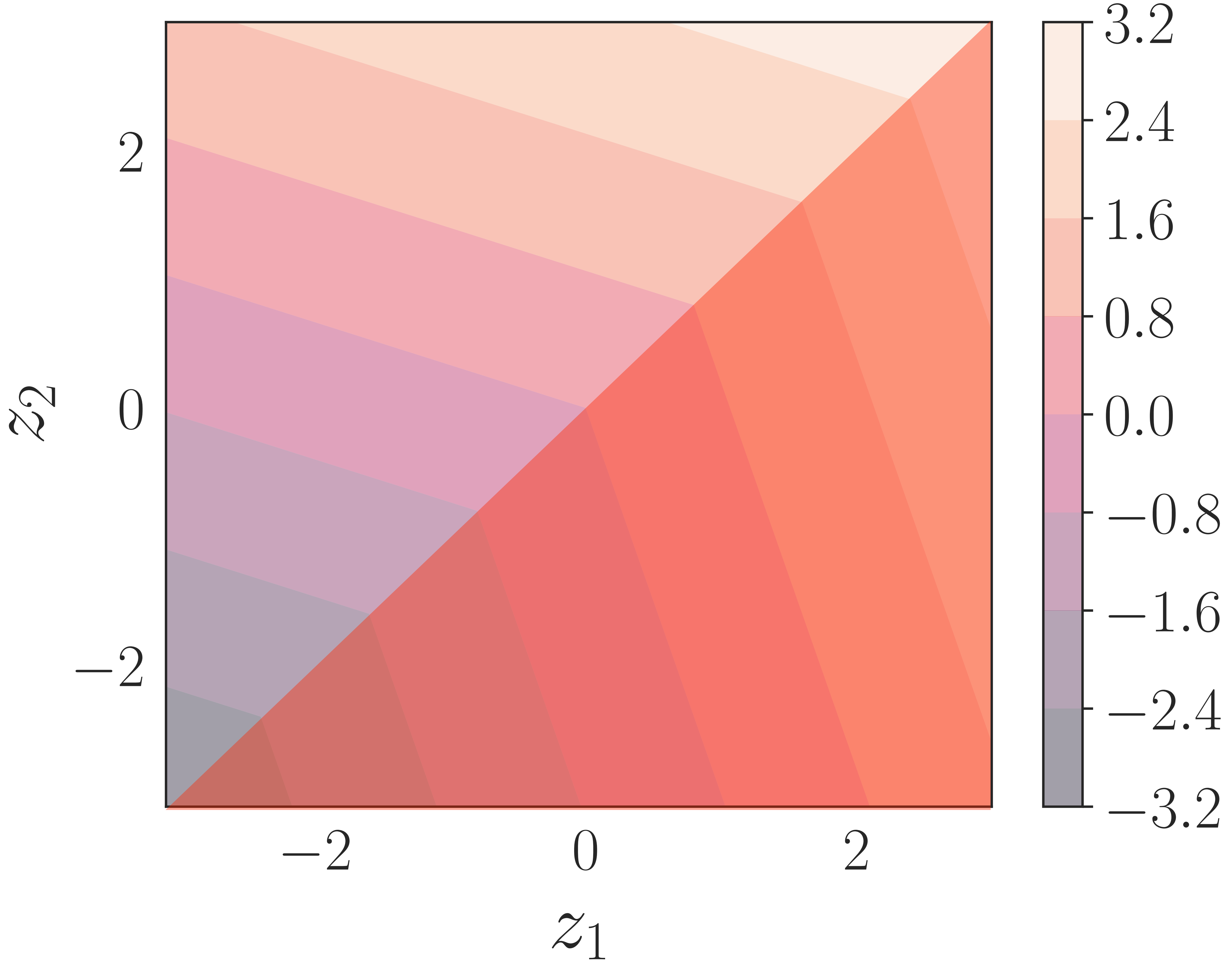
Objective is Piecewise Linear

$$f(z_1, z_2) = 0.3z_{(1)} + 0.7z_{(2)}$$
$$= 0.3z_1 + 0.7z_2$$



Objective is Piecewise Linear

$$f(z_1, z_2) = 0.3z_{(1)} + 0.7z_{(2)}$$
$$= 0.7z_1 + 0.3z_2$$



Optimization Properties

- In general:

Proposition 2. *If* $\ell_1, \ldots, \ell_n$ *are convex, the function* $\mathcal{R}_\sigma$ *is also convex, with subdifferential*

$$\partial \mathcal{R}_\sigma(w) = \operatorname{conv}\left(\bigcup_{\pi \in \operatorname{argsort}(\ell(w))} \sum_{i=1}^{n} \sigma_i \partial \ell_{\pi(i)}(w) \right),$$

where $\operatorname{argsort}(\ell(w)) = \{\pi : \ell_{\pi(1)}(w) \leq \ldots \leq \ell_{\pi(n)}(w)\}$. *Moreover, if each* ℓ_i *is G-Lipschitz continuous,* $\mathcal{R}_\sigma$ *is also G-Lipschitz continuous.*

Optimization Properties

- In general:

Proposition 2. *If* $\ell_1, \ldots, \ell_n$ *are convex, the function* $\mathcal{R}_\sigma$ *is also convex, with subdifferential*

$$\partial \mathcal{R}_\sigma(w) = \text{conv}\left(\bigcup_{\pi \in \text{argsort}(\ell(w))} \sum_{i=1}^{n} \sigma_i \partial \ell_{\pi(i)}(w) \right),$$

where $\text{argsort}\,(\ell(w)) = \left\{\pi : \ell_{\pi(1)}(w) \leq \ldots \leq \ell_{\pi(n)}(w)\right\}.$ *Moreover, if each* ℓ_i *is G-Lipschitz continuous,* $\mathcal{R}_\sigma$ *is also G-Lipschitz continuous.*

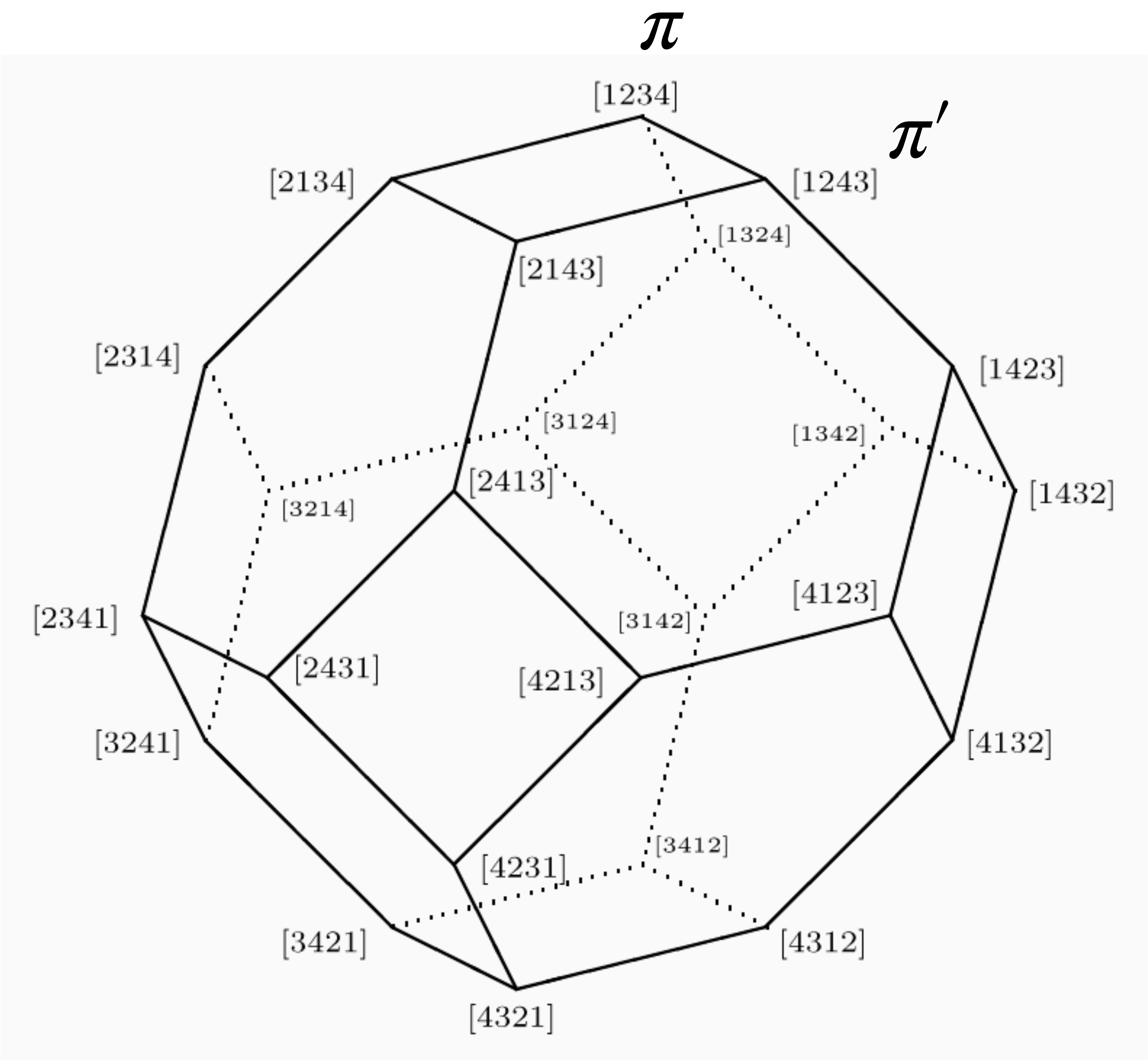
- If the losses are differentiable and $\ell_{(1)}(w) < \ldots < \ell_{(n)}(w)$, then:

$$\nabla \mathcal{R}_\sigma(w) = \sum_{i=1}^{n} \sigma_i \nabla \ell_{(i)}(w)$$

$$\ell_1(w) < \ell_2(w) < \ell_3(w) = \ell_4(w)$$

$$\downarrow$$

$$\partial R_\sigma(w) = \text{conv}\left\{ \sum_{i=1}^{4} \sigma_i \nabla \ell_{\pi(i)}(w), \sum_{i=1}^{4} \sigma_i \nabla \ell_{\pi'(i)}(w) \right\}$$



Computing Subgradients

```
l = compute_losses(w)
l_ord = torch.sort(l)[0]
risk = torch.dot(sigmas, l_ord)
g = torch.autograd.grad(risk, w)[0]
```

- Easy to compute subgradients via automatic differentiation.
- The dependence of the sorting permutation on the input is not recorded on the computation graph.

Outline

- Statistical properties of L-risks.
- Optimization properties of the L-risks.
- **Stochastic optimization algorithms.**
- Experimental evaluations.

Regularized Objective

$$\mathcal{R}_{\sigma,\mu}(w) = \mathcal{R}_{\sigma}(w) + \frac{\mu}{2}\|w\|_2^2 = \sum_{i=1}^{n} \sigma_i \ell_{(i)}(w) + \frac{\mu}{2}\|w\|_2^2$$

Algorithm 1: Minibatch SGD

- Compute a coarser discretization $\hat{\sigma}_1 \leq \ldots \leq \hat{\sigma}_m$ for $m < n$.
- At each iterate w_t:
  - Sample minibatch $\{i_1, \ldots, i_m\} \subseteq [n]$.
  - Sort the losses $\ell_{i_{(1)}}(w_t) \leq \ldots \leq \ell_{i_{(m)}}(w_t)$.
  - Update $w_{t+1} \leftarrow w_t - \eta_t \sum_{j=1}^{m} \hat{\sigma}_j \nabla \ell_{i_{(j)}}(w_t)$.

Algorithm 1 Stochastic Subgradient Method (SGD)

Require: Number of iterates T, minibatch size m, learning rate sequence $(\eta^{(t)})_{t=1}^{T}$, spectrum s, oracles $(\ell_i)_{i=1}^{n}$ and $(\nabla \ell_i)_{i=1}^{n}$, regularization $\mu > 0$.

1: Initialize $w^{(0)} = 0 \in \mathbb{R}^d$.
2: Compute $\hat{\sigma}_1, ..., \hat{\sigma}_m$, where $\hat{\sigma}_j := \int_{(j-1)/m}^{j/m} s(t)\, \mathrm{d}t$.
3: **for** $t = 0, ..., T-1$ **do**
4: Sample without replacement $(i_1, ..., i_m) \subseteq [n]$.
5: Select $\pi \in \operatorname{argsort}\left(\ell_{i_1}(w^{(t)}), ..., \ell_{i_m}(w^{(t)})\right)$.
6: Set $v_m^{(t)} = \sum_{j=1}^{m} \hat{\sigma}_j \nabla \ell_{i_{\pi(j)}}\left(w^{(t)}\right)$.
7: Set $w^{(t+1)} = (1 - \eta^{(t)}\mu) w^{(t)} - \eta^{(t)} v_m^{(t)}$.
8: **return** $\bar{w}^{(T)} = \frac{1}{T} \sum_{t=0}^{T-1} w^{(t)}$.

SGD Analysis

Proposition 2. *If the losses $\ell_1, \ldots, \ell_n$ are G-Lipschitz continuous and convex, the output w_T of Alg. 1 satisfies*

$$\mathbb{E}\left[\mathcal{R}_{\sigma,\mu}\left(w_T\right)\right] - \mathcal{R}_{\sigma,\mu}(w^*) \lesssim \underbrace{\|s-u\|_\infty B_\mu \sqrt{\frac{n-m}{mn}}}_{\text{bias term}} + \underbrace{\log T/T}_{\text{optimization term}} .$$

for $B_\mu = \sup_{w:\|w\|_2 \leq G/\mu} \max_{i=1,\ldots,n} |\ell_i(w)| < \infty.$

Algorithm 2: LSVRG

- At each epoch:
  - Store a "checkpoint" $\bar{w}$ and compute $\bar{g} = \sum_{i=1}^{n} \sigma_i \nabla \ell_{\bar{\pi}(i)}(\bar{w})$.
  - At each iterate t:
    - Uniformly randomly sample index $i_t \in [n]$.
    - Compute $v_t = n\sigma_{i_t} \nabla \ell_{\bar{\pi}(i_t)}(w_t) + n\sigma_{i_t} \nabla \ell_{\bar{\pi}(i_t)}(\bar{w}) + \bar{g}$.
    - Update $w_{t+1} \leftarrow w_t - \eta \left(v_t + \mu w_t\right)$.

Algorithm 2: LSVRG

- At each epoch:
  - Store a "checkpoint" $\bar{w}$ and compute $\bar{g} = \sum_{i=1}^{n} \sigma_i \nabla \ell_{\bar{\pi}(i)}(\bar{w})$.
  - At each iterate t:
    - Uniformly randomly sample index $i_t \in [n]$.
    - Compute $v_t = n\sigma_{i_t} \nabla \ell_{\bar{\pi}(i_t)}(w_t) + n\sigma_{i_t} \nabla \ell_{\bar{\pi}(i_t)}(\bar{w}) + \bar{g}$. mean zero w.r.t i_t
    - Update $w_{t+1} \leftarrow w_t - \eta \left(v_t + \mu w_t\right)$.

Algorithm 2: LSVRG

- At each epoch:
  - Store a "checkpoint" $\bar{w}$ and compute $\bar{g} = \sum_{i=1}^{n} \sigma_i \nabla \ell_{\bar{\pi}(i)}(\bar{w})$.
- At each iterate t:
  - Uniformly randomly sample index $i_t \in [n]$.
  - Compute $v_t = n\sigma_{i_t} \nabla \ell_{\bar{\pi}(i_t)}(w_t) + n\sigma_{i_t} \nabla \ell_{\bar{\pi}(i_t)}(\bar{w}) + \bar{g}$.
  - Update $w_{t+1} \leftarrow w_t - \eta \left(v_t + \mu w_t\right)$.

to be unbiased, we need π such that $\ell_{\pi(1)}(w_t) \leq \ldots \ell_{\pi(n)}(w_t)$

Algorithm 2 LSVRG

Require: Number of iterations T, loss functions $(\ell_i)_{i=1}^n$ and their gradient oracles, initial point $w^{(0)}$, learning rate η, sorting update frequency N, spectrum $(\sigma_i)_{i=1}^n$, regularization μ.

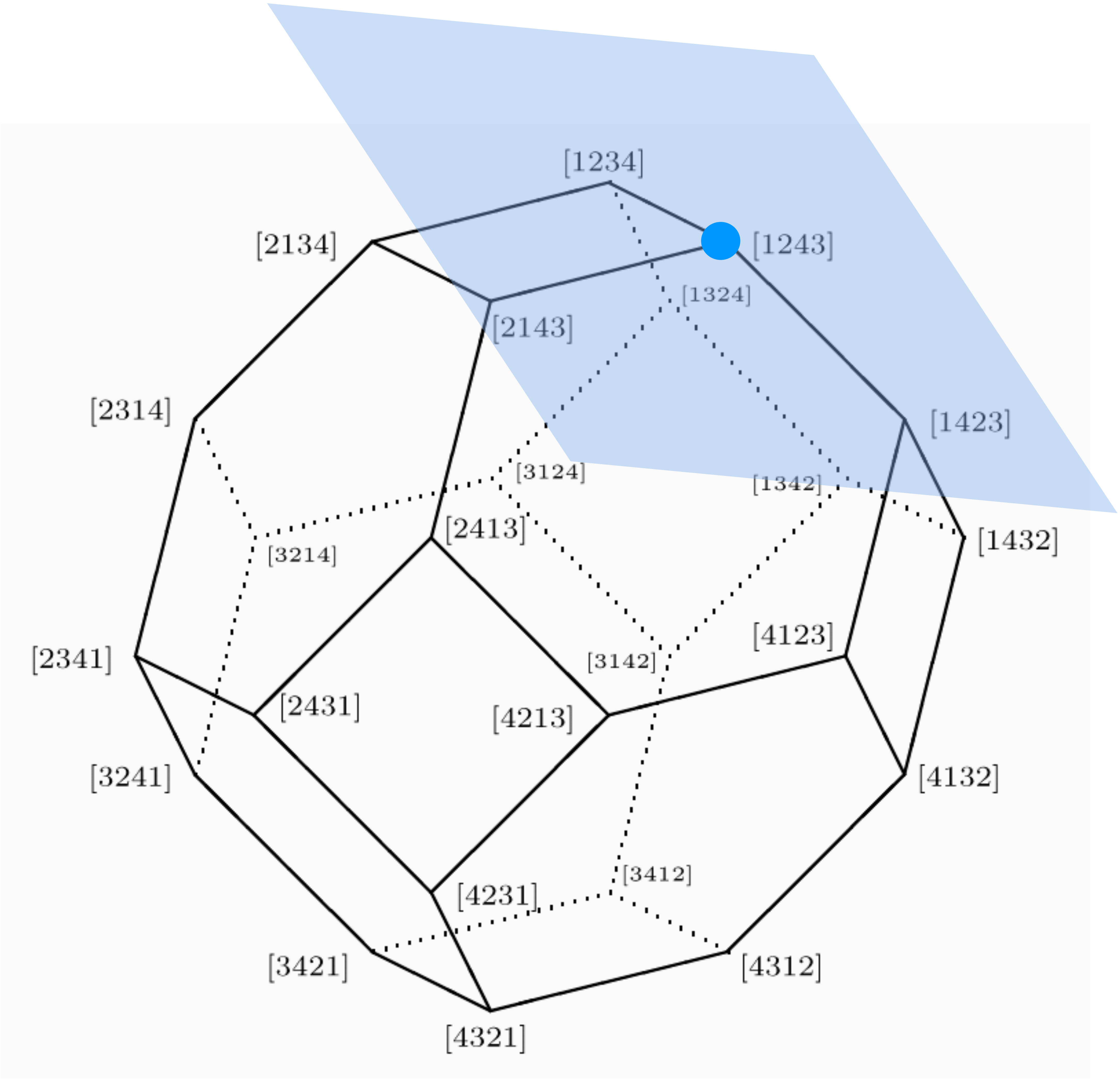
1: **for** $t = 0, ..., T-1$ **do**
2:     **if** $t \mod N = 0$ **then**
3:         Set $\bar{w} = w^{(t)}$.
4:         Select $\bar{\pi} \in \text{argsort}\,(\ell_1(\bar{w}), \ldots, \ell_n(\bar{w}))$.
5:         $\bar{g} = \sum_{i=1}^n \sigma_i \nabla \ell_{\bar{\pi}(i)}(\bar{w})$.
6:     Sample $i_t \sim p_\sigma$, where $p_\sigma(i) = \sigma_i$.
7:     $v^{(t)} = \nabla \ell_{\bar{\pi}(i_t)}(w^{(t)}) - \nabla \ell_{\bar{\pi}(i_t)}(\bar{w}) + \bar{g}$.
8:     $w^{(t+1)} = (1 - \eta\mu) w^{(t)} - \eta v^{(t)}$.
9: **return** $w^{(T)}$.

Quick Detour: Smooth Approximation

- Typical analyses of algorithms require smoothness (gradient function is Lipschitz continuous). L-Risk are not even differentiable.
- The upcoming algorithm will approximate the objective with a smoothed version.
- Notice that for $l \in \mathbb{R}^n$,

$$\sum_{i=1}^{n} \sigma_i l_{(i)} = \max_{\lambda \in \mathcal{P}(\sigma)} \sum_{i=1}^{n} \lambda_i l_i \quad (\mathcal{P}(\sigma) = \operatorname{conv}\{\text{permutations of } \sigma\}) .$$

$$\sum_{i=1}^{n} \sigma_i l_{(i)} = \max_{\lambda \in \mathcal{P}(\sigma)} \sum_{i=1}^{n} \lambda_i l_i \quad (\mathcal{P}(\sigma) = \text{conv}\,\{\text{permutations of } \sigma\})\,.$$



Quick Detour: Smooth Approximation

- Typical analyses of algorithms require smoothness (gradient function is Lipschitz continuous). L-Risk are not even differentiable.
- The upcoming algorithm will approximate the objective with a smoothed version.
- Notice that for $l \in \mathbb{R}^n$,

$$\sum_{i=1}^{n} \sigma_i l_{(i)} = \max_{\lambda \in \mathcal{P}(\sigma)} \sum_{i=1}^{n} \lambda_i l_i \quad (\mathcal{P}(\sigma) = \text{conv}\,\{\textbf{permutations of } \sigma\})\,.$$

- Consider for $\nu > 0$ the approximation:

$$h_\nu(l) = \max_{\lambda \in \mathcal{P}(\sigma)} \left\{ \sum_{i=1}^{n} \lambda_i l_i - \frac{\nu}{2} \|\lambda\|_2^2 \right\}$$

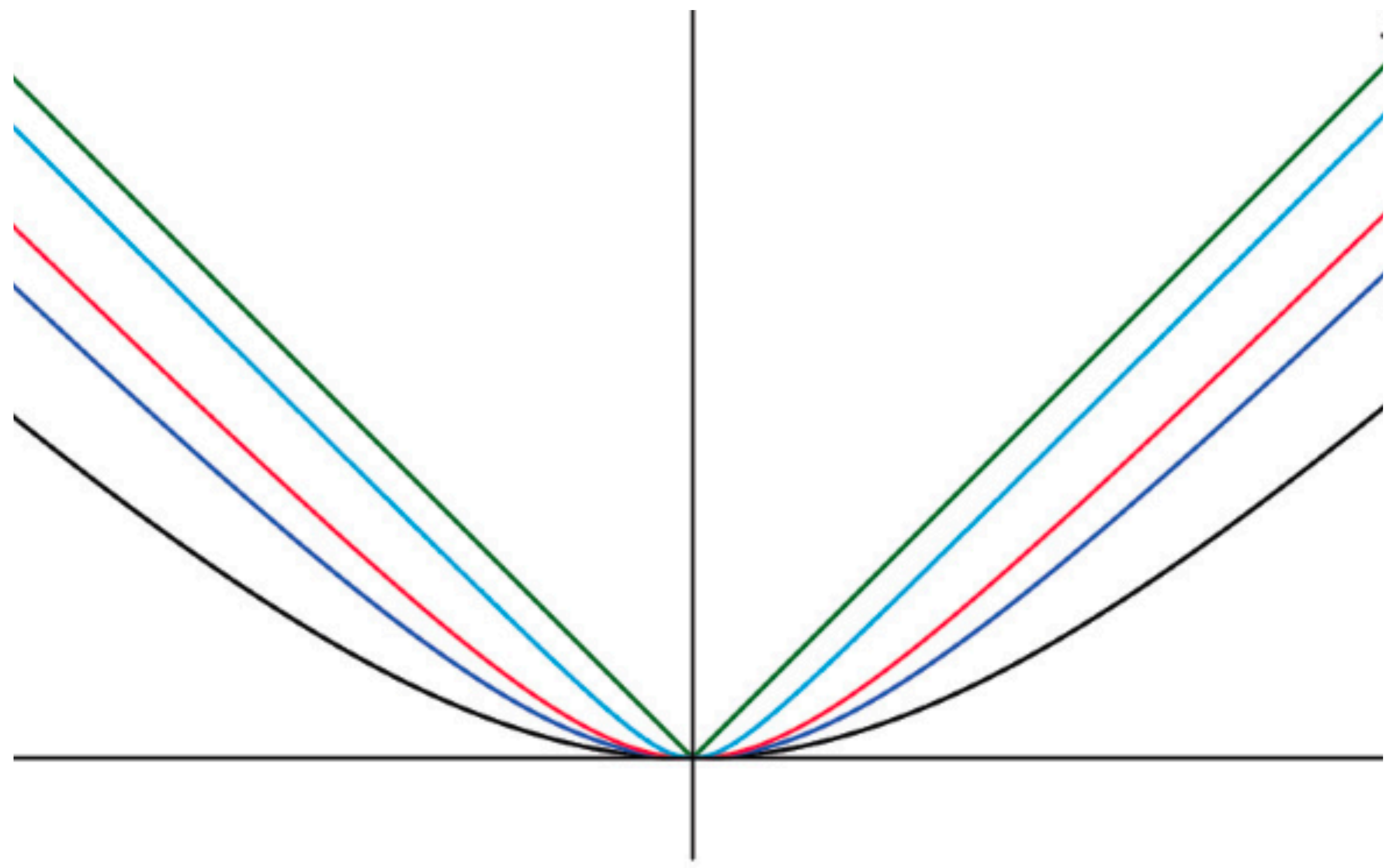
Smoothed Surrogate Objective

- Original regularized objective:

$$\mathcal{R}_{\sigma,\mu}(w) = \sum_{i=1}^{n} \sigma_i \ell_{(i)}(w) + \frac{\mu}{2} \|w\|_2^2 = \max_{\lambda \in \mathcal{P}(\sigma)} \left\{ \sum_{i=1}^{n} \lambda_i \ell_i(w) \right\} + \frac{\mu}{2} \|w\|_2^2$$

- Smoothed regularized objective:

$$\mathcal{R}_{\sigma,\mu,\nu}(w) = h_\nu(\ell(w)) + \frac{\mu}{2} \|w\|_2^2 = \max_{\lambda \in \mathcal{P}(\sigma)} \left\{ \sum_{i=1}^{n} \lambda_i \ell_i(w) - \frac{\nu}{2} \|\lambda\|_2^2 \right\} + \frac{\mu}{2} \|w\|_2^2$$



LSVRG Analysis

Theorem 3. *If ℓ_i is convex, G-Lipschitz continuous and L-smooth, for appropriately chosen epoch length N and stepsize η, we have that*

$$\mathbb{E}\|w^{(kN)} - w^*\| \leq (1/2)^k \|w^{(0)} - w^*\|$$

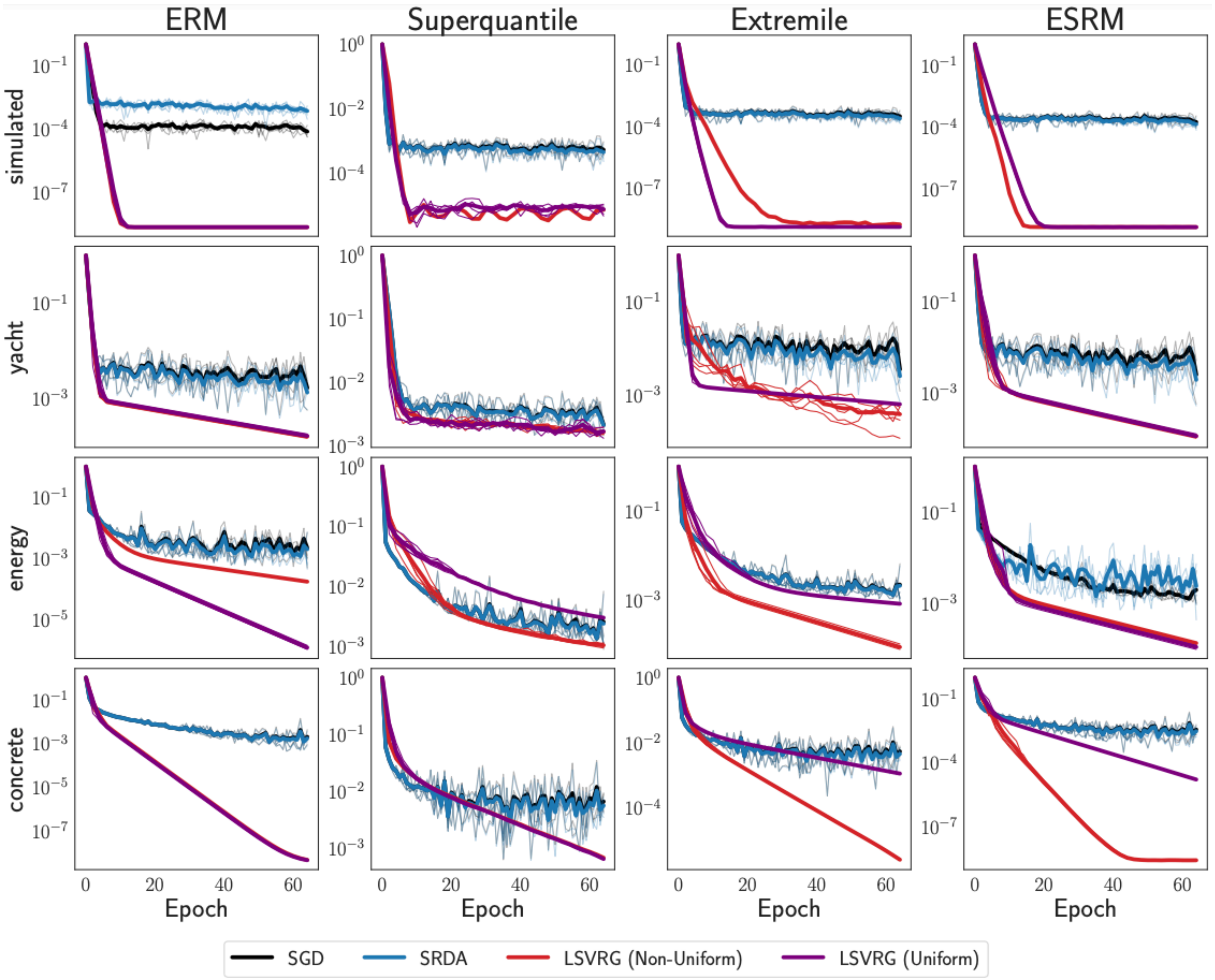
for $k \in \mathbb{N}$ and $w^* = \arg\min_{w \in \mathbb{R}^d} \mathcal{R}_{\sigma,\mu,\nu}(w)$.

Outline

- Statistical properties of L-risks.
- Optimization properties of the L-risks.
- Stochastic optimization algorithms.
- **Experimental evaluations.**

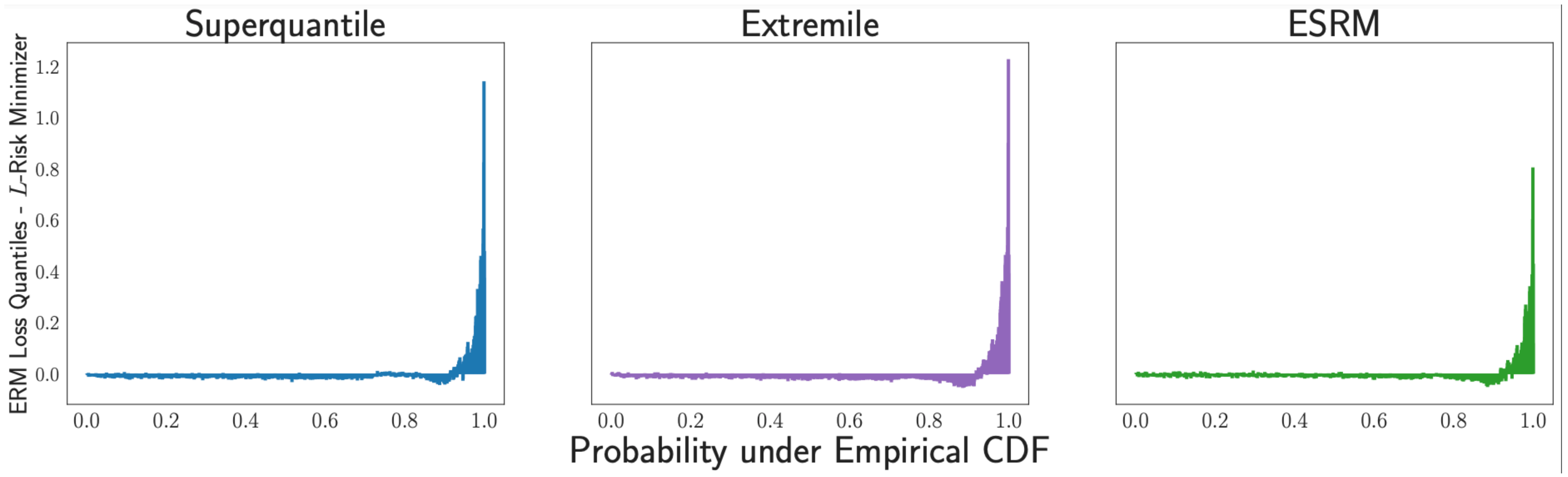
Regression

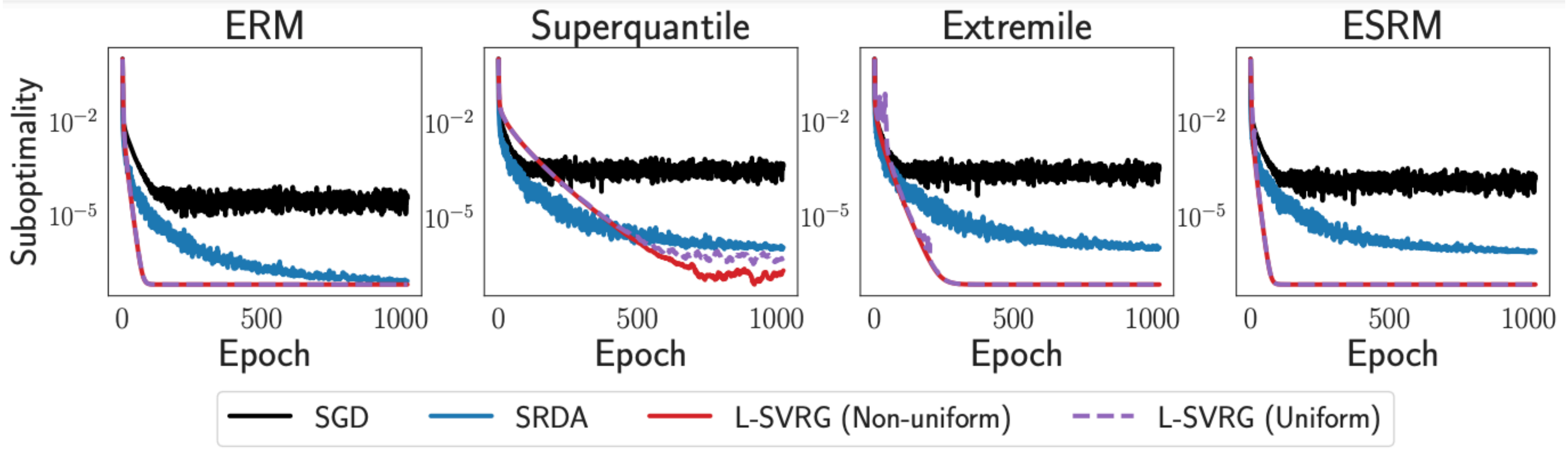
- **Setting:** Linear model and squared error loss on four UCI datasets.
- **Baselines:** Stochastic subgradient method (SGD) and stochastic regularized dual averaging (SRDA).
- **Takeaways:** Baselines do not converge due to bias and variance. Superquantile is the most difficult to optimize.



Classification

- **Setting:** Dataset of 16,000 sentences, each with one of six emotion label. Linear model applied to neural embeddings with cross entropy loss.
- **Baselines:** Stochastic subgradient method (SGD) and stochastic regularized dual averaging (SRDA).
- **Takeaways:** L-Risk minimizers control tail losses.





Summary

We present a stochastic algorithm to optimize non-smooth L-statistics of the empirical loss distribution, that

- finds an exact minimizer (is asymptotically unbiased),
- makes $O(1)$ gradient calls per update, and
- dominates out-of-the-box convex optimizers on synthetic and real data.

Future Work:

- Non-convex setting.
- Statistical properties of learned minimizers (robustness to distribution shift, etc).

Thank you!