Distributionally Robust Optimization with Bias and Variance Reduction

Ronak Mehta

October 14, 2023

Team

Ronak Mehta
University of Washington

Vincent Roulet
Google DeepMind

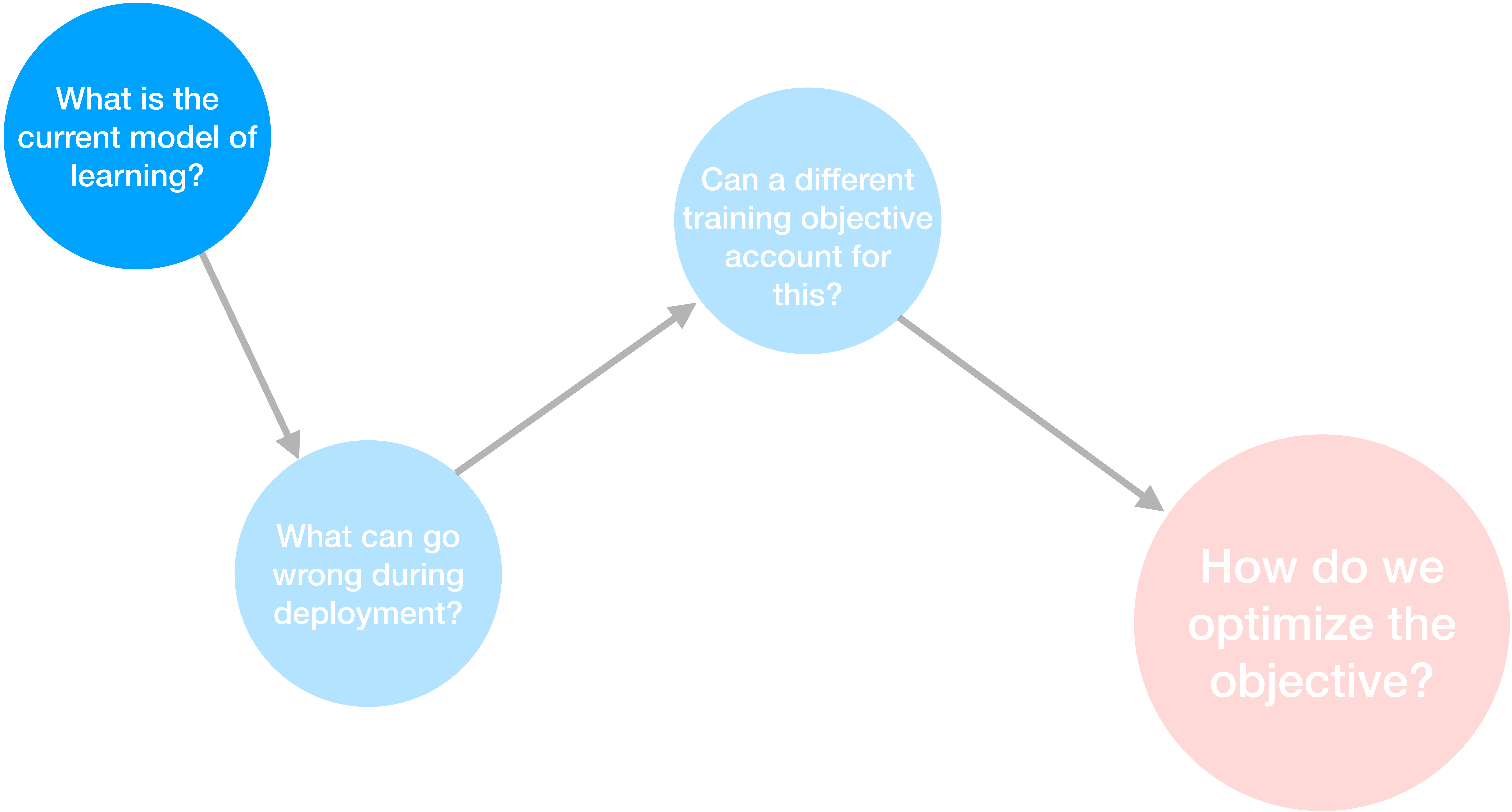
Krishna Pillutla
Google Research

Zaid Harchaoui
University of Washington

IFDS
Institute for Foundations of Data Science

NSF

W
UNIVERSITY of WASHINGTON


What is the current model of learning?
What can go wrong during deployment?
Can a different training objective account for this?
How do we optimize the objective?

Stochastic Programming is the prevailing model for machine learning.

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{Z \sim P}[\ell(w, Z)]$$

Stochastic Programming is the prevailing model for machine learning.

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{Z \sim P}[\ell(w, Z)]$$

model parameters

Stochastic Programming is the prevailing model for machine learning.

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{Z \sim P}[\ell(w, Z)]$$

loss function

data instance

Stochastic Programming is the prevailing model for machine learning.

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{Z \sim P}[\ell(w, Z)]$$

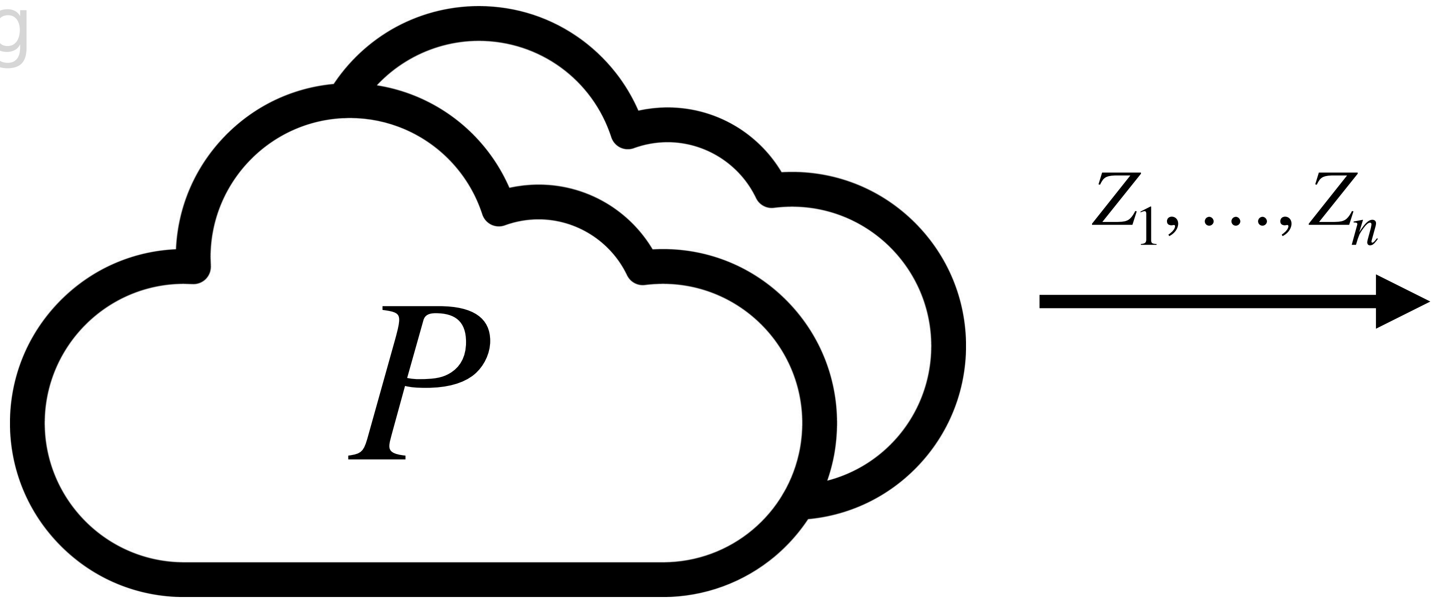
data generating distribution

Stochastic Programming is the prevailing model for machine learning.

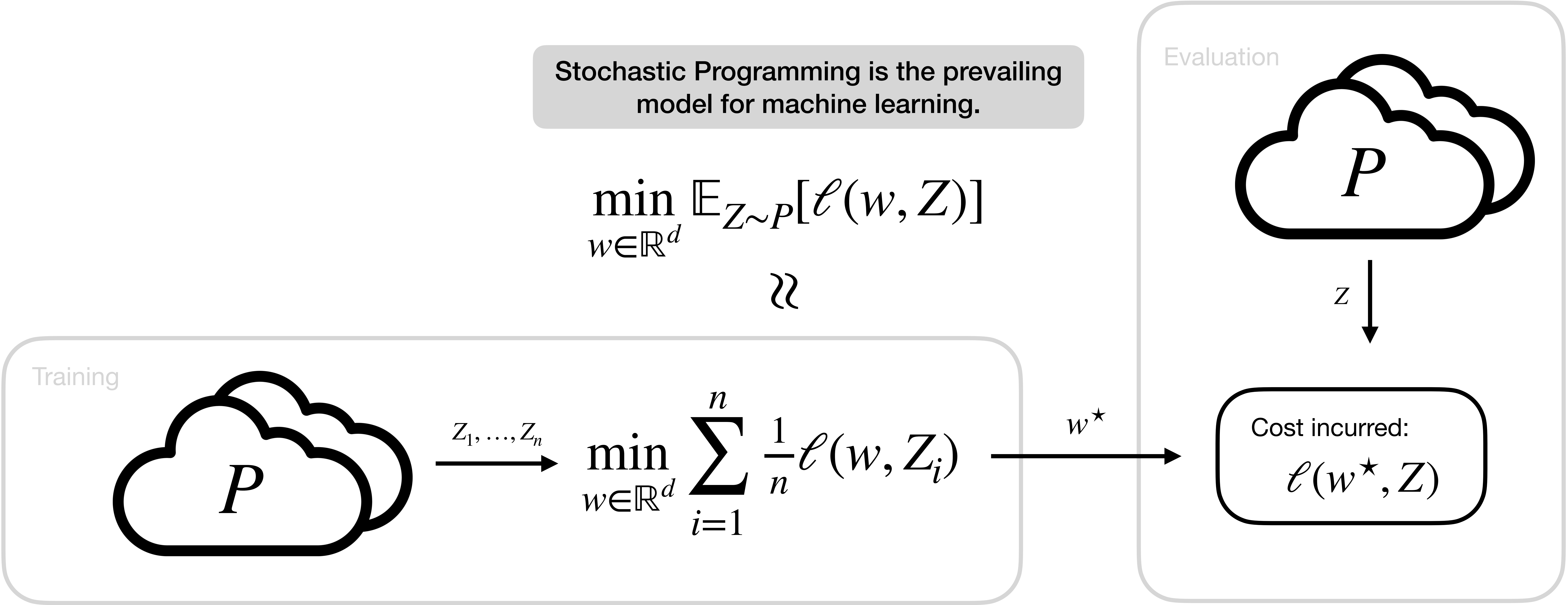
$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{Z \sim P}[\ell(w, Z)]$$

$$\approx$$

Training



$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \frac{1}{n} \ell(w, Z_i)$$



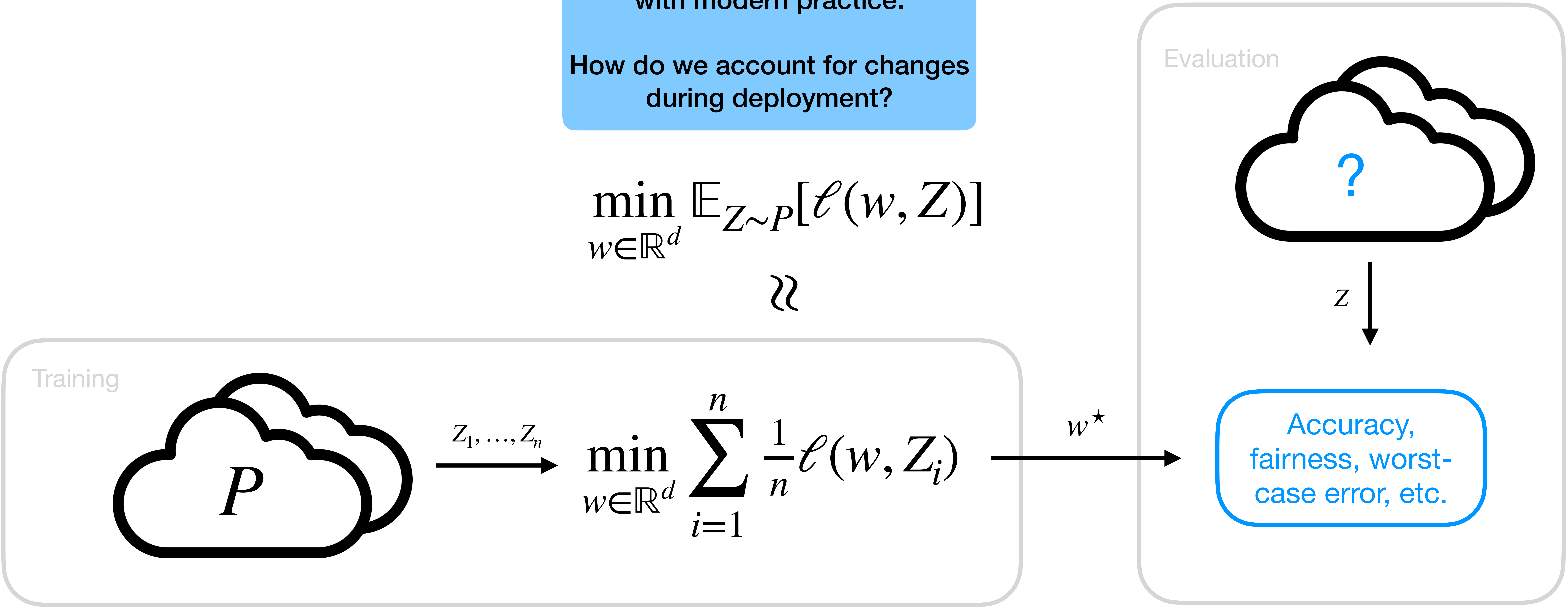
Stochastic Programming is the prevailing model for machine learning.
$\min_{w \in \mathbb{R}^d} \mathbb{E}_{Z \sim P}[\ell(w, Z)]$
$\approx$
Training
P
$Z_1, \ldots, Z_n$
$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \frac{1}{n} \ell(w, Z_i)$
$w^\star$
Evaluation
P
Z
Cost incurred:
$\ell(w^\star, Z)$

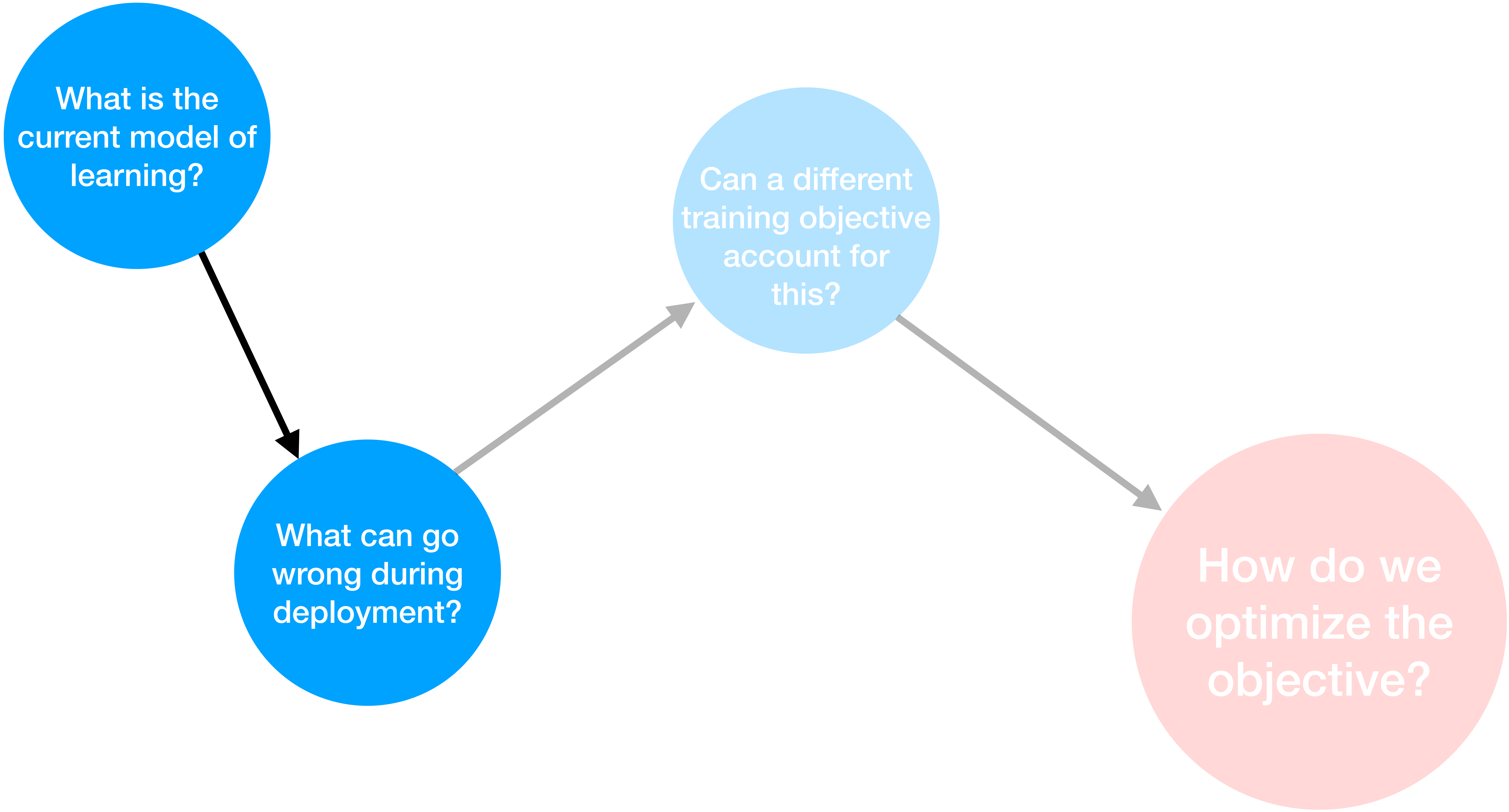
This formulation may not agree with modern practice.

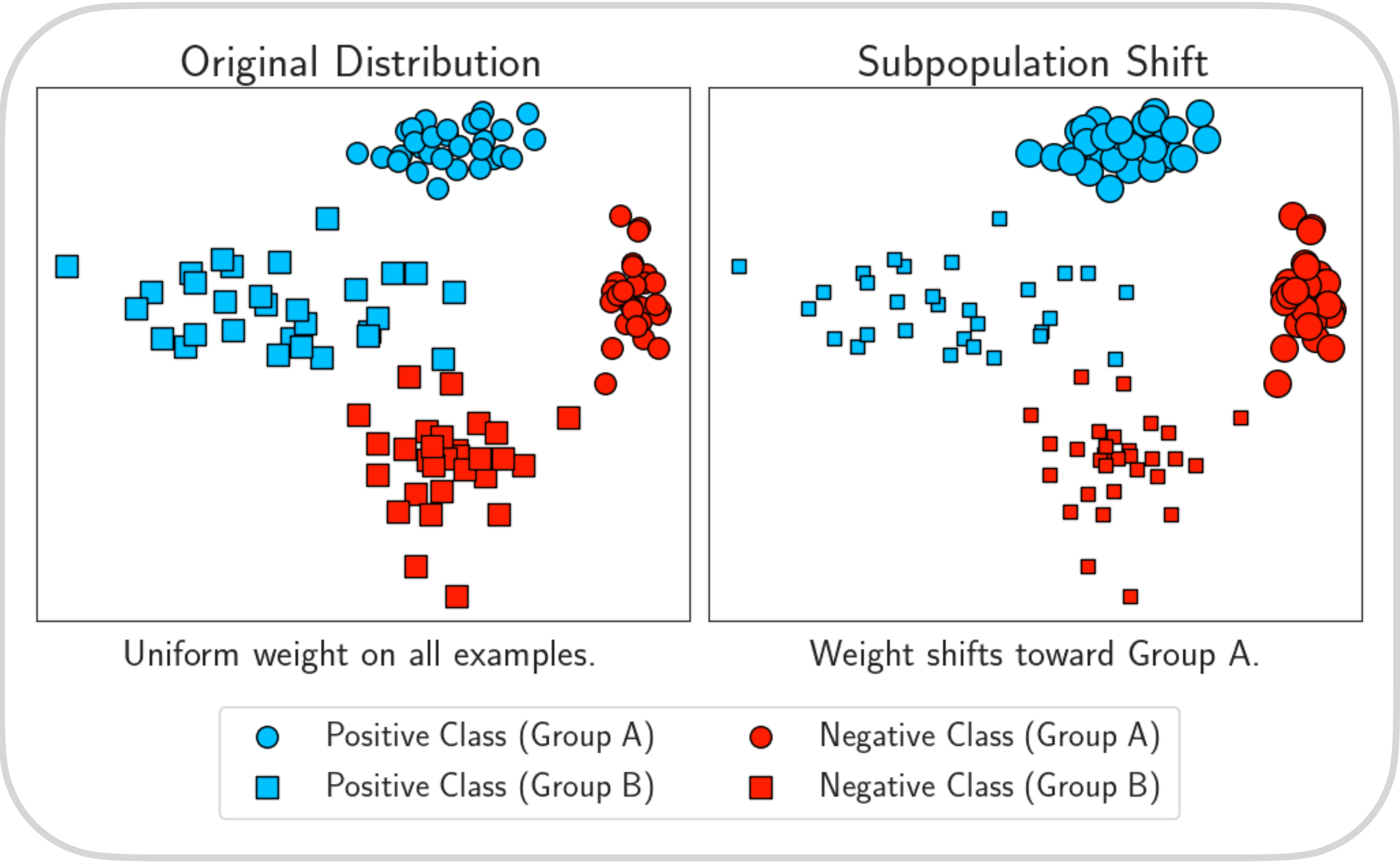
How do we account for changes during deployment?

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{Z \sim P}[\ell(w, Z)]$$

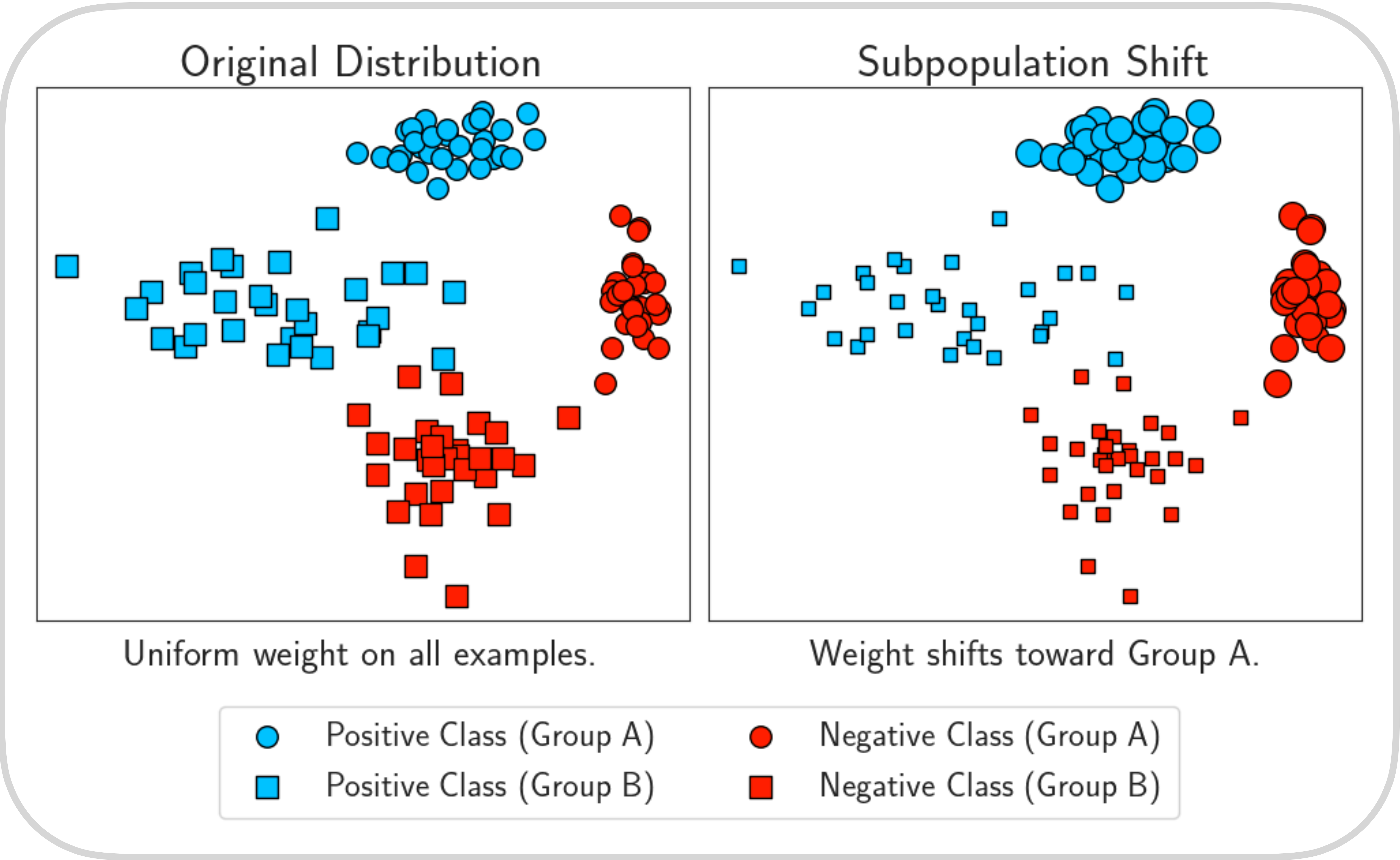
$$\approx$$



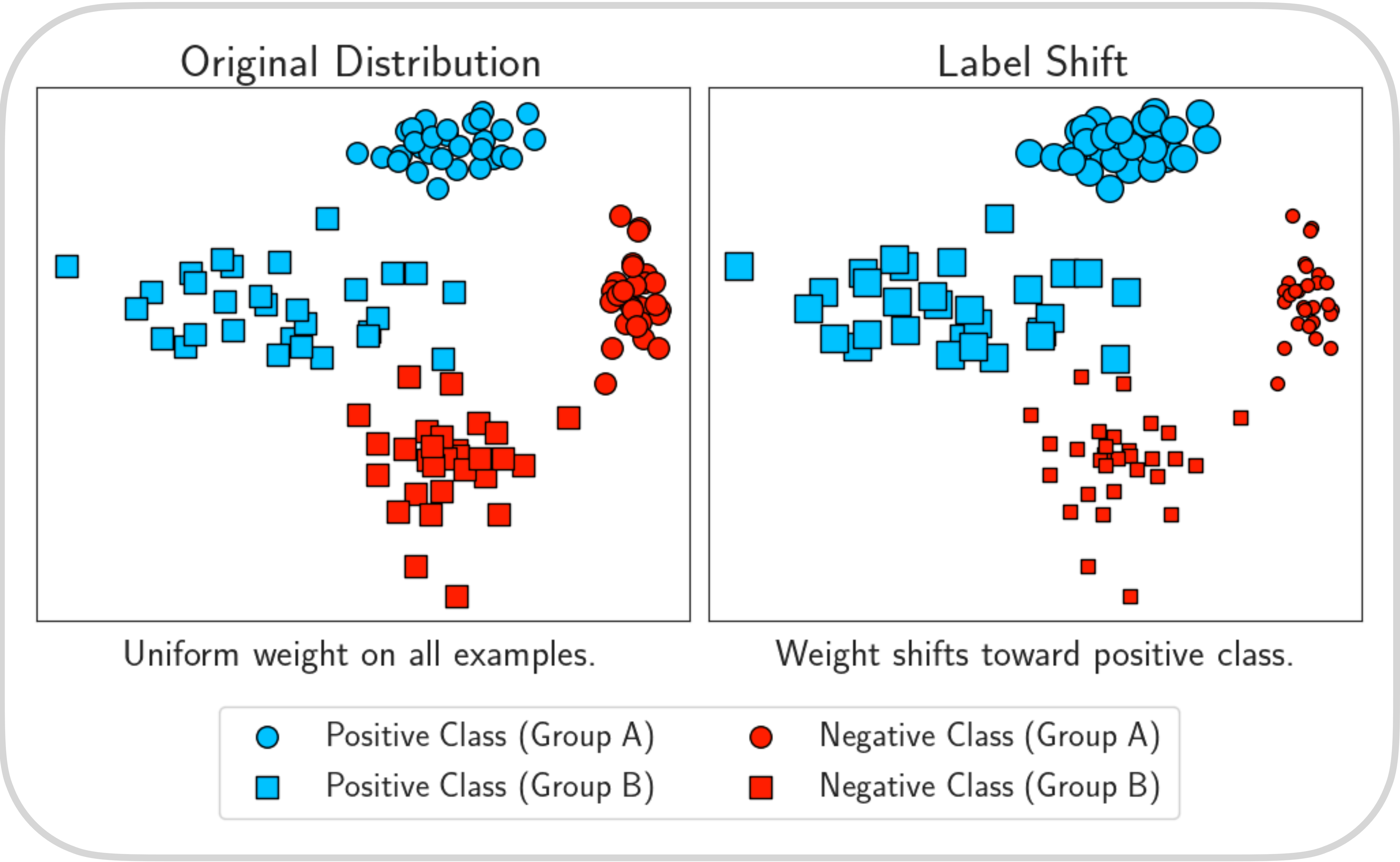

What is the current model of learning?
What can go wrong during deployment?
Can a different training objective account for this?
How do we optimize the objective?



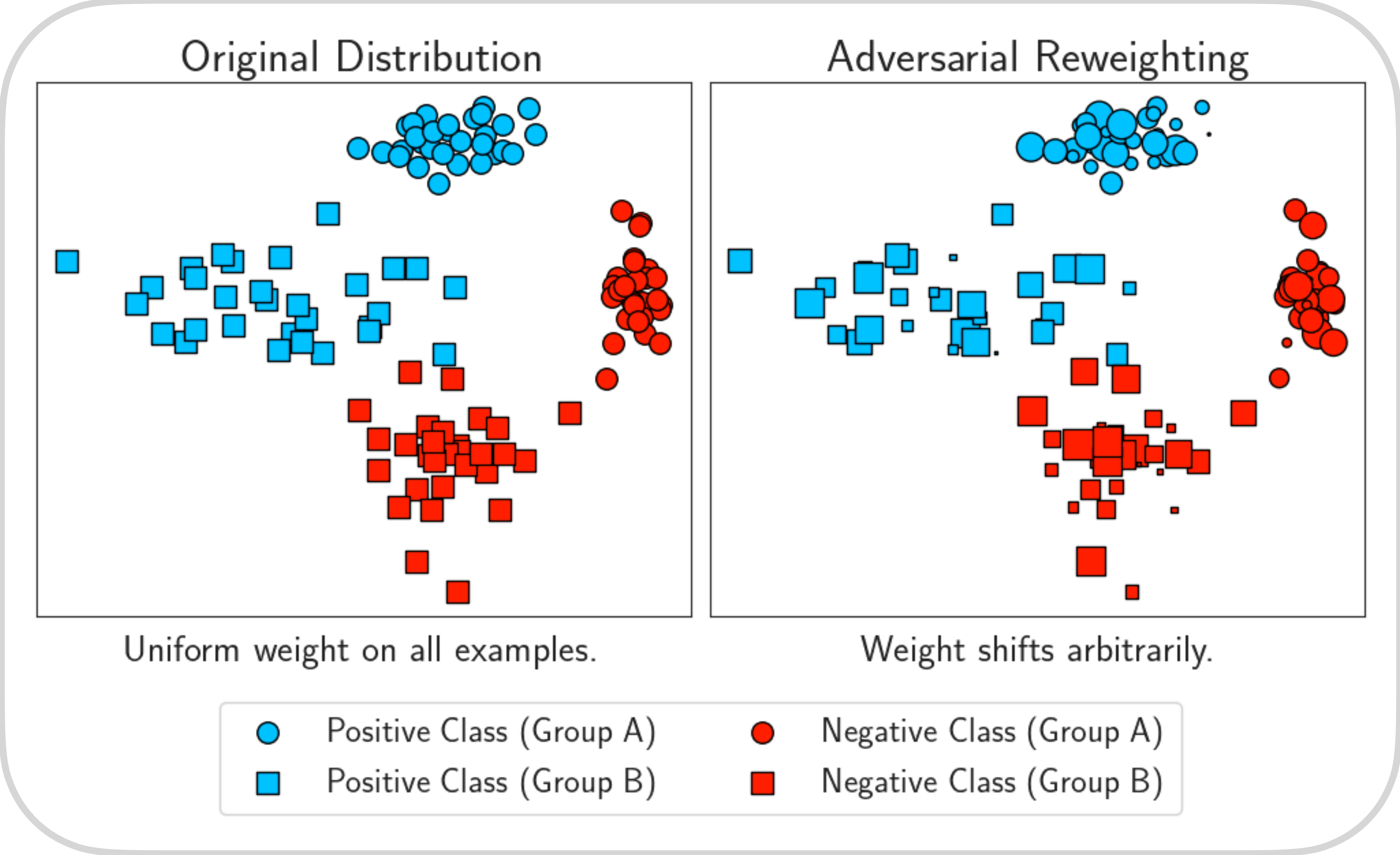
Original Distribution
Subpopulation Shift
Uniform weight on all examples.
Weight shifts toward Group A.
Positive Class (Group A)
Positive Class (Group B)
Negative Class (Group A)
Negative Class (Group B)



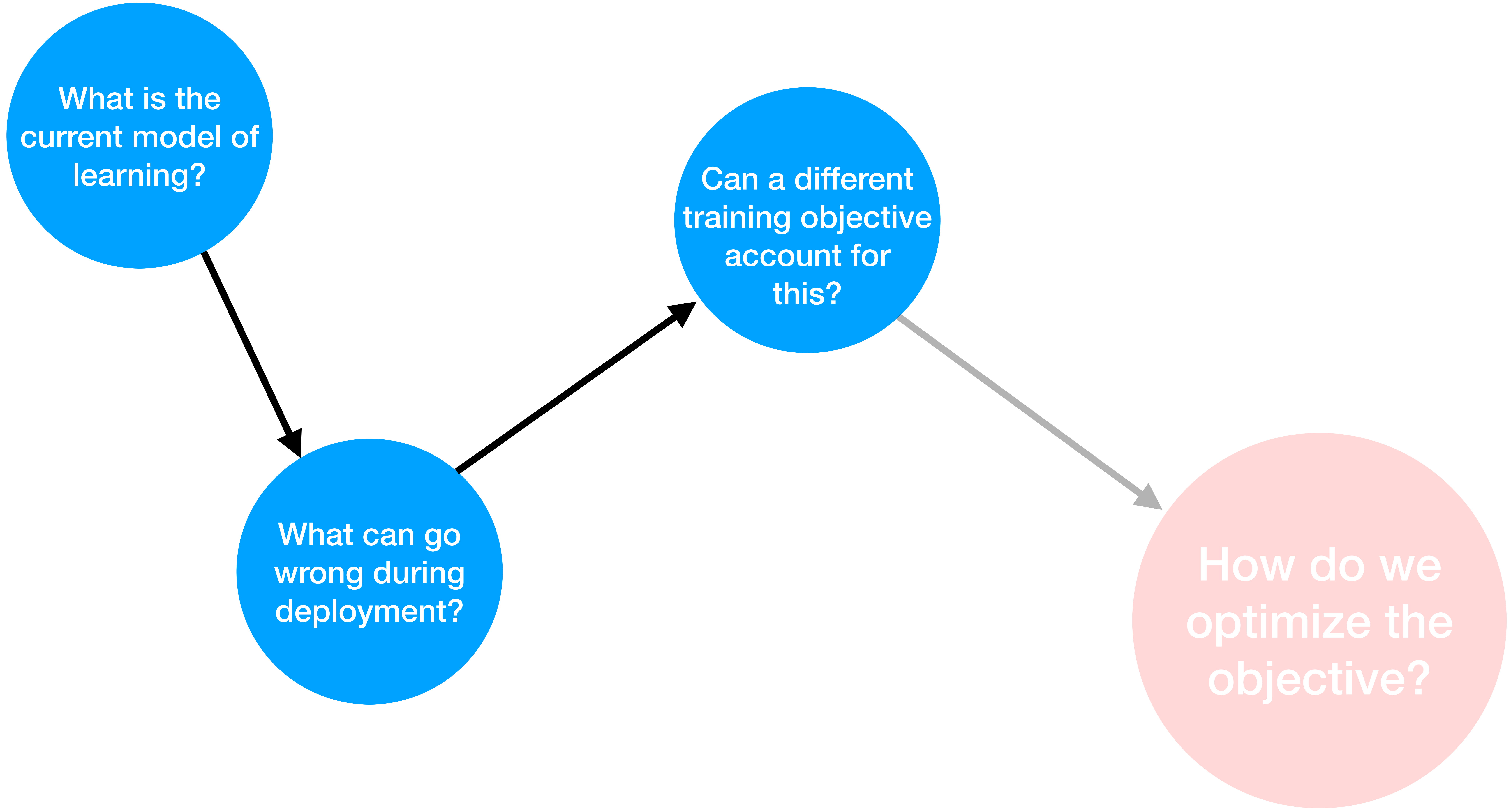
Common notions of **algorithmic fairness** impose that **model performance does not degrade drastically on any one group/ subpopulation**.



In **label shift**, the subpopulations are the labels themselves, which occur with differing frequencies than from training.



In the most general case (ours), **<u>any data point is a subpopulation.</u>**


What is the current model of learning?
What can go wrong during deployment?
Can a different training objective account for this?
How do we optimize the objective?

DR Objectives Model Reweighting Shifts

$$\min_{w \in \mathbb{R}^d} \max_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i \ell(w, Z_i) - \nu D(q \| \mathbf{1}_n / n)$$

DR Objectives Model Reweighting Shifts

$$\min_{w \in \mathbb{R}^d} \max_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i \ell(w, Z_i) - \nu D(q \| \mathbf{1}_n / n)$$

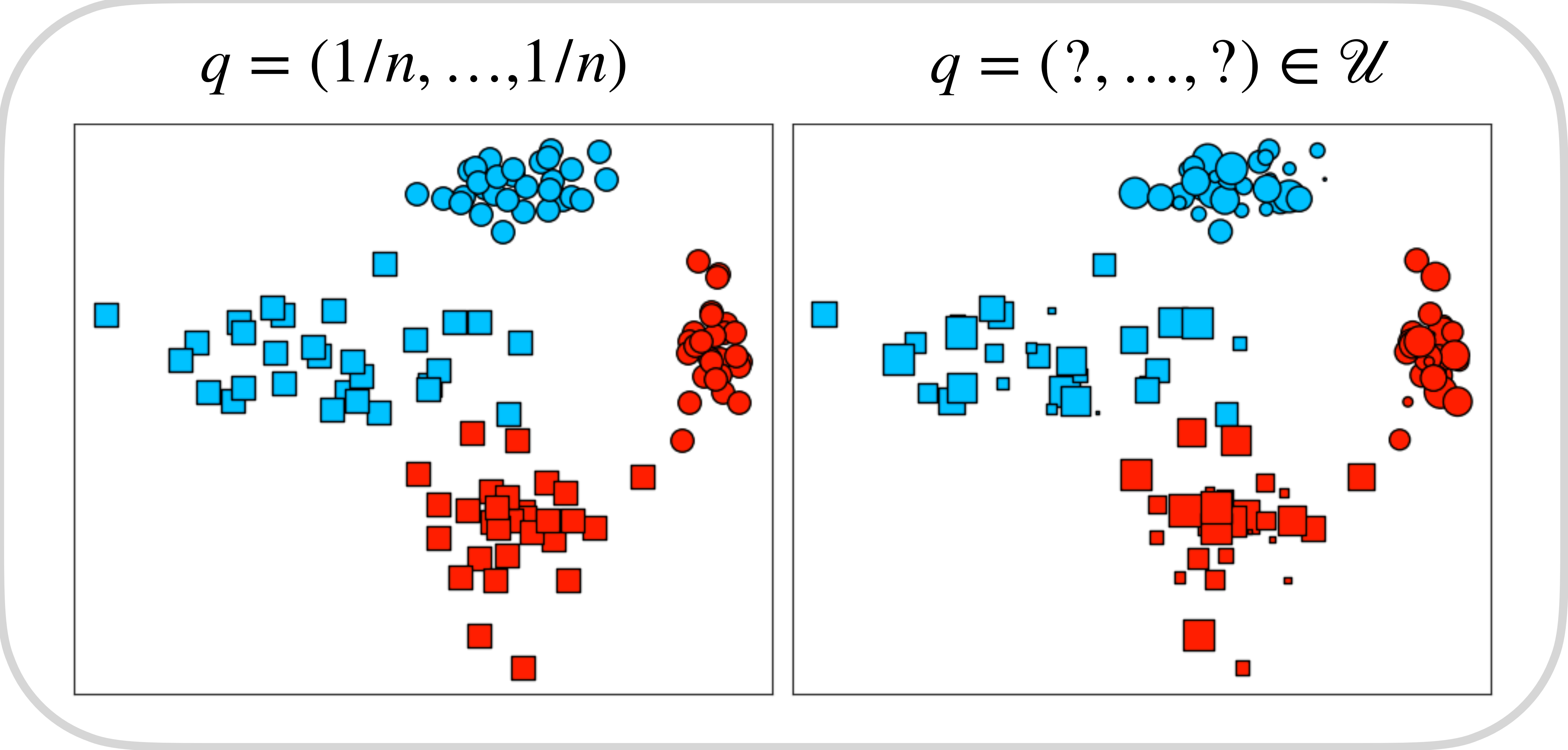
uncertainty set of possible distributions, i.e. each $q_i \geq 0$ and $\sum_{i=1}^{n} q_i = 1$

DR Objectives Model Reweighting Shifts

expected loss under q

$$\min_{w \in \mathbb{R}^d} \max_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i \ell(w, Z_i) - \nu D(q \| \mathbf{1}_n / n)$$

uncertainty set of possible distributions, i.e. each $q_i \geq 0$ and $\sum_{i=1}^{n} q_i = 1$



DR Objectives Model Reweighting Shifts

$$\min_{w \in \mathbb{R}^d} \max_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i \ell(w, Z_i) - \nu D(q \| \mathbf{1}_n / n)$$

shift cost

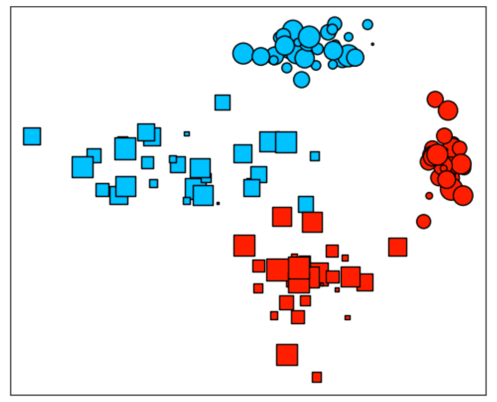
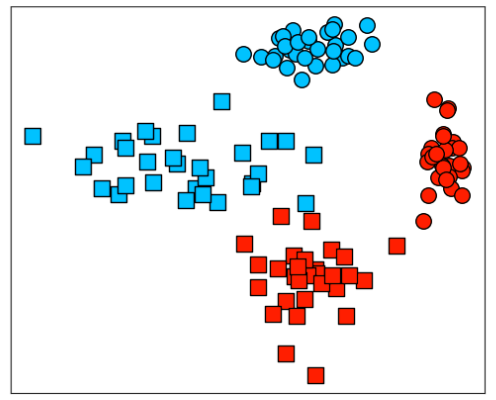
deviation of q from original distribution

DR Objectives Model Reweighting Shifts

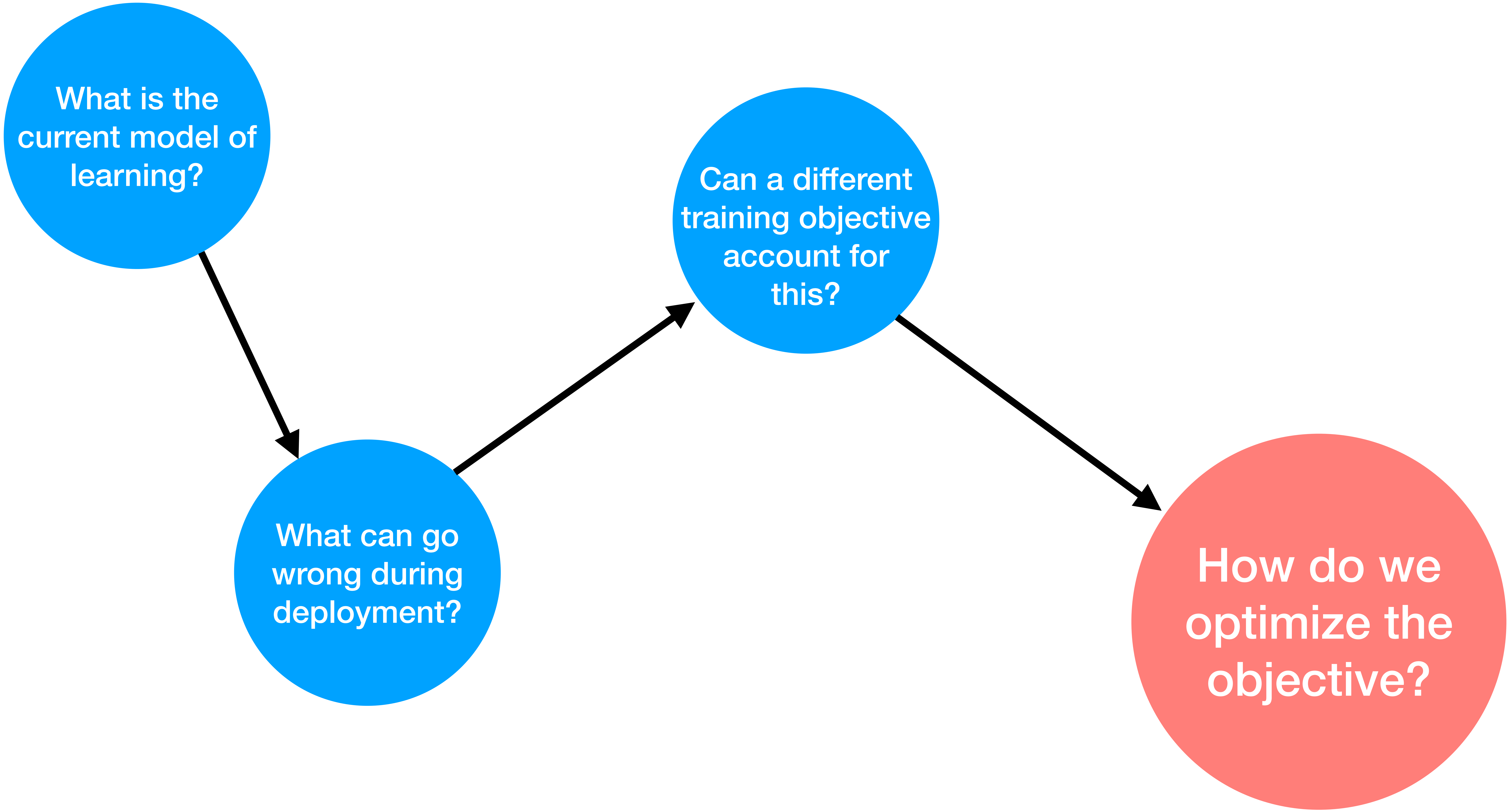
$$\min_{w \in \mathbb{R}^d} \max_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i \ell(w, Z_i) - \nu D(q \| \mathbf{1}_n / n)$$

shift cost

deviation of q from original distribution

$D(q\|\mathbf{1}_n/n) = \mathrm{dist}($  ,  $)$

q $\qquad$ $\mathbf{1}_n/n$


What is the current model of learning?
What can go wrong during deployment?
Can a different training objective account for this?
How do we optimize the objective?

Stochastic optimization is an essential ingredient for ERM, but implementing these algorithms for DRO is a key challenge.

Stochastic optimization is an essential ingredient for ERM, but implementing these algorithms for DRO is a key challenge.

$$w_{t+1} = w_t - \eta_t g_t$$

stochastic gradient estimate that only depends on $O(1)$ calls to oracles $\{\ell(\cdot, Z_i), \nabla \ell(\cdot, Z_i)\}_{i=1}^n$

stepsize sequence

Notation

$R =$ objective function

$P_n =$ sampling distribution used for g_t (e.g. mini-batch sampling)

Stochastic optimization is an essential ingredient for ERM, but implementing these algorithms for DRO is a key challenge.

$$w_{t+1} = w_t - \eta_t g_t$$

Bias

$$\mathbb{E}_{P_n}[g_t] - \nabla R(w_t)$$

Variance

$$\mathbb{E}_{P_n}\|g_t - \mathbb{E}[g_t]\|_2^2$$

Notation

$R =$ objective function

$P_n =$ sampling distribution used for g_t (e.g. mini-batch sampling)

Stochastic optimization is an essential ingredient for ERM, but implementing these algorithms for DRO is a key challenge.

$$w_{t+1} = w_t - \eta_t g_t$$

Bias

$$\mathbb{E}_{P_n}[g_t] - \nabla R(w_t)$$

Variance

$$\mathbb{E}_{P_n}\|g_t - \mathbb{E}[g_t]\|_2^2$$

Problem in ERM as well, usually handled by decreasing learning rate or variance-reduced methods.

Notation

$R =$ objective function

$P_n =$ sampling distribution used for g_t (e.g. mini-batch sampling)

Stochastic optimization is an essential ingredient for ERM, but implementing these algorithms for DRO is a key challenge.

$$w_{t+1} = w_t - \eta_t g_t$$

Unbiased estimates are used in ERM, but this is impossible for DRO, resulting in poor convergence.

Bias

$$\mathbb{E}_{P_n}[g_t] - \nabla R(w_t)$$

Variance

$$\mathbb{E}_{P_n}\|g_t - \mathbb{E}[g_t]\|_2^2$$

Is there an optimizer that converges to the minimizer of the DR objective using only $O(1)$ oracle calls per iterate?

Contributions

We propose **Prospect**, a distributionally robust optimization algorithm that:

1. Makes $O(1)$ calls to function value/gradient oracles per iteration.

2. Converges linearly for *any* positive shift cost.

3. Requires tuning a single hyperparameter (a constant learning rate).

4. Converges 2-3x faster than baselines on distribution shift/fairness benchmarks in tabular, vision, and language domains.

Quantitative Finance & Econometrics

Alternative risk measures (functionals of the loss distribution) and their axiomatic properties are well-studied.

He, 2018; Rockafellar 2007; Cotter, 2006; Acerbi, 2002; Daouia, 2019

Statistics

When $\nu = 0$, SRMs reduce to linear combinations of order statistics, or L-estimators.

Huber, 2009; Shorack, 2017

Spectral Risk Objectives in Machine Learning

Many recent examples of spectral risk-based objectives have appeared in ML, with focus on the superquantile.

Maurer, 2021; Laguel, 2021; Khim, 2020; Holland, 2022

Distributionally Robust Optimization Methods

Optimization approaches rely on full-batch gradient descent, biased SGD, or saddle-point formulations.

Levy 2020; Yu 2022; Yang 2020; Palaniappan, 2016; Kawaguchi & Lu, 2020;

Quantitative Finance & Econometrics

Alternative risk measures (functionals of the loss distribution) and their axiomatic properties are well-studied.

He, 2018; Rockafellar 2007; Cotter, 2006; Acerbi, 2002; Daouia, 2019

Statistics

When $\nu = 0$, SRMs reduce to linear combinations of order statistics, or L-estimators.

Huber, 2009; Shorack, 2017

Spectral Risk Objectives in Machine Learning

Many recent examples of spectral risk-based objectives have appeared in ML, with focus on the superquantile.

Maurer, 2021; Laguel, 2021; Khim, 2020; Holland, 2022

Distributionally Robust Optimization Methods

Optimization approaches rely on full-batch gradient descent, biased SGD, or saddle-point formulations.

Levy 2020; Yu 2022; Yang 2020; Palaniappan, 2016; Kawaguchi & Lu, 2020;

Outline

Prospect: Bias and Variance Reduction

Theoretical and Empirical Performance

Conclusion & Future Work

$$R(w) := \max_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i \ell_i(w) - \nu D(q \| \mathbf{1}_n / n)$$

How do we compute the gradient of this objective?

How do we estimate the gradient?

How do we reduce the bias and variance of the estimate?

$$R(w) := \max_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i \ell_i(w) - \nu D(q \| \mathbf{1}_n / n)$$

How do we compute the gradient of this objective?

$$\nabla R(w) := \sum_{i=1}^{n} q_i^{\ell(w)} \nabla \ell_i(w)$$

$$q^l := \text{argmax}_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i l_i - \nu D(q \| \mathbf{1}_n / n)$$

How do we compute the gradient of this objective?

Step 1: Find the "most adversarial" distribution for model performance $\ell(w)$.

$$\nabla R(w) := \sum_{i=1}^{n} q_i^{\ell(w)} \nabla \ell_i(w)$$

$$q^l := \text{argmax}_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i l_i - \nu D(q \| \mathbf{1}_n / n)$$

How do we compute the gradient of this objective?

Step 2: Take linear combination of the gradients from each loss.

$$\nabla R(w) := \sum_{i=1}^{n} q_i^{\ell(w)} \nabla \ell_i(w)$$

$$q^l := \text{argmax}_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i l_i - \nu D(q \| \mathbf{1}_n / n)$$

Bias

$$\nabla R(w) := \sum_{i=1}^{n} q_i^{\ell(w)} \nabla \ell_i(w) \approx n q_i^{\ell(w)} \nabla \ell_i(w)$$

$$q^l := \text{argmax}_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i l_i - \nu D(q \| \mathbf{1}_n / n)$$

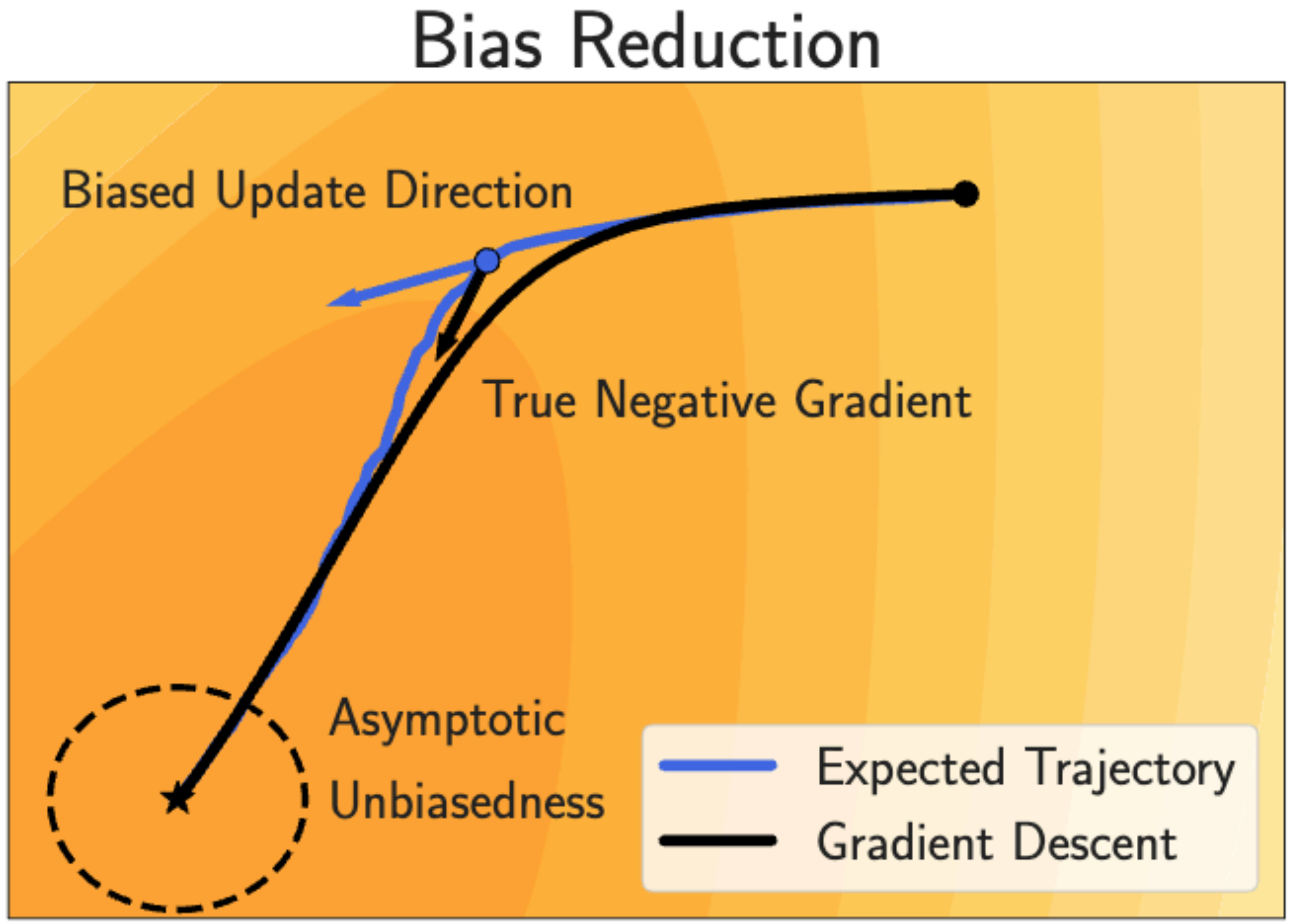
Bias

Prospect: Maintain a running table $l \in \mathbb{R}^n$ and replace l_i with $\ell_i(w)$ at each iteration

$$\nabla R(w) := \sum_{i=1}^{n} q_i^{\ell(w)} \nabla \ell_i(w) \approx n q_i^{\ell(w)} \nabla \ell_i(w) \approx n q_i^{l} \nabla \ell_i(w)$$

$$q^l := \mathrm{argmax}_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i l_i - \nu D(q \| \mathbf{1}_n / n)$$

Bias



l will approach $\ell(w)$ as $w \to w^\star$

Prospect: Maintain a running table $l \in \mathbb{R}^n$ and replace l_i with $\ell_i(w)$ at each iteration

$$\nabla R(w) := \sum_{i=1}^{n} q_i^{\ell(w)} \nabla \ell_i(w) \approx n q_i^{\ell(w)} \nabla \ell_i(w) \approx n q_i^{l} \nabla \ell_i(w)$$

$$q^l := \operatorname{argmax}_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i l_i - \nu D(q \| \mathbf{1}_n / n)$$

Variance

Prospect: Maintain a running tables $\rho \in \mathbb{R}^n$ and $g_1, \ldots, g_n \in \mathbb{R}^d$ and replace $\rho_i = q_i^l$ and $g_i = \nabla \ell_i(w)$ at each iteration

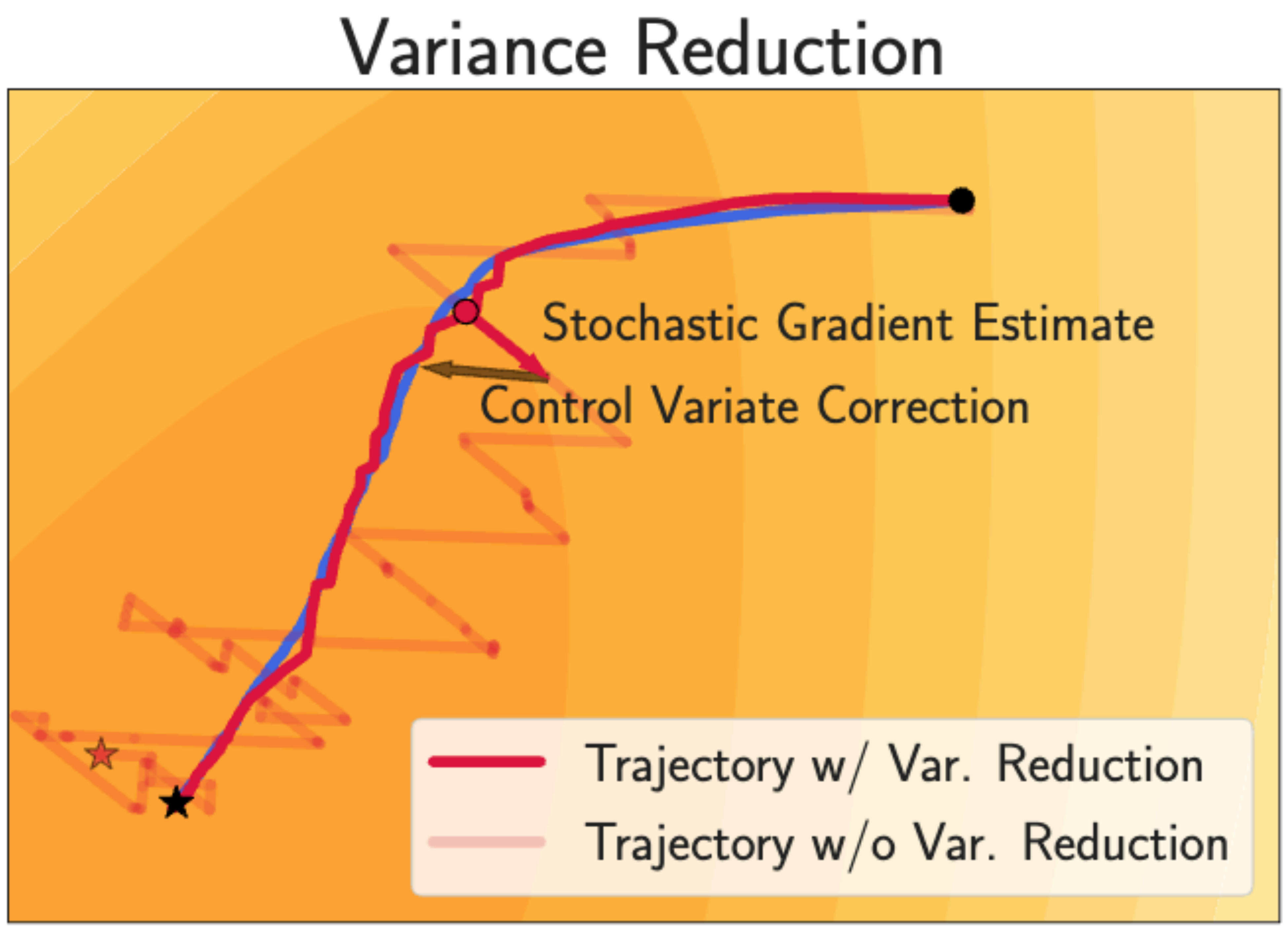
$$\nabla R(w) := \sum_{i=1}^{n} q_i^{\ell(w)} \nabla \ell_i(w) \approx n q_i^l \nabla \ell_i(w) - (n\rho_i g_i - \textstyle\sum_{j=1}^{n} \rho_j g_j)$$

Control Variate: Guesses the direction from the mean to the estimate, and subtracts off that direction.

$$q^l := \text{argmax}_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i l_i - \nu D(q \| \mathbf{1}_n / n)$$

Variance

g will approach $\nabla \ell(w)$ and ρ will approach $q^{\ell(w)}$ as iterations progress



Prospect: Maintain a running tables $\rho \in \mathbb{R}^n$ and $g_1, \ldots, g_n \in \mathbb{R}^d$ and replace $\rho_i = q_i^l$ and $g_i = \nabla \ell_i(w)$ at each iteration

$$\nabla R(w) := \sum_{i=1}^{n} q_i^{\ell(w)} \nabla \ell_i(w) \approx n q_i^l \nabla \ell_i(w) - (n\rho_i g_i - \textstyle\sum_{j=1}^{n} \rho_j g_j)$$

$$q^l := \text{argmax}_{q \in \mathcal{U}} \sum_{i=1}^{n} q_i l_i - \nu D(q \| \mathbf{1}_n / n)$$

Control Variate: Guesses the direction from the mean to the estimate, and subtracts off that direction.

Prospect Algorithm

- Initialize $w = w_0$, $l = \ell(w_0)$, $\rho = q^l$, and $g = \nabla \ell(w)$.
- For each iteration:
    - Compute $v = nq_i^l \nabla \ell_i(w) - (n\rho_i g_i - \sum_{j=1}^n \rho_j g_j)$.
    - Update $w \leftarrow w - \eta v$.
    - Recompute q^l (solve maximization), update one element of l, g, and ρ.

Outline

Prospect: Bias and Variance Reduction

Theoretical and Empirical Performance

Conclusion & Future Work

Theorem

Assume that $\ell_i(w) = f_i(w) + \frac{\mu}{2}\|w\|_2^2$,

where f is G-Lipschitz and ∇f is L-Lipschitz.

Then, **Prospect** with sufficiently small stepsize satisfies:

$$\mathbb{E}\|w_t - w^\star\|_2^2 \lesssim C\|w_0 - w^\star\|_2^2 \cdot e^{-\frac{t}{\tau}}$$

Theorem

Assume that $\ell_i(w) = f_i(w) + \frac{\mu}{2}\|w\|_2^2$,

where f is G-Lipschitz and ∇f is L-Lipschitz.

Then, **Prospect** with sufficiently small stepsize satisfies:

$$\mathbb{E}\|w_t - w^\star\|_2^2 \lesssim C\|w_0 - w^\star\|_2^2 \cdot e^{-\frac{t}{\tau}}$$

If $\nu \gtrsim G^2/\mu$, then

$\tau = n + nq_{\max}(L + \mu)/\mu$

Standard Linear Regression

← Uncertainty Sets →

y : Suboptimality

$$\frac{R(w_t) - R(w^\star)}{R(w_0) - R(w^\star)}$$

← Datasets →

x : Passes through Training Set

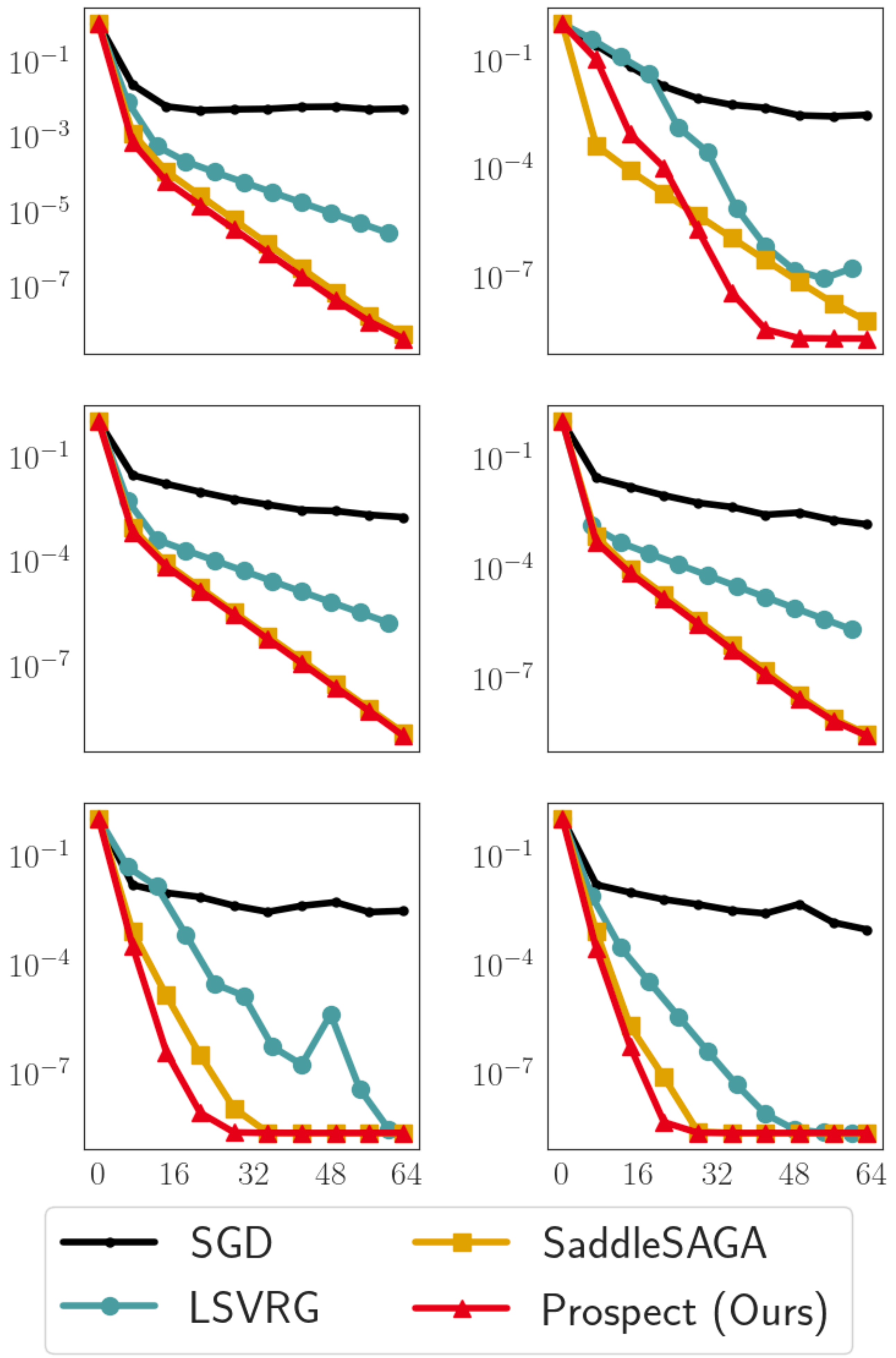
Standard Linear Regression

y : Suboptimality

$$\frac{R(w_t) - R(w^\star)}{R(w_0) - R(w^\star)}$$

← Uncertainty Sets →

↓ Datasets ↑



x : Passes through Training Set

Fairness in Binary Classification

← Uncertainty Sets →

y : Suboptimality

Optimization Metric →

y : Statistical Parity

Fairness Metric →

Statistical Parity

Task: Predict hospital re-admission of diabetes patients.

Test Metric: difference in predicted rates for men and women.

x : Passes through Training Set

Fairness in Binary Classification

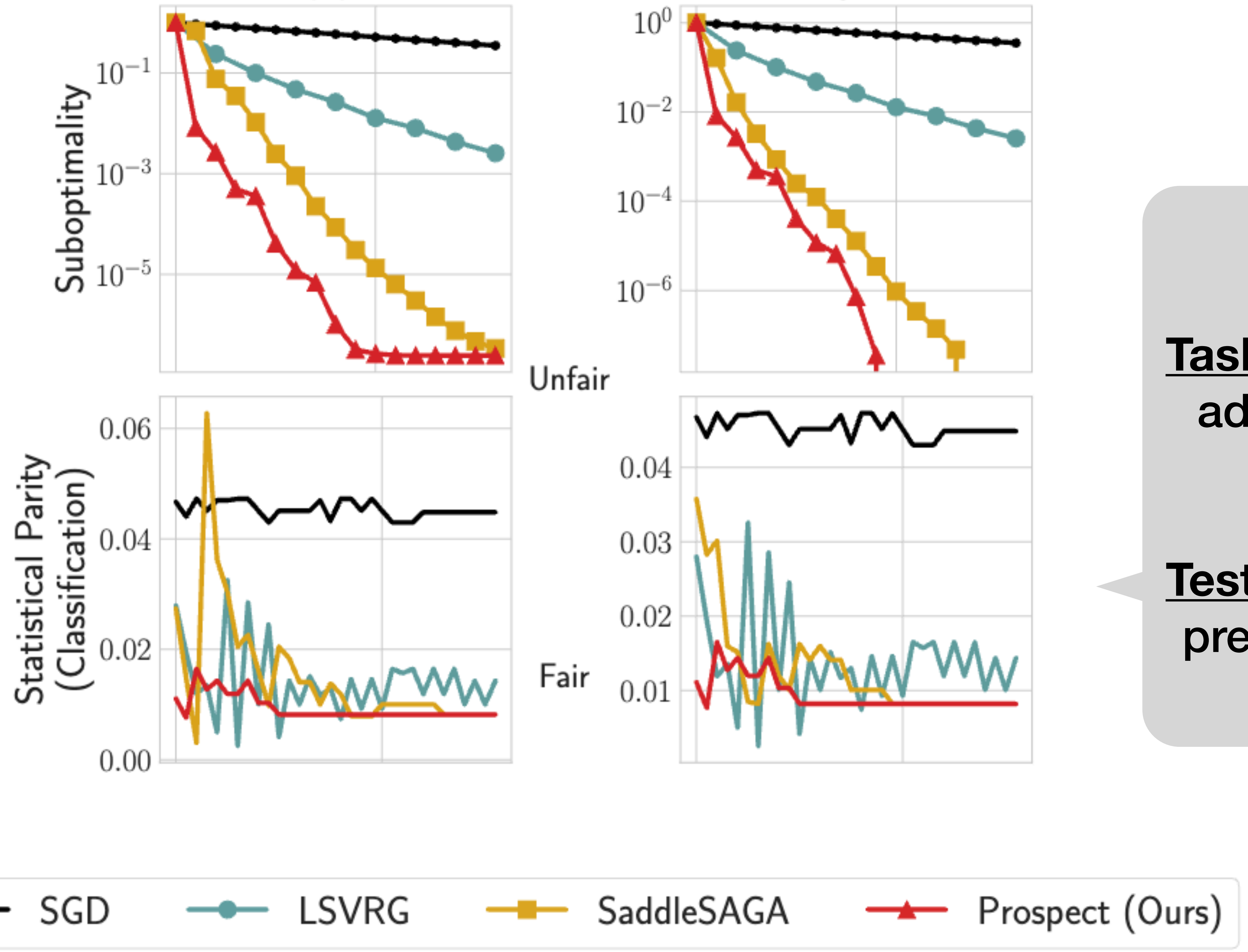
← Uncertainty Sets →

y : Suboptimality

Optimization Metric →

y : Statistical Parity

Fairness Metric →



Statistical Parity

Task: Predict hospital re-admission of diabetes patients.

Test Metric: difference in predicted rates for men and women.

x : Passes through Training Set

Distribution Shift in Text Classification

y : Suboptimality

y : Worst Group Error

x : Passes through Training Set

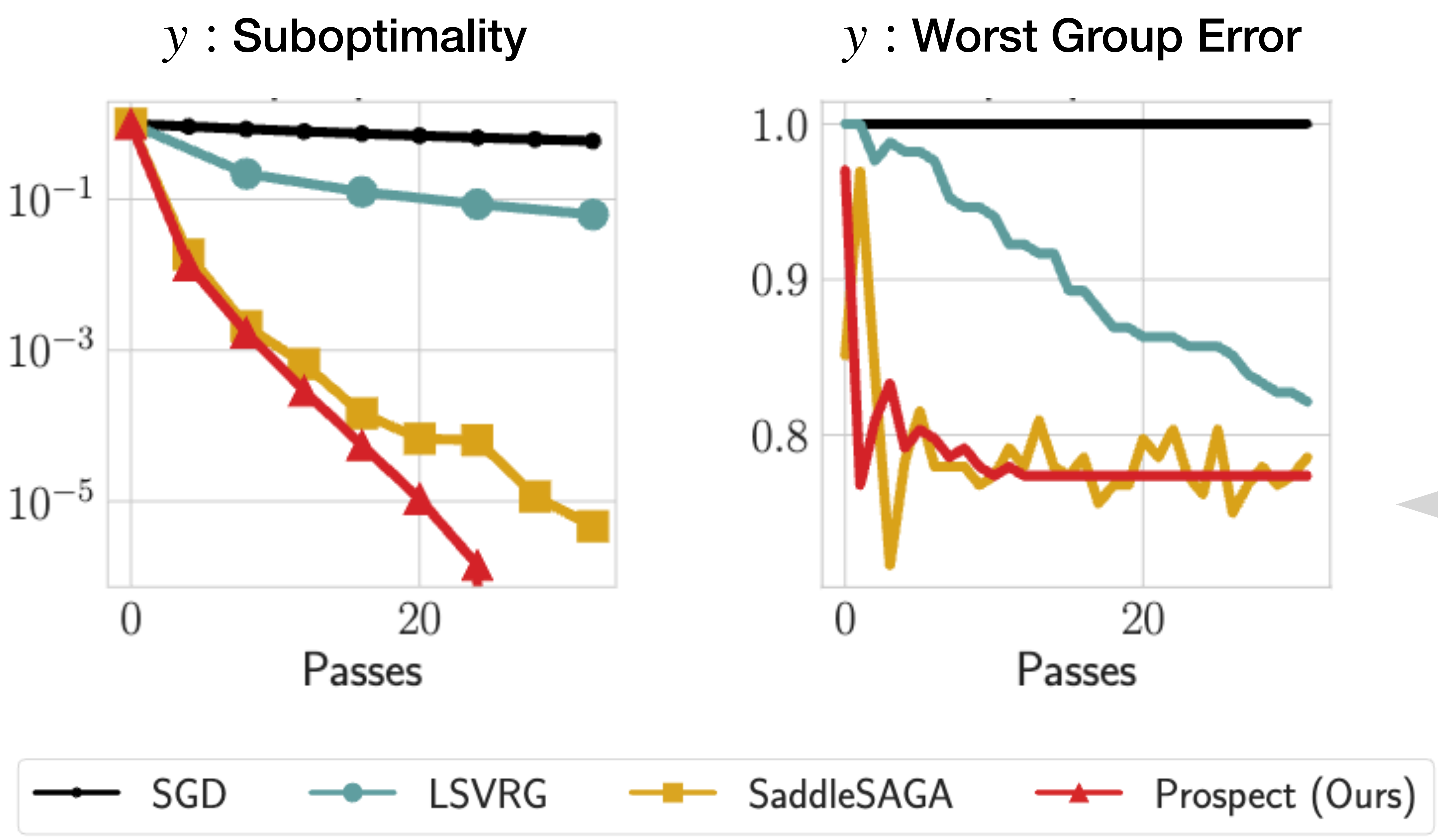
Distribution Shift

Task: Predict number of stars from Amazon reviews.

Shift: Subpopulations of reviewers are different between train, validation, and test set.

Test Metric: Worst classification error among test subpopulations.

Distribution Shift in Text Classification



x : Passes through Training Set

Distribution Shift

Task: Predict number of stars from Amazon reviews.

Shift: Subpopulations of reviewers are different between train, validation, and test set.

Test Metric: Worst classification error among test subpopulations.

Outline

Prospect: Bias and Variance Reduction

Theoretical and Empirical Performance

Conclusion & Future Work

Summary

- We present a stochastic algorithm to optimize distributionally robust of the empirical loss distribution that:
  - finds an exact minimizer/is asymptotically unbiased
  - makes $O(1)$ calls to a function/gradient oracle per update, and
  - outperforms out-of-the-box convex optimizers on real data.
- Future work includes extensions to the non-convex setting and exploring statistical properties of learned minimizers.

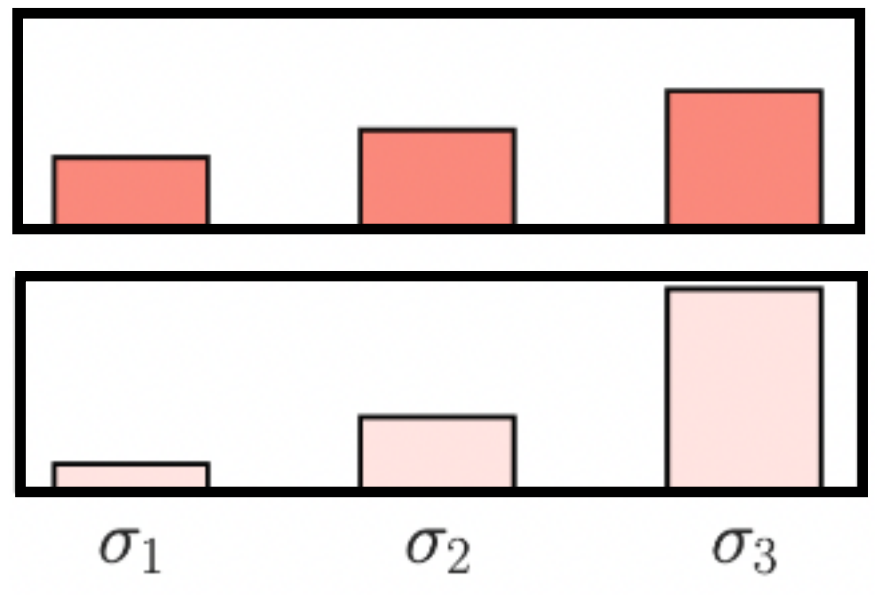
Thank you!

SCAN ME

Appendix

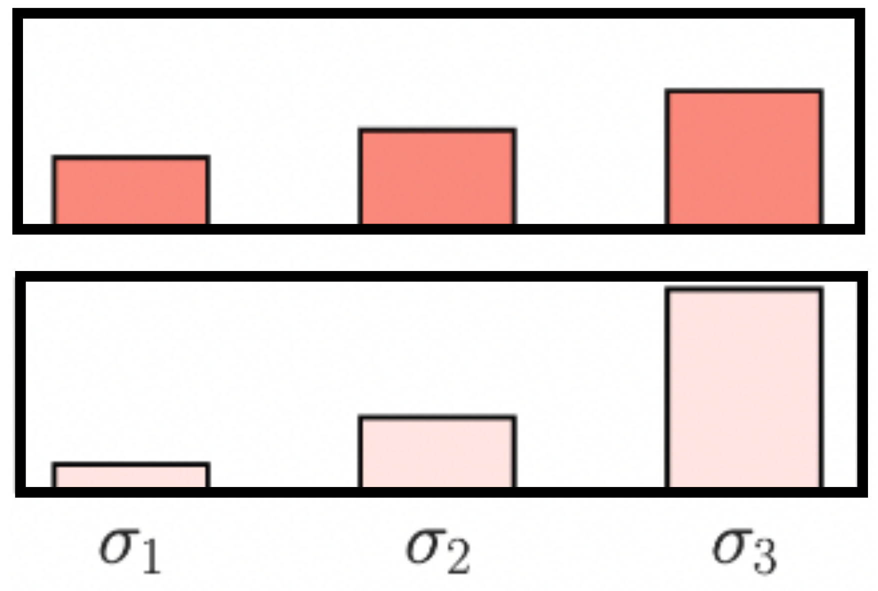
Spectral risk measures are an example of a distributionally robust objective.

$$\sum_{i=1}^{n} \sigma_i l_{(i)}$$



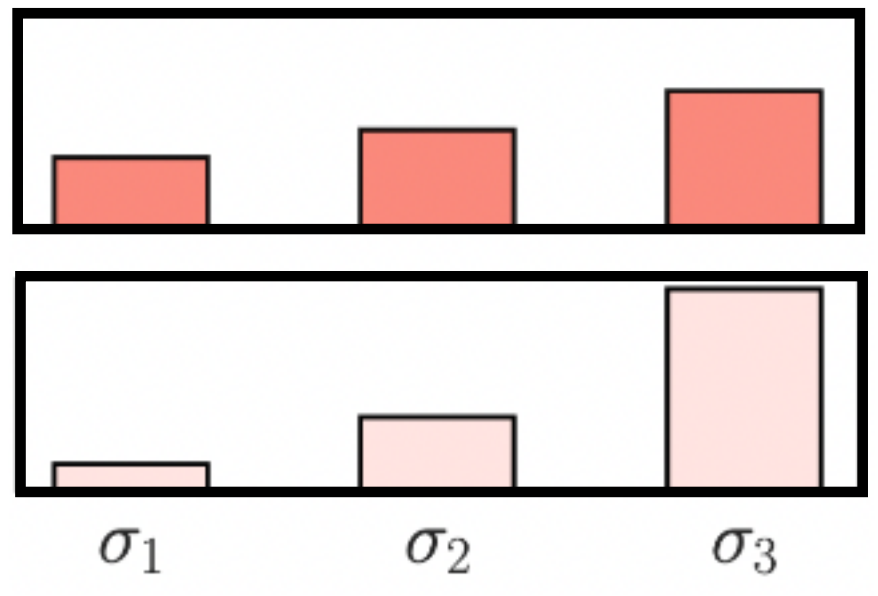
$$\sum_{i=1}^{n} \sigma_i l_{(i)}$$

Use non-negative weights $\sigma_1 \leq \ldots \leq \sigma_n$ with $\sum_{i=1}^{n} \sigma_i = 1$, and take linear combination of order statistics.



$$\sum_{i=1}^{n} \sigma_i l_{(i)} = \max_{\pi} \sum_{i=1}^{n} \sigma_{\pi(i)} l_i$$

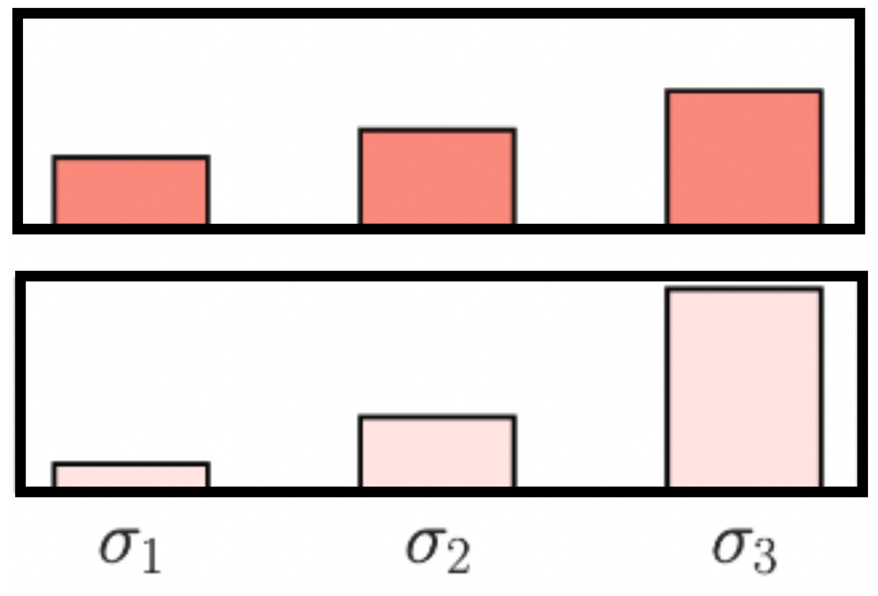
Maximize inner product over all permutations of $(\sigma_1, \ldots, \sigma_n)$ to recover the LHS quantity.

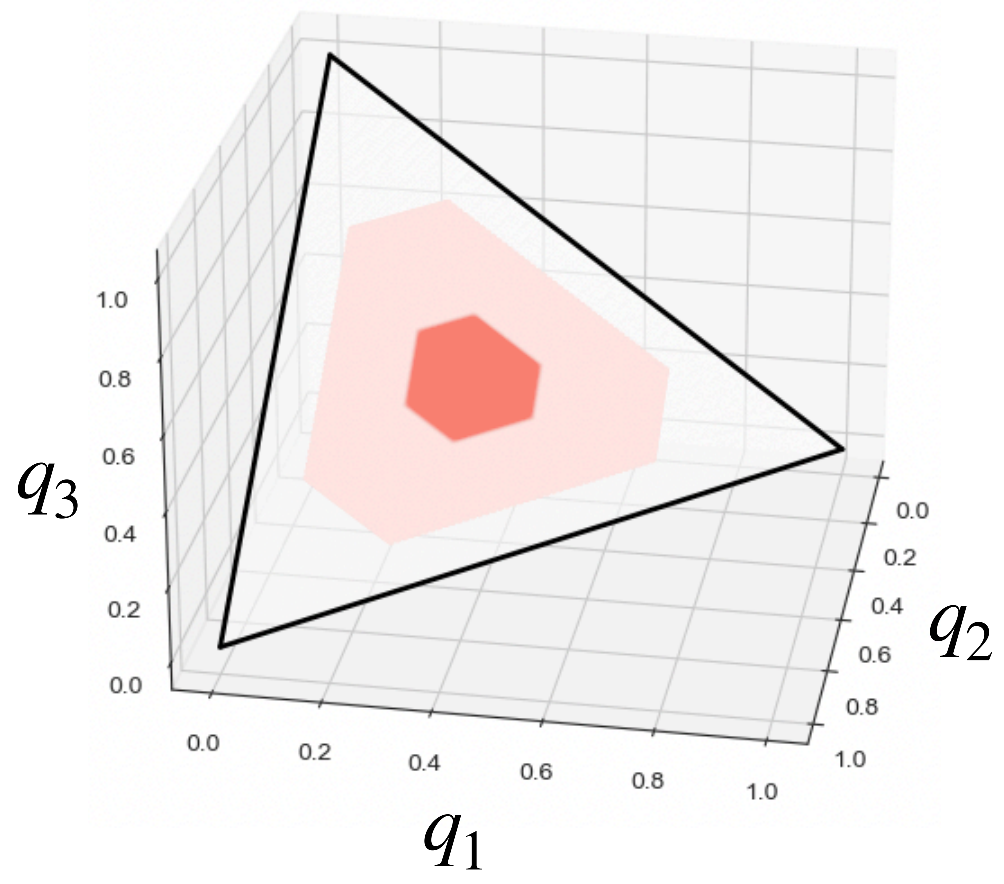


$\mathscr{P}(\sigma) :=$ {**convex hull of permutations of** σ}

$$\sum_{i=1}^{n} \sigma_i l_{(i)} = \max_{\pi} \sum_{i=1}^{n} \sigma_{\pi(i)} l_i = \max_{q \in \mathscr{P}(\sigma)} \sum_{i=1}^{n} q_i l_i$$

Maximum of linear objective over a polytope is achieved on a vertex, so we can maximize over the convex hull.





$\mathscr{P}(\sigma) :=$ {convex hull of permutations of σ}

$$\sum_{i=1}^{n} \sigma_i l_{(i)} = \max_{\pi} \sum_{i=1}^{n} \sigma_{\pi(i)} l_i = \max_{q \in \mathscr{P}(\sigma)} \sum_{i=1}^{n} q_i l_i$$

Maximum of linear objective over a polytope is achieved on a vertex, so we can maximize over the convex hull.

Spectral risk measures are generated by letting $\mathscr{U}$ be a permutahedron in $\mathbb{R}^n$.

$$\min_{w \in \mathbb{R}^d} \max_{q \in \mathscr{P}(\sigma)} \sum_{i=1}^{n} q_i \ell(w, Z_i) - \nu D(q \| \mathbf{1}_n / n)$$

Spectral risk measures are generated by letting $\mathscr{U}$ be a permutahedron in $\mathbb{R}^n$.

$$\min_{w \in \mathbb{R}^d} \max_{q \in \mathscr{P}(\sigma)} \sum_{i=1}^{n} q_i \ell(w, Z_i) - \nu D(q \| \mathbf{1}_n / n)$$