L4DC tutorial
July 15, 2024

# supervised and reinforcement learning under requirements

Miguel Calvo-Fullana
Universitat Pompeu Fabra, Spain

Luiz F. O. Chamon
Universität Stuttgart, Germany

Santiago Paternain
Rensselaer Polytechnic Institute, USA

Alejandro Ribeiro
University of Pennsylvania, USA

## Agenda

I. Constrained supervised learning
- Constrained learning theory
- Resilient constrained learning
- Robust learning

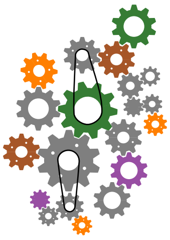
Break (30 min)

II. Constrained reinforcement learning
- Constrained RL duality
- Constrained RL algorithms

https://luizchamon.com/l4dc

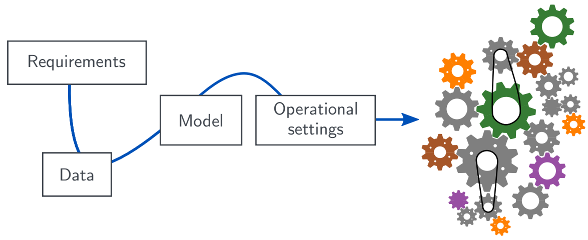


## Why requirements?



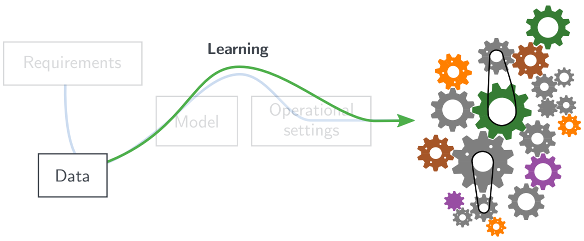


## Why requirements?



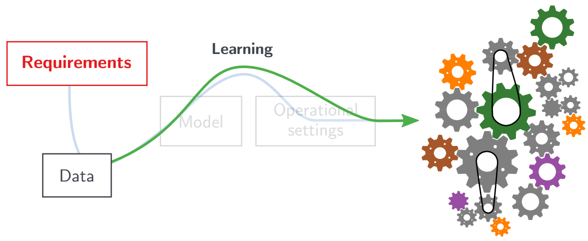



## Why requirements?





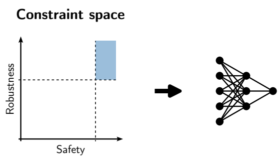

## Why requirements?






## What is a requirements?

- **Requirements are "shall" statements:** describe *necessary* features subject to verification
  - *Constraint space*: things we decide

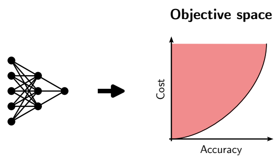



[NASA. "Systems engineering handbook," 2019]



## What is a requirements?

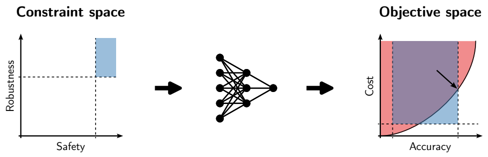
- **Requirements are "shall" statements:** describe *necessary* features subject to verification
  - *Constraint space*: things we decide
- **Goals are "should" statements:** express recommendations (once "shall" statements are satisfied)
  - *Objective space*: things the system achieves




[NASA. "Systems engineering handbook," 2019]

# What is a requirements?

- **Requirements are "shall" statements:** describe *necessary* features subject to verification
  - *Constraint space*: things we decide
- **Goals are "should" statements:** express recommendations (once "shall" statements are satisfied)
  - *Objective space*: things the system achieves




[NASA. "Systems engineering handbook," 2019]



# What is (un)constrained learning?

$$P_{\text{U}}^\star = \min_{\theta} \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big]$$

- $\ell, g$ are bounded, Lipschitz continuous (possibly non-convex) functions
- $f_{\boldsymbol{\theta}}$ is a (possibly nonlinear) parametrization [e.g., logistic classifier, (G)(C)NN]
- $\mathfrak{D}, \mathfrak{A}, \mathfrak{P}$ unknown

[Chamon et al., IEEE ICASSP'20 (best student paper); Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



# What is (un)constrained learning?

$$\begin{aligned} P^\star = \min_{\theta} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big) \leq u, \quad \mathfrak{P}\text{-a.e.} \end{aligned}$$

- $\ell, g$ are bounded, Lipschitz continuous (possibly non-convex) functions
- $f_{\boldsymbol{\theta}}$ is a (possibly nonlinear) parametrization [e.g., logistic classifier, (G)(C)NN]
- $\mathfrak{D}, \mathfrak{A}, \mathfrak{P}$ unknown

[Chamon et al., IEEE ICASSP'20 (best student paper); Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



# What about penalties?

$$\begin{aligned} P^\star = \min_{\theta} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big) \leq u, \quad \mathfrak{P}\text{-a.e.} \end{aligned}$$

$$\updownarrow$$

$$\min_{\theta} \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] + \lambda\,\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{P}}\Big[\mu(\boldsymbol{x},y)h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big]$$



# What about penalties?

NON-CONVEX

$$\begin{aligned} P^\star = \min_{\theta} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big) \leq u, \quad \mathfrak{P}\text{-a.e.} \end{aligned}$$

$$\updownarrow$$

$$\min_{\theta} \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] + \lambda\,\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{P}}\Big[\mu(\boldsymbol{x},y)h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big]$$

- ✗ There may not exist $(\lambda, \mu)$ such that the penalized solution is optimal *and* feasible
- ✗ Even if such $(\lambda, \mu)$ exist, they are not easy to find (hyperparameter search, cross-validation. . . )
- ✓ Constrained learning yields better guarantees, better performance, better trade-offs. . .



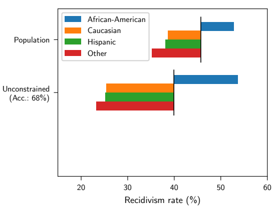
# Applications

- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23])
- Safe learning
  (e.g., [Paternain et al., IEEE TAC'23])
- Wireless resource allocation
  (e.g., [Eisen et al., IEEE TSP'19; NaderiAlizadeh et al., IEEE TSP'22; Chowdhury et al., Asilomar'23])
- . . .

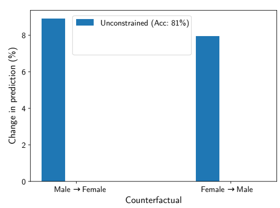


# Fairness

**Problem**
Predict whether an individual will recidivate




**Problem**
Predict whether an individual makes > \$50k




* We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.



# Fairness: "Equality" of odds

**Problem**
Predict whether an individual will recidivate at the same rate across races

$$\begin{aligned} \min_{\theta} \quad & \text{Prediction error} \\ \text{subject to} \quad & \text{Prediction rate disparity (Race)} \leq c, \\ & \text{for Race} \in \{\text{African-American, Caucasian, Hispanic, Other}\} \end{aligned}$$

* We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.
[Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23]

# Fairness: “Equality” of odds

## Problem

Predict whether an individual will recidivate at the same rate across races

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \text{Prediction rate disparity (Race)} \leq c, \\ & \quad \text{for Race} \in \{\text{African-American, Caucasian, Hispanic, Other}\} \end{aligned}$$

*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23]



# Fairness: “Equality” of odds

## Problem

Predict whether an individual will recidivate at the same rate across races

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_\theta(\boldsymbol{x}_n) = 1 \mid \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_\theta(\boldsymbol{x}_n) = 1\right] + c, \\ & \quad \text{for Race} \in \{\text{African-American, Caucasian, Hispanic, Other}\} \end{aligned}$$

*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23]



# Counterfactual fairness

## Problem

Predict whether an individual makes > $50k while being invariant to gender

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \text{Change in prediction } (\rho\boldsymbol{x}) \leq c \quad \text{a.e.} \\ & \quad (\rho : \text{Male} \leftrightarrow \text{Female}) \end{aligned}$$

*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon and Ribeiro, NeurIPS'20]



# Counterfactual fairness

## Problem

Predict whether an individual makes > $50k while being invariant to gender

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \mathrm{D}_{\mathrm{KL}}\big(f_\theta(\boldsymbol{x}_n) \| f_\theta(\rho\boldsymbol{x}_n)\big) \leq c, \quad \text{for all } n \\ & \quad (\rho : \text{Male} \leftrightarrow \text{Female}) \end{aligned}$$

*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon and Ribeiro, NeurIPS'20]



# Applications

- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23])
- Safe learning
  (e.g., [Paternain et al., IEEE TAC'23])
- Wireless resource allocation
  (e.g., [Eisen et al., IEEE TSP'19; NaderiAlizadeh et al., IEEE TSP'22; Chowdhury et al., Asilomar'23])
- . . .



# Federated learning

## Problem

Learn a common model using data using data distributed among $K$ clients

$$\min_{\theta} \quad \text{Average loss across clients}$$


- $k$-th client loss: $\text{Loss}_k(f_\theta) = \dfrac{1}{N_k}\displaystyle\sum_{n_k=1}^{N_k} \text{Loss}\big(f_\theta(\boldsymbol{x}_{n_k}), y_{n_k}\big)$

[Shen et al., ICLR'22]



# Federated learning

## Problem

Learn a common model using data using data distributed among $K$ clients

$$\min_{\theta} \quad \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_\theta)$$


- $k$-th client loss: $\text{Loss}_k(f_\theta) = \dfrac{1}{N_k}\displaystyle\sum_{n_k=1}^{N_k} \text{Loss}\big(f_\theta(\boldsymbol{x}_{n_k}), y_{n_k}\big)$

[Shen et al., ICLR'22]



# Federated learning

## Problem

Learn a common model using data using data distributed among $K$ clients

$$\min_{\theta} \quad \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_\theta)$$


- $k$-th client loss: $\text{Loss}_k(f_\theta) = \dfrac{1}{N_k}\displaystyle\sum_{n_k=1}^{N_k} \text{Loss}\big(f_\theta(\boldsymbol{x}_{n_k}), y_{n_k}\big)$

[Shen et al., ICLR'22]

# Federated learning

**Problem**
Learn a common model using data using data distributed among $K$ clients

$$\begin{aligned}\min_{\theta}\quad & \frac{1}{K}\sum_{k=1}^{K}\text{Loss}_k(f_\theta)\\ \text{subject to}\quad & \text{Loss disparity } (k\text{-th client}) \leq c,\\ & k=1,\dots,K\end{aligned}$$


- $k$-th client loss: $\text{Loss}_k(f_\theta) = \frac{1}{N_k}\sum_{n_k=1}^{N_k}\text{Loss}\big(f_\theta(x_{n_k}), y_{n_k}\big)$

[Shen et al., ICRL'22]



# Federated learning

**Problem**
Learn a common model using data using data distributed among $K$ clients

$$\begin{aligned}\min_{\theta}\quad & \frac{1}{K}\sum_{k=1}^{K}\text{Loss}_k(f_\theta)\\ \text{subject to}\quad & \text{Loss}_k(f_\theta) \leq \frac{1}{K}\sum_{k=1}^{K}\text{Loss}_k(f_\theta) + c,\\ & k=1,\dots,K\end{aligned}$$


- $k$-th client loss: $\text{Loss}_k(f_\theta) = \frac{1}{N_k}\sum_{n_k=1}^{N_k}\text{Loss}\big(f_\theta(x_{n_k}), y_{n_k}\big)$

[Shen et al., ICRL'22]



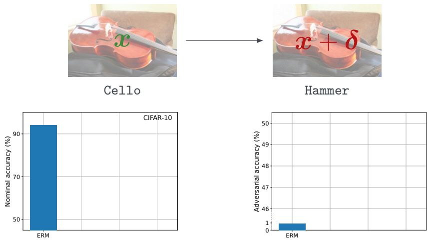
# Applications

- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23])
- Safe learning
  (e.g., [Paternain et al., IEEE TAC'23])
- Wireless resource allocation
  (e.g., [Eisen et al., IEEE TSP'19; NaderiAlizadeh et al., IEEE TSP'22; Chowdhury et al., Asilomar'23])
- ...



# Robustness

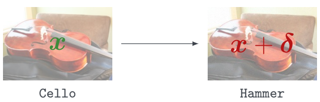
**Problem**
Learn a classifier that is robust to input perturbations






# Robustness

**Problem**
Learn a classifier that is robust to input perturbations



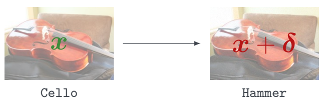

$$\begin{aligned}\min_{\theta}\quad & \text{Nominal accuracy}\\ \text{subject to}\quad & \text{Robustness} \leq c\end{aligned}$$

[Chamon and Ribeiro, NeurIPS'20; Robey*, Chamon*, Pappas, Hassani, and Ribeiro, NeurIPS'21; Chamon, Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]



# Robustness

**Problem**
Learn a classifier that is robust to input perturbations



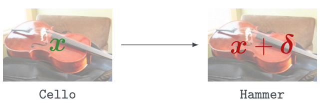

$$\begin{aligned}\min_{\theta}\quad & \frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_\theta(x_n), y_n\big)\\ \text{subject to}\quad & \text{Robustness} \leq c\end{aligned}$$

[Chamon and Ribeiro, NeurIPS'20; Robey*, Chamon*, Pappas, Hassani, and Ribeiro, NeurIPS'21; Chamon, Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]



# Robustness

**Problem**
Learn a classifier that is robust to input perturbations



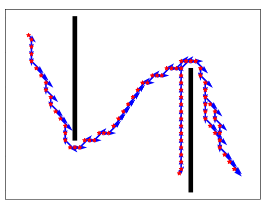

$$\begin{aligned}\min_{\theta}\quad & \frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_\theta(x_n), y_n\big)\\ \text{subject to}\quad & \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\|\delta\|_\infty\leq\epsilon}\text{Loss}\big(f_\theta(x_n+\delta), y_n\big)\right] \leq c\end{aligned}$$

[Chamon and Ribeiro, NeurIPS'20; Robey*, Chamon*, Pappas, Hassani, and Ribeiro, NeurIPS'21; Chamon, Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]



# (Manifold) smoothness

**Problem**
Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories

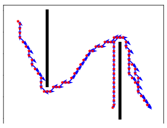


[Cerviño et al., ICML'23]

# (Manifold) smoothness

## Problem

Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories



$$\begin{array}{ll} \underset{\boldsymbol{\theta}}{\min} & \text{Imitation error} \\ \text{subject to} & \text{Smoothness in free space} \leq L \end{array}$$

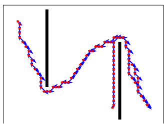
[Cerviño et al., ICML'23]



# (Manifold) smoothness

## Problem

Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories



$$\begin{array}{ll} \underset{\boldsymbol{\theta}}{\min} & \frac{1}{N} \sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), \boldsymbol{u}_n\big) \\ \text{subject to} & \text{Smoothness in free space} \leq L \end{array}$$

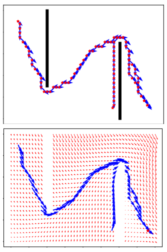
[Cerviño et al., ICML'23]



# (Manifold) smoothness

## Problem

Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories



$$\begin{array}{ll} \underset{\boldsymbol{\theta}}{\min} & \frac{1}{N} \sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), \boldsymbol{u}_n\big) \\ \text{subject to} & \max_{\boldsymbol{x} \in \mathcal{M}} \|\nabla_{\mathcal{M}} f_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2 \leq L \end{array}$$

[Cerviño et al., ICML'23]



# Applications

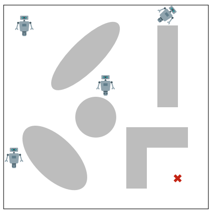
- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23])
- Safe learning
  (e.g., [Paternain et al., IEEE TAC'23])
- Wireless resource allocation
  (e.g., [Eisen et al., IEEE TSP'19; NaderiAlizadeh et al., IEEE TSP'22; Chowdhury et al., Asilomar'23])
- ...



# Safety

## Problem

Find a control policy that navigates the environment effectively and safely

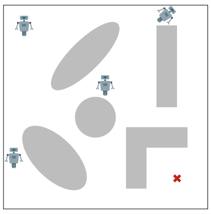




# Safety

## Problem

Find a control policy that navigates the environment effectively and safely



$$\begin{array}{ll} \underset{\pi \in \mathcal{P}(\mathcal{S})}{\text{maximize}} & \text{Task reward} \\ \text{subject to} & \Pr[\text{Colliding with } \mathcal{O}_i] \leq \delta, \\ & \text{for } i = 1, 2, \ldots \end{array}$$

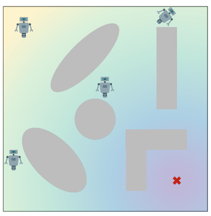
[Paternain et al., IEEE TAC'23]



# Safety

## Problem

Find a control policy that navigates the environment effectively and safely



$$\begin{array}{ll} \underset{\pi \in \mathcal{P}(\mathcal{S})}{\text{maximize}} & \mathbb{E}_{s,a \sim \pi}\left[\frac{1}{T} \sum_{t=0}^{T-1} r_0(s_t, a_t)\right] \\ \text{subject to} & \Pr[\text{Colliding with } \mathcal{O}_i] \leq \delta, \\ & \text{for } i = 1, 2, \ldots \end{array}$$

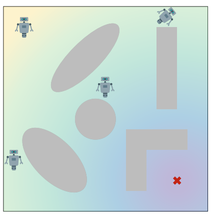
[Paternain et al., IEEE TAC'23]



# Safety

## Problem

Find a control policy that navigates the environment effectively and safely



$$\begin{array}{ll} \underset{\pi \in \mathcal{P}(\mathcal{S})}{\text{maximize}} & \mathbb{E}_{s,a \sim \pi}\left[\frac{1}{T} \sum_{t=0}^{T-1} r_0(s_t, a_t)\right] \\ \text{subject to} & \Pr\left(\bigcap_{t=0}^{T-1} \{s_t \notin \mathcal{O}_i\} \,\middle|\, \pi\right) \geq 1 - \delta_i, \\ & \text{for } i = 1, 2, \ldots \end{array}$$

[Paternain et al., IEEE TAC'23]

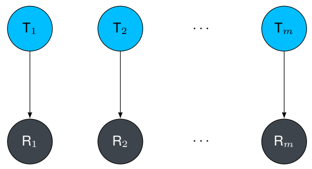
# Applications

- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23])
- Safe learning
  (e.g., [Paternain et al., IEEE TAC'23])
- Wireless resource allocation
  (e.g., [Eisen et al., IEEE TSP'19; NaderiAlizadeh et al., IEEE TSP'22; Chowdhury et al., Asilomar'23])
- . . .



# Wireless resource allocation

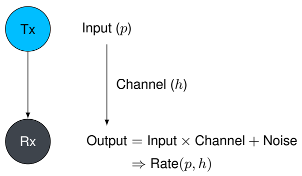
Problem
Allocate the least transmit power to $m$ device pairs to achieve a communication rate






# Wireless resource allocation

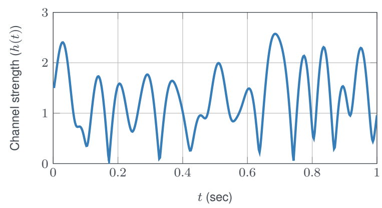
Problem
Allocate the least transmit power to $m$ device pairs to achieve a communication rate






# Wireless resource allocation

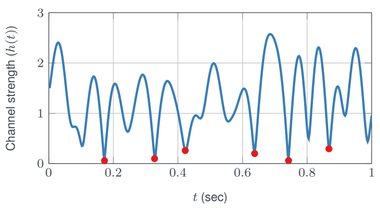
Problem
Allocate the least transmit power to $m$ device pairs to achieve a communication rate






# Wireless resource allocation

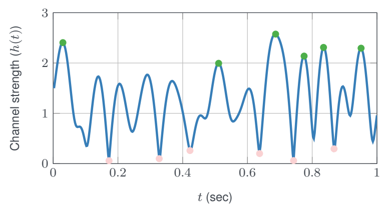
Problem
Allocate the least transmit power to $m$ device pairs to achieve a communication rate






# Wireless resource allocation

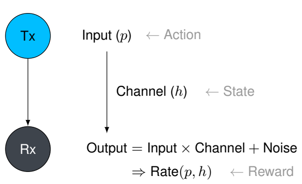
Problem
Allocate the least transmit power to $m$ device pairs to achieve a communication rate






# Wireless resource allocation

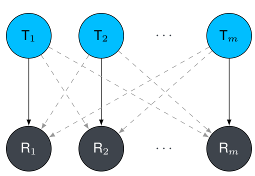
Problem
Allocate the least transmit power to $m$ device pairs to achieve a communication rate






# Wireless resource allocation

Problem
Allocate the least transmit power to $m$ device pairs to achieve a communication rate

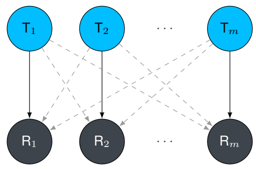



$$\begin{array}{ll} \min\limits_{p} & \text{Total transmit power} \\ \text{s. to} & \text{Rate } T_i \to R_i \geq c_i \end{array}$$

[Eisen, Zhang, Chamon, Lee, and Ribeiro, IEEE TSP'19]

## Wireless resource allocation

**Problem**
Allocate the least transmit power to $m$ device pairs to achieve a communication rate



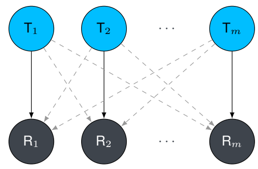

$$\min_{\boldsymbol{p}} \sum_{i=1}^{m} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} p_i(h_t)\right]$$

$$\text{s. to} \quad \text{Rate } T_i \to R_i \geq c_i$$



## Wireless resource allocation

**Problem**
Allocate the least transmit power to $m$ device pairs to achieve a communication rate



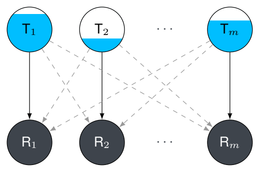

$$\min_{\boldsymbol{p}} \sum_{i=1}^{m} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} p_i(h_t)\right]$$

$$\text{s. to} \quad \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} \text{Rate}_i\big(\boldsymbol{p}(h_t), h_t\big)\right] \geq c_i$$



## Wireless resource allocation

**Problem**
Allocate the least transmit power to $m$ device pairs to achieve a communication rate



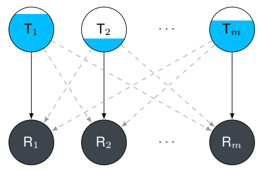

$$\min_{\boldsymbol{p}} \quad \text{Total probability of depleting battery}$$

$$\text{s. to} \quad \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} \text{Rate}_i\big(\boldsymbol{p}(h_t), h_t\big)\right] \geq c_i$$



## Wireless resource allocation

**Problem**
Allocate the least transmit power to $m$ device pairs to achieve a communication rate




$$\min_{\boldsymbol{p}} \sum_{i=1}^{m} \Pr\left[\bigcap_{t=0}^{T-1} \{b_{i,t} = 0\}\right]$$

$$\text{s. to} \quad \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} \text{Rate}_i\big(\boldsymbol{p}(h_t), h_t\big)\right] \geq c_i$$



## And many more...

- Precision, recall, churn (e.g., [Cotter et al., JMLR'19])
- Scientific priors (e.g., [Lu et al., SIAM J. Sci. Comp.'21])
- Continual learning (e.g., [Peng et al., ICML'23])
- Active learning (e.g., [Elenter et al., NeurIPS'22])
- Data augmentation (e.g., [Hounie et al., ICML'23])
- Semi-supervised learning (e.g., [Cerviño et al., ICML'23])
- Minimum norm interpolation, SVM...



# Constrained supervised learning

## What is (un)constrained learning?

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \quad \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)$$

$$\text{subject to} \quad \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c$$

$$h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r), y_r\big) \leq u, \quad r = 1, \ldots, N$$

- $\ell, g$ are bounded, Lipschitz continuous (possibly non-convex) functions
- $f_{\boldsymbol{\theta}}$ is a (possibly nonlinear) parametrization [e.g., logistic classifier, (G)(C)NN]
- $(\boldsymbol{x}_n, y_n) \sim \mathfrak{D}$, $(\boldsymbol{x}_m, y_m) \sim \mathfrak{A}$, $(\boldsymbol{x}_r, y_r) \sim \mathfrak{P}$ (i.i.d.)



## What is (un)constrained learning?

$$P^\star = \min_{\boldsymbol{\theta}} \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big]$$

$$\text{subject to} \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \leq c$$

$$h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big) \leq u, \quad \mathfrak{P}\text{-a.e.}$$

- $\ell, g$ are bounded, Lipschitz continuous (possibly non-convex) functions
- $f_{\boldsymbol{\theta}}$ is a (possibly nonlinear) parametrization [e.g., logistic classifier, (G)(C)NN]
- $\mathfrak{D}, \mathfrak{A}, \mathfrak{P}$ unknown

## Constrained learning challenges

$$\hat{P}^\star = \min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_\theta(x_n), y_n\big) \qquad \xrightarrow{?} \qquad P^\star = \min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\ell\big(f_\theta(x), y\big)\Big]$$

$$\text{subject to} \ \frac{1}{N}\sum_{m=1}^{N} g\big(f_\theta(x_m), y_m\big) \leq c \qquad \text{subject to} \ \mathbb{E}_{(x,y)\sim\mathfrak{A}}\Big[g\big(f_\theta(x), y\big)\Big] \leq c$$

$$h\big(f_\theta(x_r), y_r\big) \leq u \qquad h\big(f_\theta(x), y\big) \leq u \text{ a.e.}$$

**Challenges**

1) *Statistical*: does the solution of the constrained empirical problem generalize?



## Constrained learning challenges

$$\hat{P}^\star = \min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_\theta(x_n), y_n\big) \qquad \xrightarrow{?} \qquad P^\star = \min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\ell\big(f_\theta(x), y\big)\Big]$$

$$\text{subject to} \ \frac{1}{N}\sum_{m=1}^{N} g\big(f_\theta(x_m), y_m\big) \leq c \qquad \text{subject to} \ \mathbb{E}_{(x,y)\sim\mathfrak{A}}\Big[g\big(f_\theta(x), y\big)\Big] \leq c$$

$$h\big(f_\theta(x_r), y_r\big) \leq u \qquad h\big(f_\theta(x), y\big) \leq u \text{ a.e.}$$

**Challenges**

1) *Statistical*: does the solution of the constrained empirical problem generalize?

2) *Computational*: can we solve the constrained empirical problem?



## Constrained learning challenges

$$\hat{P}^\star = \min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_\theta(x_n), y_n\big) \qquad \xrightarrow{?} \qquad P^\star = \min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\ell\big(f_\theta(x), y\big)\Big]$$

$$\text{subject to} \ \frac{1}{N}\sum_{m=1}^{N} g\big(f_\theta(x_m), y_m\big) \leq c \qquad \text{subject to} \ \mathbb{E}_{(x,y)\sim\mathfrak{A}}\Big[g\big(f_\theta(x), y\big)\Big] \leq c$$

$$h\big(f_\theta(x_r), y_r\big) \leq u \qquad h\big(f_\theta(x), y\big) \leq u \text{ a.e.}$$

**Challenges**

1) *Statistical*: does the solution of the constrained empirical problem generalize?

2) *Computational*: can we solve the constrained empirical problem?



## Agenda

Constrained learning theory

Constrained learning algorithms

Resilient constrained learning



## Constrained learning challenges

$$\hat{P}^\star = \min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_\theta(x_n), y_n\big) \qquad \xrightarrow{?} \qquad P^\star = \min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\ell\big(f_\theta(x), y\big)\Big]$$

$$\text{subject to} \ \frac{1}{N}\sum_{m=1}^{N} g\big(f_\theta(x_m), y_m\big) \leq c \qquad \text{subject to} \ \mathbb{E}_{(x,y)\sim\mathfrak{A}}\Big[g\big(f_\theta(x), y\big)\Big] \leq c$$

$$h\big(f_\theta(x_r), y_r\big) \leq u \qquad h\big(f_\theta(x), y\big) \leq u \text{ a.e.}$$

**Challenges**

1) *Statistical*: does the solution of the constrained empirical problem generalize?

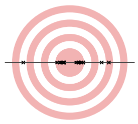
2) *Computational*: can we solve the constrained empirical problem?



## What classical learning theory says?

$$\min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n), y_n\big) \ \xrightarrow{\text{"LLN"}} \ \min_{\theta} \ \mathbb{E}\Big[\text{Loss}\big(f_\theta(x), y\big)\Big]$$

- ✓ $f_\theta$ is *probably approximately correct (PAC)* learnable
  e.g., linear functions, smooth functions (finite RKHS norm, bandlimited), NNs...
  $(N \approx 1/\epsilon^2)$



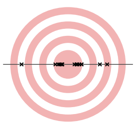
[Rostamizadeh, Talwalkar, Mohri. Foundations of machine learning, 2012]; [Ben-David, Shalev-Shwartz. Understanding machine learning..., 2014]



## What classical learning theory says?

$$\min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n), y_n\big) \ \xrightarrow{\text{"LLN"}} \ \min_{\theta} \ \mathbb{E}\Big[\text{Loss}\big(f_\theta(x), y\big)\Big]$$

- ✓ $f_\theta$ is *probably approximately correct (PAC)* learnable
  e.g., linear functions, smooth functions (finite RKHS norm, bandlimited), NNs...
  $(N \approx 1/\epsilon^2)$
- ✗ Requirements?



[Rostamizadeh, Talwalkar, Mohri. Foundations of machine learning, 2012]; [Ben-David, Shalev-Shwartz. Understanding machine learning..., 2014]

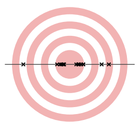


## What's in a solution?

Definition (PAC learnability)

$f_\theta$ is a *probably approximately correct (PAC)* learnable if for every $\epsilon, \delta$ and every distributions $\mathfrak{D}, \mathfrak{A}$, we can obtain $f_{\theta^\dagger}$ from $N_f(\epsilon, \delta)$ samples such that, with prob. $1-\delta$,

- near-optimal $$P^\star - \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\ell\big(f_{\theta^\dagger}(x), y\big)\Big] \ \leq \epsilon$$



[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]

## What's in a solution?

**Definition (PACC learnability)**

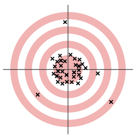
$f_{\boldsymbol{\theta}}$ is a *probably approximately correct constrained (PACC)* learnable if for every $\epsilon, \delta$ and every distributions $\mathfrak{D}, \mathfrak{A}$, we can obtain $f_{\boldsymbol{\theta}^\dagger}$ from $N_f(\epsilon, \delta)$ samples such that, with prob. $1-\delta$,

- near-optimal

$$\left| P^\star - \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\ell\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big)\right] \right| \leq \epsilon$$

- approximately feasible

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\left[g\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big)\right] \leq c + \epsilon$$



[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



## When is constrained learning possible?

$$\begin{aligned} \hat{P}^\star = \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \end{aligned} \quad \xrightarrow{?} \quad \begin{aligned} P^\star = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\right] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\left[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\right] \leq c \end{aligned}$$

**Proposition**

$f_{\boldsymbol{\theta}}$ is PAC learnable $\not\Rightarrow$ $f_{\boldsymbol{\theta}}$ is PACC learnable

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



## ECRM is not a PACC learner

**Counter-example**

$$\begin{aligned} P^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) \\ \text{subject to} \quad & \theta_2\, \mathbb{E}_\tau[\tau] \leq \theta_1 - 1 \\ & -\theta_1\, \mathbb{E}_\tau[\tau] \leq \theta_2 - 1 \end{aligned} \qquad J(\boldsymbol{\theta}) = \begin{cases} 1/16, & \boldsymbol{\theta} = [1/2, 1/2] \\ 1/8, & \boldsymbol{\theta} = [1, 1] \\ 1/4, & \boldsymbol{\theta} = [1, 0] \end{cases}$$

- $\tau \sim$ Uniform$\big(-1/2, 1/2\big)$



## ECRM is not a PACC learner

**Counter-example**

$$\begin{aligned} P^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) = \frac{1}{8} \\ \text{subject to} \quad & \theta_2\, \mathbb{E}_\tau[\tau] \leq \theta_1 - 1 \Rightarrow \theta_1 \geq 1 \\ & -\theta_1\, \mathbb{E}_\tau[\tau] \leq \theta_2 - 1 \Rightarrow \theta_2 \leq 1 \end{aligned} \qquad J(\boldsymbol{\theta}) = \begin{cases} 1/16, & \boldsymbol{\theta} = [1/2, 1/2] \\ 1/8, & \boldsymbol{\theta} = [1, 1] \\ 1/4, & \boldsymbol{\theta} = [1, 0] \end{cases}$$

- $\tau \sim$ Uniform$\big(-1/2, 1/2\big)$



## ECRM is not a PACC learner

**Counter-example**

$$\begin{aligned} P^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) = \frac{1}{8} \\ \text{subject to} \quad & \theta_2\, \mathbb{E}_\tau[\tau] \leq \theta_1 - 1 \Rightarrow \theta_1 \geq 1 \\ & -\theta_1\, \mathbb{E}_\tau[\tau] \leq \theta_2 - 1 \Rightarrow \theta_2 \leq 1 \end{aligned} \qquad J(\boldsymbol{\theta}) = \begin{cases} 1/16, & \boldsymbol{\theta} = [1/2, 1/2] \\ 1/8, & \boldsymbol{\theta} = [1, 1] \\ 1/4, & \boldsymbol{\theta} = [1, 0] \end{cases}$$

$$\begin{aligned} \hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) \\ \text{subject to} \quad & \theta_2 \bar{\tau}_N \leq \theta_1 - 1 \\ & -\theta_1 \bar{\tau}_N \leq 1 - \theta_2 \end{aligned} \qquad \Pr\left[|\hat{P}^\star - P^\star| \leq 1/32\right] = \Pr\left[\bar{\tau}_N = 0\right] = 0$$

- $\tau \sim$ Uniform$\big(-1/2, 1/2\big)$ $\rightarrow \bar{\tau}_N = \frac{1}{N}\sum_{n=1}^{N} \tau_n$



## ECRM is not a PACC learner

**Counter-example**

$$\begin{aligned} P^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) = \frac{1}{8} \\ \text{subject to} \quad & \theta_2\, \mathbb{E}_\tau[\tau] \leq \theta_1 - 1 \Rightarrow \theta_1 \geq 1 \\ & -\theta_1\, \mathbb{E}_\tau[\tau] \leq \theta_2 - 1 \Rightarrow \theta_2 \leq 1 \end{aligned} \qquad J(\boldsymbol{\theta}) = \begin{cases} 1/16, & \boldsymbol{\theta} = [1/2, 1/2] \\ 1/8, & \boldsymbol{\theta} = [1, 1] \\ 1/4, & \boldsymbol{\theta} = [1, 0] \end{cases}$$

$$\begin{aligned} \hat{P}_r^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) \\ \text{subject to} \quad & \theta_2 \bar{\tau}_N \leq \theta_1 - 1 + r_1 \\ & -\theta_1 \bar{\tau}_N \leq 1 - \theta_2 + r_2 \end{aligned} \qquad \Pr\left[|\hat{P}_r^\star - P^\star| \leq 1/32\right] \leq 4e^{-0.001N},$$

unless $\bar{\tau}_N \leq r_1 < \dfrac{\bar{\tau}_N + 1}{2}$ and $r_2 \geq \bar{\tau}_N$

- $\tau \sim$ Uniform$\big(-1/2, 1/2\big)$ $\rightarrow \bar{\tau}_N = \frac{1}{N}\sum_{n=1}^{N} \tau_n$



## Constrained learning challenges

$$\begin{aligned} \hat{P}^\star = \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r), y_r\big) \leq u \end{aligned} \quad \xrightarrow{\text{PAC}} \quad \begin{aligned} P^\star = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\right] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\left[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\right] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big) \leq u \text{ a.e.} \end{aligned}$$

**Challenges**

1) *Statistical*: does the solution of the constrained empirical problem generalize?

2) *Computational*: can we solve the constrained empirical problem?



## Constrained learning challenges

$$\begin{aligned} \hat{P}^\star = \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r), y_r\big) \leq u \end{aligned} \quad \xrightarrow{\text{PAC}} \quad \begin{aligned} P^\star = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\right] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\left[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\right] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big) \leq u \text{ a.e.} \end{aligned}$$

**Challenges**

1) *Statistical*: does the solution of the constrained empirical problem generalize?

2) *Computational*: can we solve the constrained empirical problem?

## Duality

PRIMAL

$\updownarrow$

DUAL



## Duality

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \text{ subject to } \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c$$

$\updownarrow$

DUAL



## Duality

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \text{ subject to } \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c$$

$\updownarrow$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda\left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$



## Duality

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \text{ subject to } \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c$$

$\updownarrow$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda\left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

- In general, $\hat{D}^\star \leq \hat{P}^\star$
- But in some cases, $\hat{D}^\star = \hat{P}^\star$ (strong duality) [e.g., convex optimization]



## Duality

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \text{ subject to } \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c$$

$\not\updownarrow$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda\left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

- In general, $\hat{D}^\star \leq \hat{P}^\star$
- ~~But in some cases, $\hat{D}^\star = \hat{P}^\star$ (strong duality) [e.g., convex optimization]~~



## An alternative path

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \quad \text{s. to} \quad \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c$$

PRIMAL $\nleftrightarrow$ DUAL

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

$\not\updownarrow$ PAC

$$P^\star = \min_{\boldsymbol{\theta}\in\Theta} \ \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \quad \text{s. to} \quad \mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \leq c$$

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



## An alternative path

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \quad \text{s. to} \quad \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c$$

PRIMAL $\nleftrightarrow$ DUAL

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

$\not\updownarrow$ PAC

$$P^\star = \min_{\boldsymbol{\theta}\in\Theta} \ \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \quad \text{s. to} \quad \mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \leq c$$

$\big\downarrow \ \mathcal{H}_\theta \subset \mathcal{H}$

$$\tilde{P}^\star = \min_{\phi\in\mathcal{H}} \ \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] \quad \text{s. to} \quad \mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] \leq c \quad \overset{?}{\longleftrightarrow} \quad \tilde{D}^\star = \max_{\lambda\geq 0} \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] - c\right)$$

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



## Non-convex variational duality

| | |
|---|---|
| Convex optimization: | Primal $\longleftrightarrow$ Dual |
| Non-convex, finite dimensional optimization: | Primal $\nleftrightarrow$ Dual |

# Non-convex variational duality

Convex optimization: Primal ⟷ Dual

Non-convex, finite dimensional optimization: Primal ⟷ Dual

**Non-convex**, **infinite** dimensional optimization: Primal ⟷ Dual

[Chamon et al., IEEE TSP'20]



# Sparse logistic regression

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \boldsymbol{\theta}^T \boldsymbol{x}_n\right)\right]$$

$$\text{s. to } \|\boldsymbol{\theta}\|_0 = \sum_{t=1}^{p} \mathbb{I}\left[\boldsymbol{\theta}_t \neq 0\right] \leq k$$

**Discrete, non-convex**

[Chen et al., JMLR'19]: NP-hard



# Sparse logistic regression

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \boldsymbol{\theta}^T \boldsymbol{x}_n\right)\right]$$

$$\text{s. to } \|\boldsymbol{\theta}\|_0 = \sum_{t=1}^{p} \mathbb{I}\left[\boldsymbol{\theta}_t \neq 0\right] \leq k$$

**Discrete, non-convex**

[Chen et al., JMLR'19]: NP-hard

$$\min_{\theta\in L_2} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \int \theta(t) x_n(t) dt\right)\right]$$

$$\text{s. to } \|\theta\|_{L_0} = \int \mathbb{I}\left[\theta(t) \neq 0\right] dt \leq \frac{k}{p}$$

**Continuous, non-convex**

[Chamon et al., IEEE TSP'20]: tractable



# Sparse logistic regression

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \boldsymbol{\theta}^T \boldsymbol{x}_n\right)\right]$$

$$\text{s. to } \|\boldsymbol{\theta}\|_0 = \sum_{t=1}^{p} \mathbb{I}\left[\boldsymbol{\theta}_t \neq 0\right] \leq k$$

**Discrete**, non-convex

[Chen et al., JMLR'19]: NP-hard

$$\min_{\theta\in L_2} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \int \theta(t) x_n(t) dt\right)\right]$$

$$\text{s. to } \|\theta\|_{L_0} = \int \mathbb{I}\left[\theta(t) \neq 0\right] dt \leq \frac{k}{p}$$

**Continuous**, non-convex

[Chamon et al., IEEE TSP'20]: tractable



# An alternative path

PRIMAL

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \quad \text{s. to } \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c$$

DUAL

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

PAC

$$P^\star = \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \leq c$$

$$\tilde{P}^\star = \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] \leq c \quad \xleftrightarrow{=} \quad \tilde{D}^\star = \max_{\lambda\geq 0} \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] - c\right)$$

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



# An alternative path

PRIMAL

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \quad \text{s. to } \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c$$

DUAL

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

PAC

$$P^\star = \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \leq c \quad \xleftrightarrow{\epsilon_0} \quad D^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] - c\right)$$

$\epsilon_0$ $\epsilon_0$

$$\tilde{P}^\star = \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] \leq c \quad \xleftrightarrow{=} \quad \tilde{D}^\star = \max_{\lambda\geq 0} \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] - c\right)$$

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



# An alternative path

PRIMAL

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \quad \text{s. to } \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c$$

DUAL

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

$O(\epsilon)$

PAC

$$P^\star = \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \leq c \quad \xleftrightarrow{\epsilon_0} \quad D^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] - c\right)$$

$\epsilon_0$ $\epsilon_0$

$$\tilde{P}^\star = \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] \leq c \quad \xleftrightarrow{=} \quad \tilde{D}^\star = \max_{\lambda\geq 0} \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] - c\right)$$

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



# Dual (near-)PACC learning

**Theorem**

Let $f$ be $\nu$-universal, i.e., for each $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and $\gamma \in [0,1]$ there exists $\boldsymbol{\theta}$ such that

$$\mathbb{E}\left[\left|\gamma f_{\boldsymbol{\theta}_1}(\boldsymbol{x}) + (1-\gamma) f_{\boldsymbol{\theta}_2}(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(\boldsymbol{x})\right|\right] \leq \nu$$

$\left[\{f_\theta\}\right.$ is a good covering of $\left.\overline{\operatorname{conv}}(\{f_\theta\})\right]$

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]

# Dual (near-)PACC learning

**Theorem**

Let $f$ be $\nu$-universal, i.e., for each $\theta_1$, $\theta_2$, and $\gamma \in [0,1]$ there exists $\theta$ such that

$$\mathbb{E}\Big[\big|\gamma f_{\theta_1}(x) + (1-\gamma) f_{\theta_2}(x) - f_{\theta}(x)\big|\Big] \le \nu$$

Then $\hat{D}^\star$ is a (near-)PACC learner, i.e., there exists a solution $\theta^\dagger$ that, with probability $1-\delta$,

Near-optimal: $\big|P^\star - \hat{D}^\star\big| \le \tilde{O}\left(\nu + \frac{1}{\sqrt{N}}\right)$

Approximately feasible: $\mathbb{E}\Big[g\big(f_{\theta^\dagger}(x), y\big)\Big] \le c + \tilde{O}\left(\frac{1}{\sqrt{N}}\right)$

(mild conditions apply)



# Dual (near-)PACC learning

**Theorem**

Let $f$ be $\nu$-universal, i.e., for each $\theta_1$, $\theta_2$, and $\gamma \in [0,1]$ there exists $\theta$ such that

$$\mathbb{E}\Big[\big|\gamma f_{\theta_1}(x) + (1-\gamma) f_{\theta_2}(x) - f_{\theta}(x)\big|\Big] \le \nu$$

Then $\hat{D}^\star$ is a (near-)PACC learner, i.e., there exists a solution $\theta^\dagger$ that, with probability $1-\delta$,

Near-optimal: $\big|P^\star - \hat{D}^\star\big| \le \tilde{O}\left(\nu + \frac{1}{\sqrt{N}}\right)$

Approximately feasible: $\mathbb{E}\Big[g\big(f_{\theta^\dagger}(x), y\big)\Big] \le c + \tilde{O}\left(\frac{1}{\sqrt{N}}\right)$

(if losses are convex) $h\big(f_{\theta^\dagger}(x), y\big) \le r$, with $\mathfrak{P}$-prob. $1 - \tilde{O}\left(\frac{1}{\sqrt{N}}\right)$

(mild conditions apply)



# Dual (near-)PACC learning

**Theorem**

Let $f$ be $\nu$-universal with VC dimension $d_{\text{VC}} < \infty$. There exists $(\theta^\dagger, \lambda^\dagger)$ achieving $\hat{D}^\star$ such that $f_{\theta^\dagger}$ is a (near-)PACC solution of (P-CSL), i.e., with probability at least $1-\delta$,

$$\big|P^\star - \hat{D}^\star\big| \le (1+\Delta)(\epsilon_0 + \epsilon)$$

$$\mathbb{E}\Big[g\big(f_{\theta^\dagger}(x), y\big)\Big] \le c + \epsilon$$

$\epsilon_0 = M\nu$ $\quad \epsilon = B\sqrt{\frac{1}{N}\left[1 + \log\left(\frac{4m(2N)^{d_{\text{VC}}}}{\delta}\right)\right]}$ $\quad \Delta = \max\left(\|\lambda^\star\|_1, \|\hat{\lambda}^\star\|_1, \|\tilde{\lambda}^\star\|_1\right)$

**Sources of error**

parametrization richness ($\nu$) sample size ($N$) requirements difficulty ($\lambda^\star$)



# Dual (near-)PACC learning

**Theorem**

Let $f$ be $\nu$-universal with VC dimension $d_{\text{VC}} < \infty$. There exists $(\theta^\dagger, \lambda^\dagger)$ achieving $\hat{D}^\star$ such that $f_{\theta^\dagger}$ is a (near-)PACC solution of (P-CSL), i.e., with probability at least $1-\delta$,

$$\big|P^\star - \hat{D}^\star\big| \le (1+\Delta)(\epsilon_0 + \epsilon)$$

$$\mathbb{E}\Big[g\big(f_{\theta^\dagger}(x), y\big)\Big] \le c + \epsilon$$

$\epsilon_0 = M\nu$ $\quad \epsilon = B\sqrt{\frac{1}{N}\left[1 + \log\left(\frac{4m(2N)^{d_{\text{VC}}}}{\delta}\right)\right]}$ $\quad \Delta = \max\left(\|\lambda^\star\|_1, \|\hat{\lambda}^\star\|_1, \|\tilde{\lambda}^\star\|_1\right)$

**Sources of error**

parametrization richness ($\nu$) sample size ($N$) requirements difficulty ($\lambda^\star$)



# Dual (near-)PACC learning

**Theorem**

Let $f$ be $\nu$-universal with VC dimension $d_{\text{VC}} < \infty$. There exists $(\theta^\dagger, \lambda^\dagger)$ achieving $\hat{D}^\star$ such that $f_{\theta^\dagger}$ is a (near-)PACC solution of (P-CSL), i.e., with probability at least $1-\delta$,

$$\big|P^\star - \hat{D}^\star\big| \le (1+\Delta)(\epsilon_0 + \epsilon)$$

$$\mathbb{E}\Big[g\big(f_{\theta^\dagger}(x), y\big)\Big] \le c + \epsilon$$

$\epsilon_0 = M\nu$ $\quad \epsilon = B\sqrt{\frac{1}{N}\left[1 + \log\left(\frac{4m(2N)^{d_{\text{VC}}}}{\delta}\right)\right]}$ $\quad \Delta = \max\left(\|\lambda^\star\|_1, \|\hat{\lambda}^\star\|_1, \|\tilde{\lambda}^\star\|_1\right)$

**Sources of error**

parametrization richness ($\nu$) sample size ($N$) requirements difficulty ($\lambda^\star$)



# Dual (near-)PACC learning

**Theorem**

Let $f$ be $\nu$-universal with VC dimension $d_{\text{VC}} < \infty$. There exists $(\theta^\dagger, \lambda^\dagger)$ achieving $\hat{D}^\star$ such that $f_{\theta^\dagger}$ is a (near-)PACC solution of (P-CSL), i.e., with probability at least $1-\delta$,

$$\big|P^\star - \hat{D}^\star\big| \le (1+\Delta)(\epsilon_0 + \epsilon)$$

$$\mathbb{E}\Big[g\big(f_{\theta^\dagger}(x), y\big)\Big] \le c + \epsilon$$

$\epsilon_0 = M\nu$ $\quad \epsilon = B\sqrt{\frac{1}{N}\left[1 + \log\left(\frac{4m(2N)^{d_{\text{VC}}}}{\delta}\right)\right]}$ $\quad \Delta = \max\left(\|\lambda^\star\|_1, \|\hat{\lambda}^\star\|_1, \|\tilde{\lambda}^\star\|_1\right)$

**Sources of error**

parametrization richness ($\nu$) sample size ($N$) requirements difficulty ($\lambda^\star$)



# Dual (near-)PACC learning

**Theorem**

Let $f$ be $\nu$-universal with VC dimension $d_{\text{VC}} < \infty$. There exists $(\theta^\dagger, \lambda^\dagger)$ achieving $\hat{D}^\star$ such that $f_{\theta^\dagger}$ is a (near-)PACC solution of (P-CSL), i.e., with probability at least $1-\delta$,

$$\big|P^\star - \hat{D}^\star\big| \le (1+\Delta)(\epsilon_0 + \epsilon)$$

$$\mathbb{E}\Big[g\big(f_{\theta^\dagger}(x), y\big)\Big] \le c + \epsilon$$

$\epsilon_0 = M\nu$ $\quad \epsilon = B\sqrt{\frac{1}{N}\left[1 + \log\left(\frac{4m(2N)^{d_{\text{VC}}}}{\delta}\right)\right]}$ $\quad \Delta = \max\left(\|\lambda^\star\|_1, \|\hat{\lambda}^\star\|_1, \|\tilde{\lambda}^\star\|_1\right)$

**Sources of error**

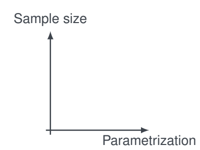
parametrization richness ($\nu$) sample size ($N$) requirements difficulty ($\lambda^\star$)



# Dual learning trade-offs

- Unconstrained learning

  parametrization $\times$ sample size



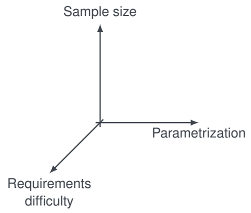
# Dual learning trade-offs

- Unconstrained learning

  parametrization × sample size

- Constrained learning

  parametrization × sample size × requirements




[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



# When is constrained learning possible?

Corollary

$f_\theta$ is PAC learnable $\approx^*$ $f_\theta$ is PACC learnable

Constrained learning is **essentially as hard as** unconstrained learning

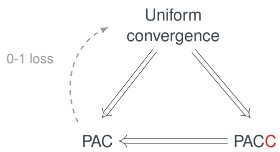
[mild conditions apply]

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



# When is constrained learning possible?

Corollary




[mild conditions apply]

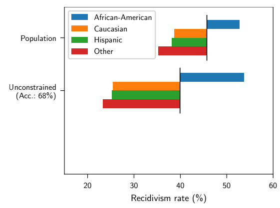
[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]



# Fairness

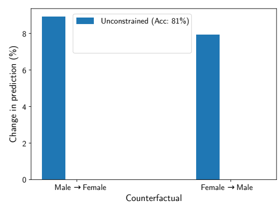
Problem

Predict whether an individual will recidivate




Problem

Predict whether an individual makes > $50k



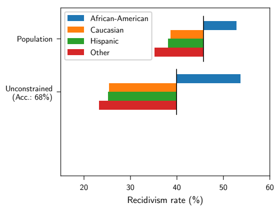

*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.



# Fairness

Problem

Predict whether an individual will recidivate




Problem

Predict whether an individual makes > $50k




*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.



# Fairness: "Equality" of odds

Problem

Predict whether an individual will recidivate at the same rate across races

$$\min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n), y_n\big)$$

$$\text{subject to} \ \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_\theta(x_n) = 1 \mid \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_\theta(x_n) = 1\right] + c,$$

for Race $\in$ {African-American, Caucasian, Hispanic, Other}

*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.
[Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23]



# Fairness: "Equality" of odds

Problem

Predict whether an individual will recidivate at the same rate across races

$$\min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n), y_n\big)$$

$$\text{subject to} \ \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_\theta(x_n) = 1 \mid \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_\theta(x_n) = 1\right] + c,$$

for Race $\in$ {African-American, Caucasian, Hispanic, Other}

*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.
[Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23]



# Fairness: "Equality" of odds

Problem

Predict whether an individual will recidivate at the same rate across races

$$\min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n), y_n\big)$$

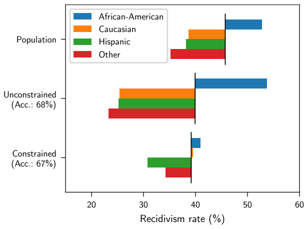
$$\text{subject to} \ \frac{1}{N}\sum_{n=1}^{N} \sigma\big(f_\theta(x_n) - 0.5\big)\, \mathbb{I}\left[x_n \in \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \sigma\big(f_\theta(x_n) - 0.5\big) + c,$$

for Race $\in$ {African-American, Caucasian, Hispanic, Other}

*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.
[Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23]

# Fairness: “Equality” of odds

Problem
Predict whether an individual will recidivate at the same rate across races

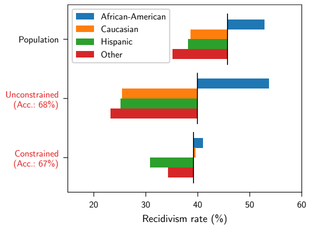



*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon et al., IEEE TIT'23]



# Fairness: “Equality” of odds

Problem
Predict whether an individual will recidivate at the same rate across races




*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon et al., IEEE TIT'23]



# Fairness: “Equality” of odds

*Prediction*

| | True label | Unconstrained 0 | Unconstrained 1 | Constrained 0 | Constrained 1 |
|---|---|---|---|---|---|
| African-American | 0 | 31% | 16% | 36% | 11% |
| | 1 | 16% | 37% | 23% | 30% |
| Caucasian | 0 | 52% | 9% | 44% | 17% |
| | 1 | 23% | 16% | 16% | 23% |

*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon et al., IEEE TIT'23]



# Fairness: “Equality” of odds

*Prediction*

| | True label | Unconstrained 0 | Unconstrained 1 | Constrained 0 | Constrained 1 |
|---|---|---|---|---|---|
| African-American | 0 | 31% | 16% | 36% | 11% |
| | 1 | 16% | 37% | 23% | 30% |
| Caucasian | 0 | 52% | 9% | 44% | 17% |
| | 1 | 23% | 16% | 16% | 23% |

*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon et al., IEEE TIT'23]



# Fairness: “Equality” of odds

*Prediction*

| | True label | Unconstrained 0 | Unconstrained 1 | Constrained 0 | Constrained 1 |
|---|---|---|---|---|---|
| African-American | 0 | 31% | 16% | 36% | 11% |
| | 1 | 16% | 37% | 23% | 30% |
| Caucasian | 0 | 52% | 9% | 44% | 17% |
| | 1 | 23% | 16% | 16% | 23% |

*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon et al., IEEE TIT'23]

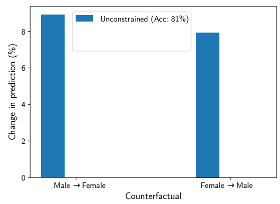


# Fairness

Problem
Predict whether an individual will recidivate




Problem
Predict whether an individual makes > $50k




*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.



# Counterfactual fairness

Problem
Predict whether an individual makes > $50k while being invariant to gender

$$\min_{\theta} \; \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)$$

$$\text{subject to} \quad D_{\text{KL}}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) \| f_{\boldsymbol{\theta}}(\rho \boldsymbol{x}_n)\right) \le c, \quad \text{for all } n$$

($\rho$ : Male ↔ Female)

[Chamon and Ribeiro, NeurIPS'20]



# Counterfactual fairness

Problem
Predict whether an individual makes > $50k while being invariant to gender

$$\min_{\theta} \; \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)$$

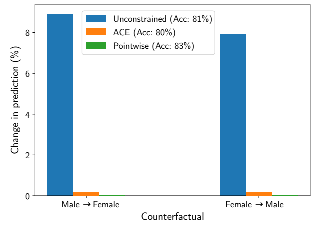
$$\text{subject to} \quad \frac{1}{N}\sum_{n=1}^{N} D_{\text{KL}}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) \| f_{\boldsymbol{\theta}}(\rho \boldsymbol{x}_n)\right) \le c \quad \text{for all } n$$

($\rho$ : Male ↔ Female)

[Chamon and Ribeiro, NeurIPS'20]

# Counterfactual fairness

**Problem**
Predict whether an individual makes > $50k while being invariant to gender




[Chamon and Ribeiro, NeurIPS'20]



# Counterfactual fairness

**Problem**
Predict whether an individual makes > $50k while being invariant to gender

$$\begin{aligned} &\min_{\boldsymbol{\theta}} && \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ &\text{subject to} && \mathrm{D}_{\mathrm{KL}}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) \| f_{\boldsymbol{\theta}}(\rho \boldsymbol{x}_n)\right) \leq c, \quad \text{for all } n \end{aligned}$$

($\rho$ : Male ↔ Female)

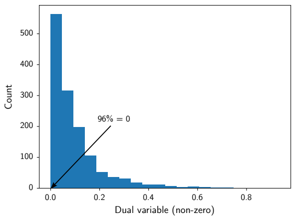
$$\max_{\lambda_n \geq 0} \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \sum_{n=1}^{N} \lambda_n \Big[\mathrm{D}_{\mathrm{KL}}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) \| f_{\boldsymbol{\theta}}(\rho \boldsymbol{x}_n)\right) - c\Big]$$

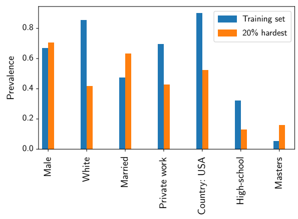
[Chamon and Ribeiro, NeurIPS'20]



# Counterfactual fairness

**Problem**
Predict whether an individual makes > $50k while being invariant to gender




[Chamon and Ribeiro, NeurIPS'20]



# Agenda

Constrained learning theory

Constrained learning algorithms

Resilient constrained learning



# Constrained optimization methods

$$\begin{aligned} \hat{P}^\star = &\min_{\boldsymbol{\theta}} && \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ &\text{subject to} && \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & && h\big(f_{\boldsymbol{\theta}}(x_r), y_r\big) \leq u \end{aligned}$$



# Constrained optimization methods

$$\begin{aligned} \hat{P}^\star = &\min_{\boldsymbol{\theta}} && \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ &\text{subject to} && \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & && h\big(f_{\boldsymbol{\theta}}(x_r), y_r\big) \leq u \end{aligned}$$

- Feasible update methods
  e.g., conditional gradients (Frank-Wolfe)
- Interior point methods
  e.g., barriers, projection, polyhedral approx.



# Constrained optimization methods

$$\begin{aligned} \hat{P}^\star = &\min_{\boldsymbol{\theta}} && \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ &\text{subject to} && \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & && h\big(f_{\boldsymbol{\theta}}(x_r), y_r\big) \leq u \end{aligned}$$

- Feasible update methods
  e.g., conditional gradients (Frank-Wolfe)
  - ✗ Tractability [non-convex constraints]
  - ✓ Feasible candidate solution
- Interior point methods
  e.g., barriers, projection, polyhedral approx.
  - ✗ Tractability [non-convex constraints]
  - ✓ Feasible candidate solution



# Constrained optimization methods

$$\begin{aligned} \hat{P}^\star = &\min_{\boldsymbol{\theta}} && \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ &\text{subject to} && \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & && h\big(f_{\boldsymbol{\theta}}(x_r), y_r\big) \leq u \end{aligned}$$

- Feasible update methods
  e.g., conditional gradients (Frank-Wolfe)
  - ✗ Tractability [non-convex constraints]
  - ✓ Feasible candidate solution
- Interior point methods
  e.g., barriers, projection, polyhedral approx.
  - ✗ Tractability [non-convex constraints]
  - ✓ Feasible candidate solution
- Duality
  e.g., (augmented) Lagrangian
  - ✓ Tractability
  - ✓ (near-)feasible solution [small duality gap]

## Dual learning algorithm

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{N} \sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N} \sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$



## Dual learning algorithm

- Minimize the primal ($\equiv$ **ERM**)

$$\boldsymbol{\theta}^\dagger \in \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{N} \sum_{n=1}^{N} \left[\ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right)\right]$$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{N} \sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N} \sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$



## Dual learning algorithm

- Minimize the primal ($\equiv$ **ERM**)

$$\boldsymbol{\theta}^+ \approx \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)\Big], \quad n = 1, 2, \dots$$

[Haeffele et al., CVPR'17; Ge et al., ICLR'18; Mei et al., PNAS'18; Kawaguchi et al., AISTATS'20…]

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{N} \sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N} \sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$



## Dual learning algorithm

- Minimize the primal ($\equiv$ **ERM**)

$$\boldsymbol{\theta}^+ \approx \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)\Big], \quad n = 1, 2, \dots$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}^+}(\boldsymbol{x}_m), y_m\big) - c\right)\right]_+$$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{N} \sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N} \sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$



## A (near-)PACC learner

**Theorem**

Suppose $\boldsymbol{\theta}^\dagger$ is a $\rho$-approximate solution of the regularized ERM:

$$\boldsymbol{\theta}^\dagger \approx \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{N} \sum_{n=1}^{N} \Big(\ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right)\Big).$$

Then, after $T = \left\lceil \frac{\|\boldsymbol{\lambda}^\star\|^2}{2\eta M \nu} \right\rceil + 1$ dual iterations with step size $\eta \leq \frac{2\epsilon}{m B^2}$,

the iterates $\left(\boldsymbol{\theta}^{(T)}, \boldsymbol{\lambda}^{(T)}\right)$ are such that

$$\left|P^\star - L\Big(\boldsymbol{\theta}^{(T)}, \boldsymbol{\lambda}^{(T)}\Big)\right| \leq (2 + \Delta)(\epsilon_0 + \epsilon) + \rho$$

with probability $1 - \delta$ over sample sets.

[Chamon et al., IEEE TIT'23]



## In practice…

- Minimize the primal ($\equiv$ **ERM**)

$$\boldsymbol{\theta}^+ \approx \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)\Big], \quad n = 1, 2, \dots$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}^+}(\boldsymbol{x}_m), y_m\big) - c\right)\right]_+$$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{N} \sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N} \sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$



## In practice…

- Minimize the primal ($\equiv$ **ERM**)

$$\boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)\Big], \quad n = 1, 2, \dots, N$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}^+}(\boldsymbol{x}_m), y_m\big) - c\right)\right]_+$$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{N} \sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N} \sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$



## In practice…

Initialize: $\boldsymbol{\theta}_0, \lambda_0$

**for** $t = 1, \dots, T$

  $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$

  **for** $n = 1, \dots, N$

    $\boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n - \eta_\theta \nabla_{\boldsymbol{\beta}} \Big[\ell\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) + \lambda_{t-1} g\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big)\Big]$

  **end**

  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$

SGD

  $\lambda_t = \left[\lambda_{t-1} + \eta_\lambda \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_n\big) - c\right)\right]_+$

Dual update

**end**

Output: $\boldsymbol{\theta}_T, \lambda_T$

PyTorch

https://github.com/lfochamon/csl

# In practice...

Initialize: $\boldsymbol{\theta}_0$, $\lambda_0$
**for** $t = 1, \dots, T$
  $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$
  **for** $n = 1, \dots, N$
    $\boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n - \eta_\theta \nabla_{\boldsymbol{\beta}} \left[ \ell\left(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\right) + \lambda_{t-1} g\left(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\right) \right]$
  **end**
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[ \lambda_{t-1} + \eta_\lambda \left( \frac{1}{N} \sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_m\right) - c \right) \right]_+$
**end**
Output: $\boldsymbol{\theta}_T$, $\lambda_T$

Use adaptive method (e.g., ADAM)

PyTorch
https://github.com/lfochamon/csl



# In practice...

Initialize: $\boldsymbol{\theta}_0$, $\lambda_0$
**for** $t = 1, \dots, T$
  $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$
  **for** $n = 1, \dots, N$
    $\boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n - \eta_\theta \nabla_{\boldsymbol{\beta}} \left[ \ell\left(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\right) + \lambda_{t-1} g\left(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\right) \right]$
  **end**
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[ \lambda_{t-1} + \eta_\lambda \left( \frac{1}{N} \sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_m\right) - c \right) \right]_+$
**end**
Output: $\boldsymbol{\theta}_T$, $\lambda_T$

Use adaptive method (e.g., ADAM)

Use different time-scales ($\eta_\lambda = 0.1\eta_\theta$)

PyTorch
https://github.com/lfochamon/csl

60

# In practice...

Initialize: $\boldsymbol{\theta}_0$, $\lambda_0$
**for** $t = 1, \dots, T$
  $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$
  **for** $n = 1, \dots, N$
    $\boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n - \eta_\theta \nabla_{\boldsymbol{\beta}} \left[ \ell\left(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\right) + \lambda_{t-1} g\left(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\right) \right]$
  **end**
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[ \lambda_{t-1} + \eta_\lambda \left( \frac{1}{N} \sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_m\right) - c \right) \right]_+$
**end**
Output: $\boldsymbol{\theta}_T$, $\lambda_T$

Check slack:
- feasibility: $s_t \leq 0$
- "duality gap": $\lambda_t s_t$

$$s_t = \frac{1}{N} \sum_{n=1}^{N} g\left(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_n), y_n\right) - c$$

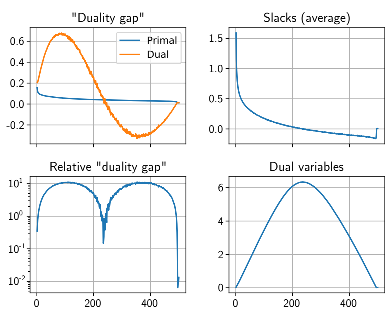
Use adaptive method (e.g., ADAM)
Use different time-scales ($\eta_\lambda = 0.1\eta_\theta$)

PyTorch
https://github.com/lfochamon/csl

60

# In practice...

Initialize: $\boldsymbol{\theta}_0$, $\lambda_0$
**for** $t = 1, \dots, T$
  $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$
  **for** $n = 1, \dots,$
    $\boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n$
  **end**
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[ \lambda_{t-1} + \right.$
**end**
Output: $\boldsymbol{\theta}_T$, $\lambda_T$




ethod (e.g., ADAM)
ne-scales ($\eta_\lambda = 0.1\eta_\theta$)

rch
https://github.com/lfochamon/csl

60

# Penalty-based vs. dual learning

**Penalty-based learning**

$$\boldsymbol{\theta}^\dagger \in \operatorname*{argmin}_{\boldsymbol{\theta}} \ \text{Loss}(\boldsymbol{\theta}) + \lambda \cdot \text{Penalty}(\boldsymbol{\theta})$$

- Parameter: $\lambda$ (data-dependent)
- Generalizes with respect to Loss + $\lambda$Penalty

**Dual learning**

$$\boldsymbol{\theta}^\dagger \in \operatorname*{argmin}_{\boldsymbol{\theta}} \ \text{Loss}(\boldsymbol{\theta}) + \lambda \cdot \text{Penalty}(\boldsymbol{\theta})$$

$$\lambda^+ = \left[ \lambda + \eta \left( \text{Penalty}(\boldsymbol{\theta}^\dagger) - c \right) \right]_+$$

- Parameter: $c$ (requirement-dependent)
- Generalizes with respect to Loss and Penalty $\leq c$



# Agenda

Constrained learning theory

Constrained learning algorithms

Resilient constrained learning



# Heterogeneous federated learning

Problem
Learn a common model using data using data distributed among $K$ clients

$$\min_{\boldsymbol{\theta}} \ \frac{1}{K} \sum_{k=1}^{K} \text{Loss}_k(f_{\boldsymbol{\theta}})$$

$$\text{subject to} \quad \text{Loss}_k(f_{\boldsymbol{\theta}}) \leq \frac{1}{K} \sum_{k=1}^{K} \text{Loss}_k(f_{\boldsymbol{\theta}}) + c, \quad k = 1, \dots, K$$

Clients


- $k$-th client loss: $\text{Loss}_k(\phi) = \frac{1}{N_k} \sum_{n_k=1}^{N_k} \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{n_k}), y_{n_k}\right)$



# Heterogeneous federated learning

Problem
Learn a common model using data using data distributed among $K$ clients

$$\min_{\boldsymbol{\theta}} \ \frac{1}{K} \sum_{k=1}^{K} \text{Loss}_k(f_{\boldsymbol{\theta}})$$

$$\text{subject to} \quad \text{Loss}_k(f_{\boldsymbol{\theta}}) \leq \frac{1}{K} \sum_{k=1}^{K} \text{Loss}_k(f_{\boldsymbol{\theta}}) + c, \quad k = 1, \dots, K$$

Clients


- $k$-th client loss: $\text{Loss}_k(\phi) = \frac{1}{N_k} \sum_{n_k=1}^{N_k} \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{n_k}), y_{n_k}\right)$

# Heterogeneous federated learning

Problem
Learn a common model using data using data distributed among $K$ clients

$$\begin{aligned}\min_{\boldsymbol{\theta}} \quad & \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_{\boldsymbol{\theta}}) \\ \text{subject to} \quad & \text{Loss}_k(f_{\boldsymbol{\theta}}) \leq \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_{\boldsymbol{\theta}}) + c_k, \\ & k = 1, \dots, K\end{aligned}$$


- $k$-th client loss: $\text{Loss}_k(\phi) = \frac{1}{N_k}\sum_{n_k=1}^{N_k} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{n_k}), y_{n_k}\big)$



# Resilient constrained learning

Definition (Resilience)
(ecology) ability of an ecosystem to adapt its function to accommodate operating conditions



# Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties



# Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties

$$\begin{aligned}P^\star = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}_i}\Big[g_i\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big)\Big] \leq c_i\end{aligned}$$



# Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties

$$\begin{aligned}P^\star(\boldsymbol{r}) = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}_i}\Big[g_i\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big)\Big] \leq c_i + r_i\end{aligned}$$



# Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties

$$\begin{aligned}P^\star(\boldsymbol{r}) = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}_i}\Big[g_i\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big)\Big] \leq c_i + r_i\end{aligned}$$

- Larger relaxations $\boldsymbol{r}$ decrease the objective $P^\star(\boldsymbol{r})$ (benefit), but increase specification violation $c_i + r_i$ (cost)



# Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties

$$\begin{aligned}P^\star(\boldsymbol{r}) = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}_i}\Big[g_i\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big)\Big] \leq c_i + r_i\end{aligned}$$

- Larger relaxations $\boldsymbol{r}$ decrease the objective $P^\star(\boldsymbol{r})$ (benefit), but increase specification violation $c_i + r_i$ (cost)
- Resilience is a compromise!



# Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties

$$\begin{aligned}P^\star(\boldsymbol{r}) = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}_i}\Big[g_i\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big)\Big] \leq c_i + r_i\end{aligned}$$

- Larger relaxations $\boldsymbol{r}$ decrease the objective $P^\star(\boldsymbol{r})$ (benefit), but increase specification violation $c_i + r_i$ (cost) $\Rightarrow h(\boldsymbol{r})$
- Resilience is a compromise!

## Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) \quad \leftarrow (\partial\text{: subdifferential})$$

**In words**: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing

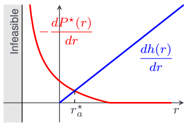


## Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) \quad \leftarrow (\partial\text{: subdifferential})$$

**In words**: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing



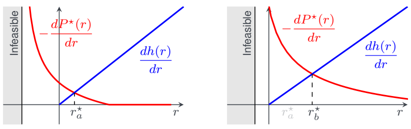



## Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) \quad \leftarrow (\partial\text{: subdifferential})$$

**In words**: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing



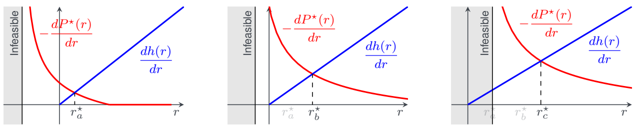



## Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) \quad \leftarrow (\partial\text{: subdifferential})$$

**In words**: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing






## Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) = \boldsymbol{\lambda}^\star(\boldsymbol{r}^\star)$$

**In words**: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing

- After relaxing, $\boldsymbol{\lambda}^\star(\boldsymbol{r}^\star)$ is *smaller* than $\boldsymbol{\lambda}^\star(\mathbf{0})$
  $\Rightarrow$ Resilient constrained learning "generalizes better" (lower sample complexity)



## Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) = \boldsymbol{\lambda}^\star(\boldsymbol{r}^\star)$$

**In words**: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing

- After relaxing, $\boldsymbol{\lambda}^\star(\boldsymbol{r}^\star)$ is *smaller* than $\boldsymbol{\lambda}^\star(\mathbf{0})$
  $\Rightarrow$ Resilient constrained learning "generalizes better" (lower sample complexity)
- The resilient equilibrium *exists and is unique* (because $h$ is strictly convex)



## Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

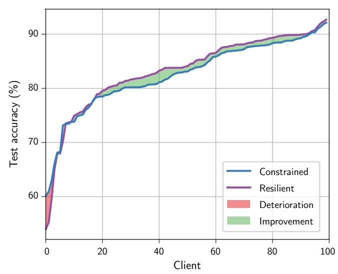
$$\begin{aligned} P^\star(\boldsymbol{r}^\star) = \min_{\boldsymbol{\theta}, \boldsymbol{r}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] + h(\boldsymbol{r}) \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}_i}\Big[g_i\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big)\Big] \le c_i + r_i \end{aligned}$$

**In words**: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing

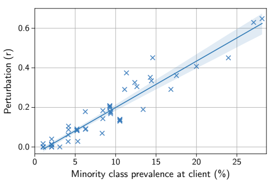
- After relaxing, $\boldsymbol{\lambda}^\star(\boldsymbol{r}^\star)$ is *smaller* than $\boldsymbol{\lambda}^\star(\mathbf{0})$
  $\Rightarrow$ Resilient constrained learning "generalizes better" (lower sample complexity)
- The resilient equilibrium *exists and is unique* (because $h$ is strictly convex)

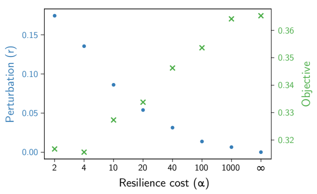


## Heterogeneous federated learning



# Heterogeneous federated learning

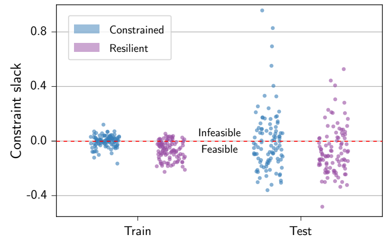



[Hounie et al., NeurIPS'23]



# Heterogeneous federated learning




[Hounie et al., NeurIPS'23]



# Summary

- **Constrained learning is the a tool to learn under requirements**
- **Constrained learning is hard...**
- **...but possible. How?**



# Summary

- **Constrained learning is the a tool to learn under requirements**
  Constrained learning imposes generalizable requirements organically during training, e.g., fairness [Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23], heterogeneity [Shen et al., ICRL'22]...
- **Constrained learning is hard...**
- **...but possible. How?**

72

# Summary

- **Constrained learning is the a tool to learn under requirements**
  Constrained learning imposes generalizable requirements organically during training, e.g., fairness [Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23], heterogeneity [Shen et al., ICRL'22]...
- **Constrained learning is hard...**
  Constrained, non-convex, statistical optimization problem
- **...but possible. How?**

72

# Summary

- **Constrained learning is the a tool to learn under requirements**
  Constrained learning imposes generalizable requirements organically during training, e.g., fairness [Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23], heterogeneity [Shen et al., ICRL'22]...
- **Constrained learning is hard...**
  Constrained, non-convex, statistical optimization problem
- **...but possible. How?**
  We can learn under requirements (essentially) whenever we can learn at all by solving *(penalized) ERM problems*. Resilient learning can then be used to adapt the requirements to the task difficulty [Hounie et al., NeurIPS'23]



# Robustness constraints

# Agenda

Adversarially robust learning

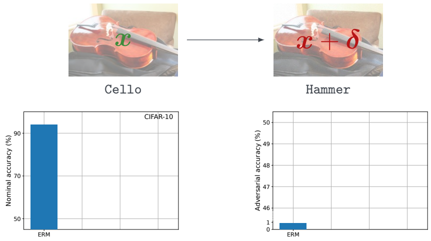
Semi-infinite learning

Probabilistic robustness

# Robust learning

## Problem

Learn an image classifier that is robust to input perturbations






# Adversarial training

## Problem

Learn an image classifier that is robust to input perturbations

- Adversarial training [Szegedy et al., ICLR'14; Goodfellow et al., ICLR'15; Madry et al., ICLR'18; ...]

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(x_n), y_n\big) \longrightarrow \min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\delta\|_\infty \le \epsilon} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$

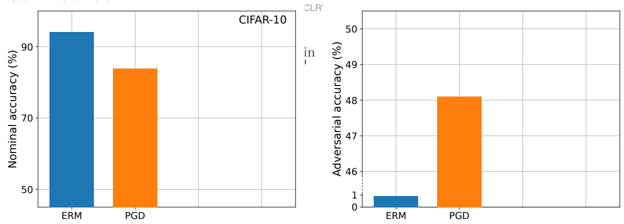




# Adversarial training

## Problem

Learn an image classifier that is robust to input perturbations




[Robey et al., NeurIPS'21]



# Adversarial training

## Problem

Learn an image classifier that is robust to input perturbations

- Adversarial training [Szegedy et al., ICLR'14; Goodfellow et al., ICLR'15; Madry et al., ICLR'18; ...]

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(x_n), y_n\big) \qquad \min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\delta\|_\infty \le \epsilon} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$

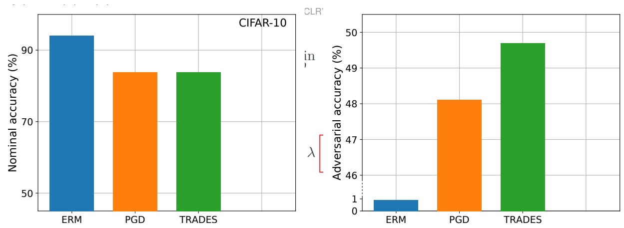
$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(x_n), y_n\big) + \lambda \left[\max_{\|\delta\|_\infty \le \epsilon} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$



# Adversarial training

## Problem

Learn an image classifier that is robust to input perturbations




[Zhang et al., ICML'19]



# Constrained learning for robustness

## Problem

Learn an image classifier that is robust to input perturbations

$$\begin{aligned} &\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(x_n), y_n\big) \\ &\text{subject to} \quad \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\delta\|_\infty \le \epsilon} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right] \le c \end{aligned}$$

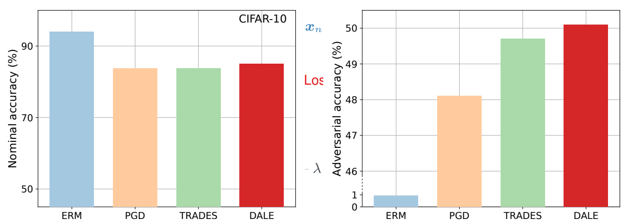
[Chamon and Ribeiro, NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23]



# Constrained learning for robustness

## Problem

Learn an image classifier that is robust to input perturbations




[Chamon and Ribeiro, NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23]



# Constrained learning for robustness

## Problem

Learn an image classifier th


[Chamon and Ribeiro, NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23]

# Penalty-based vs. dual learning

**Penalty-based learning**

$$\theta^\dagger \in \underset{\theta}{\operatorname{argmin}} \ \text{Loss}(\theta) + \lambda \cdot \text{Penalty}(\theta)$$

- Parameter: $\lambda$ (data-dependent)
- Generalizes with respect to Loss + $\lambda$Penalty

**Dual learning**

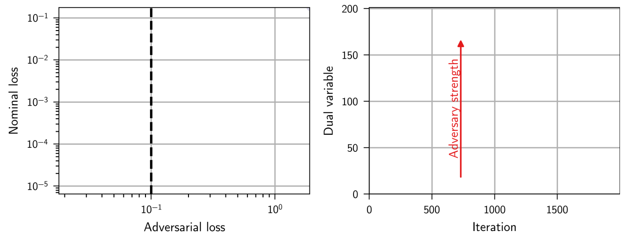
$$\theta^\dagger \in \underset{\theta}{\operatorname{argmin}} \ \text{Loss}(\theta) + \lambda \cdot \text{Penalty}(\theta)$$

$$\lambda^+ = \left[\lambda + \eta\left(\text{Penalty}(\theta^\dagger) - c\right)\right]_+$$

- Parameter: $c$ (requirement-dependent)
- Generalizes with respect to Loss and Penalty $\leq c$



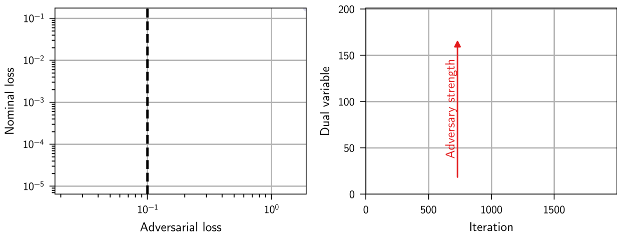
# Constrained learning for robustness




[Chamon et al., IEEE TIT'23]



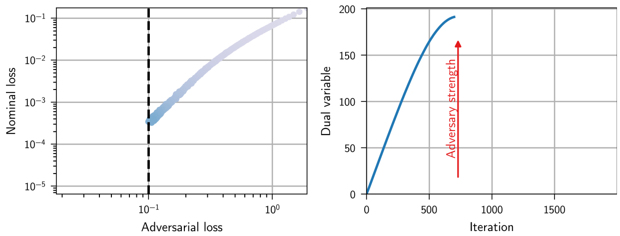
# Constrained learning for robustness




[Chamon et al., IEEE TIT'23]



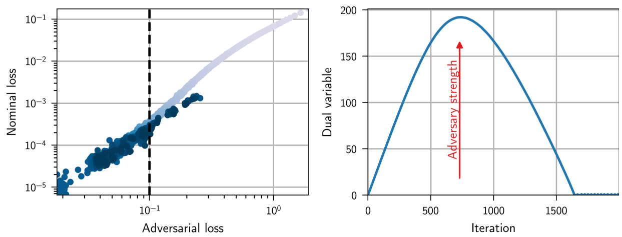
# Constrained learning for robustness




[Chamon et al., IEEE TIT'23]



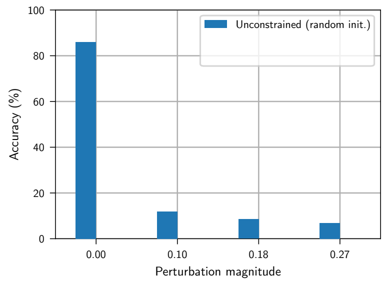
# Constrained learning for robustness




Empirical observations: [Zhang et al., ICML'20; Sitawarin, arXiv'20]



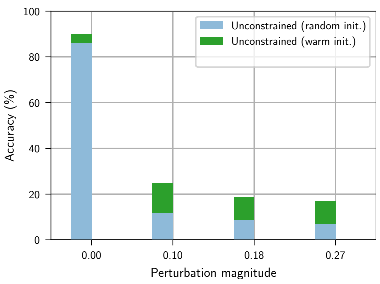
# Constrained learning for robustness




[Chamon et al., IEEE TIT'23]



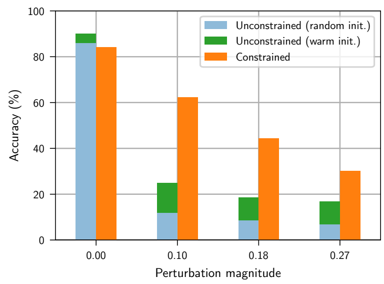
# Constrained learning for robustness




[Chamon et al., IEEE TIT'23]



# Constrained learning for robustness




[Chamon et al., IEEE TIT'23]

# Constrained learning for robustness

**Problem**
Learn an image classifier that is robust to input perturbations

$$\max_{\lambda \geq 0} \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda \left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right]$$

- ✓ Balancing nominal accuracy and robustness ⇒ Dual constrained learning



# Constrained learning for robustness

**Problem**
Learn an image classifier that is robust to input perturbations

$$\max_{\lambda \geq 0} \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda \left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right]$$

- ✓ Balancing nominal accuracy and robustness ⇒ Dual constrained learning
- ✗ Computing the worst-case perturbations



# Adversarial training

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right]$$

- "PGD" [Mądry et al., ICLR'18]

$\boldsymbol{\delta}^1 \leftarrow \boldsymbol{\delta}_{t-1}$
**for** $k = 1, \dots, K$
$\quad \boldsymbol{\delta}^{k+1} \leftarrow \text{proj}_{\Delta}\left[\boldsymbol{\delta}^k + \eta \,\text{sign}\left(\nabla_{\boldsymbol{\delta}} \text{Loss}\big(f_{\boldsymbol{\theta}^k}(\boldsymbol{x} + \boldsymbol{\delta}^k), y\big)\right)\right]$
**end**
$\boldsymbol{\delta}_t \leftarrow \boldsymbol{\delta}^{K+1}$
$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}_t), y\big)$



# Adversarial training

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right]$$

- "PGD" [Mądry et al., ICLR'18]

$\boldsymbol{\delta}^1 \leftarrow \boldsymbol{\delta}_{t-1}$
**for** $k = 1, \dots, K$
$\quad \boldsymbol{\delta}^{k+1} \leftarrow \text{proj}_{\Delta}\left[\boldsymbol{\delta}^k + \eta \,\text{sign}\left(\nabla_{\boldsymbol{\delta}} \text{Loss}\big(f_{\boldsymbol{\theta}^k}(\boldsymbol{x} + \boldsymbol{\delta}^k), y\big)\right)\right]$
**end**
$\boldsymbol{\delta}_t \leftarrow \boldsymbol{\delta}^{K+1}$
$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}_t), y\big)$

- Random initialization
- Restarts
- Pruning
- Adaptive step size

[Dhillon et al., ICLR'18; Carmon et al., NeurIPS'19; Wu et al., NeurIPS'20; Cheng et al., IJCAI'22]



# Constrained learning for robustness

**Problem**
Learn an image classifier that is robust to input perturbations

$$\max_{\lambda \geq 0} \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda \left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right]$$

- ✓ Balancing nominal accuracy and robustness ⇒ Dual constrained learning
- ✗ Computing the worst-case perturbations
  - ~~gradient ascent → non-convex, underparametrized~~



# Agenda

Adversarially robust learning

Semi-infinite learning

Probabilistic robustness



# Semi-infinite constrained learning

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right]$$



# Semi-infinite constrained learning

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \left[t(\boldsymbol{x}_n, y_n)\right] \\ \text{subject to} \quad & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) \leq t(\boldsymbol{x}_n, y_n), \\ & \text{for all } (\boldsymbol{x}_n, y_n) \text{ and } \boldsymbol{\delta} \in \Delta \end{aligned}$$

- Epigraph formulation:

$$\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big) \leq t \iff \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big) \leq t, \text{ for all } \|\boldsymbol{\delta}\|_\infty \leq \epsilon$$

## Semi-infinite constrained learning

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[t(\boldsymbol{x}_n, y_n)\right]$$

$$\begin{aligned}\text{subject to}\quad & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_0), y_n\big) \le t(\boldsymbol{x}_n, y_n)\\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\sqrt{2}}), y_n\big) \le t(\boldsymbol{x}_n, y_n)\\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_e), y_n\big) \le t(\boldsymbol{x}_n, y_n)\\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_\pi), y_n\big) \le t(\boldsymbol{x}_n, y_n)\\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_4), y_n\big) \le t(\boldsymbol{x}_n, y_n)\\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^2}), y_n\big) \le t(\boldsymbol{x}_n, y_n)\\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\pi^e}), y_n\big) \le t(\boldsymbol{x}_n, y_n)\\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^\pi}), y_n\big) \le t(\boldsymbol{x}_n, y_n)\\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{25}), y_n\big) \le t(\boldsymbol{x}_n, y_n)\end{aligned}$$

- Epigraph formulation:
  $$\max_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big) \le t \iff \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big) \le t, \text{ for all } \|\boldsymbol{\delta}\|_\infty \le \epsilon$$
- Semi-infinite program



## Duality

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]$$

$$\Updownarrow =$$

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[t(\boldsymbol{x}_n, y_n)\right] \text{ s. to } \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big) \le t(\boldsymbol{x}_n, y_n), \forall(\boldsymbol{x}_n, y_n, \boldsymbol{\delta})$$

$$\Updownarrow =$$

$$\min_{\boldsymbol{\theta}} \sup_{\mu\in\mathcal{P}} \underbrace{\frac{1}{N}\sum_{n=1}^{N}\int_\Delta \mu_n(\boldsymbol{\delta})\, \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big) d\boldsymbol{\delta}}_{L(\theta,\mu_n)}$$



## Duality

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]$$

$$\Updownarrow =$$

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[t(\boldsymbol{x}_n, y_n)\right] \text{ s. to } \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big) \le t(\boldsymbol{x}_n, y_n), \forall(\boldsymbol{x}_n, y_n, \boldsymbol{\delta})$$

$$\Updownarrow =$$

$$\min_{\boldsymbol{\theta}} \sup_{\mu\in\mathcal{P}} \underbrace{\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{\delta}\sim\mu(\cdot|\boldsymbol{x}_n,y_n)}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]}_{L(\theta,\mu_n)}$$



## From optimization to sampling

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]$$

$$\Updownarrow \approx$$

$$\min_{\boldsymbol{\theta}} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{\delta}\sim\mu_\gamma(\cdot|\boldsymbol{x}_n,y_n)}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]}_{L(\theta,\mu)}$$

**Proposition**

For all $\epsilon > 0$, there exists $\gamma(\boldsymbol{x}, y) < \max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big)$ s.t. $L(\boldsymbol{\theta}, \mu_\gamma) \ge \sup_{\mu\in\mathcal{P}^2} L(\boldsymbol{\theta}, \mu) - \xi$ for

$$\mu_\gamma(\boldsymbol{\delta}|\boldsymbol{x}, y) \propto \Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big) - \gamma(\boldsymbol{x}, y)\Big]_+$$



## From optimization to sampling

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]$$

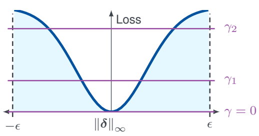
$$\Updownarrow \approx$$

$$\min_{\boldsymbol{\theta}} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{\delta}\sim\mu_\gamma(\cdot|\boldsymbol{x}_n,y_n)}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]}_{L(\theta,\mu)}$$

**Proposition**

For any approximation error, $\exists\, \gamma(\boldsymbol{x}, y)$ such that

$$\mu_\gamma(\boldsymbol{\delta}|\boldsymbol{x}, y) \propto \Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big) - \gamma(\boldsymbol{x}, y)\Big]_+$$




[Robey et al., NeurIPS'21]



## From optimization to sampling

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]$$

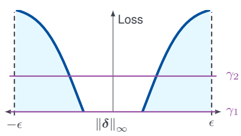
$$\Updownarrow \approx$$

$$\min_{\boldsymbol{\theta}} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{\delta}\sim\mu_\gamma(\cdot|\boldsymbol{x}_n,y_n)}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]}_{L(\theta,\mu)}$$

**Proposition**

For any approximation error, $\exists\, \gamma(\boldsymbol{x}, y)$ such that

$$\mu_\gamma(\boldsymbol{\delta}|\boldsymbol{x}, y) \propto \Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big) - \gamma(\boldsymbol{x}, y)\Big]_+$$




[Robey et al., NeurIPS'21]



## From optimization to sampling

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]$$

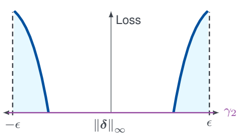
$$\Updownarrow \approx$$

$$\min_{\boldsymbol{\theta}} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{\delta}\sim\mu_\gamma(\cdot|\boldsymbol{x}_n,y_n)}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]}_{L(\theta,\mu)}$$

**Proposition**

For any approximation error, $\exists\, \gamma(\boldsymbol{x}, y)$ such that

$$\mu_\gamma(\boldsymbol{\delta}|\boldsymbol{x}, y) \propto \Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big) - \gamma(\boldsymbol{x}, y)\Big]_+$$




[Robey et al., NeurIPS'21]



## From optimization to sampling

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]$$

$$\Updownarrow =$$

$$\min_{\boldsymbol{\theta}} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{\delta}\sim\mu_\gamma(\cdot|\boldsymbol{x}_n,y_n)}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]}_{L(\theta,\mu)}$$

**Proposition**

For any approximation error, $\exists\, \gamma(\boldsymbol{x}, y)$ such that

$$\mu_\gamma(\boldsymbol{\delta}|\boldsymbol{x}, y) \propto \Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big) - \gamma(\boldsymbol{x}, y)\Big]_+$$


[Robey et al., NeurIPS'21]

# From optimization to sampling

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]$$

PRIMAL

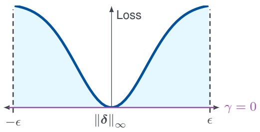
$$\updownarrow \approx$$

$$\min_{\boldsymbol{\theta}} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{\delta}\sim\mu_\gamma(\cdot|\boldsymbol{x}_n,y_n)}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]}_{L(\theta,\mu)}$$

PRIMAL

**Proposition**
For any approximation error, $\exists\, \gamma(\boldsymbol{x}, y)$ such that

$$\mu_0(\boldsymbol{\delta}|\boldsymbol{x}, y) \propto \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big)$$




[Robey et al., NeurIPS'21]



# Constrained learning for robustness

**Problem**
Learn an image classifier that is robust to input perturbations

$$\max_{\lambda\geq 0}\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda\left[\max_{\boldsymbol{\delta}\in\Delta}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]$$

- ✓ Balancing nominal accuracy and robustness ⇒ Dual constrained learning
- ✗ Computing the worst-case perturbations
  - gradient ascent → non-convex, underparametrized



# Constrained learning for robustness

**Problem**
Learn an image classifier that is robust to input perturbations

$$\max_{\lambda\geq 0}\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda\left[\cancel{\max_{\boldsymbol{\delta}\in\Delta}}^{\mathbb{E}_{\boldsymbol{\delta}\sim\mu_0(\cdot|\boldsymbol{x}_n,y_n)}}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\right]$$

- ✓ Balancing nominal accuracy and robustness ⇒ Dual constrained learning
- ✓ Computing the worst-case perturbations
  - ~~gradient ascent → non-convex, underparametrized~~ ⇒ sampling



# Dual Adversarial LEarning

**for** $n = 1, \dots, N$:
- $\boldsymbol{\delta}_n \sim \text{Random}(\Delta)$
- **for** $k = 1, \dots, K$:
  - $\zeta \sim \text{Laplace}(0, I)$
  - $\boldsymbol{\delta}_n \leftarrow \text{proj}_\Delta\left[\boldsymbol{\delta}_n + \eta\,\text{sign}\left[\nabla_{\boldsymbol{\delta}}\log\Big(\text{Loss}\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big)\Big)\right] + \sqrt{2\eta T}\zeta\right]$
- **end**

HMC sampling: $\boldsymbol{\delta} \sim \mu_0(\cdot|\boldsymbol{x}_n, y_n)$

- $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big)\right]$ — SGD

**end**

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big) - c\right)\right]_+$ — GA

[Robey et al., NeurIPS'21]



# Dual Adversarial LEarning

**for** $n = 1, \dots, N$:
- $\boldsymbol{\delta}_n \sim \text{Random}(\Delta)$
- **for** $k = 1, \dots, K$:
  - $\zeta \sim \text{Laplace}(0, I)$
  - $\boldsymbol{\delta}_n \leftarrow \text{proj}_\Delta\left[\boldsymbol{\delta}_n + \eta\,\text{sign}\left[\nabla_{\boldsymbol{\delta}}\log\Big(\text{Loss}\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big)\Big)\right] + \sqrt{2\eta T}\zeta\right]$
- **end**

HMC sampling: $\boldsymbol{\delta} \sim \mu_0(\cdot|\boldsymbol{x}_n, y_n)$

- $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big)\right]$ — SGD

**end**

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big) - c\right)\right]_+$ — GA

[Robey et al., NeurIPS'21]



# Dual Adversarial LEarning

**for** $n = 1, \dots, N$:
- $\boldsymbol{\delta}_n \sim \text{Random}(\Delta)$
- **for** $k = 1, \dots, K$:
  - $\zeta \sim \text{Laplace}(0, I)$
  - $\boldsymbol{\delta}_n \leftarrow \text{proj}_\Delta\left[\boldsymbol{\delta}_n + \eta\,\text{sign}\left[\nabla_{\boldsymbol{\delta}}\log\Big(\text{Loss}\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big)\Big)\right] + \sqrt{2\eta T}\zeta\right]$
- **end**

HMC sampling: $\boldsymbol{\delta} \sim \mu_0(\cdot|\boldsymbol{x}_n, y_n)$

- $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big)\right]$ — SGD

**end**

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big) - c\right)\right]_+$ — GA

[Robey et al., NeurIPS'21]



# Dual Adversarial LEarning

**for** $n = 1, \dots, N$:
- $\boldsymbol{\delta}_n \sim \text{Random}(\Delta)$
- **for** $k = 1, \dots, K$:
  - $\zeta \sim \text{Laplace}(0, I)$
  - $\boldsymbol{\delta}_n \leftarrow \text{proj}_\Delta\left[\boldsymbol{\delta}_n + \eta\,\text{sign}\left[\nabla_{\boldsymbol{\delta}}\log\Big(\text{Loss}\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big)\Big)\right] + \sqrt{2\eta T}\zeta\right]$
- **end**

HMC sampling: $\boldsymbol{\delta} \sim \mu_0(\cdot|\boldsymbol{x}_n, y_n)$

- $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big)\right]$ — SGD

**end**

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big) - c\right)\right]_+$ — GA

[Robey et al., NeurIPS'21]



# Dual Adversarial LEarning

**for** $n = 1, \dots, N$:
- $\boldsymbol{\delta}_n \sim \text{Random}(\Delta)$
- **for** $k = 1, \dots, K$:
  - $\zeta \sim \text{Laplace}(0, I)$
  - $\boldsymbol{\delta}_n \leftarrow \text{proj}_\Delta\left[\boldsymbol{\delta}_n + \eta\,\text{sign}\left[\nabla_{\boldsymbol{\delta}}\log\Big(\text{Loss}\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big)\Big)\right] + \sqrt{2\eta T}\zeta\right]$
- **end**

HMC sampling: $\boldsymbol{\delta} \sim \mu(\cdot|\boldsymbol{x}_n, y_n)$

- $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big)\right]$ — SGD

**end**

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_n), y_n\big) - c\right)\right]_+$ — GA

[Robey et al., NeurIPS'21]

# Dual Adversarial LEarning

for $n = 1, \dots, N$:

$\quad \boldsymbol{\delta}_n \sim \text{Random}(\Delta)$

$\quad$ for $k = 1, \dots, K$:

$\qquad \zeta \sim \text{Laplace}(0, I)$

$\qquad \boldsymbol{\delta}_n \leftarrow \text{proj}_{\Delta}\left[\boldsymbol{\delta}_n + \eta\, \text{sign}\left[\nabla_{\boldsymbol{\delta}} \log\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_n), y_n\big)\right]\right] + \sqrt{2\eta T}\zeta\right]$

$\quad$ end

$\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_n), y_n\big)\right]$

end

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_n), y_n\big) - c\right)\right]_+$

Gaussian
[Lopes et al., arXiv'19]
[Rusak et al., ECCV'20]
Patches
[Zhong et al., AAAI'20]
[Yun et al., ICCV'19]
...

SGD

GA



# Dual Adversarial LEarning

for $n = 1, \dots, N$:

$\quad \boldsymbol{\delta}_n \sim \text{Random}(\Delta)$

$\quad$ for $k = 1, \dots, K$:

$\qquad \zeta \sim \text{Laplace}(0, I)$

$\qquad \boldsymbol{\delta}_n \leftarrow \text{proj}_{\Delta}\left[\boldsymbol{\delta}_n + \eta\, \text{sign}\left[\nabla_{\boldsymbol{\delta}} \log\left(\text{Loss}\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_n + \boldsymbol{\delta}_n), y_n\big)\right)\right] + \sqrt{2\eta T}\zeta\right]$

$\quad$ end

$\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_n), y_n\big)\right]$

end

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_n), y_n\big) - c\right)\right]_+$

$T \to 0$: "PGD"
[Szegedy et al., ICLR'14]
[Goodfellow et al., ICLR'15]
[Madry et al., ICLR'18]

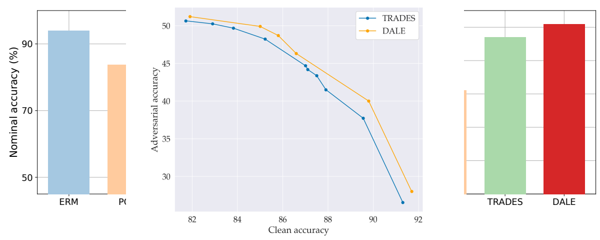
SGD

GA



# Dual Adversarial LEarning

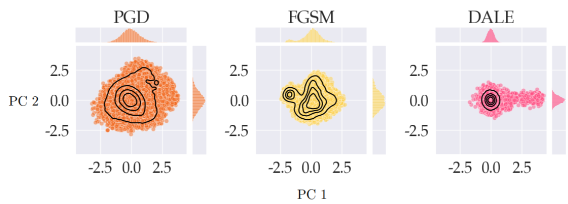
Problem
Learn an image classifier th






# Dual Adversarial LEarning

Problem
Learn an image classifier that is robust to input perturbations

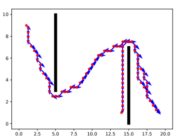





# (Manifold) smoothness

Problem
Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories

- Labeled data ({State, Action})

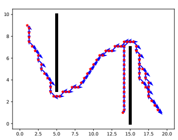

Dataset

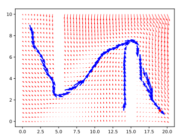


# (Manifold) smoothness

Problem
Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories

- Labeled data ({State, Action})


Dataset

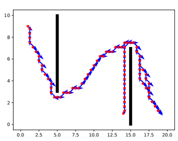

ERM
(85% success)

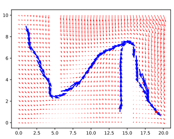


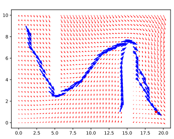
# (Manifold) smoothness

Problem
Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories

- Labeled data ({State, Action})


Dataset


ERM
(85% success)

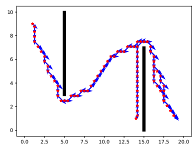

Lipschitz regularization
(66% success)

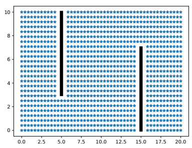


# (Manifold) smoothness

Problem
Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories

- Labeled data ({State, Action}) and unlabeled data ({State in free space})


Dataset


Dataset

# (Manifold) smoothness

Problem
Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories

- Labeled data ({State, Action}) and unlabeled data ({State in free space})
- Use {State in free space} to estimate a data manifold $\mathcal{M}$

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \|f_{\theta}(x_n) - u_n\|^2 \\ \text{subject to} \quad & \max_{x} \|\nabla_{\mathcal{M}} f_{\theta}(x)\|^2 \leq c \end{aligned}$$

[Cerviño et al., ICML'23]



# (Manifold) smoothness

Problem
Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories

- Labeled data ({State, Action}) and unlabeled data ({State in free space})
- Use {State in free space} to estimate a data manifold $\mathcal{M}$

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \|f_{\theta}(x_n) - u_n\|^2 \\ \text{subject to} \quad & \cancel{\max_{x}} \|\nabla_{\mathcal{M}} f_{\theta}(x)\|^2 \leq c \\ & \mathbb{E}_{x\sim\mu_0} \end{aligned}$$

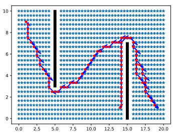
[Cerviño et al., ICML'23]



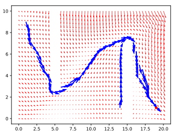
# (Manifold) smoothness

Problem
Learn a smooth (Lipschitz on a manifold) controller that imitates a behavior from limited trajectories

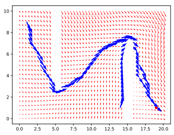
- Labeled data ({Position, Action}) and unlabeled data ({Position})



Dataset



ERM
(85% success)



Manifold smoothness
(94% success)

[Cerviño et al., ICML'23]



# Agenda

Adversarially robust learning

Semi-infinite learning

Probabilistic robustness



# Constrained learning challenges

$$\begin{array}{lcl} \displaystyle\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\theta}(x_n), y_n\big) & \xrightarrow{\text{PAC}} & \displaystyle\min_{\theta} \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\ell\big(f_{\theta}(x_n), y_n\big)\Big] \\ & \xrightarrow{\text{PACC}} & \\ \text{s. to } \displaystyle\frac{1}{N}\sum_{n=1}^{N}\Big[\max_{\delta\in\Delta} \ell\big(f_{\theta}(x_n+\delta), y_n\big)\Big] \leq c & \xrightarrow[\text{PAC}]{} & \text{s. to } \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\max_{\delta\in\Delta} \ell\big(f_{\theta}(x_n+\delta), y_n\big)\Big] \leq c \end{array}$$

**Challenges**

1) *Statistical*: does the solution of the constrained empirical problem generalize?
2) *Computational*: can we solve the constrained empirical problem?



# Constrained learning challenges

$$\begin{array}{lcl} \displaystyle\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\theta}(x_n), y_n\big) & \xrightarrow{\text{PAC}} & \displaystyle\min_{\theta} \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\ell\big(f_{\theta}(x_n), y_n\big)\Big] \\ & \text{PACC} & \\ \text{s. to } \displaystyle\frac{1}{N}\sum_{n=1}^{N}\Big[\max_{\delta\in\Delta} \ell\big(f_{\theta}(x_n+\delta), y_n\big)\Big] \leq c & \xrightarrow[\text{PAC?}]{} & \text{s. to } \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\max_{\delta\in\Delta} \ell\big(f_{\theta}(x_n+\delta), y_n\big)\Big] \leq c \end{array}$$

**Challenges**

1) *Statistical*: does the solution of the constrained empirical problem generalize?
2) *Computational*: can we solve the constrained empirical problem?



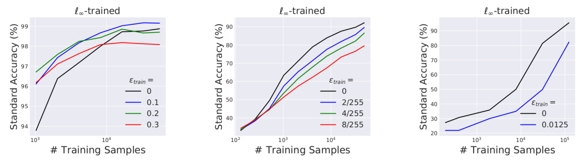
# Statistical complexity

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N}\Big[\max_{\delta\in\Delta} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\Big] \xrightarrow{?} \min_{\theta} \mathbb{E}_{(x,y)}\Big[\max_{\delta\in\Delta} \text{Loss}\big(f_{\theta}(x+\delta), y\big)\Big]$$

- Is robust learning harder than non-robust learning? Do we need more samples?
  **A:** YES *and* NO
  [Cullina, Bhagoji, Mittal. PAC-learning in the presence of evasion adversaries, NeurIPS'18] ✔ ✘
  [Yin, Ramchandran, Bartlett. Rademacher Complexity for Adversarially Robust Generalization, ICML'19] ✔
  [Montasser, Hanneke, Srebro. VC Classes are Adversarially Robustly Learnable, but Only Improperly, COLT'19] ✔ ✘
  [Awasthi, Frank, Mohri. Adversarial Learning Guarantees for Linear Hypotheses and Neural Networks, ICML'20] ✔
  [Montasser, Hanneke, Srebro. Adversarially robust learning: A generic minimax optimal learner & characterization, NeurIPS'22] ✔ ✘



# Nominal performance of robust models




(a) MNIST (b) CIFAR-10 (c) Restricted ImageNet

[Tsipras et al., ICLR'19]

# “Softer” robustness

- Softmax or *log-sum-exp* [Li et al., ICLR'21]

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\frac{1}{\tau}\log\left(\mathbb{E}_{\boldsymbol{\delta}\sim\mathfrak{m}}\left[e^{\tau\cdot\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right)}\right]\right)\right]$$

  - $\tau \to 0$: classical learning (with randomized data augmentation)
  - $\tau \to \infty$: adversarial robustness (ess sup)

- $L_p$ norms [Rice et al., NeurIPS'21]

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\mathbb{E}_{\boldsymbol{\delta}\sim\mathfrak{m}}\left[\left|\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right)\right|^{\tau}\right]^{1/\tau}\right]$$

  - $\tau = 1$: classical learning (with randomized data augmentation)
  - $\tau \to \infty$: adversarial robustness (ess sup)



# “Softer” robustness

- Softmax or *log-sum-exp* [Li et al., ICLR'21]

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\frac{1}{\tau}\log\left(\mathbb{E}_{\boldsymbol{\delta}\sim\mathfrak{m}}\left[e^{\tau\cdot\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right)}\right]\right)\right]$$

- $L_p$ norms [Rice et al., NeurIPS'21]

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\mathbb{E}_{\boldsymbol{\delta}\sim\mathfrak{m}}\left[\left|\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right)\right|^{\tau}\right]^{1/\tau}\right]$$

✖ Computationally challenging (especially as $\tau \to \infty$, i.e., stronger robustness)

✖ No guaranteed advantages (lower sample complexity? improved trade-offs?)



# Towards probabilistic robustness

$$\begin{aligned}
\min_{\boldsymbol{\theta}}\ & \frac{1}{N}\sum_{n=1}^{N}[t(\boldsymbol{x}_n,y_n)] \\
\text{subject to}\ & \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_0),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_1),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\sqrt{2}}),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_e),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_\pi),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_4),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^2}),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\pi^e}),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^\pi}),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{25}),y_n\right) \leq t(\boldsymbol{x}_n,y_n)
\end{aligned}$$

- Epigraph formulation:

$$\max_{\|\boldsymbol{\delta}\|_\infty\leq\epsilon} \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right) \leq t \iff \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right) \leq t, \text{ for all } \|\boldsymbol{\delta}\|_\infty \leq \epsilon$$

- Semi-infinite program



# Towards probabilistic robustness

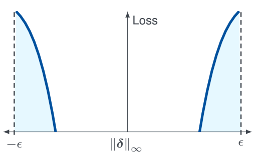
$$\begin{aligned}
\min_{\boldsymbol{\theta}}\ & \frac{1}{N}\sum_{n=1}^{N}[t(\boldsymbol{x}_n,y_n)] \\
\text{subject to}\ & \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_0),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_1),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \cancel{\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\sqrt{2}}),y_n\right) \leq t(\boldsymbol{x}_n,y_n)} \\
& \cancel{\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_e),y_n\right) \leq t(\boldsymbol{x}_n,y_n)} \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_\pi),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_4),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \cancel{\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^2}),y_n\right) \leq t(\boldsymbol{x}_n,y_n)} \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\pi^e}),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \cancel{\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^\pi}),y_n\right) \leq t(\boldsymbol{x}_n,y_n)} \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{25}),y_n\right) \leq t(\boldsymbol{x}_n,y_n)
\end{aligned}$$



# Towards probabilistic robustness

$$\begin{aligned}
\min_{\boldsymbol{\theta}}\ & \frac{1}{N}\sum_{n=1}^{N}[t(\boldsymbol{x}_n,y_n)] \\
\text{subject to}\ & \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_0),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_1),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\sqrt{2}}),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_e),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_\pi),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_4),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^2}),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\pi^e}),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^\pi}),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{25}),y_n\right) \leq t(\boldsymbol{x}_n,y_n)
\end{aligned}$$

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\overbrace{\max_{\boldsymbol{\delta}\in\Delta}}^{\approx\,\mathbb{E}_{\boldsymbol{\delta}\sim\mu_0(\cdot|\boldsymbol{x},y)}} \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right)\right]$$



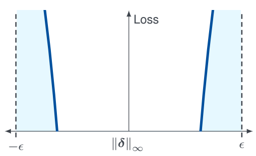

[Robey et al., ICML'22 (spotlight)]



# Towards probabilistic robustness

$$\begin{aligned}
\min_{\boldsymbol{\theta}}\ & \frac{1}{N}\sum_{n=1}^{N}[t(\boldsymbol{x}_n,y_n)] \\
\text{subject to}\ & \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_0),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_1),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \cancel{\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\sqrt{2}}),y_n\right) \leq t(\boldsymbol{x}_n,y_n)} \\
& \cancel{\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_e),y_n\right) \leq t(\boldsymbol{x}_n,y_n)} \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_\pi),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_4),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \cancel{\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^2}),y_n\right) \leq t(\boldsymbol{x}_n,y_n)} \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\pi^e}),y_n\right) \leq t(\boldsymbol{x}_n,y_n) \\
& \cancel{\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^\pi}),y_n\right) \leq t(\boldsymbol{x}_n,y_n)} \\
& \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{25}),y_n\right) \leq t(\boldsymbol{x}_n,y_n)
\end{aligned}$$

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\overbrace{\max_{\boldsymbol{\delta}\in\Delta}}^{\approx\,\mathbb{E}_{\boldsymbol{\delta}\sim\mu_0(\cdot|\boldsymbol{x},y)}} \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right)\right]$$




[Robey et al., ICML'22 (spotlight)]

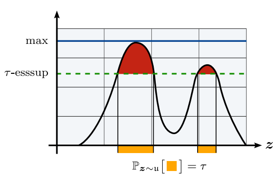


# Probabilistic robustness

- Probabilistic robustness

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\tau\text{-}\underset{\boldsymbol{\delta}\in\Delta}{\text{esssup}}\ \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right)\right]$$

  - $\tau = 1/2$: classical learning (for symmetric m)
  - $\tau = 0$: adversarial robustness (ess sup)



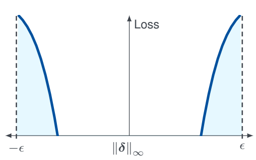

[Robey et al., ICML'22 (spotlight)]

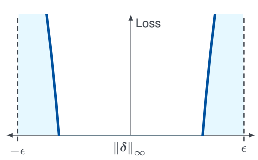


# Probabilistic robustness

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\overbrace{\max_{\boldsymbol{\delta}\in\Delta}}^{\mathbb{E}_{\boldsymbol{\delta}\sim\mu_0(\cdot|\boldsymbol{x},y)}} \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right)\right]$$

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\overbrace{\tau\text{-}\underset{\boldsymbol{\delta}\in\Delta}{\text{esssup}}}^{\mathbb{E}_{\boldsymbol{\delta}\sim\mu_0(\cdot|\boldsymbol{x},y)}} \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),y\right)\right]$$



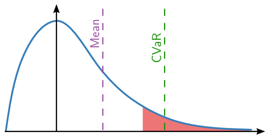

[Robey et al., ICML'22 (spotlight)]

# Probabilistic robustness and Risk

- Conditional value at risk:

$$\mathrm{CVaR}_\rho(f) = \mathbb{E}_z\left[f(z) \mid f(z) \geq F_z^{-1}(\rho)\right]$$
$$= \inf_{\alpha \in \mathbb{R}} \alpha + \frac{\mathbb{E}_z\left[[f(z)-\alpha]_+\right]}{1-\rho}$$

  - $\mathrm{CVaR}_0(f) = \mathbb{E}_z[f(z)]$
  - $\mathrm{CVaR}_1(f) = \mathrm{ess\,sup}_z\, f(z)$




**Proposition**
CVaR is the tightest *convex* upper bound of $\tau$-esssup, i.e.,
$\tau\text{-esssup}_z\, f(z) \leq \mathrm{CVaR}_{1-\tau}(f)$ with equality when $\rho = 0$ or $\rho = 1$.

[Shapiro et al. Lectures on Stochastic Programming, 2014; Kalogerias et al., IEEE ICASSP'20]



# Probabilistically robust learning

**for** $n = 1, \dots, N$:
  $\alpha_0 = 0$
  **for** $t = 1, \dots, T$:
    $\delta_t \sim \text{Random}(\Delta)$
    $\alpha \leftarrow \alpha - \frac{\eta}{\tau}\left(\tau - \mathbb{I}\left[\text{Loss}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \delta_t), y_n) \geq \alpha\right]\right)$
  **end**
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \underbrace{\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_T), y_n\big) - \alpha\right]_+}_{\approx \mathrm{CVaR}_{1-\tau}\left[\text{Loss}(f_{\boldsymbol{\theta}}(x_n+\boldsymbol{\delta}), y_n)\right]}$
**end**

SGD (CVaR)

SGD ($\boldsymbol{\theta}$)

[Robey et al., ICML'22 (spotlight)]



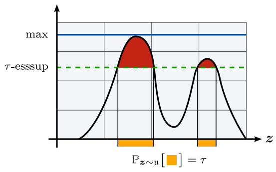
# Probabilistic robustness

- Probabilistic robustness

$$\min_{\theta} \mathbb{E}_{(x,y)}\left[\tau\text{-esssup}_{\boldsymbol{\delta} \in \Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big)\right]$$

  - $\tau = 1/2$: classical learning (for symmetric $\mathfrak{m}$)
  - $\tau = 0$: adversarial robustness (ess sup)

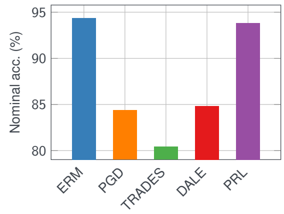
  - Potentially better sample complexity
    [Robey et al., ICML'22 (spotlight)]
    [Raman et al., NeurIPS ML Safety Workshop'22]
  - Better performance trade-off
    [Robey et al., ICML'22 (spotlight)]

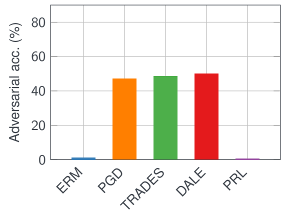



[Robey et al., ICML'22 (spotlight)]



# Probabilistically robust learning

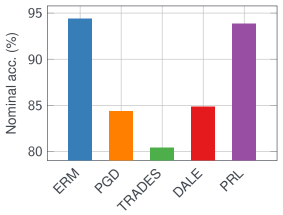


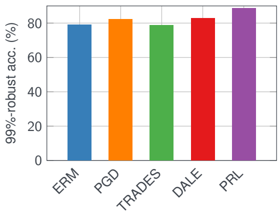




[Robey et al., ICML'22 (spotlight)]



# Probabilistically robust learning




[Robey et al., ICML'22 (spotlight)]



# Summary

- **Semi-infinite constrained learning is the a tool to enforce worst-case requirements**
- **Semi-infinite constrained learning…**
- **…but possible. How?**



# Summary

- **Semi-infinite constrained learning is the a tool to enforce worst-case requirements**
  e.g., robustness [Robey et al., NeurIPS'21], invariance [Hounie et al., ICML'23], smoothness [Cerviño et al., ICML'23]…
- **Semi-infinite constrained learning…**
- **…but possible. How?**



# Summary

- **Semi-infinite constrained learning is the a tool to enforce worst-case requirements**
  e.g., robustness [Robey et al., NeurIPS'21], invariance [Hounie et al., ICML'23], smoothness [Cerviño et al., ICML'23]…
- **Semi-infinite constrained learning…**
  Learning problem with an infinite number of constraints
- **…but possible. How?**

## Summary

- **Semi-infinite constrained learning is the a tool to enforce worst-case requirements**
  e.g., robustness [Robey et al., NeurIPS'21], invariance [Hounie et al., ICML'23], smoothness [Cerviño et al., ICML'23]. . .

- **Semi-infinite constrained learning. . .**
  Learning problem with an infinite number of constraints

- **. . . but possible. How?**
  Using a hybrid sampling–optimization algorithm or, in the case of probabilistic robustness, a *tight* convex relaxation (CVaR) [Robey et al., ICML'22]



## Break