# Optimization for more robust, resilient, responsible AI

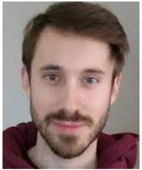
Jérôme MALICK

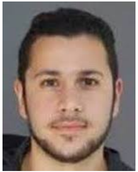
CNRS, Lab. Jean Kuntzmann & MIAI (Institut IA de Grenoble)

SPOT – Toulouse – Oct. 2023

Based on joint work with

Franck Iutzeler ! Waïss Azizian Yassine Laguel Zaid Harchaoui Krishna Pillutla





## Look at how impressive deep learning can be !

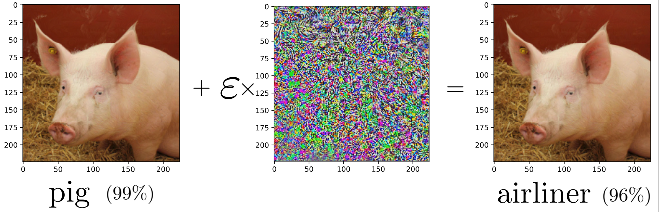
Spectacular success of deep learning, in many fields/applications... E.g. in generation

**Ex:** picture generated with stable diffusion (`https://stablediffusionweb.com`)

"A way towards more robust, resilient, responsible decisions"

# Don't forget how fragile deep learning can be !

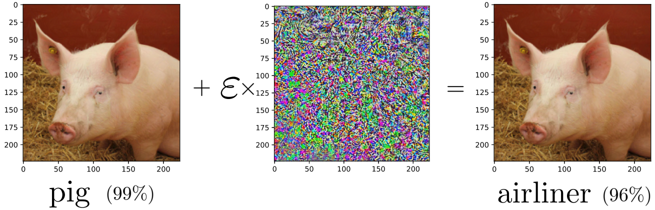
**Example 1:** Flying pigs (notebooks of NeurIPS 2018, tutorial on robustness)




"ML is a wonderful technology: it makes pigs fly" [Kolter, Madry '18]

# Don't forget how fragile deep learning can be !

**Example 1:** Flying pigs (notebooks of NeurIPS 2018, tutorial on robustness)

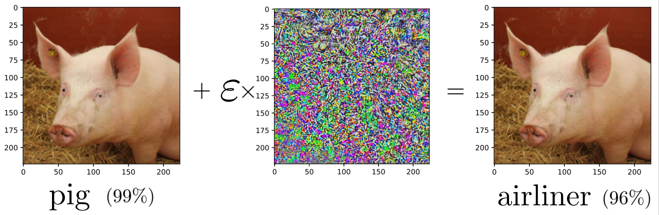



"ML is a wonderful technology: it makes pigs fly" [Kolter, Madry '18]

**Example 2:** Attacks against self-driving cars [@ CVPR '18]

# Don't forget how fragile deep learning can be !

**Example 1:** Flying pigs (notebooks of NeurIPS 2018, tutorial on robustness)

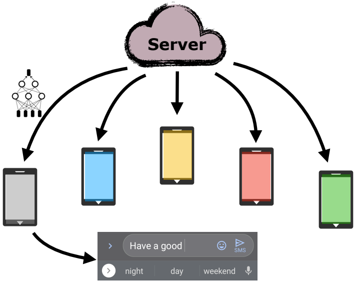


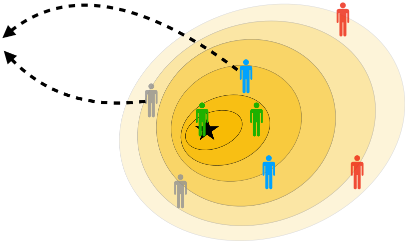

"ML is a wonderful technology: it makes pigs fly" [Kolter, Madry '18]

**Example 2:** Attacks against self-driving cars [@ ICLR '19]

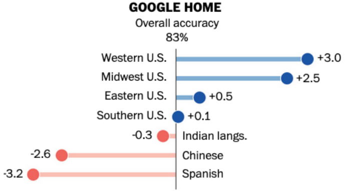
# Observe also that ML can perform poorly

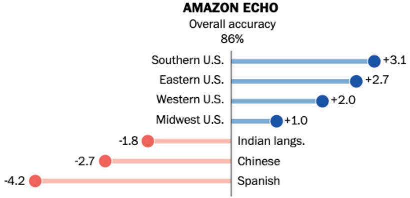
**Example:** Global model is deployed on *individual* clients



[Washington Post '19] "the accent gap"



## Toward robust, responsible learning: set-up of the optim. perspective

- Training data: $\xi_1, \ldots, \xi_N$ (in theory: sampled from $\mathbb{P}_{\text{train}}$ unknown)

  e.g. in supervised learning: labeled data $\xi_i = (a_i, y_i)$ feature, label

- Train model: $f(x, \cdot)$ the loss function with $x$ the parameter/decision $(\omega, \beta, \theta, \ldots)$

  e.g. least-square regression: $f\big(x, (a, y)\big) = (x^\top a - y)^2$

- Compute $x$ via empirical risk minimization (a.k.a SAA)

  (minimize the average loss on training data)

$$\min_x \frac{1}{N} \sum_{i=1}^{N} f(x, \xi_i)$$

# Toward robust, responsible learning: set-up of the optim. perspective

- Training data: $\xi_1, \ldots, \xi_N$ (in theory: sampled from $\mathbb{P}_{\text{train}}$ unknown)
  e.g. in supervised learning: labeled data $\xi_i = (a_i, y_i)$ feature, label
- Train model: $f(x, \cdot)$ the loss function with $x$ the parameter/decision $(\omega, \beta, \theta, \ldots)$
  e.g. least-square regression: $f\big(x, (a, y)\big) = (x^\top a - y)^2$
- Compute $x$ via empirical risk minimization (a.k.a SAA)
  (minimize the average loss on training data)

$$\min_x \frac{1}{N}\sum_{i=1}^{N} f(x, \xi_i) = \mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x, \xi)] \qquad \text{with } \widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\xi_i}$$

- Prediction with $x$ for different data $\xi$
  - Adversarial attacks (e.g. flying pigs, driving cakes...)
  - Presence of bias, e.g. heterogeneous data
  - Distributional shifts: $\mathbb{P}_{\text{train}} \neq \mathbb{P}_{\text{test}}$
  - Generalization: computations with $\widehat{\mathbb{P}}_N$ and guarantees on $\mathbb{P}_{\text{train}}$
- Solution: take possible variations into account during training

# (Distributionally) robust optimization

Optimize expected loss for the worst probability in a set of perturbations

rather than $\min_{x} \mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x,\xi)]$ solve instead $\min_{x} \max_{\mathbb{Q}\in\mathcal{U}} \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$

with $\mathcal{U}$ a neighborhood of $\widehat{\mathbb{P}}_N$ (called ambiguity set)

- $\mathcal{U} = \left\{\widehat{\mathbb{P}}_N\right\}$ : $\min_{x} \frac{1}{N}\sum_{i=1}^{N} f(x,\xi_i)$ standard ERM
- $\mathcal{U}$ defined by moments e.g. [Delage, Ye, '10] [Jegelka *et al.* '19]
- $\mathcal{U} = \left\{\mathbb{Q} : \; d(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho\right\}$ for various distances or divergences
  E.g. KL-div., $\chi_2$-div., max-mean-discrepancy... e.g. [Namkoong, Duchi '17]
- $\mathcal{U} = \left\{\mathbb{Q} : \; W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho\right\}$ Wasserstein distance [Kuhn *et al.* '18] (popular in OT)

$\boxed{\text{modeling vs. computational tractability}}$

## Simple illustration of the gain in robustness

### Example : basic classification (linear, 2D, 2 classes...)

- Training data : $\xi_i = (a_i, y_i) \in \mathbb{R}^2 \times \{-1, +1\}$
  sampled from two Gaussian distributions with variances $\sigma = 1$ and $\sigma = 5$
- Testing data : reverse variance $\sigma = 5$ and $\sigma = 1$
- Compute standard separator by min logistic loss $f(x, \xi) = \log(1 + \exp(-y\, a^\top x))$

$$\min_x \frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-y_i\, a_i{}^\top x))$$

- Compute a robust separator (Wassertein DRO w. $c((a,y),(a',y')) = \|a - a'\| + \kappa 1_{y=y'}$)

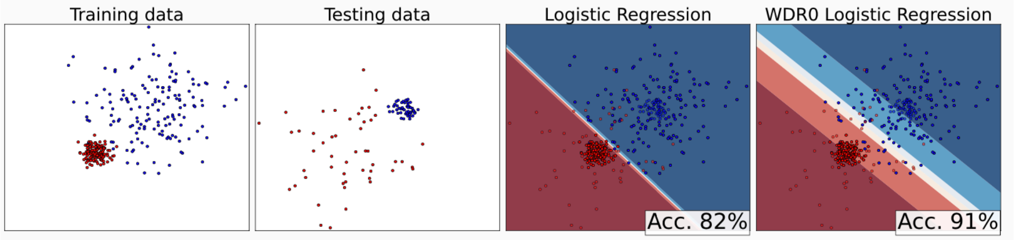
# Simple illustration of the gain in robustness

## Example : basic classification (linear, 2D, 2 classes...)

- Training data : $\xi_i = (a_i, y_i) \in \mathbb{R}^2 \times \{-1, +1\}$
  sampled from two Gaussian distributions with variances $\sigma = 1$ and $\sigma = 5$
- Testing data : reverse variance $\sigma = 5$ and $\sigma = 1$
- Compute standard separator by min logistic loss $f(x, \xi) = \log(1 + \exp(-y\, a^\top x))$

$$\min_x \frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-y_i\, a_i^\top x))$$

- Compute a robust separator (Wassertein DRO w. $c((a, y), (a', y')) = \|a - a'\| + \kappa 1_{y=y'}$)



## DRO, at the intersection of OR, ML, Optim

DRO is very attractive

- Statistical/theoretical properties
  e.g. [Blanchet *et al.* '18] and [Blanchet and Shapiro '23]
- Computable in many cases
  e.g. [Kuhn *et al.* '18], [Zhao Guan '18]...
- Natural in many applications
  back to [Scarf 1958] ! + (...) + recent trend in learning, e.g. [Kuhn *et al.* '20]
- Interprets up to first-order as a penalization by $\|\nabla_\xi f(x, \xi)\|$ e.g. [Gao *et al.* '18]

## DRO, at the intersection of OR, ML, Optim

DRO is very attractive

- Statistical/theoretical properties – warning : dimensionality ! (spotlight #1)
  e.g. [Blanchet *et al.* '18] and [Blanchet and Shapiro '23]
- Computable in many cases – on-going research ! (Franck's talk)
  e.g. [Kuhn *et al.* '18], [Zhao Guan '18]...
- Natural in many applications – towards fairness (spotlight #2)
  back to [Scarf 1958] ! + (...) + recent trend in learning, e.g. [Kuhn *et al.* '20]
- Interprets up to first-order as a penalization by $\|\nabla_\xi f(x, \xi)\|$ e.g. [Gao *et al.* '18]

# Spotlight #1 : Statistical guarantees of optimal-transport-based DRO

Azizian Waiss, Franck Iutzeler, and Jérôme Malick
Excat generalization guarantees for (regularized) WDRO models
Just accepted in NeurIPS, 2023

# Wasserstein comes into play

**Def:** Wasserstein distance (given a cost function $c$)

$$W(\mathbb{P}, \mathbb{Q}) = \min_{\boldsymbol{\pi}} \left\{ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi, \xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P} \text{ and } [\boldsymbol{\pi}]_2 = \mathbb{Q} \right\}$$

# Wasserstein comes into play

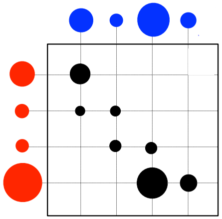
**Def:** Wasserstein distance (given a cost function $c$)

$$W(\mathbb{P},\mathbb{Q}) = \min_{\boldsymbol{\pi}}\left\{ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P} \text{ and } [\boldsymbol{\pi}]_2 = \mathbb{Q} \right\}$$

**Demystification:** in the discrete case

e.g. $\mathbb{P} = (p_1, \ldots, p_N)$ and $\mathbb{Q} = (q_1, \ldots, q_N)$ in the simplex

$$\begin{cases} \min_{\boldsymbol{\pi}} & \sum_{i,j=1}^N c_{i,j}\, \boldsymbol{\pi}_{i,j} \\ & \sum_{j=1}^N \boldsymbol{\pi}_{i,j} = p_i \quad i = 1,\ldots,N \\ & \sum_{i=1}^N \boldsymbol{\pi}_{i,j} = q_j \quad j = 1,\ldots,N \\ & \boldsymbol{\pi}_{i,j} \geqslant 0 \quad i,j = 1,\ldots,N \end{cases}$$



linear assignment !

# Wasserstein comes into play

**Def:** Wasserstein distance (given a cost function $c$)

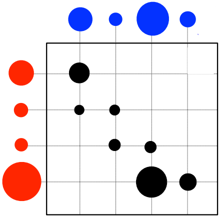
$$W(\mathbb{P},\mathbb{Q}) = \min_{\boldsymbol{\pi}} \left\{ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P} \text{ and } [\boldsymbol{\pi}]_2 = \mathbb{Q} \right\}$$

**Demystification:** in the discrete case

e.g. $\mathbb{P} = (p_1, \dots, p_N)$ and $\mathbb{Q} = (q_1, \dots, q_N)$ in the simplex

$$\begin{cases} \min_{\boldsymbol{\pi}} \quad \sum_{i,j=1}^N c_{i,j}\, \boldsymbol{\pi}_{i,j} \\ \quad \sum_{j=1}^N \boldsymbol{\pi}_{i,j} = p_i \quad i = 1,\dots,N \\ \quad \sum_{i=1}^N \boldsymbol{\pi}_{i,j} = q_j \quad j = 1,\dots,N \\ \quad \boldsymbol{\pi}_{i,j} \geqslant 0 \quad i,j = 1,\dots,N \end{cases}$$

linear assignment !



**Wasserstein-DRO objective** for given $\mathbb{P}$ and $\rho$

$$\begin{cases} \max_{\mathbb{Q}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ \quad W(\mathbb{P},\mathbb{Q}) \leqslant \rho \end{cases}$$

# Wasserstein comes into play

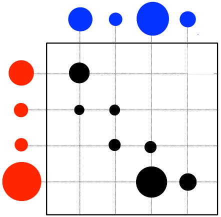
**Def:** Wasserstein distance (given a cost function $c$)

$$W(\mathbb{P},\mathbb{Q}) = \min_{\boldsymbol{\pi}}\left\{\, \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P} \text{ and } [\boldsymbol{\pi}]_2 = \mathbb{Q}\right\}$$

**Demystification:** in the discrete case

e.g. $\mathbb{P} = (p_1, \dots, p_N)$ and $\mathbb{Q} = (q_1, \dots, q_N)$ in the simplex

$$\begin{cases} \min_{\boldsymbol{\pi}} \quad \sum_{i,j=1}^N c_{i,j}\,\boldsymbol{\pi}_{i,j} \\ \quad \sum_{j=1}^N \boldsymbol{\pi}_{i,j} = p_i \quad i = 1,\dots,N \\ \quad \sum_{i=1}^N \boldsymbol{\pi}_{i,j} = q_j \quad j = 1,\dots,N \\ \quad \boldsymbol{\pi}_{i,j} \geqslant 0 \quad i,j = 1,\dots,N \end{cases}$$



linear assignment !

**Wasserstein-DRO objective** for given $\mathbb{P}$ and $\rho$

$$\begin{cases} \max_{\mathbb{Q}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ W(\mathbb{P},\mathbb{Q}) \leqslant \rho \end{cases} \Leftrightarrow \begin{cases} \max_{\mathbb{Q},\boldsymbol{\pi}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \mathbb{P},\, [\boldsymbol{\pi}]_2 = \mathbb{Q} \\ \min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{cases}$$

## Wasserstein comes into play

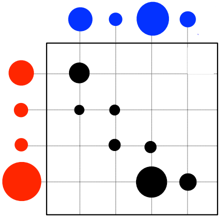
**Def:** Wasserstein distance (given a cost function $c$)

$$W(\mathbb{P},\mathbb{Q}) = \min_{\pi}\left\{ \mathbb{E}_{\pi}[c(\xi,\xi')] : \pi \text{ with marginals } [\pi]_1 = \mathbb{P} \text{ and } [\pi]_2 = \mathbb{Q}\right\}$$

**Demystification:** in the discrete case

e.g. $\mathbb{P} = (p_1, \dots, p_N)$ and $\mathbb{Q} = (q_1, \dots, q_N)$ in the simplex

$$\begin{cases} \min_{\pi} \quad \sum_{i,j=1}^{N} c_{i,j}\, \pi_{i,j} \\ \quad \sum_{j=1}^{N} \pi_{i,j} = p_i \quad i = 1, \dots, N \\ \quad \sum_{i=1}^{N} \pi_{i,j} = q_j \quad j = 1, \dots, N \\ \quad \pi_{i,j} \geqslant 0 \quad i,j = 1, \dots, N \end{cases}$$



linear assignment !

**Wasserstein-DRO objective** for given $\mathbb{P}$ and $\rho$

$$\begin{cases} \max_{\mathbb{Q}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ W(\mathbb{P},\mathbb{Q}) \leqslant \rho \end{cases} \Leftrightarrow \begin{cases} \max_{\mathbb{Q},\pi} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ [\pi]_1 = \mathbb{P}, [\pi]_2 = \mathbb{Q} \\ \min_{\pi} \mathbb{E}_{\pi}[c(\xi,\xi')] \leqslant \rho \end{cases} \Leftrightarrow \begin{cases} \max_{\pi} \ \mathbb{E}_{[\pi]_2}[f(x,\xi)] \\ [\pi]_1 = \mathbb{P} \\ \mathbb{E}_{\pi}[c(\xi,\xi')] \leqslant \rho \end{cases}$$

# Wasserstein comes into play

**Def:** Wasserstein distance (given a cost function $c$)

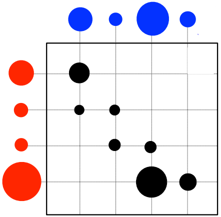
$$W(\mathbb{P},\mathbb{Q}) = \min_{\boldsymbol{\pi}} \big\{ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P} \text{ and } [\boldsymbol{\pi}]_2 = \mathbb{Q} \big\}$$

**Demystification:** in the discrete case

e.g. $\mathbb{P} = (p_1, \dots, p_N)$ and $\mathbb{Q} = (q_1, \dots, q_N)$ in the simplex

$$\begin{cases} \min_{\boldsymbol{\pi}} \quad \sum_{i,j=1}^{N} c_{i,j}\, \boldsymbol{\pi}_{i,j} \\ \quad \sum_{j=1}^{N} \boldsymbol{\pi}_{i,j} = p_i \quad i = 1,\dots,N \\ \quad \sum_{i=1}^{N} \boldsymbol{\pi}_{i,j} = q_j \quad j = 1,\dots,N \\ \quad \boldsymbol{\pi}_{i,j} \geqslant 0 \quad i,j = 1,\dots,N \end{cases}$$

linear assignment !



**Wasserstein-DRO objective** for given $\mathbb{P}$ and $\rho$

$$\begin{cases} \max_{\mathbb{Q}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ W(\mathbb{P},\mathbb{Q}) \leqslant \rho \end{cases} \Leftrightarrow \begin{cases} \max_{\mathbb{Q},\boldsymbol{\pi}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \mathbb{P}, [\boldsymbol{\pi}]_2 = \mathbb{Q} \\ \min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{cases} \Leftrightarrow \begin{cases} \max_{\boldsymbol{\pi}} \ \mathbb{E}_{[\boldsymbol{\pi}]_2}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \mathbb{P} \\ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{cases}$$

duality...

Franck's talk!

# Existing statistical guarantees of WDRO

- Suppose $\xi_1, \ldots, \xi_N \sim \mathbb{P}_{\text{train}}$ (where $\xi \in \mathbb{R}^d$)
- Computations with $\widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\xi_i}$ and guarantees with $\mathbb{P}_{\text{train}}$ ?
- We manipulate the WDRO risk : $R_\rho(x) = \max_{W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho} \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$
- Obviously, if $\rho, N$ large enough such that $W(\mathbb{P}_{\text{train}}, \widehat{\mathbb{P}}_N) \leqslant \rho$, then

$$\underbrace{R_\rho(x)}_{\text{can compute \& optimize}} \geqslant \underbrace{\mathbb{E}_{\mathbb{P}_{\text{train}}}[f(x,\xi)]}_{\text{cannot access}}$$

# Existing statistical guarantees of WDRO

- Suppose $\xi_1, \ldots, \xi_N \sim \mathbb{P}_{\text{train}}$ (where $\xi \in \mathbb{R}^d$)
- Computations with $\widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\xi_i}$ and guarantees with $\mathbb{P}_{\text{train}}$ ?
- We manipulate the WDRO risk : $R_\rho(x) = \max_{W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho} \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$
- Obviously, if $\rho, N$ large enough such that $W(\mathbb{P}_{\text{train}}, \widehat{\mathbb{P}}_N) \leqslant \rho$, then

$$\underbrace{R_\rho(x)}_{\text{can compute \& optimize}} \geqslant \underbrace{\mathbb{E}_{\mathbb{P}_{\text{train}}}[f(x,\xi)]}_{\text{cannot access}}$$

- It requires $\rho \propto 1/\sqrt[d]{N}$ [Fournier and Guillin '15] (issue)
- Not optimal: $\rho \propto 1/\sqrt{N}$ suffices
  - asymptotically [Blanchet *et al* '22]
  - in particular cases [Shafieez-Adehabadeh *et al* '19]
  - or with error terms [Gao '22]

# Extended exact generalization guarantees of WDRO

Our approach : a direct "optimization" approach

(work to get a concentration result on the (dual) objective in the $\ell_2$-case)

**Theorem ([Azizian, Iutzeler, M. '23])**

*Assumptions : compactness on $\xi$ + compactness on $f$ + quad. growth of $f$ near its minimizers*

*For* $\delta \in (0,1)$, *if* $\rho \geqslant O\left(\sqrt{\frac{\log 1/\delta}{N}}\right)$

*Generalization guarantee: w.p.* $1-\delta$, $\qquad R_\rho(x) \geqslant \mathbb{E}_{\mathbb{P}_{train}}[f(x,\xi)]$

# Extended exact generalization guarantees of WDRO

Our approach : a direct "optimization" approach

(work to get a concentration result on the (dual) objective in the $\ell_2$-case)

**Theorem ([Azizian, Iutzeler, M. '23])**

*Assumptions : compactness on $\xi$ + compactness on $f$ + quad. growth of $f$ near its minimizers*

*For* $\delta \in (0,1)$, *if* $\rho \geqslant O\left(\sqrt{\frac{\log 1/\delta}{N}}\right)$

*Generalization guarantee: w.p.* $1-\delta$, $\quad R_\rho(x) \geqslant \mathbb{E}_{\mathbb{P}_{train}}[f(x,\xi)]$

*Distribution shifts: w.p.* $1-\delta$,

$$W(\mathbb{P},\mathbb{Q})^2 \leqslant \rho\left(\rho - O\left(\sqrt{\frac{\log 1/\delta}{N}}\right)\right) \quad \textit{it holds} \quad R_\rho(x) \geqslant \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$$

# Extended exact generalization guarantees of WDRO

Our approach : a direct "optimization" approach
(work to get a concentration result on the (dual) objective in the $\ell_2$-case)

**Theorem ([Azizian, Iutzeler, M. '23])**

*Assumptions : compactness on $\xi$ + compactness on $f$ + quad. growth of $f$ near its minimizers*

*For* $\delta \in (0,1)$, *if* $\rho \geqslant O\left(\sqrt{\frac{\log 1/\delta}{N}}\right)$

*Generalization guarantee: w.p.* $1-\delta$, $\quad R_\rho(x) \geqslant \mathbb{E}_{\mathbb{P}_{train}}[f(x,\xi)]$

*Distribution shifts: w.p.* $1-\delta$,

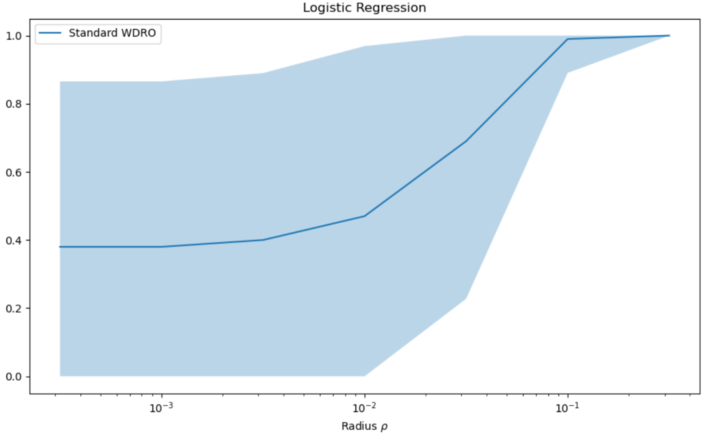
$$W(\mathbb{P},\mathbb{Q})^2 \leqslant \rho\left(\rho - O\left(\sqrt{\frac{\log 1/\delta}{N}}\right)\right) \quad \textit{it holds} \quad R_\rho(x) \geqslant \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$$

Assumptions valid in many cases: linear/logistic regression, kernel models, smooth neural networks, family of invertible mappings (e.g. normalizing flows)

# Illustration

On logistic regression:

- for each $\rho$, sample 200 training datasets
- solve the WDRO problem on each of them [Blanchet *et al* '22]
- plot the proba of $R_\rho(f) - \mathbb{E}_{\mathbb{P}_{\text{train}}}[f] \geqslant 0$ (average, standard deviation)
- the training robust loss is indeed an upper-bound on the true loss



# Spotlight #2 : Robust Federated Learning

Krishna Pillutla, Yassine Laguel, Jérôme Malick, Zaid Harchaoui
Federated Learning with Superquantile Aggregation for Heterogeneous Data
Machine Learning Journal, 2023

## Setting: federated learning in a nutschell

- Standard learning : get all the data and learn your model on it
- Efficient... but is privacy invasive (hospitals, compagnies...)
- Idea : move the model not the data !

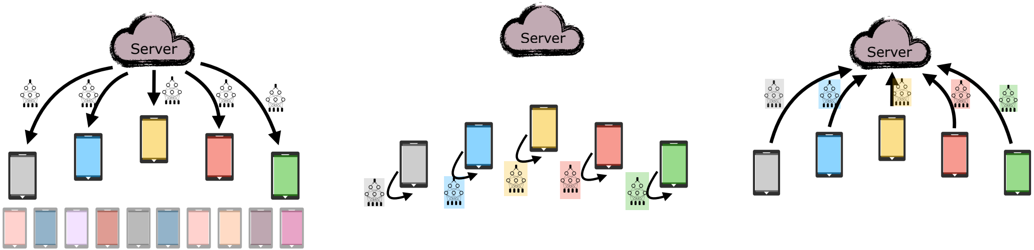
## Setting: federated learning in a nutschell

- Standard learning : get all the data and learn your model on it
- Efficient... but is privacy invasive (hospitals, compagnies...)
- Idea : move the model not the data !
- Usual learning algorithm : FedAvg [McMahan *et al* 2017]
  (based on old ideas, e.g. [Mangasarian 1995])

*Step 1 of 3: Server broadcasts global model to sampled clients*

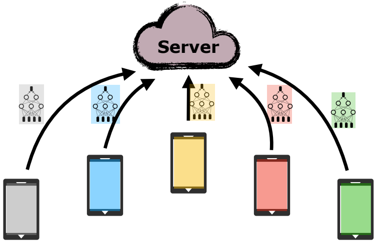
*Step 2 of 3: Clients perform some local SGD steps on their local data*

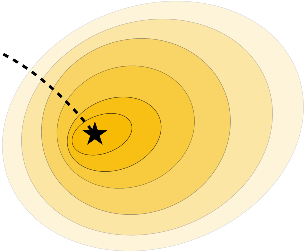
*Step 3 of 3: Aggregate client updates securely*



# Issue of heterogeneous users

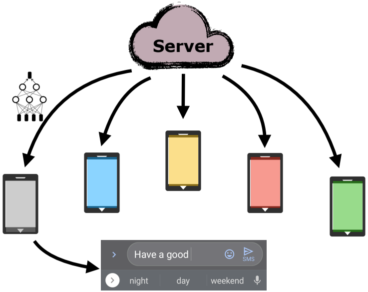
Global model is trained on *average distribution* across clients (ERM)

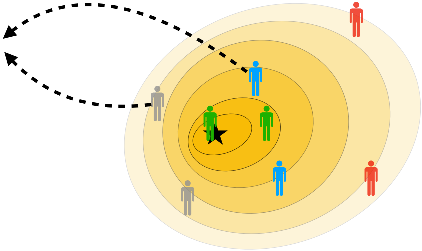


# Issue of heterogeneous users

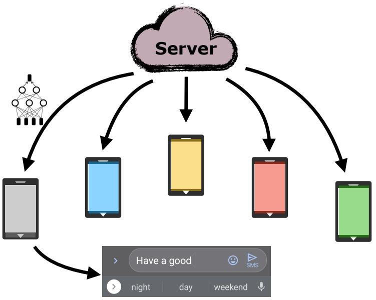
Global model is deployed on *individual* clients

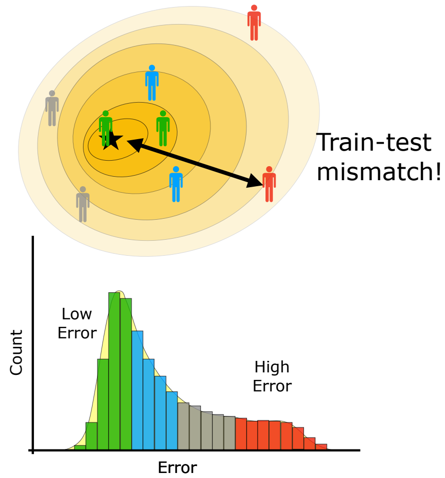


# Issue of heterogeneous users

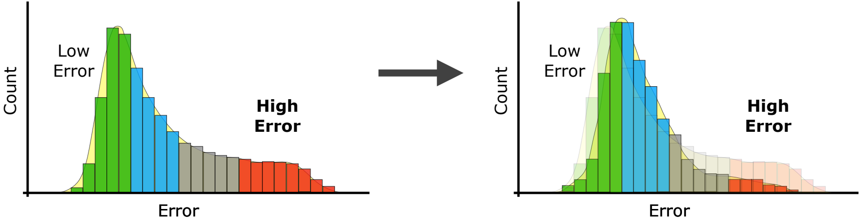
Global model is deployed on *individual* clients



# Robust approach over the users

Our goal: reduce the tail error

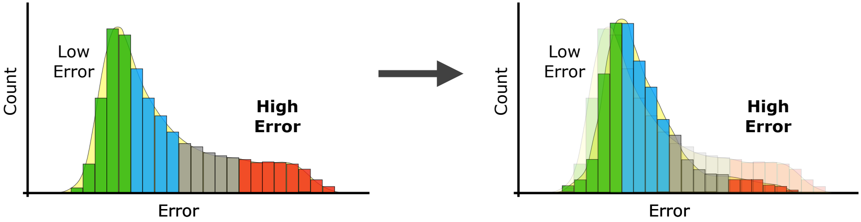



Risk measure: Superquantile [Rockafellar *et al* '00] (a.k.a. Conditional Value-at-Risk)

(Recent applications in learning [Pillutla, Laguel, M., Harchaoui '21] [Bondel *et al* '22])

# Robust approach over the users

Our goal: reduce the tail error

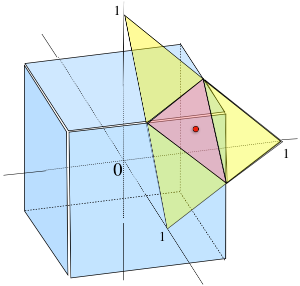



Risk measure: Superquantile [Rockafellar *et al* '00] (a.k.a. Conditional Value-at-Risk)
(Recent applications in learning [Pillutla, Laguel, M., Harchaoui '21] [Bondel *et al* '22])

Duality gives a DRO formulation

$$R_\theta(x) = \max_{\mathbb{Q} \in \mathcal{U}} \mathbb{E}_{\mathbb{Q}}[F(x)]$$

$$= \max_{\pi \in \Delta_n} \left\{ \sum_{i=1}^{n} \pi_i \, F_i(x) : \ \|\pi\|_\infty \leqslant \frac{1}{n\theta} \right\}$$

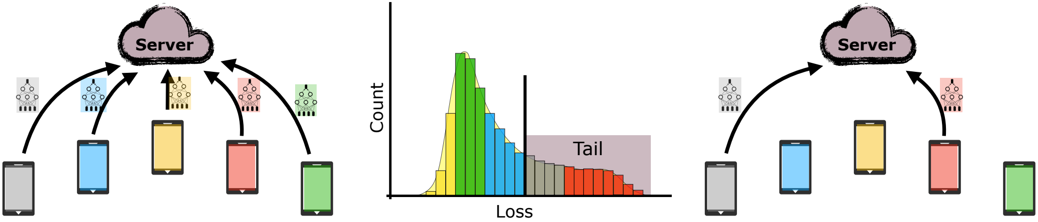


# DRO/superquantile in action in federated learning

Only step 3 differs between Standard ERM approach and our DRO approach

*Step 3 of 3: Aggregate updates contributed by **all clients***

*Step 3 of 3: Aggregate updates contributed by **tail clients** only*




DRO approach is fully compatible with secure aggregation and differential privacy
[Pillutla, Laguel, M., Harchaoui '22]

# Convergence analysis

Analysis when $F_i$ are smooth (and nonconvex)

Challenges: non-smoothness of $R_\theta$, biais due to local participation,...

**Theorem ([Pillutla, Laguel, M., Harchaoui '23])**

*Suppose $F_i$ are $G$-Lipschitz and with gradients $L$-Lipshitz*

$$\mathbb{E}\|\nabla \Phi_\theta^{2L}(x_t)\|^2 \leqslant \sqrt{\frac{\Delta L G^2}{t}} + (1-\tau)^{1/3}\left(\frac{\Delta L G}{t}\right)^{2/3} + \frac{\Delta L}{t}$$

*with $t$: nb comm. rounds, $\tau$: nb local updates, and $\Delta$: initial error*

where $\Phi_\theta^\mu(x) = \inf_y \left\{ \bar{R}_\theta(y) + \frac{\mu}{2}\|y-x\|^2 \right\}$ (Moreau♡ enveloppe) [Davis Drus. '21]

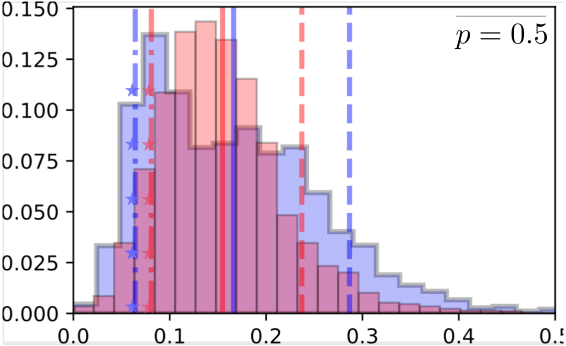
$\bar{R}_\theta$ an approximation of $R_\theta$ with unbiased gradient [Levy *et al* '21]

\+ result of linear convergence when $F_i$ are convex (add smoothing and regularization)

# Illustration: DRO does reshape test histograms

Classification task – ConvNet with EMNIST dataset (1730 users, 179 images/users)

Histogram over users of test misclassification error: standard vs. DRO
(dashed lines: 10%/90%-quantiles)




Recall the goal:

# Conclusion

## Main take-aways

- ML works well, unless it does not. Work needed. Optimization is in the game
  Distributionally robust optimization DRO is rich, active topic
- Spotlight #1: WDRO has nice generalization properties
- Spotlight #2: DRO works in practice (code: `github.com/krishnap25/sqwash`)

```python
import torch.nn.functional as F
from sqwash import reduce_superquantile

for x, y in dataloader:
    y_hat = model(x)
    batch_losses = F.cross_entropy(y_hat, y, reduction='none')  # must set `reduction='none'`
    loss = reduce_superquantile(batch_losses, superquantile_tail_fraction=0.5)  # Additional line
    loss.backward()  # Proceed as usual from here
    ...
```

# Conclusion

## Main take-aways

- ML works well, unless it does not. Work needed. Optimization is in the game
  Distributionally robust optimization DRO is rich, active topic
- Spotlight #1: WDRO has nice generalization properties
- Spotlight #2: DRO works in practice (code: `github.com/krishnap25/sqwash`)

```python
import torch.nn.functional as F
from sqwash import reduce_superquantile

for x, y in dataloader:
    y_hat = model(x)
    batch_losses = F.cross_entropy(y_hat, y, reduction='none')  # must set `reduction='none'`
    loss = reduce_superquantile(batch_losses, superquantile_tail_fraction=0.5)  # Additional line
    loss.backward()  # Proceed as usual from here
    ...
```

## What's next ? Can't wait for Franck's talk !

- WDRO is popular... But requires numerical work
- How to dealing with nonsmooth objective $R_\rho(x) = \max_{W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho} \mathbb{E}_{\mathbb{Q}}[f(x, \xi)]$

thank you all ☺