Distributionally Robust Optimization with Data Geometry

Jiashuo Liu, Jiayun Wu, Bo Li, Peng Cui

Department of Computer Science and Technology

Tsinghua University

Over-Pessimism Problem of DRO

NEURAL INFORMATION PROCESSING SYSTEMS

- The objective function of DRO:

$$\min_{\theta} \sup_{Q \in \mathcal{P}(P_{tr})} \mathbb{E}_Q[\ell(f_\theta(X), Y)]$$

  - $\mathcal{P}(P_{tr})$ is the distribution set defined via some distance metric as:

$$\mathcal{P}(P_{tr}) = \{Q: Dist(Q, P_{tr}) \leq \rho\}$$

- When the testing distribution is included in $\mathcal{P}(P_{tr})$, the testing performance is guaranteed.
- When the distribution set $\mathcal{P}(P_{tr})$ is overwhelmingly large, the learned model will predict with ***low-confidence***.

→ *Over-Pessimism*

Over-pessimism Problem in DRO

- When the uncertainty set is overwhelmingly large, the learned model predicts with low confidence.

$$\min_\theta \sup_{P:Dist(P,P_{tr})\leq\epsilon} \mathbb{E}_P[\ell(\theta;X,Y)]$$

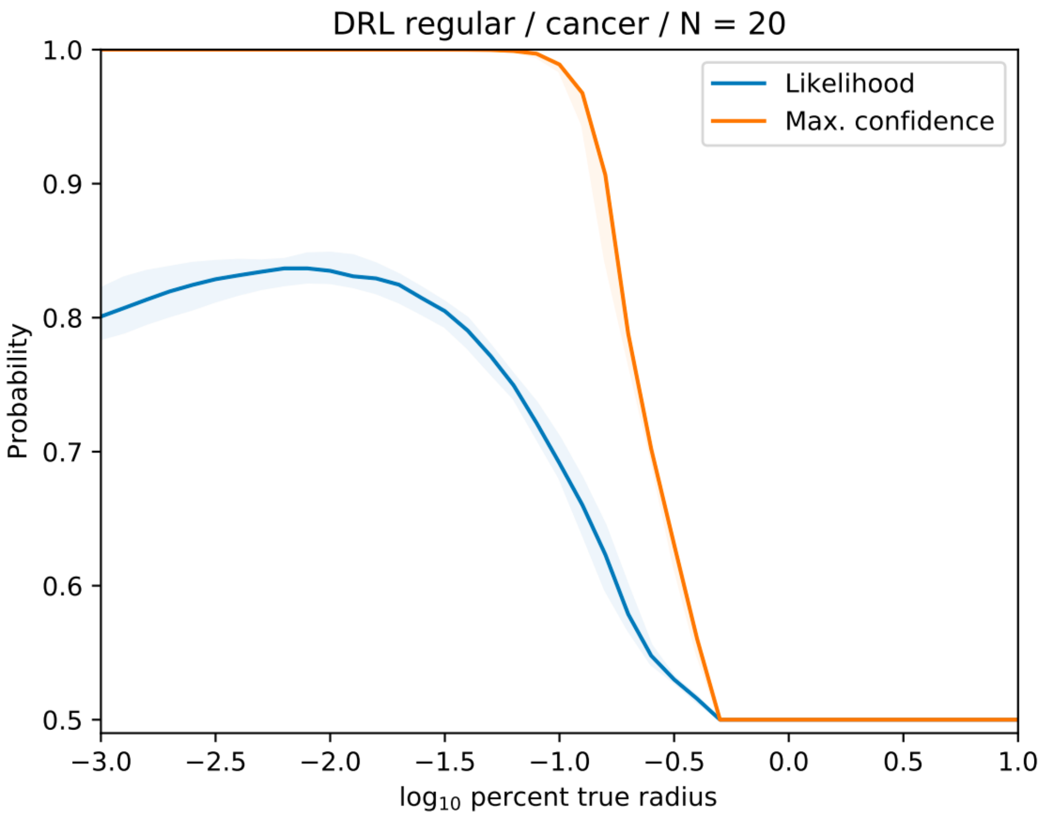
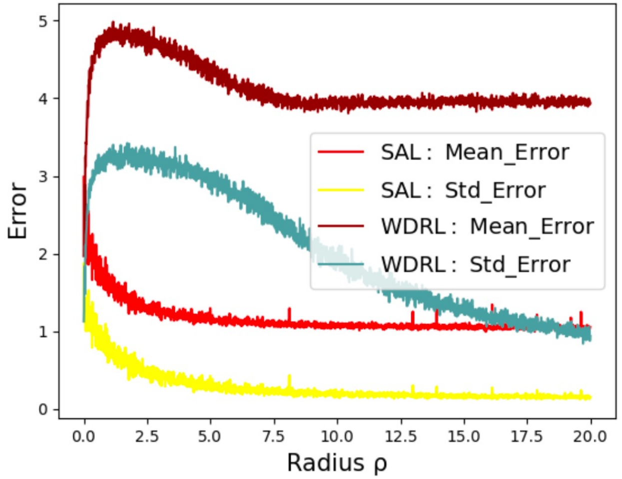


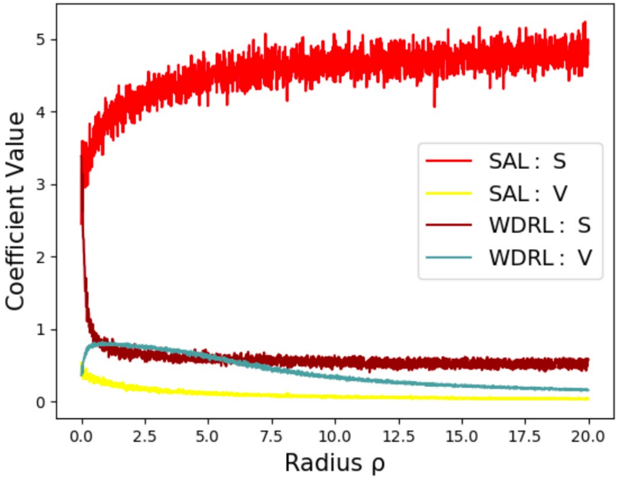
Figure from Frogner, C., Claici, S., Chien, E., & Solomon, J. (2019). Incorporating unlabeled data into distributionally robust learning. JMLR.

Over-pessimism Problem in DRO

- WDRO: another low confidence example for in linear setting:



(b) Testing performance with respect to radius



(c) The learned coefficients of S and V w.r.t. radius

- f-DRO (or joint DRO): exactly fits the training distribution in classification

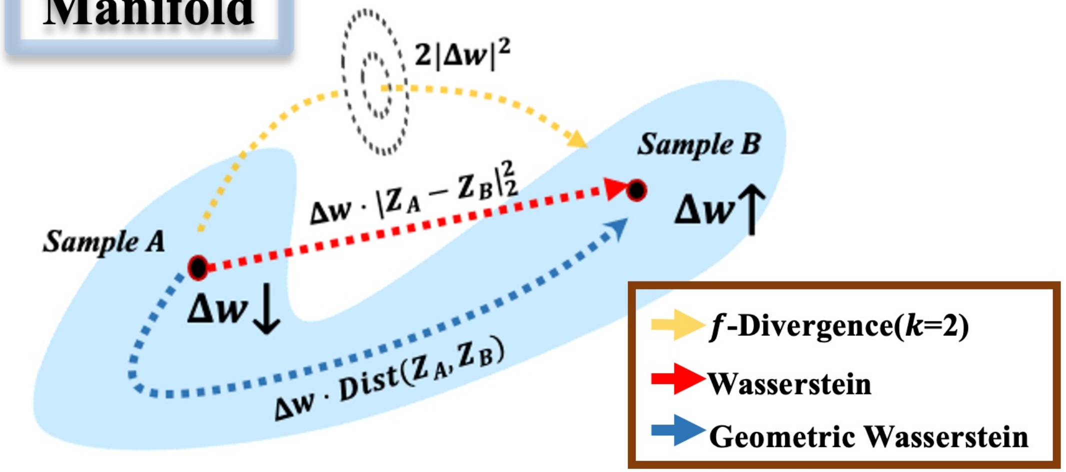
Liu et al. Stable Adversarial Learning under Distributional Shifts.
Hu et al. Does Distributionally Robust Supervised Learning Give Robust Classifiers?

What Caused the Over-pessimism?

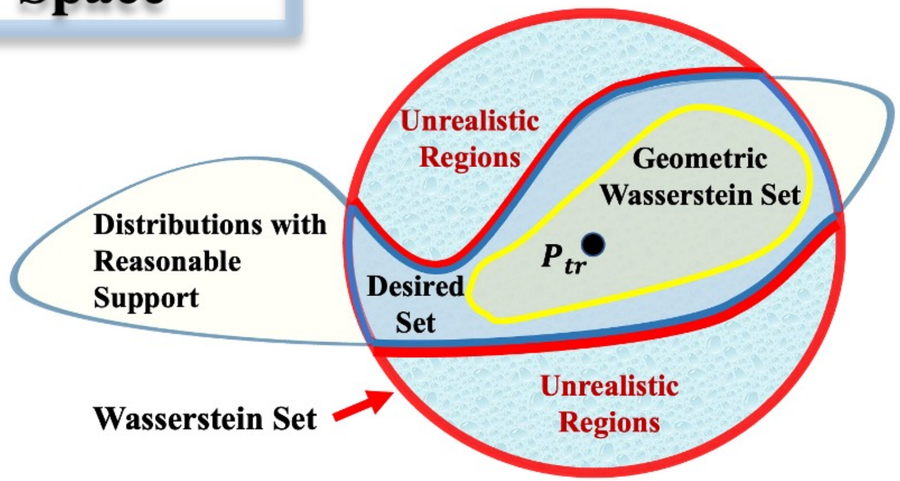
—— *from the distance metric perspective*

NEURAL INFORMATION PROCESSING SYSTEMS

Data Manifold



Wasserstein Space



Leverage the data geometry to form a more reasonable distribution set.

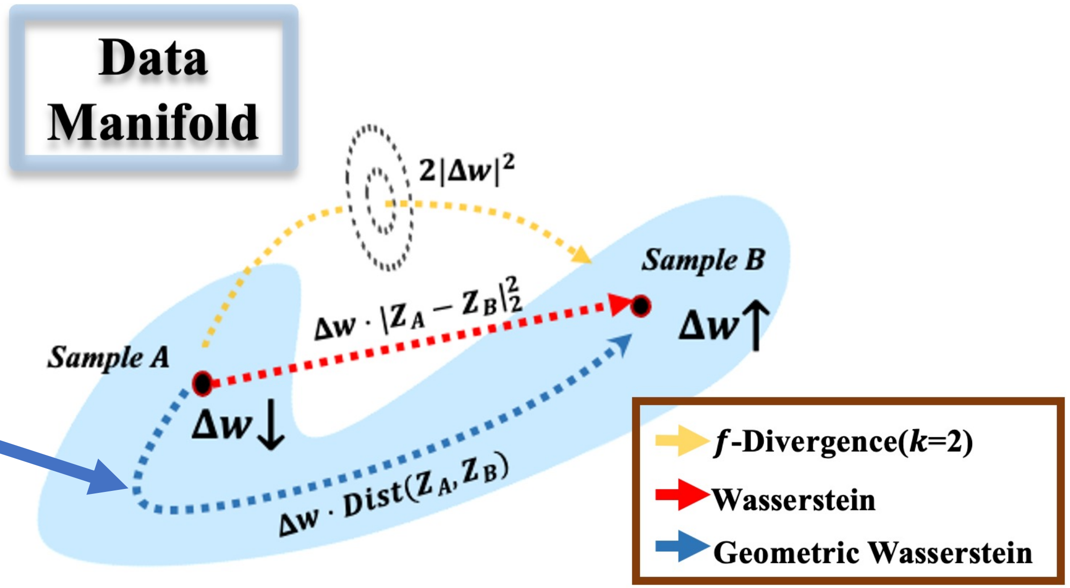
Geometric Wasserstein Distance

Definition 3.1 (Discrete Geometric Wasserstein Distance $\mathcal{GW}_{G_0}(\cdot,\cdot)$ [4]). *Given a finite graph G_0, for any pair of distributions $p^0, p^1 \in \mathscr{P}_o(G_0)$, define the Geometric Wasserstein Distance:*

$$\mathcal{GW}^2_{G_0}(p^0, p^1) := \inf_v \left\{ \int_0^1 \frac{1}{2} \sum_{(i,j)\in E} \kappa_{ij}(p) v_{ij}^2 dt \;\middle|\; \frac{dp}{dt} + div_{G_0}(pv) = 0, p(0) = p^0, p(1) = p^1 \right\}, \quad (2)$$

where $v \in \mathbb{R}^{n\times n}$ denotes the velocity field on G_0, p is a continuously differentiable curve $p(t) : [0,1] \to \mathscr{P}_o(G_0)$, and $\kappa_{ij}(p)$ is a pre-defined interpolation function between p_i and p_j.

The density transfers smoothly along the data manifold.



Geometric Wasserstein DRO

- Objective function:

$$\theta^* = \arg\min_{\theta\in\Theta} \sup_{P:\mathcal{GW}^2_{G_0}(\hat{P}_{tr},P)\le\epsilon} \left\{ \mathcal{R}_n(\theta,p) = \sum_{i=1}^{n} p_i \ell(f_\theta(x_i),y_i) - \beta \sum_{i=1}^{n} p_i \log p_i \right\}.$$

- Sample weights updating:

$$\frac{dp_i}{dt} = \sum_{j:(i,j)\in E} w_{ij}\kappa_{ij}(\ell_i - \ell_j) + \beta \sum_{j:(i,j)\in E} w_{ij}\kappa_{ij}(\log p_j - \log p_i),$$

Algorithm 1 Geometric Wasserstein Distributionally Robust Optimization (GDRO)

Input: Training Dataset $D_{tr} = \{(x_i, y_i)\}_{i=1}^n$, learning rate α_θ, gradient flow iterations T, entropy term β, manifold representation G_0 (learned by kNN algorithm from D_{tr}).
Initialization: Sample weights initialized as $(1/n, \ldots, 1/n)^T$. Predictor's parameters initialized as $\theta^{(0)}$.

for $i = 0$ **to** Epochs **do**
  1. Simulate gradient flow for T time steps according to Equation 5~6 to learn an approximate worst-case probability weight p^T.
  2. $\theta^{(i+1)} \leftarrow \theta^{(i)} - \alpha_\theta \nabla_\theta (\sum_i p_i^T \ell_i(\theta))$
end for

Theoretical Properties

- Global Error Rate Bound:

Theorem 3.2 (Global Error Rate Bound). *Given the model parameter θ, denote the approximate worst-case by gradient descent in Equation 6 after time t as $p^t(\theta)$, and $\epsilon(\theta) = \mathcal{GW}^2_{G_0}(\hat{P}_{tr}, p^t(\theta))$ denotes the distance between our approximation p^t and the training distribution $\hat{P}_{tr}$. Then denote the real worst-case distribution within the $\epsilon(\theta)$-radius discrete Geometric Wasserstein-ball as $p^*(\theta)$, that is,*

$$p^*(\theta) = \arg \sup_{p:\mathcal{GW}^2_{G_0}(\hat{P}_{tr},p)\le\epsilon(\theta)} \sum_{i=1}^{n} p_i \ell_i - \beta \sum_{i=1}^{n} p_i \log p_i. \tag{8}$$

Here we derive the bound w.r.t. the error ratio of objective function $R_n(\theta, p)$ (abbr. $\mathcal{R}(p)$). For $\theta \in \Theta$, there exists $C > 0$ such that

$$\text{Error Rate} = \left(\mathcal{R}(p^*) - \mathcal{R}(p^t)\right) / \left(\mathcal{R}(p^*) - \mathcal{R}(\hat{P}_{tr})\right) < e^{-Ct}, \tag{9}$$

- Convergence:

Theorem 3.3 (Convergence of Algorithm 1). *Denote the objective function for the predictor as:*

$$F(\theta) = \sup_{\mathcal{GW}^2_{G_0}(\hat{P}_{tr},p)\le\epsilon(\theta)} \mathcal{R}_n(\theta, p), \tag{10}$$

which is assumed as L-smooth and $\mathcal{R}_n(\theta, p)$ satisfies L_p-smoothness such that $\|\nabla_p \mathcal{R}_n(\theta, p) - \nabla_p \mathcal{R}_n(\theta, p')\|_2 \le L_p \|p - p'\|_2$. $\epsilon(\theta)$ follows the definition in Theorem 3.2. Take a constant $\Delta_F \ge F(\theta^{(0)}) - \inf_\theta F(\theta)$ and set step size as $\alpha = \sqrt{\Delta_F/(LK)}$. For $t \ge T_0$ where T_0 is a constant, denote the upper bound of $\|p^t - p^*\|_2^2$ as γ and train the model for K steps, we have:

$$\frac{1}{K}\mathbb{E}\left[\sum_{k=1}^{K} \|\nabla_\theta F(\theta^{(k)})\|_2^2\right] - \frac{(1 + 2\sqrt{L\Delta_F/K})}{1 - 2\sqrt{L\Delta_F/K}} L_p^2 \gamma \le \frac{2\Delta_F}{\sqrt{\Delta_F K} - 2L\Delta_F}. \tag{11}$$

Experiment: Selection Bias

- Data Generation:

Data Generation The input features $X = [S, U, V]^T \in \mathbb{R}^{10}$ are comprised of stable features $S \in \mathbb{R}^5$, noisy features $U \in \mathbb{R}^4$ and the spurious feature $V \in \mathbb{R}$:

$$S \sim \mathcal{N}(0, 2\mathbb{I}_5) \in \mathbb{R}^5, \quad U \sim \mathcal{N}(0, 2\mathbb{I}_4) \in \mathbb{R}^4, \quad Y = \theta_S^T S + 0.1 \cdot S_1 S_2 S_3 + \mathcal{N}(0, 0.5), \tag{13}$$

$$V \sim \text{Laplace}(\text{sign}(r) \cdot Y, \frac{1}{5 \ln |r|}) \in \mathbb{R}, \tag{14}$$

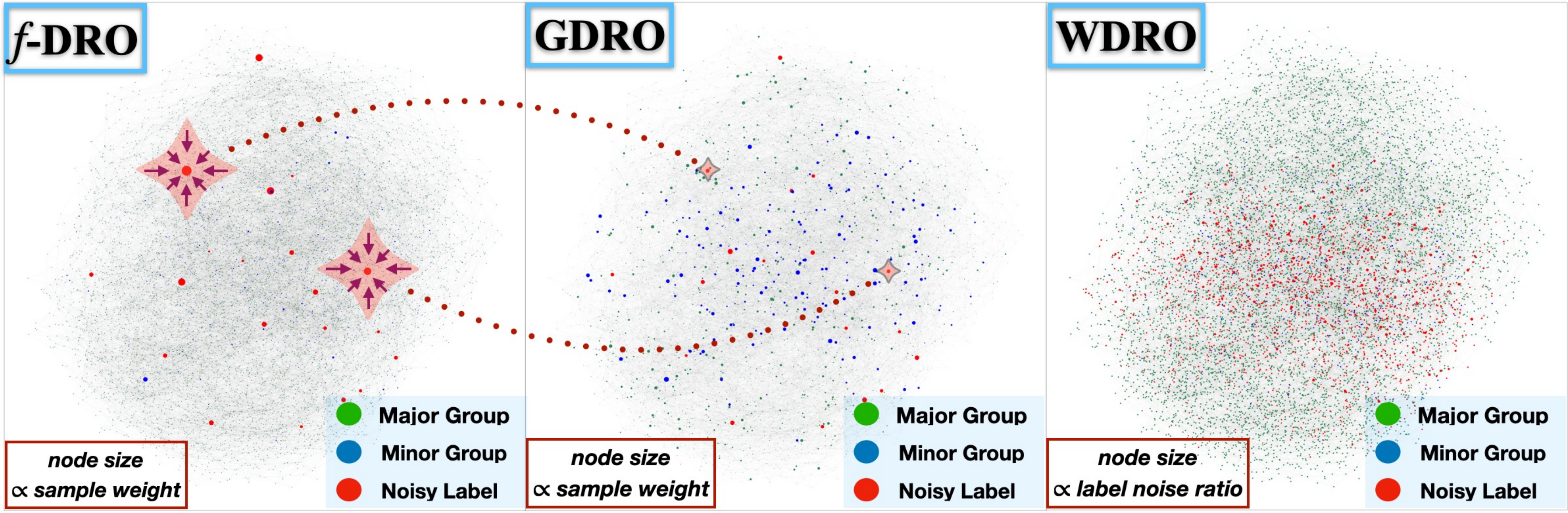
where $\theta_S \in \mathbb{R}^5$ is the coefficient of the true model. $|r| > 1$ is a factor for each sub-population. S are *stable features* with the invariant relationship with Y. U are *noisy features* such that $U \perp Y$.

- Results:

Table 1: Results on the Selection Bias Experiments. We report the root mean square errors.

Simulation 1: regression data without label noises									
	Train(major)	Train(minor)	Test						Parameter
Bias Ratio r	$r = 1.9$	$r = -1.3$	$r = -1.5$	$r = -1.7$	$r = -1.9$	$r = -2.3$	$r = -2.7$	$r = -3.0$	Est Error
ERM	**0.339**	0.876	0.892	0.884	0.864	0.880	0.843	0.888	0.423
WDRO	**0.339**	0.877	0.894	0.885	0.865	0.882	0.844	0.890	0.424
χ^2-DRO	0.411	0.744	0.757	0.741	0.733	0.742	0.714	0.755	0.367
KL-DRO	0.370	0.713	0.728	0.716	0.708	0.713	0.685	0.724	0.319
GDRO	0.493	**0.492**	**0.508**	**0.489**	**0.501**	**0.483**	**0.486**	**0.496**	**0.033**
Simulation 2: regression data with label noises									
ERM	**0.335**	0.845	0.885	0.879	0.874	0.884	0.882	0.876	0.422
WDRO	**0.335**	0.896	0.887	0.880	0.875	0.886	0.884	0.877	0.423
χ^2-DRO	0.375	0.866	0.855	0.856	0.843	0.860	0.854	0.845	0.408
KL-DRO	0.393	0.879	0.868	0.866	0.856	0.876	0.866	0.861	0.391
GDRO	0.542	**0.537**	**0.553**	**0.549**	**0.534**	**0.539**	**0.555**	**0.550**	**0.058**

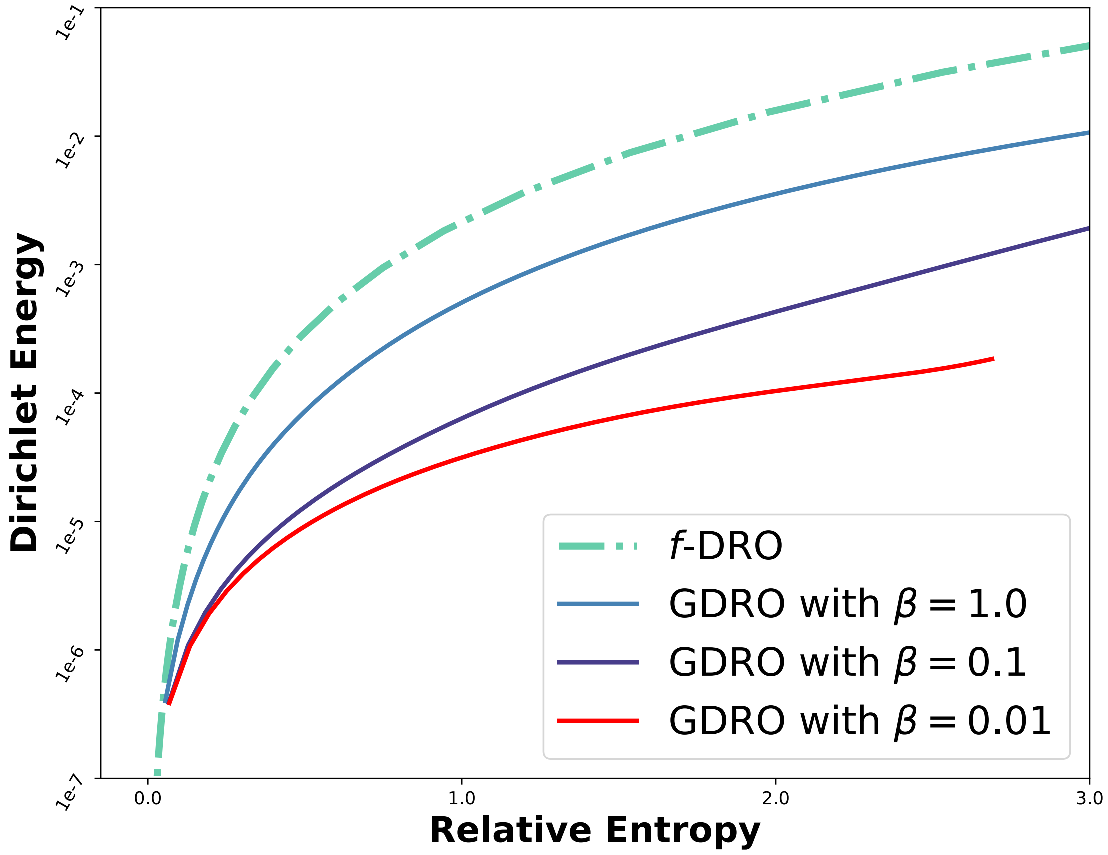
Visualize the Worst-Cases under Label Noises



f-DRO put much more weights on *noisy data points*. ➡ ***Over-Pessimism*** ⬅ **WDRO introduces much more *noisy data points*.**

Smoothness of Sample Weights Along the Manifold

measure the smoothness



much smoother

Jiashuo Liu

- Page: ljsthu.github.io
- Email: liujiashuo77@gmail.com
- Twitter: @liujiashuo77
- WeChat: jiashuo200819