

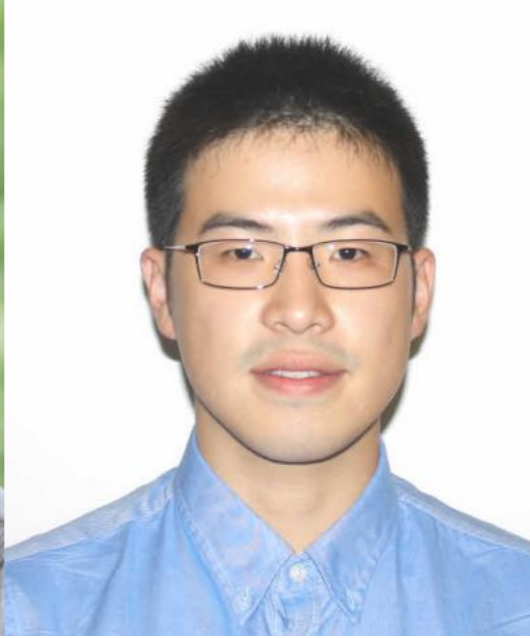
# Robust Statistical Learning and Generative Adversarial Networks

Yuan YAO  
HKUST





**Chao GAO**  
U. Chicago



**Jiye LIU**  
Yale U.

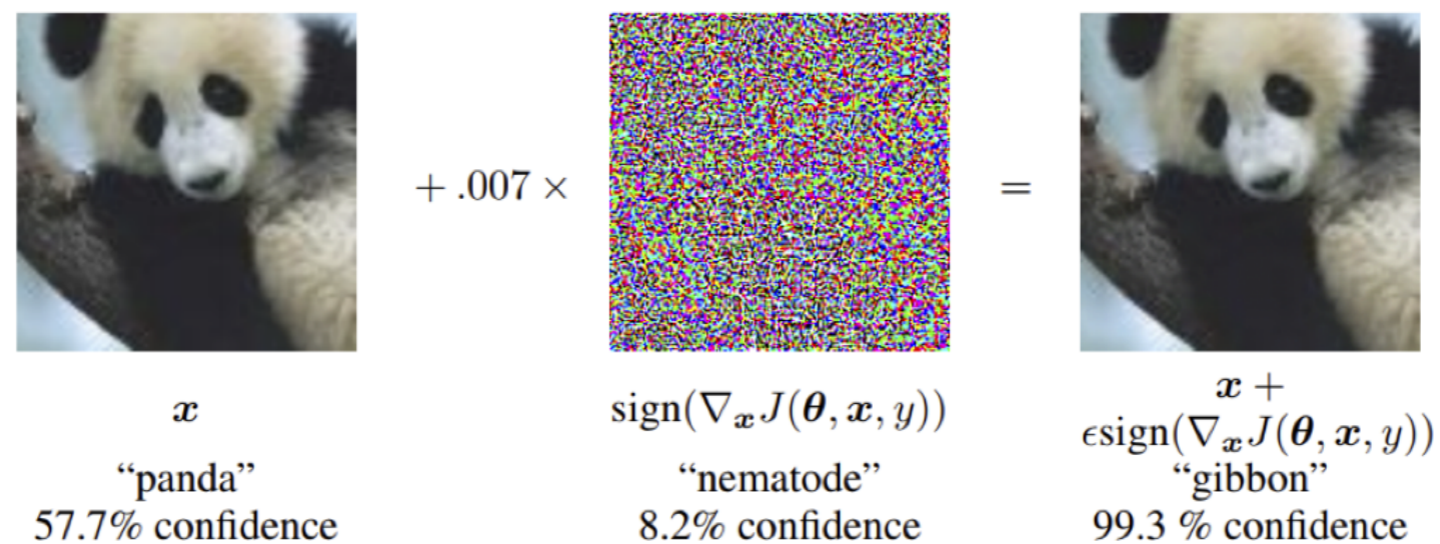


**Weizhi ZHU**  
HKUST

Cryo-EM Applications:

Hanlin GU (HKUST), Yin XIAN,  
Xuhui HUANG (HKUST, UW-Madison)

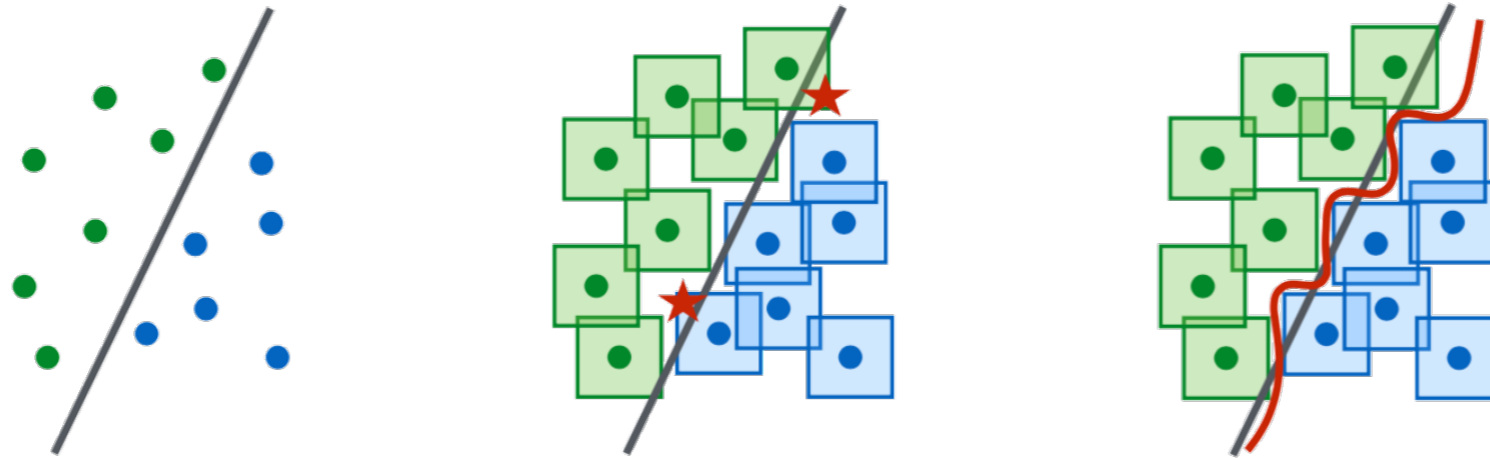
# Deep Learning is Notoriously Not Robust!



[Goodfellow et al., 2014]

- Imperceptible adversarial examples are ubiquitous to fail neural networks
- How can one achieve (adversarial) robustness?

# Robust Optimization



- Traditional training:

$$\min_{\theta} J_n(\theta, \mathbf{z} = (x_i, y_i)_{i=1}^n)$$

- e.g. square or cross-entropy loss as negative log-likelihood of logit models
- Robust optimization (Madry et al. ICLR'2018):

$$\min_{\theta} \max_{\|\epsilon_i\| \leq \delta} J_n(\theta, \mathbf{z} = (x_i + \epsilon_i, y_i)_{i=1}^n)$$

- robust to any distributions, yet computationally hard

# Distributionally Robust Optimization (DRO)

- Distributional Robust Optimization:

$$\min_{\theta} \max_{\epsilon} \mathbb{E}_{\mathbf{z} \sim P_{\epsilon} \in \mathcal{D}} [J_n(\theta, \mathbf{z})]$$

- $\mathcal{D}$  is a set of ambiguous distributions, e.g. Wasserstein ambiguity set

$$\mathcal{D} = \{P_{\epsilon} : W_2(P_{\epsilon}, \text{uniform distribution}) \leq \epsilon\}$$

where DRO may be reduced to **regularized maximum likelihood** estimates (Shafieezadeh-Abadeh, Esfahani, Kuhn, NIPS'2015) that are convex optimizations and tractable

# Wasserstein Distributionally Robust Optimization

Wasserstein-DRO:

$$\min_{\theta} \max_{P_{\epsilon}: W_p(P_{\epsilon}, P_n) \leq \epsilon} \mathbb{E}_{z \sim P_{\epsilon} \in \mathcal{D}} [\ell_{\theta}(z)]$$

where

$$W_p(\mathbb{P}, \mathbb{Q}) = \begin{cases} \left( \min_{\gamma \in \Gamma(\mathbb{P}, \mathbb{Q})} \left\{ \int_{\mathcal{Z} \times \mathcal{Z}} d^p(z, z') \gamma(dz, dz') \right\} \right)^{1/p}, & \text{if } 1 \leq p < \infty, \\ \inf_{\gamma \in \Gamma(\mathbb{P}, \mathbb{Q})} \{ \gamma - \text{esssup}_{\mathcal{Z} \times \mathcal{Z}} d(z, z') \}, & \text{if } p = \infty, \end{cases}$$

- For a broad class of loss functions, Wasserstein-DRO is asymptotically equivalent to the following regularization problem

$$\min_{\theta} \mathbb{E}_{(x,y) \sim P_n} [\ell_{\theta}(x, y)] + \alpha \cdot \left\| \nabla_{(x,y)} \ell_{\theta} \right\|_{\mathbb{P}_n, p_*}$$

where  $p_* = \frac{p}{p-1}$  and the penalty term  $\left\| \nabla_{(x,y)} \ell_{\theta} \right\|_{\mathbb{P}_n, p_*}$  represents the empirical dual  $p_*$ -norm of the gradient of the loss function with respect to the data (input Lipschitz).

- Gao, Kleywegt (2016); Gao, Chen, Kleywegt (2017); Blanchet, Kang, Murphy (2016), et al.

# Certified Adversarial Robustness of Lasso

Theorem (Blanchet, Kang, Murphy (2016))

Consider the cost for  $z = (x, y)$ :

$$c((x, y), (x', y')) = \begin{cases} \|x - x'\|_p^2 & \text{if } y = y' \\ \infty & \text{if } y \neq y' \end{cases}$$

For  $p = \infty$  and linear regression, Wasserstein-DRO is equivalent to SQRT-Lasso:

$$\begin{aligned} & \min_{\beta} \max_{P: W_c(P, P_n) \leq \delta} E_P \left( (Y - \beta^T X)^2 \right) \\ & = \min_{\beta} \left\{ E_{P_n}^{1/2} \left[ (Y - \beta^T X)^2 \right] + \sqrt{\delta} \|\beta\|_{p_*} \right\}^2 \end{aligned}$$

where  $p_* = \frac{p}{p-1} = 1$ .

- Generalized to asymmetric cost (Bregman divergence, Asymmetric Mahalannobis) by [Blanchet, Glynn, Hui, Xie \(2022\)](#)

# TV-neighborhood

- Now how about the TV-uncertainty set?

$$\mathcal{D} = \{P_\epsilon : TV(P_\epsilon, \text{uniform distribution}) \leq \epsilon\}?$$

- an example from *robust statistics* ...



# Huber's Model

$$X_1, \dots, X_n \sim (1 - \epsilon)P_\theta + \epsilon Q$$

contamination proportion

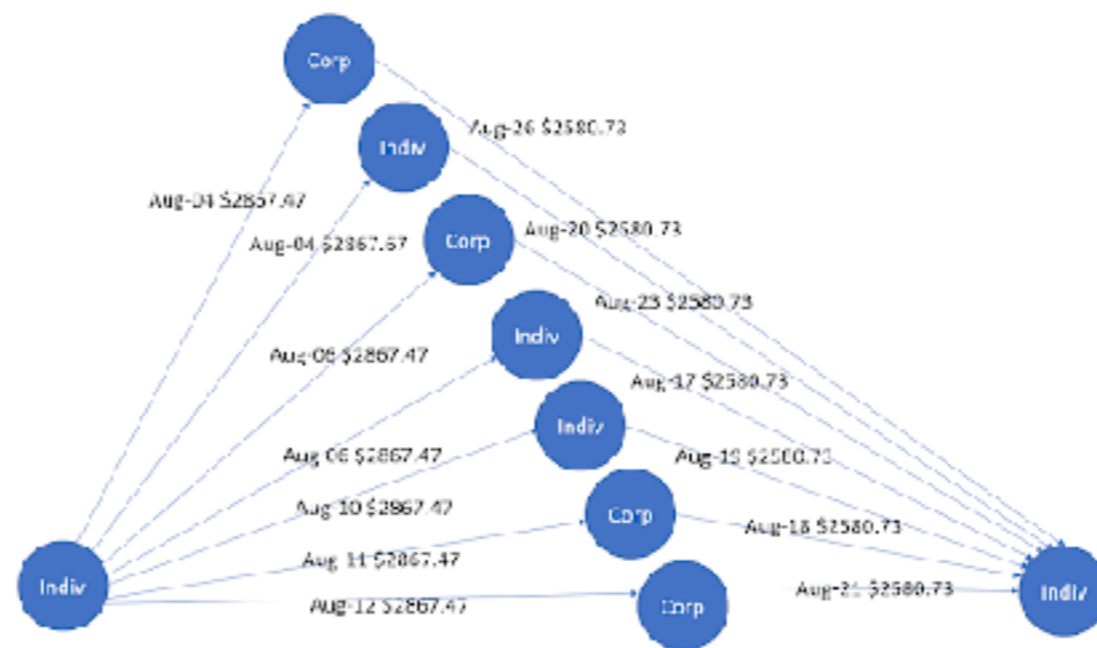
arbitrary contamination

parameter of interest

*[Huber 1964]*

# Example: Financial Fraud

- $P$  represent normal transactions
- $Q$  represent fraudulent transactions, e.g. money laundering, which is sparse and **arbitrarily close** to  $P$
- Finding  $P$  and its dual problem in finding  $Q$ ?



# An Example

$$X_1, \dots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon Q.$$

**how to estimate ?**

# Medians

## 1. Coordinatewise median (M-estimator)

$$\hat{\theta} = (\hat{\theta}_j), \text{ where } \hat{\theta}_j = \text{Median}(\{X_{ij}\}_{i=1}^n);$$

## 2. Tukey's median (1975)

$$\hat{\theta} = \arg \max_{\eta \in \mathbb{R}^p} \min_{\|u\|=1} \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{u^T X_i > u^T \eta\}.$$

# Comparisons

	Coordinatewise Median	Tukey's Median
breakdown point	$1/2$	$1/3$
statistical precision (no contamination)	$\frac{p}{n}$	$\frac{p}{n}$
statistical precision (with contamination)	$\frac{p}{n} + p\epsilon^2$	$\frac{p}{n} + \epsilon^2$ : <b>optimal</b> [Chen-Gao-Ren'15]
computational complexity	Polynomial	<b>NP-hard</b> [Amenta et al. '00]

Note: R-package for Tukey median can not deal with more than **10** dimensions!

[<https://github.com/ChenMengjie/DepthDescent>]

# Depth and Statistical Properties

# Multivariate Location Depth

$$\left\{ \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{u^T X_i > u^T \eta\} \wedge \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{u^T X_i \leq u^T \eta\} \right\}$$
$$= \arg \max_{\eta \in \mathbb{R}^p} \min_{\|u\|=1} \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{u^T X_i > u^T \eta\}.$$

*[Tukey, 1975]*

# Regression Depth

model

$$y|X \sim N(X^T \beta, \sigma^2)$$

embedding

$$Xy|X \sim N(XX^T \beta, \sigma^2 XX^T)$$

projection

$$u^T Xy|X \sim N(u^T XX^T \beta, \sigma^2 u^T XX^T u)$$

$$\left\{ \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{u^T X_i (y_i - X_i^T \eta) > 0\} \wedge \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{u^T X_i (y_i - X_i^T \eta) \leq 0\} \right\}$$



Tukey's depth is not a special case of regression depth.

# Multi-task Regression Depth

$$(X, Y) \in \mathbb{R}^p \times \mathbb{R}^m \sim \mathbb{P}$$

$$B \in \mathbb{R}^{p \times m}$$

population version:

$$\mathcal{D}_{\mathcal{U}}(B, \mathbb{P}) = \inf_{U \in \mathcal{U}} \mathbb{P} \{ \langle U^T X, Y - B^T X \rangle \geq 0 \}$$

empirical version:

$$\mathcal{D}_{\mathcal{U}}(B, \{(X_i, Y_i)\}_{i=1}^n) = \inf_{U \in \mathcal{U}} \frac{1}{n} \sum_{i=1}^n \mathbb{I} \{ \langle U^T X_i, Y_i - B^T X_i \rangle \geq 0 \}$$

# Multi-task Regression Depth

$$\mathcal{D}_{\mathcal{U}}(B, \mathbb{P}) = \inf_{U \in \mathcal{U}} \mathbb{P} \{ \langle U^T X, Y - B^T X \rangle \geq 0 \}$$

$$p = 1, X = 1 \in \mathbb{R},$$

$$\mathcal{D}_{\mathcal{U}}(b, \mathbb{P}) = \inf_{u \in \mathcal{U}} \mathbb{P} \{ u^T (Y - b) \geq 0 \}$$

$$m = 1,$$

$$\mathcal{D}_{\mathcal{U}}(\beta, \mathbb{P}) = \inf_{U \in \mathcal{U}} \mathbb{P} \{ u^T X (y - \beta^T X) \geq 0 \}$$

# Statistical Errors of Multi-task Regression Depth

**Estimation Error.** For any  $\delta > 0$ ,

$$\sup_{B \in \mathbb{R}^{p \times m}} |\mathcal{D}(B, \mathbb{P}_n) - \mathcal{D}(B, \mathbb{P})| \leq C \sqrt{\frac{pm}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}},$$

with probability at least  $1 - 2\delta$ .

**Contamination Error.**

$$\sup_{B, Q} |\mathcal{D}(B, (1 - \epsilon)P_{B^*} + \epsilon Q) - \mathcal{D}(B, P_{B^*})| \leq \epsilon$$

# Statistical Optimality of Multi-task Regression Depth

$$(X, Y) \sim P_B$$

$$(X_1, Y_1), \dots, (X_n, Y_n) \sim (1 - \epsilon)P_B + \epsilon Q$$

**Theorem [G17].** For some  $C > 0$ ,

$$\text{Tr}((\hat{B} - B)^T \Sigma (\hat{B} - B)) \leq C \sigma^2 \left( \frac{pm}{n} \vee \epsilon^2 \right),$$

$$\|\hat{B} - B\|_{\text{F}}^2 \leq C \frac{\sigma^2}{\kappa^2} \left( \frac{pm}{n} \vee \epsilon^2 \right),$$

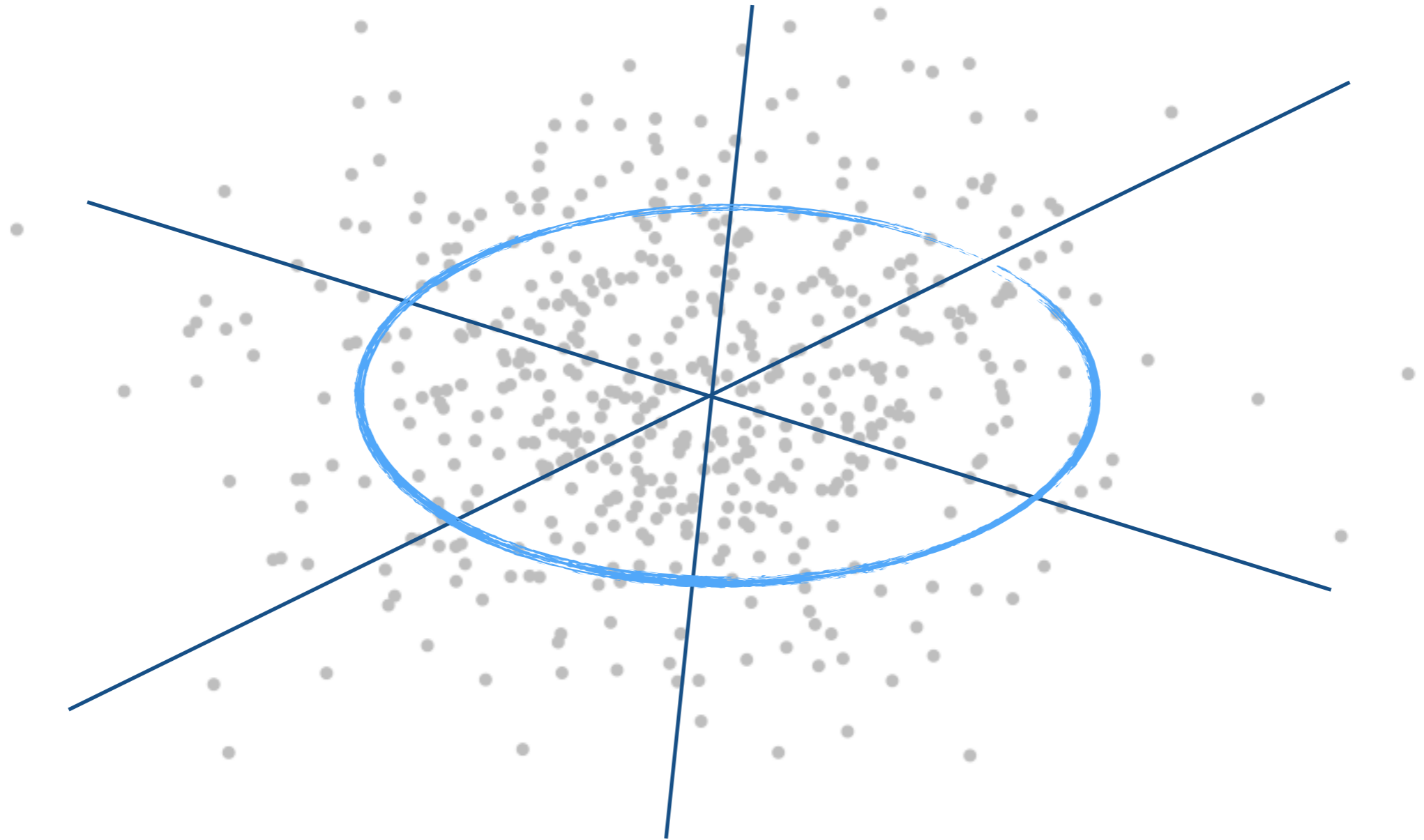
with high probability uniformly over  $B, Q$ .

# Covariance Matrix

$$X_1, \dots, X_n \sim (1 - \epsilon)N(0, \Sigma) + \epsilon Q.$$

**how to estimate ?**

# Covariance Matrix



# Covariance Matrix

$$\mathcal{D}(\Gamma, \{X_i\}_{i=1}^n) = \min_{\|u\|=1} \min \left\{ \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{|u^T X_i|^2 \geq u^T \Gamma u\}, \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{|u^T X_i|^2 < u^T \Gamma u\} \right\}$$

$$\hat{\Gamma} = \arg \max_{\Gamma \succeq 0} \mathcal{D}(\Gamma, \{X_i\}_{i=1}^n) \quad \hat{\Sigma} = \hat{\Gamma} / \beta$$

**Theorem [CGR15].** For some  $C > 0$ ,

$$\|\hat{\Sigma} - \Sigma\|_{\text{op}}^2 \leq C \left( \frac{p}{n} \vee \epsilon^2 \right)$$

with high probability uniformly over  $\Sigma, Q$ .



# Summary

mean	$\ \cdot\ ^2$	$\frac{p}{n} \sqrt{\epsilon^2}$
reduced rank regression	$\ \cdot\ _F^2$	$\frac{\sigma^2}{\kappa^2} \frac{r(p+m)}{n} \sqrt{\frac{\sigma^2}{\kappa^2} \epsilon^2}$
Gaussian graphical model	$\ \cdot\ _{\ell_1}^2$	$\frac{s^2 \log(ep/s)}{n} \sqrt{s\epsilon^2}$
covariance matrix	$\ \cdot\ _{\text{op}}^2$	$\frac{p}{n} \sqrt{\epsilon^2}$
sparse PCA	$\ \cdot\ _F^2$	$\frac{s \log(ep/s)}{n\lambda^2} \sqrt{\frac{\epsilon^2}{\lambda^2}}$

# Computation

# Computational Challenges

$$X_1, \dots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon Q.$$

Lai, Rao, Vempala

Diakonikolas, Kamath, Kane, Li, Moitra, Stewart

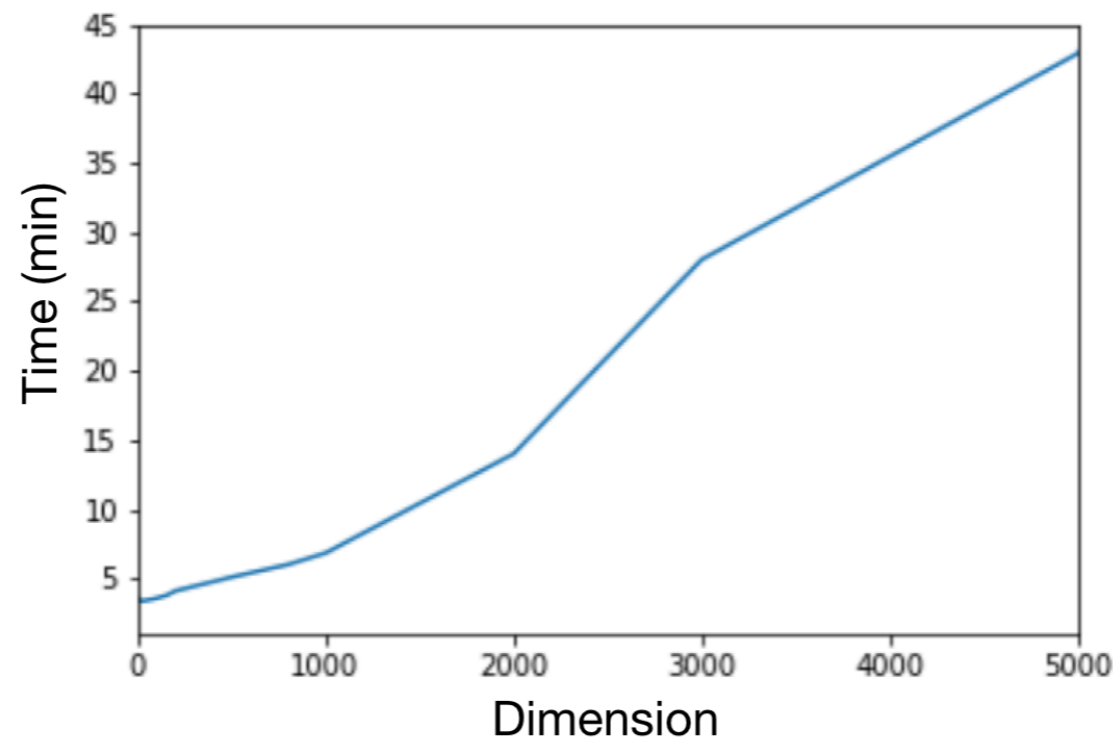
Balakrishnan, Du, Singh

- Polynomial algorithms are proposed [[Diakonikolas et al.'16](#), [Lai et al. 16](#)] of minimax optimal statistical precision
  - needs information on second or higher order of moments
  - some priori knowledge about  $\epsilon$

# Advantages of Tukey Median

A practically good algorithm?

# Generative Adversarial Networks [Goodfellow et al. 2014]



Note: R-package for Tukey median can not deal with more than 10 dimensions [<https://github.com/ChenMengjie/DepthDescent>]

# Robust Learning of Gaussian Distributions

$$(1 - \epsilon)N(0_p, I_p) + \epsilon Q$$

$Q$	$n$	$p$	$\epsilon$	TV-GAN	JS-GAN	Dimension Halving	Iterative Filtering
$N(0.5 * 1_p, I_p)$	50,000	100	.2	<b>0.0953 (0.0064)</b>	0.1144 (0.0154)	0.3247 (0.0058)	0.1472 (0.0071)
$N(0.5 * 1_p, I_p)$	5,000	100	.2	<b>0.1941 (0.0173)</b>	0.2182 (0.0527)	0.3568 (0.0197)	0.2285 (0.0103)
$N(0.5 * 1_p, I_p)$	50,000	200	.2	<b>0.1108 (0.0093)</b>	0.1573 (0.0815)	0.3251 (0.0078)	0.1525 (0.0045)
$N(0.5 * 1_p, I_p)$	50,000	100	.05	0.0913 (0.0527)	0.1390 (0.0050)	0.0814 (0.0056)	<b>0.0530 (0.0052)</b>
$N(5 * 1_p, I_p)$	50,000	100	.2	2.7721 (0.1285)	<b>0.0534 (0.0041)</b>	0.3229 (0.0087)	0.1471 (0.0059)
$N(0.5 * 1_p, \Sigma)$	50,000	100	.2	0.1189 (0.0195)	<b>0.1148 (0.0234)</b>	0.3241 (0.0088)	0.1426 (0.0113)
Cauchy( $0.5 * 1_p$ )	50,000	100	.2	0.0738 (0.0053)	<b>0.0525 (0.0029)</b>	0.1045 (0.0071)	0.0633 (0.0042)

**Table:** Comparison of various robust mean estimation methods. The smallest error of each case is highlighted in bold.

- *Dimension Halving:* [Lai et al.'16]  
<https://github.com/kal2000/AgnosticMeanAndCovarianceCode>.
- *Iterative Filtering:* [Diakonikolas et al.'17]  
<https://github.com/hoonose/robust-filter>.

# Robust Learning of Cauchy Distributions

Table 4: Comparison of various methods of robust location estimation under Cauchy distributions. Samples are drawn from  $(1 - \epsilon)\text{Cauchy}(0_p, I_p) + \epsilon Q$  with  $\epsilon = 0.2, p = 50$  and various choices of  $Q$ . Sample size: 50,000. Discriminator net structure: 50-50-25-1. Generator  $g_\omega(\xi)$  structure: 48-48-32-24-12-1 with absolute value activation function in the output layer.

Contamination $Q$	JS-GAN ( $G_1$ )	JS-GAN ( $G_2$ )	Dimension Halving	Iterative Filtering
Cauchy( $1.5 * 1_p, I_p$ )	<b>0.0664 (0.0065)</b>	0.0743 (0.0103)	0.3529 (0.0543)	0.1244 (0.0114)
Cauchy( $5.0 * 1_p, I_p$ )	<b>0.0480 (0.0058)</b>	0.0540 (0.0064)	0.4855 (0.0616)	0.1687 (0.0310)
Cauchy( $1.5 * 1_p, 5 * I_p$ )	0.0754 (0.0135)	<b>0.0742 (0.0111)</b>	0.3726 (0.0530)	0.1220 (0.0112)
Normal( $1.5 * 1_p, 5 * I_p$ )	<b>0.0702 (0.0064)</b>	0.0713 (0.0088)	0.3915 (0.0232)	0.1048 (0.0288)

- *Dimension Halving*: [Lai et al.'16]  
<https://github.com/kal2000/AgnosticMeanAndCovarianceCode>.
- *Iterative Filtering*: [Diakonikolas et al.'17]  
<https://github.com/hoonose/robust-filter>.



# f-GAN

Given a strictly convex function  $f$  that satisfies  $f(1) = 0$ , the  $f$ -divergence between two probability distributions  $P$  and  $Q$  is defined by

$$D_f(P\|Q) = \int f\left(\frac{p}{q}\right) dQ. \quad (8)$$

Let  $f^*$  be the convex conjugate of  $f$ . A variational lower bound of (8) is

$$D_f(P\|Q) \geq \sup_{T \in \mathcal{T}} [\mathbb{E}_P T(X) - \mathbb{E}_Q f^*(T(X))]. \quad (9)$$

where equality holds whenever the class  $\mathcal{T}$  contains the function  $f'(p/q)$ .

[Nowozin-Cseke-Tomioka'16]  $f$ -GAN minimizes the variational lower bound (9)

$$\hat{P} = \arg \min_{Q \in \mathcal{Q}} \sup_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^n T(X_i) - \mathbb{E}_Q f^*(T(X)) \right]. \quad (10)$$

with i.i.d. observations  $X_1, \dots, X_n \sim P$ .

# From f-GAN to Tukey's Median: f-learning (GLYZ'18)

Consider the special case


$$\mathcal{T} = \left\{ f' \left( \frac{\tilde{q}}{q} \right) : \tilde{q} \in \tilde{\mathcal{Q}} \right\}. \quad (11)$$

which is tight if  $P \in \tilde{\mathcal{Q}}$ . The sample version leads to the following  $f$ -learning

$$\hat{P} = \arg \min_{Q \in \mathcal{Q}} \sup_{\tilde{Q} \in \tilde{\mathcal{Q}}} \left[ \frac{1}{n} \sum_{i=1}^n f' \left( \frac{\tilde{q}(X_i)}{q(X_i)} \right) - \mathbb{E}_Q f^* \left( f' \left( \frac{\tilde{q}(X)}{q(X)} \right) \right) \right]. \quad (12)$$

- If  $f(x) = x \log x$ ,  $\mathcal{Q} = \tilde{\mathcal{Q}}$ , (12)  $\Rightarrow$  **Maximum Likelihood Estimate**
- If  $f(x) = (x - 1)_+$ , then  $D_f(P \| Q) = \frac{1}{2} \int |p - q|$  is the TV-distance,  $f^*(t) = t \mathbb{I}\{0 \leq t \leq 1\}$ ,  $f$ -GAN  $\Rightarrow$  **TV-GAN**
  - $\mathcal{Q} = \{N(\eta, I_p) : \eta \in \mathbb{R}^p\}$  and  $\tilde{\mathcal{Q}} = \{N(\tilde{\eta}, I_p) : \|\tilde{\eta} - \eta\| \leq r\}$ , (12)  $\xrightarrow{r \rightarrow 0}$  **Tukey's Median**

# TV-GAN

$$\hat{\theta} = \operatorname{argmin}_{\eta} \sup_{w,b} \left[ \frac{1}{n} \sum_{i=1}^n \frac{1}{1 + e^{-w^T X_i - b}} - E_{\eta} \frac{1}{1 + e^{-w^T X - b}} \right]$$


**logistic regression classifier**

**Theorem [GLYZ18].** For some  $C > 0$ ,

$$\|\hat{\theta} - \theta\|^2 \leq C \left( \frac{p}{n} \vee \epsilon^2 \right)$$

with high probability uniformly over  $\theta \in \mathbb{R}^p, Q$ .

# TV-GAN

## rugged landscape!

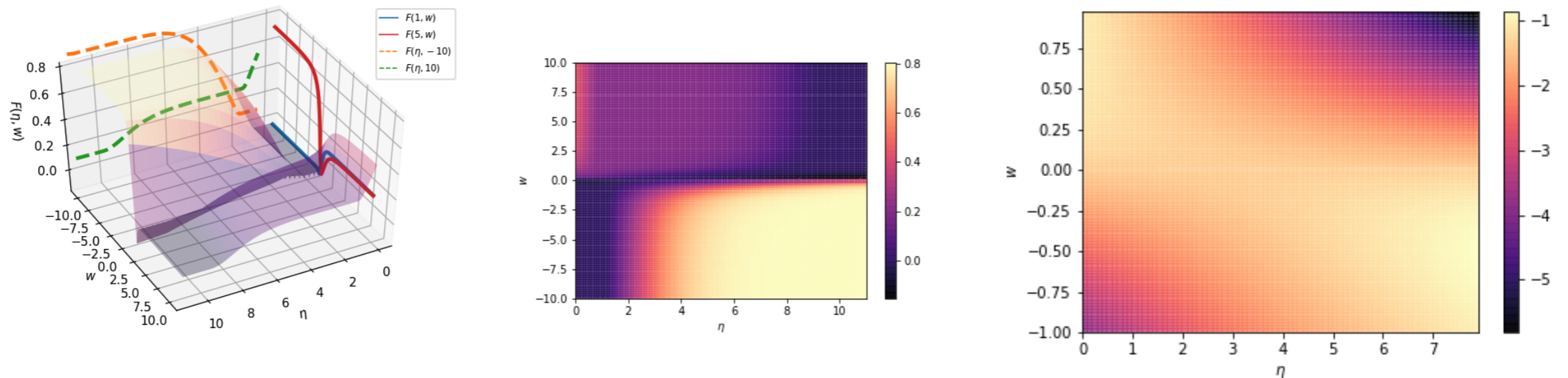


Figure 1: Landscape of TV-GAN objective function  $F(\eta, w) = \sup_b [E_P \text{sigmoid}(wX + b) - E_{N(\eta, 1)} \text{sigmoid}(wX + b)]$ , where  $b$  is maximized out for visualization. Samples are drawn from  $P = (1 - \epsilon)N(1, 1) + \epsilon N(10, 1)$  with  $\epsilon = 0.2$ . Left: a surface plot of  $F(\eta, w)$ . The solid curves are marginal functions for fixed  $\eta$ 's:  $F(1, w)$  (red) and  $F(5, w)$  (blue), and the dash curves are marginal functions for fixed  $w$ 's:  $F(\eta, -10)$  (orange) and  $F(\eta, 10)$  (green). Right: a heatmap of  $F(\eta, w)$ . It is clear that  $\tilde{F}(w) = F(\eta, w)$  has two local maxima for a given  $\eta$ , achieved at  $w = +\infty$  and  $w = -\infty$ . In fact, the global maximum for  $\tilde{F}(w)$  has a phase transition from  $w = +\infty$  to  $w = -\infty$  as  $\eta$  grows. For example, the maximum is achieved at  $w = +\infty$  when  $\eta = 1$  (blue solid) and is achieved at  $w = -\infty$  when  $\eta = 5$  (red solid). Unfortunately, even if we initialize with  $\eta_0 = 1$  and  $w_0 > 0$ , gradient ascents on  $\eta$  will only increase the value of  $\eta$  (green dash), and thus as long as the discriminator cannot reach the global maximizer,  $w$  will be stuck in the positive half space  $\{w : w > 0\}$  and further increase the value of  $\eta$ .

# JS-GAN

[Goodfellow et al. 2014] For  $f(x) = x \log x - (x + 1) \log \frac{x+1}{2}$ ,

$$\hat{\theta} = \arg \min_{\eta \in \mathbb{R}^p} \max_{D \in \mathcal{D}} \left[ \frac{1}{n} \sum_{i=1}^n \log D(X_i) + \mathbb{E}_{\mathcal{N}(\eta, I_p)} \log(1 - D(X)) \right] + \log 4. \quad (15)$$

What are  $\mathcal{D}$ , the class of discriminators?

- Single layer (no hidden layer):

$$\mathcal{D} = \left\{ D(x) = \text{sigmoid}(w^T x + b) : w \in \mathbb{R}^p, b \in \mathbb{R} \right\}$$

- One-hidden or Multiple layer:

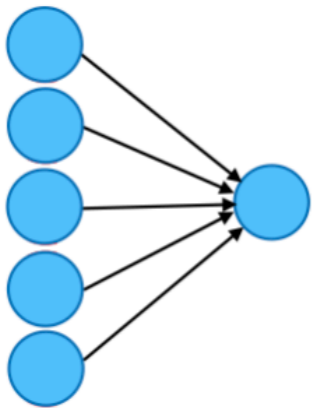
$$\mathcal{D} = \left\{ D(x) = \text{sigmoid}(w^T g(X)) \right\}$$

# Deep JS-GAN

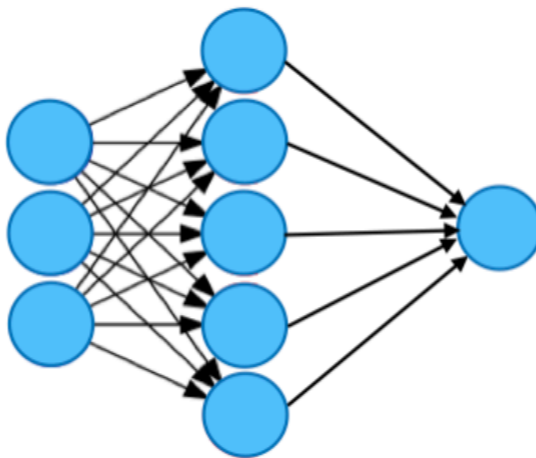
$$\hat{\theta} = \operatorname{argmin}_{\eta \in \mathbb{R}^p} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^n \log T(X_i) + E_{\eta} \log(1 - T(X)) \right] + \log 4$$

**numerical  
experiment**

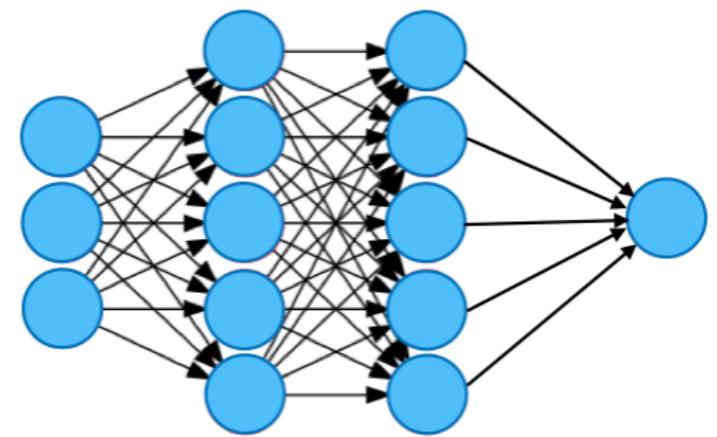
$$X_1, \dots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon N(\tilde{\theta}, I_p)$$



$$\hat{\theta} \approx (1 - \epsilon)\theta + \epsilon\tilde{\theta}$$



$$\hat{\theta} \approx \theta$$



$$\hat{\theta} \approx \theta$$



# JS-GAN

**A classifier with hidden layers leads to robustness. Why?**

$$\text{JS}_g(\mathbb{P}, \mathbb{Q}) = \max_{w \in \mathbb{R}^d} \left[ \mathbb{P} \log \frac{1}{1 + e^{-w^T g(X)}} + \mathbb{Q} \log \frac{1}{1 + e^{w^T g(X)}} \right] + \log 4.$$

**Proposition.**

$$\text{JS}_g(\mathbb{P}, \mathbb{Q}) = 0 \iff \mathbb{P}g(X) = \mathbb{Q}g(X)$$

# JS-GAN

$$\hat{\theta} = \operatorname{argmin}_{\eta \in \mathbb{R}^p} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^n \log T(X_i) + E_{\eta} \log(1 - T(X)) \right] + \log 4$$

**Theorem [GLYZ18].** For a neural network class  $\mathcal{T}$  with at least one hidden layer and appropriate regularization, we have

$$\|\hat{\theta} - \theta\|^2 \lesssim \begin{cases} \frac{p}{n} + \epsilon^2 & \text{(indicator/sigmoid/ramp)} \\ \frac{p \log p}{n} + \epsilon^2 & \text{(ReLU+sigmoid features)} \end{cases}$$

with high probability uniformly over  $\theta \in \mathbb{R}^p, Q$ .



# JS-GAN: Adaptation to Unknown Covariance

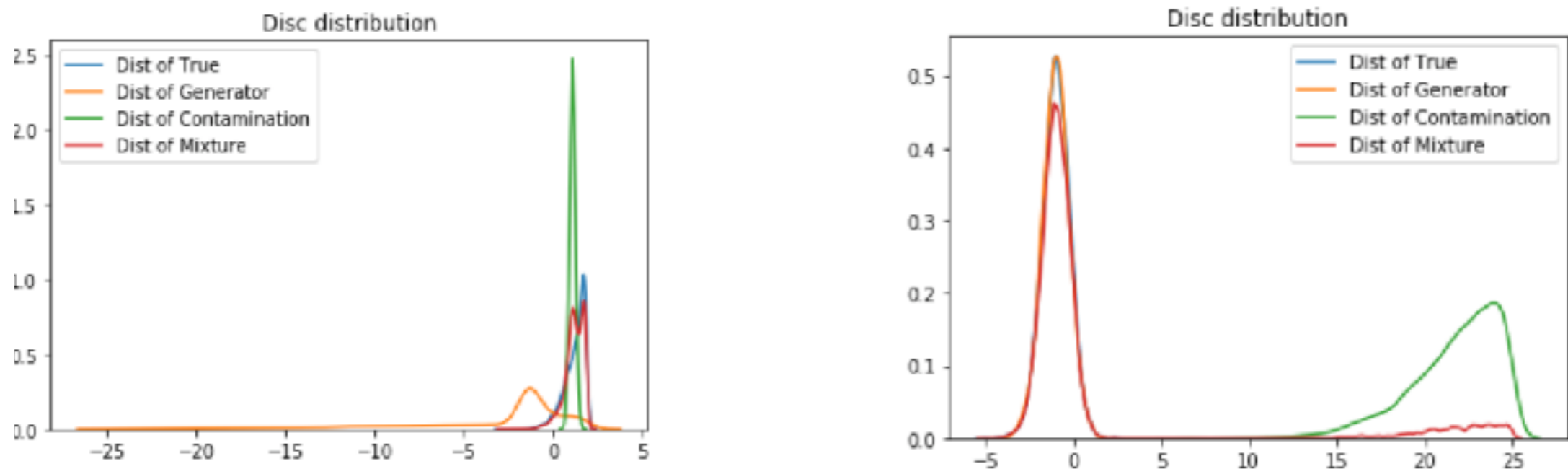
**unknown  
covariance?**

$$X_1, \dots, X_n \sim (1 - \epsilon)N(\theta, \Sigma) + \epsilon Q$$

$$(\hat{\theta}, \hat{\Sigma}) = \operatorname{argmin}_{\eta, \Gamma} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^n \log T(X_i) + \mathbb{E}_{X \sim N(\eta, \Gamma)} \log(1 - T(X)) \right]$$

no need to change the discriminator class

# Discriminator identifies outliers



$$(1 - \epsilon)N(0_p, I_p) + \epsilon Q$$
$$N(5 * 1_p, I_p)$$

- Discriminator helps identify outliers or contaminated samples
- Generator fits uncontaminated portion of true samples

# Generalization

## Strong Contamination model:

$X_1, \dots, X_n \stackrel{iid}{\sim} P$  for some  $P$  satisfying  $\text{TV}(P, E(\theta, \Sigma, H)) \leq \epsilon$

$$(\hat{\theta}, \hat{\Sigma}, \hat{H}) = \underset{\eta \in \mathbb{R}^p, \Gamma \in \mathcal{E}_p(M), H \in \mathcal{H}(M')}{\text{argmin}} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^n S(T(X_i), 1) + \mathbb{E}_{X \sim E(\eta, \Gamma, G)} S(T(X), 0) \right]$$

- ▶ We are going to replace the log likelihoods in JS-GAN by some scoring functions

$$\log t \mapsto S(t, 1) : [0, 1] \rightarrow \mathbb{R}$$

$$\log(1 - t) \mapsto S(t, 0) : [0, 1] \rightarrow \mathbb{R}$$

that map the probability (likelihood) to some real numbers.

# Fisher Consistency: Proper Scoring Rule

- ▶ With a Bernoulli experiment of probability  $p$  observing 1, define the expected score

$$S(t, p) = pS(t, 1) + (1 - p)S(t, 0)$$

- ▶ Like likelihood functions, as a function of  $t$ , we hope that  $S(t, p)$  is maximized at  $t = p$

$$\max_t S(t, p) = S(p, p) =: G(p)$$

- ▶ Such a score is called **Proper Scoring Rule**.

# Savage Representation of Proper Scoring Rule

Lemma (Savage representation)

- ▶ For a proper scoring rule  $S(t, p)$ :
  - $G(t) = S(t, t)$  is convex
  - $S(t, 0) = G(t) - tG'(t)$
  - $S(t, 1) = G(t) + (1 - t)G'(t)$
  - $S(t, p) = pS(t, 1) + (1 - p)S(t, 0) = G(t) + G'(t)(p - t)$

# Divergence

$$D_{\mathcal{T}}(P, Q) = \max_{T \in \mathcal{T}} \left[ \frac{1}{2} \mathbb{E}_{X \sim P} S(T(X), 1) + \frac{1}{2} \mathbb{E}_{X \sim Q} S(T(X), 0) \right] - G(1/2),$$

**Proposition 1** *Given any regular proper scoring rule  $\{S(\cdot, 1), S(\cdot, 0)\}$  and any class  $\mathcal{T} \ni \{\frac{1}{2}\}$ ,  $D_{\mathcal{T}}(P, Q)$  is a divergence function, and*

$$D_{\mathcal{T}}(P, Q) \leq D_f \left( P \parallel \frac{1}{2}P + \frac{1}{2}Q \right), \quad (4)$$

where  $f(t) = G(t/2) - G(1/2)$ . Moreover, whenever  $\mathcal{T} \ni \frac{dP}{dP+dQ}$ , the inequality above becomes an equality.

- ▶ A scoring rule  $S$  is *regular* if both  $S(\cdot, 0)$  and  $S(\cdot, 1)$  are real-valued, except possibly that  $S(0, 1) = -\infty$  or  $S(1, 0) = -\infty$ .

# Example 1: Log Score and JS-GAN

1. *Log Score*. The log score is perhaps the most commonly used rule because of its various intriguing properties [31]. The scoring rule with  $S(t, 1) = \log t$  and  $S(t, 0) = \log(1 - t)$  is regular and strictly proper. Its Savage representation is given by the convex function  $G(t) = t \log t + (1 - t) \log(1 - t)$ , which is interpreted as the negative Shannon entropy of Bernoulli( $t$ ). The corresponding divergence function  $D_{\mathcal{T}}(P, Q)$ , according to Proposition 3.1, is a variational lower bound of the Jensen-Shannon divergence

$$\text{JS}(P, Q) = \frac{1}{2} \int \log \left( \frac{dP}{dP + dQ} \right) dP + \frac{1}{2} \int \log \left( \frac{dQ}{dP + dQ} \right) dQ + \log 2.$$

Its sample version (13) is the original GAN proposed by [25] that is widely used in learning distributions of images.

# Example 2: Zero-One Score and TV-GAN

2. *Zero-One Score.* The zero-one score  $S(t, 1) = 2\mathbb{I}\{t \geq 1/2\}$  and  $S(t, 0) = 2\mathbb{I}\{t < 1/2\}$  is also known as the misclassification loss. This is a regular proper scoring rule but not strictly proper. The induced divergence function  $D_{\mathcal{T}}(P, Q)$  is a variational lower bound of the total variation distance

$$\text{TV}(P, Q) = P\left(\frac{dP}{dQ} \geq 1\right) - Q\left(\frac{dP}{dQ} \geq 1\right) = \frac{1}{2} \int |dP - dQ|.$$

The sample version (13) is recognized as the TV-GAN that is extensively studied by [21] in the context of robust estimation.



# Example 3: Quadratic Score and LS-GAN

3. *Quadratic Score*. Also known as the Brier score [6], the definition is given by  $S(t, 1) = -(1 - t)^2$  and  $S(t, 0) = -t^2$ . The corresponding convex function in the Savage representation is given by  $G(t) = -t(1 - t)$ . By Proposition 2.1, the divergence function (3) induced by this regular strictly proper scoring rule is a variational lower bound of the following divergence function,

$$\Delta(P, Q) = \frac{1}{8} \int \frac{(dP - dQ)^2}{dP + dQ},$$

known as the triangular discrimination. The sample version (5) belongs to the family of least-squares GANs proposed by [39].

# Example 4: Boosting Score

4. *Boosting Score.* The boosting score was introduced by [7] with  $S(t, 1) = -\left(\frac{1-t}{t}\right)^{1/2}$  and  $S(t, 0) = -\left(\frac{t}{1-t}\right)^{1/2}$  and has an connection to the AdaBoost algorithm. The corresponding convex function in the Savage representation is given by  $G(t) = -2\sqrt{t(1-t)}$ . The induced divergence function  $D_{\mathcal{T}}(P, Q)$  is thus a variational lower bound of the squared Hellinger distance

$$H^2(P, Q) = \frac{1}{2} \int \left( \sqrt{dP} - \sqrt{dQ} \right)^2.$$

# Example 5: Beta Score and new GANs

5. *Beta Score.* A general Beta family of proper scoring rules was introduced by [7] with  $S(t, 1) = - \int_t^1 c^{\alpha-1} (1-c)^\beta dc$  and  $S(t, 0) = - \int_0^t c^\alpha (1-c)^{\beta-1} dc$  for any  $\alpha, \beta > -1$ . The log score, the quadratic score and the boosting score are special cases of the Beta score with  $\alpha = \beta = 0$ ,  $\alpha = \beta = 1$ ,  $\alpha = \beta = -1/2$ . The zero-one score is a limiting case of the Beta score by letting  $\alpha = \beta \rightarrow \infty$ . Moreover, it also leads to asymmetric scoring rules with  $\alpha \neq \beta$ .

# Smooth Proper Scores

## Assumption (Smooth Proper Scoring Rules)

We assume that

- ▶  $G^{(2)}(1/2) > 0$  and  $G^{(3)}(t)$  is continuous at  $t = 1/2$ ;
- ▶ Moreover, there is a universal constant  $c_0 > 0$ , such that  $2G^{(2)}(1/2) \geq G^{(3)}(1/2) + c_0$ .
  - The condition  $2G^{(2)}(1/2) \geq G^{(3)}(1/2) + c_0$  is automatically satisfied by a symmetric scoring rule, because  $S(t, 1) = S(1 - t, 0)$  immediately implies that  $G^{(3)}(1/2) = 0$ .
  - For the Beta score with  $S(t, 1) = -\int_t^1 c^{\alpha-1}(1-c)^\beta dc$  and  $S(t, 0) = -\int_0^t c^\alpha(1-c)^{\beta-1} dc$  for any  $\alpha, \beta > -1$ , it is easy to check that such a  $c_0$  (only depending on  $\alpha, \beta$ ) exists as long as  $|\alpha - \beta| < 1$ .


# Statistical Optimality

**Theorem [GYZ19].** For a neural network class  $\mathcal{T}$  with at least one hidden layer and appropriate regularization, we have


$$\begin{aligned}\|\hat{\theta} - \theta\|^2 &\leq C \left( \frac{p}{n} \vee \epsilon^2 \right), \\ \|\hat{\Sigma} - \Sigma\|_{\text{op}}^2 &\leq C \left( \frac{p}{n} \vee \epsilon^2 \right),\end{aligned}$$

# Application: Cryo-EM image Denoising

**The Nobel Prize in Chemistry 2017**



**NOBEL PRIZE**  
FOR 3D STRUCTURES



**DUBOCHET'S VITRIFICATION METHOD**

- 1 The aqueous biological sample and vitrification medium are mixed.
- 2 The vitrification medium is cooled rapidly to -180°C.
- 3 The vitrification medium is cooled rapidly to -180°C.

**1** Randomly oriented particles are frozen by the vitrification medium, forming a thin layer of ice.

**2** The vitrified particles are imaged by the electron microscope, producing a series of 2D images.

Using thousands of images, the computer generates a high-resolution 3D image.

The computer software then reconstructs the 3D image from the 2D images and generates high-resolution 3D structures.

© Nobel Media. ILL N. Elmehed  
**Jacques Dubochet**  
Prize share: 1/3

© Nobel Media. ILL N. Elmehed  
**Joachim Frank**  
Prize share: 1/3

© Nobel Media. ILL N. Elmehed  
**Richard Henderson**  
Prize share: 1/3





Illustration: W. H. Rindler-Schjerve  
The Royal Society, London, UK

The 2017 Nobel prize in chemistry has been awarded to three scientists (Jacques Dubochet; Joachim Frank from; and Richard Henderson from Switzerland, US & UK, respectively) for developing **cryo-electron microscopy** for the high-resolution structure determination of biomolecules in solution.

# Challenges of Cryo-EM image Denoising

- Low signal vs. high noise
- unknown contamination (e.g. ice and so on)
- conformational heterogeneity

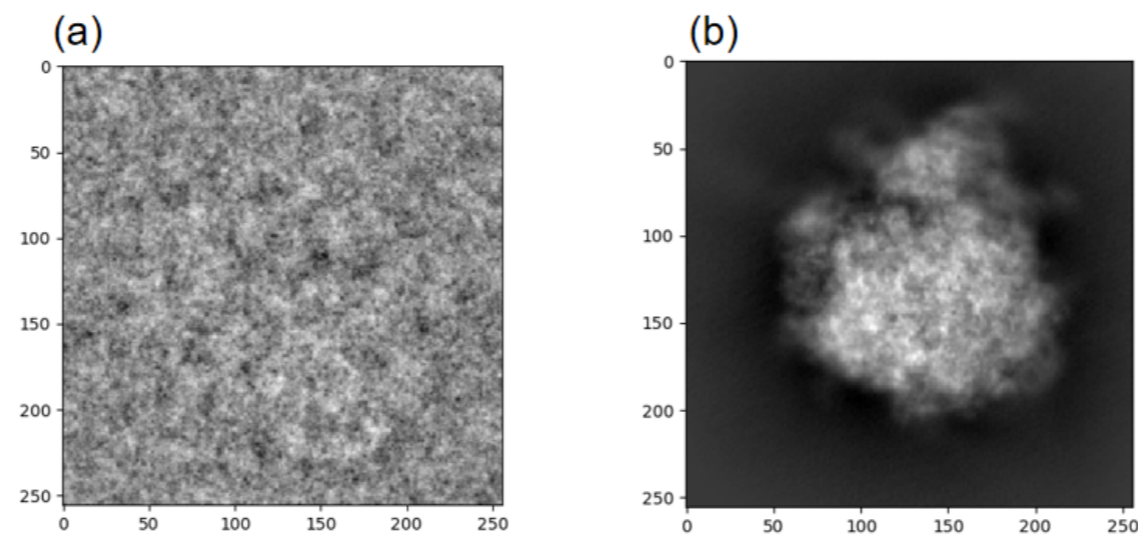


Fig. 1: (a) a noisy Cryo-EM image (b) a reference image

# GAN jointly trained with AutoEncoder

$$\hat{\mathcal{L}}_{GAN}(x, \hat{x}) + \lambda \|x - \hat{x}\|_p^p,$$

- $\beta$ -GAN facilitates Robust denoising against unknown contaminations
- Joint training with  $\ell_p$ -AutoEncoder stabilizes the minimax training of GANs

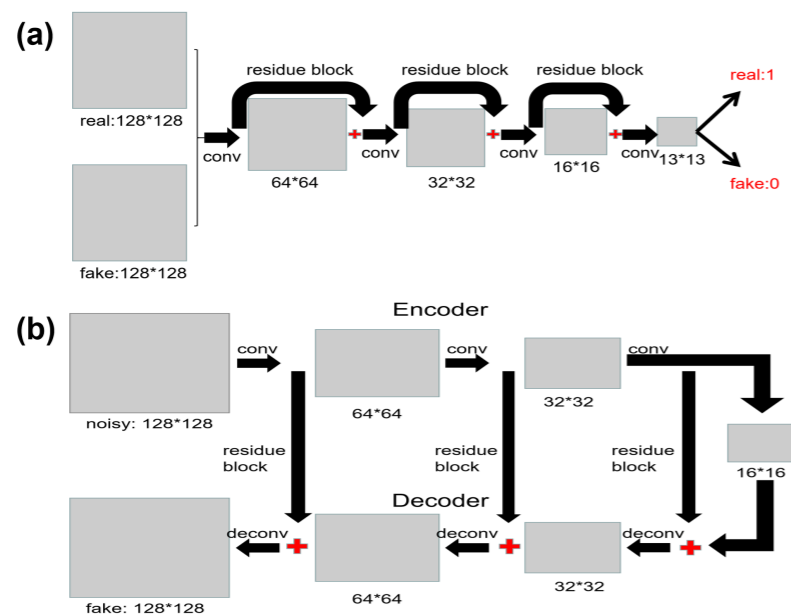


Fig. 4: The architectures of (a) discriminator  $D$  and (b) generator  $G$ , which borrow the residue structure. The input image size ( $128 \times 128$ ) here is adapted to RNAP dataset, while input image size of EMPIAR-10028 dataset is  $256 \times 256$ .

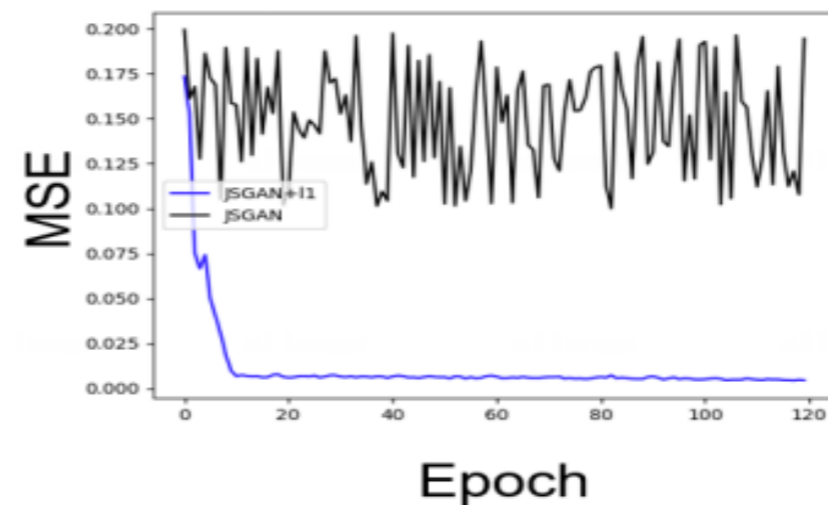


Figure: illustration of training stability of  $\beta$ -GAN and  $\beta$ -GAN+autoencoder. Autoencoder can stabilize the training process of GAN.



# Synthetic RNA Polymerase (RNAP) dataset

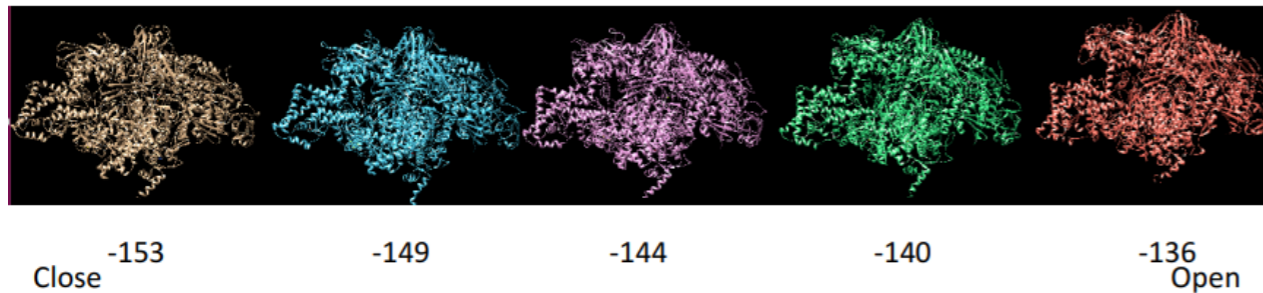


Fig. 3: Five conformations in RNAP heterogeneous dataset, from left to right are close conformation to open conformation of different angles.

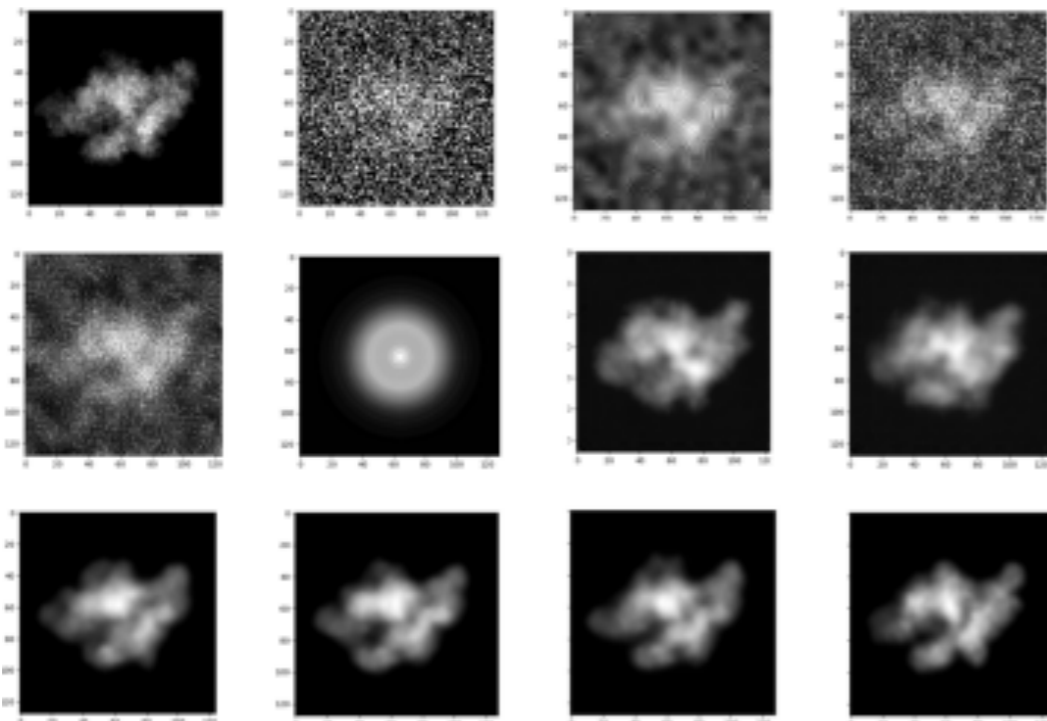


Figure: denoised images with different methods (left to right, top to bottom): clean, noisy, BM3D, KSVD, non-local means, CWF,  $l_1$ -Autoencoder,  $l_2$ -Autoencoder, (1, 1)-GAN+  $l_1$ , (0,0)-GAN+  $l_1$ , (0.5, 0.5)-GAN+  $l_1$ , and WGANgp+  $l_1$ .

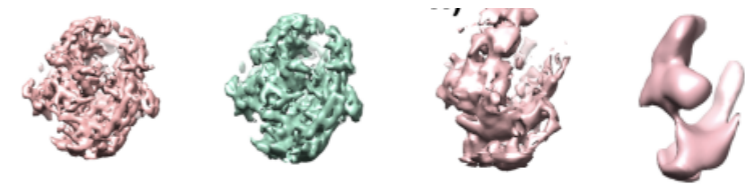


Figure: 3D reconstruction (left to right): clean images, (0.5, 0.5)-GAN+  $l_1$ , (0.5, 0.5)-GAN+  $l_1$  under noise contamination, and  $l_2$ -Autoencoder under noise contamination.

Method	MSE		PSNR	
	SNR=0.1	SNR=0.05	SNR=0.1	SNR=0.05
BM3D	3.52e-2	5.87e-2	14.54	12.13
CWF	2.53e-2	9.28e-3	16.06	20.31
$l_1$ -Autoencoder	3.13e-3	4.02e-3	25.10	23.67
$l_2$ -Autoencoder	3.16e-3	4.23e-3	25.05	23.80
<b>WGANgp+ <math>l_1</math></b>	<b>2.95e-3</b>	<b>4.00e-3</b>	<b>25.42</b>	<b>24.06</b>
<b>(0.5, 0.5)-GAN+ <math>l_1</math></b>	<b>3.01e-3</b>	<b>3.98e-3</b>	<b>25.27</b>	<b>24.07</b>

# EMPIAR-10028 dataset: robust denoising under contaminations

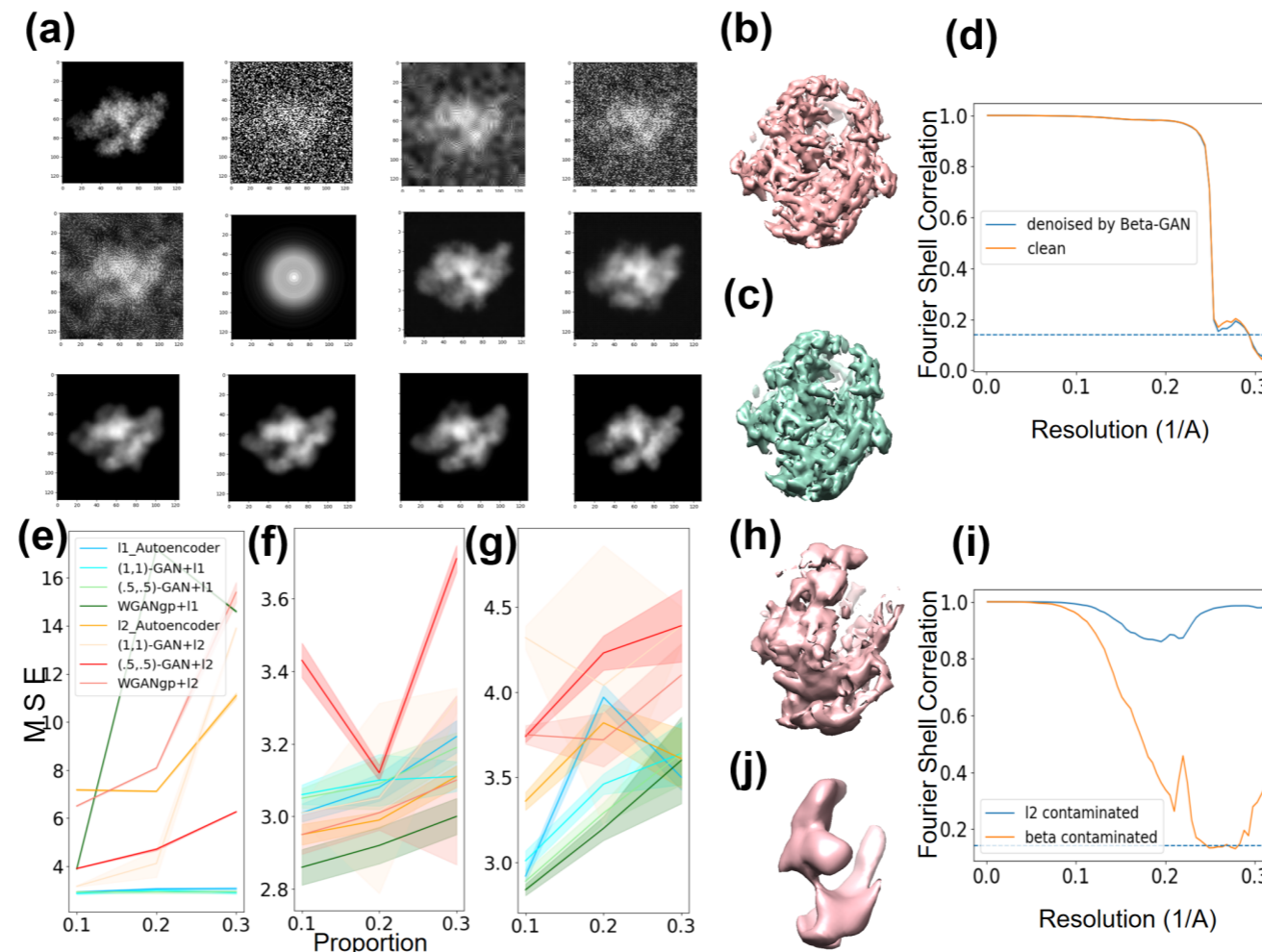


Fig. 5: Results for RNAP dataset. (a) is denoised images in different denoised methods (from left to right, top to bottom): Clean, Noisy, BM3D, KSVD, Non-local means, CWF,  $\ell_1$ -Autoencoder,  $\ell_2$ -Autoencoder,  $(1,1)$ -GAN +  $\ell_1$ ,  $(0,0)$ -GAN +  $\ell_1$ ,  $(.5, .5)$ -GAN +  $\ell_1$  and WGANgp +  $\ell_1$ . (b) and (c) are reconstruction of clean images and  $(.5, .5)$ -GAN +  $\ell_1$  denoised images. (d) is FSC curve of (b) and (c). (e), (f) and (g) are robustness tests of various methods under  $\epsilon \in \{0.1, 0.2, 0.3\}$ -proportion contamination in three types of contamination: (e) Type A: replacing the reference images with random noise; (f) Type B: replacing the noisy images with random noise; (g) Type C: replacing both with random noise. (h) and (j) are reconstructions of images with  $(.5, .5)$ -GAN +  $\ell_1$  and  $\ell_2$ -Autoencoder under type A contamination, respectively, where  $\ell_2$ -Autoencoder totally fails but  $(.5, .5)$ -GAN +  $\ell_1$  is robust. (i) shows FSC curves of (h) and (j).

# Summary

- A family of Generative Adversarial Networks ( $\beta$ -GAN), with a proper network architecture design, may provide statistically optimal robust estimators under TV-ambiguity set of contaminations
  - A minimax-type estimator like Tukey's median, we don't know any M-estimator can provide statistical optimality for arbitrary contamination
  - Discriminators help to identify contaminated outliers while generator adapt to the true distributions
  - Application examples in Robust PCA of S&P500 and Robust Cryo-EM image Denoising

# Reference

- Gao, Liu, Yao, Zhu, Robust Estimation and Generative Adversarial Networks, *ICLR 2019*, <https://arxiv.org/abs/1810.02030>
- Gao, Yao, Zhu, Generative Adversarial Networks for Robust Scatter Estimation: A Proper Scoring Rule Perspective, *Journal of Machine Learning Research*, 21(160):1-48, 2020. <https://arxiv.org/abs/1903.01944>
- Gu, H., Y. Xian, I. Unarta, and Y. Yao, Generative adversarial networks for robust Cryo-EM image denoting. *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, Springer, to appear. <https://arxiv.org/abs/2008.07307>

# Thank You

