# Adversarial Projections for Inverse Problems

Samy Wu Fung* Howard Heaton* Alex Lin*

Stanley Osher Wotao Yin

University of California Los Angeles

*ISU MDL Collective*

October 2, 2020

# Agenda

1. Background
2. Adversarial Projections
3. Numerical Experiments
4. Conclusions

# Agenda

1 Background

2 Adversarial Projections

3 Numerical Experiments

4 Conclusions

# Background

Forward problem is given by

$$b = A(u^\star) + \epsilon,$$

where

- $b \in \mathbb{R}^m$ = measurement data
- $A \colon \mathbb{R}^n \to \mathbb{R}^m$ = forward operator
- $\epsilon \in \mathbb{R}^m$ = measurement noise
- $u^\star \in \mathbb{R}^n$ = true signal

# Background

Forward problem is given by

$$b = A(u^\star) + \epsilon,$$

where

- $b \in \mathbb{R}^m$ = measurement data
- $A \colon \mathbb{R}^n \to \mathbb{R}^m$ = forward operator
- $\epsilon \in \mathbb{R}^m$ = measurement noise
- $u^\star \in \mathbb{R}^n$ = true signal

**Inverse Problem:** recover signal $u^\star$ given noisy measurements $b$

## Inverse Problem

$$\tilde{u} \in \arg\min_{u} \quad \ell(A(u), b) + J(u),$$

where

- $\ell \colon \mathbb{R}^m \times \mathbb{R}^m \to \overline{\mathbb{R}}$ is a fidelity term (e.g., Least Squares)
- $J \colon \mathbb{R}^m \times \mathbb{R}^m \to \overline{\mathbb{R}}$ is a regularizer (e.g., TV, L1, Learned)

## Inverse Problem

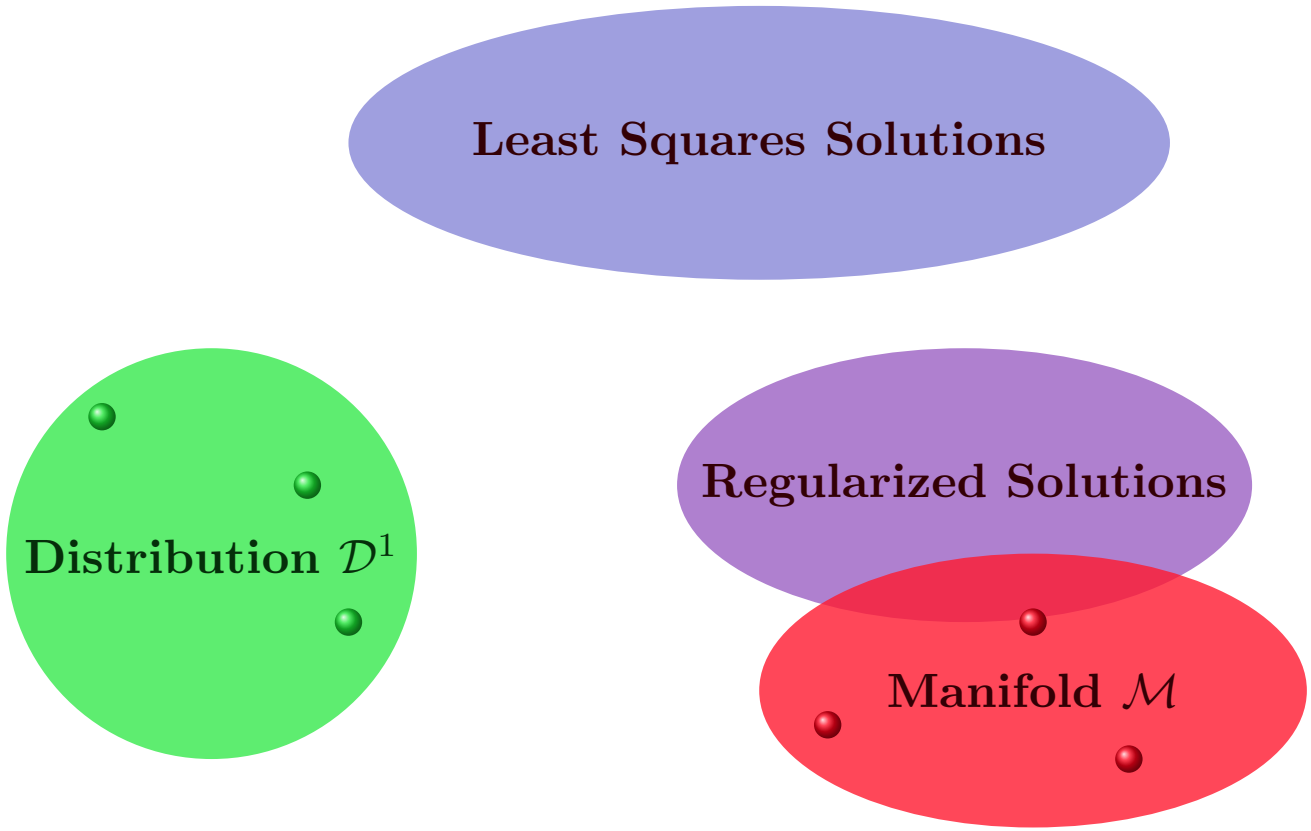
$$\tilde{u} \in \arg\min_{u} \quad \ell(A(u), b) + J(u),$$

where

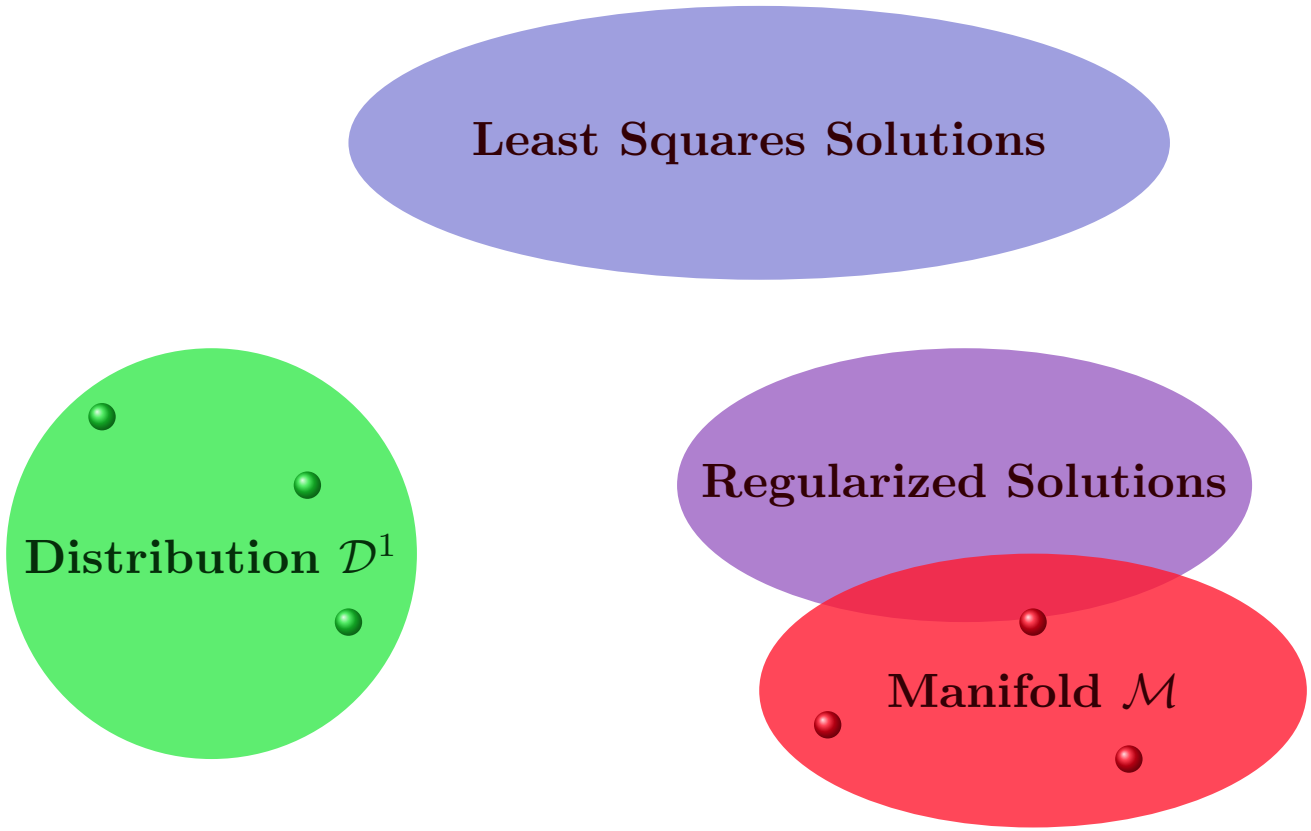
- $\ell \colon \mathbb{R}^m \times \mathbb{R}^m \to \overline{\mathbb{R}}$ is a fidelity term (e.g., Least Squares)
- $J \colon \mathbb{R}^m \times \mathbb{R}^m \to \overline{\mathbb{R}}$ is a regularizer (e.g., TV, L1, Learned)

**Remark:** regularizers often picked heuristically.

# Distribution Illustration



# Distribution Illustration




**Key question:** how can we guarantee reconstructed signal estimates are in the manifold of true signals?

# Agenda

1 Background

2 Adversarial Projections

3 Numerical Experiments

4 Conclusions

# Adversarial Projections

**Recurring theme:** signals represented in high dimensional spaces often exhibit redundancies, i.e., they lie on some low-dimensional manifold.

# Adversarial Projections

**Recurring theme:** signals represented in high dimensional spaces often exhibit redundancies, i.e., they lie on some low-dimensional manifold.

**Our Task:** learn an operator that projects onto manifold of true signals in unsupervised manner.

# Manifold Assumptions

## Assumption 1

The true distribution $\mathcal{D}_{\text{true}}$ is supported on a convex, compact set $\mathcal{M} \in \mathbb{R}^n$.

# Manifold Assumptions

## Assumption 1

The true distribution $\mathcal{D}_{\text{true}}$ is supported on a convex, compact set $\mathcal{M} \in \mathbb{R}^n$.

The projection operator $P_{\mathcal{M}} : \mathbb{R}^n \to \mathbb{R}^n$ gives the closest point in $\mathcal{M}$ to the point $u$, i.e.,

$$P_{\mathcal{M}}(u) := \arg\min_{v \in \mathcal{M}} \|v - u\|.$$

## Assumption 1

The true distribution $\mathcal{D}_{\text{true}}$ is supported on a convex, compact set $\mathcal{M} \in \mathbb{R}^n$.

The projection operator $P_{\mathcal{M}} : \mathbb{R}^n \to \mathbb{R}^n$ gives the closest point in $\mathcal{M}$ to the point $u$, i.e.,

$$P_{\mathcal{M}}(u) := \arg\min_{v \in \mathcal{M}} \|v - u\|.$$

The distance function is given by

$$d_{\mathcal{M}}(u) := \inf_{v \in \mathcal{M}} \|v - u\| = \|P_{\mathcal{M}}(u) - u\|.$$

# Manifold Assumptions

## Assumption 2

The initial distribution $\mathcal{D}^1$ is uniformly bounded.

## Assumption 3

The push forward of the projection operation $P_{\mathcal{M}}$ applied to $\mathcal{D}^1$ recovers the true distribution $\mathcal{D}_{\text{true}}$ up to a set of measure zero, i.e.,

$$\mathcal{D}_{\text{true}} = (P_{\mathcal{M}})_{\#}(\mathcal{D}^1) := \{P_{\mathcal{M}}(u) : u \in \mathcal{D}^1\}$$

## Assumption 2

The initial distribution $\mathcal{D}^1$ is uniformly bounded.

## Assumption 3

The push forward of the projection operation $P_{\mathcal{M}}$ applied to $\mathcal{D}^1$ recovers the true distribution $\mathcal{D}_{\text{true}}$ up to a set of measure zero, i.e.,

$$\mathcal{D}_{\text{true}} = (P_{\mathcal{M}})_{\#}(\mathcal{D}^1) := \{P_{\mathcal{M}}(u) : u \in \mathcal{D}^1\}$$

**Remark:** We do not assume each signal can be recovered, but instead the overall distribution is not "too noisy."

## Distance Function Theorem

Under the previous assumptions,

$$d_{\mathcal{M}} = \arg\min_{J \in \mathcal{I}} \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}} [J(u)] - \mathbb{E}_{u \sim \tilde{\mathcal{D}}} [J(u)] + \tau \cdot \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}} [(J(u))^p],$$

where $\tau \in (0, \infty)$, $p > 1$, $\tilde{\mathcal{D}}$ is the approximate distribution, and $\mathcal{I}$ is the set of all nonnegative 1-Lipschitz functions.

Lunz, et al. Adversarial Regularizers in Inverse Problems. NeurIPS 2018

## Distance Function Theorem

Under the previous assumptions,

$$d_{\mathcal{M}} = \arg\min_{J \in \mathcal{I}} \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}[J(u)] - \mathbb{E}_{u \sim \tilde{\mathcal{D}}}[J(u)] + \tau \cdot \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}[(J(u))^p],$$

where $\tau \in (0, \infty)$, $p > 1$, $\tilde{\mathcal{D}}$ is the approximate distribution, and $\mathcal{I}$ is the set of all nonnegative 1-Lipschitz functions.

**Remarks:**

- $d_{\mathcal{M}}$ is obtained without explicit representation of true manifold $\mathcal{M}$

Lunz, et al. Adversarial Regularizers in Inverse Problems. NeurIPS 2018

## Distance Function Theorem

Under the previous assumptions,

$$d_{\mathcal{M}} = \arg\min_{J \in \mathcal{I}} \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}[J(u)] - \mathbb{E}_{u \sim \tilde{\mathcal{D}}}[J(u)] + \tau \cdot \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}[(J(u))^p],$$

where $\tau \in (0, \infty)$, $p > 1$, $\tilde{\mathcal{D}}$ is the approximate distribution, and $\mathcal{I}$ is the set of all nonnegative 1-Lipschitz functions.

**Remarks:**

- $d_{\mathcal{M}}$ is obtained without explicit representation of true manifold $\mathcal{M}$
- we force $J$ to be zero on the manifold with $\tau > 0$

Lunz, et al. Adversarial Regularizers in Inverse Problems. NeurIPS 2018

## Distance Function Theorem

Under the previous assumptions,

$$d_{\mathcal{M}} = \arg\min_{J \in \mathcal{I}} \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}[J(u)] - \mathbb{E}_{u \sim \tilde{\mathcal{D}}}[J(u)] + \tau \cdot \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}[(J(u))^p],$$

where $\tau \in (0, \infty)$, $p > 1$, $\tilde{\mathcal{D}}$ is the approximate distribution, and $\mathcal{I}$ is the set of all nonnegative 1-Lipschitz functions.

**Remarks:**

- $d_{\mathcal{M}}$ is obtained without explicit representation of true manifold $\mathcal{M}$
- we force $J$ to be zero on the manifold with $\tau > 0$
- $\mathcal{I} =$ *all* 1-Lipschitz functions and $\tau = 0 \implies$ WGAN problem.

Lunz, et al. Adversarial Regularizers in Inverse Problems. NeurIPS 2018

## Distance Function Theorem

Under the previous assumptions,

$$d_{\mathcal{M}} = \arg\min_{J \in \mathcal{I}} \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}} [J(u)] - \mathbb{E}_{u \sim \tilde{\mathcal{D}}} [J(u)] + \tau \cdot \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}} [(J(u))^p],$$

where $\tau \in (0, \infty)$, $p > 1$, $\tilde{\mathcal{D}}$ is the approximate distribution, and $\mathcal{I}$ is the set of all nonnegative 1-Lipschitz functions.

**Remarks:**

- $d_{\mathcal{M}}$ is obtained without explicit representation of true manifold $\mathcal{M}$
- we force $J$ to be zero on the manifold with $\tau > 0$
- $\mathcal{I} =$ *all* 1-Lipschitz functions and $\tau = 0 \implies$ WGAN problem.
- we will parameterize $J$ by a neural network

Lunz, et al. Adversarial Regularizers in Inverse Problems. NeurIPS 2018

# Distribution Updates

For step size $\lambda_k$, define the operator

$$g_k(u) = u + \frac{\lambda_k(u)}{d_{\mathcal{M}}(u)} \left(P_{\mathcal{M}}(u) - u\right)$$

# Distribution Updates

For step size $\lambda_k$, define the operator

$$g_k(u) = u + \frac{\lambda_k(u)}{d_{\mathcal{M}}(u)} \left(P_{\mathcal{M}}(u) - u\right) \in u - \lambda_k(u) \partial d_{\mathcal{M}}(u),$$

# Distribution Updates

For step size $\lambda_k$, define the operator

$$g_k(u) = u + \frac{\lambda_k(u)}{d_{\mathcal{M}}(u)} \left(P_{\mathcal{M}}(u) - u\right) \in u - \lambda_k(u) \partial d_{\mathcal{M}}(u),$$

**Remarks:** In practice,

- we only obtain an approximation of $d_{\mathcal{M}}$
- we numerically differentiate to obtain $\partial d_{\mathcal{M}}(u)$

# Distribution Updates

For step size $\lambda_k$, define the operator

$$g_k(u) = u + \frac{\lambda_k(u)}{d_{\mathcal{M}}(u)} \left(P_{\mathcal{M}}(u) - u\right) \in u - \lambda_k(u)\partial d_{\mathcal{M}}(u),$$

**Remarks:** In practice,

- we only obtain an approximation of $d_{\mathcal{M}}$
- we numerically differentiate to obtain $\partial d_{\mathcal{M}}(u)$

To update an individual signal, we use a Halpern update

$$u^{k+1} = \gamma_k u^1 + (1 - \gamma_k) g_k(u^k),$$

where $\gamma_k \in (0, 1]$, $\gamma_k \to 0$, and $\sum_{k \in \mathbb{N}} \gamma_k = \infty$.

# Distribution Updates

For step size $\lambda_k$, define the operator

$$g_k(u) = u + \frac{\lambda_k(u)}{d_{\mathcal{M}}(u)} \left(P_{\mathcal{M}}(u) - u\right) \in u - \lambda_k(u)\partial d_{\mathcal{M}}(u),$$

**Remarks:** In practice,

- we only obtain an approximation of $d_{\mathcal{M}}$
- we numerically differentiate to obtain $\partial d_{\mathcal{M}}(u)$

To update an individual signal, we use a Halpern update

$$u^{k+1} = \gamma_k u^1 + (1 - \gamma_k) g_k(u^k),$$

where $\gamma_k \in (0, 1]$, $\gamma_k \to 0$, and $\sum_{k\in\mathbb{N}} \gamma_k = \infty$.

The distribution is updated as

$$\mathcal{D}^k := \{u^k(\omega) : \omega \in \Omega\}, \quad \text{for all } k \in \mathbb{N},$$

# Distribution Updates

For step size $\lambda_k$, define the operator

$$g_k(u) = u + \frac{\lambda_k(u)}{d_{\mathcal{M}}(u)} \left(P_{\mathcal{M}}(u) - u\right) \in u - \lambda_k(u)\partial d_{\mathcal{M}}(u),$$

**Remarks:** In practice,

- we only obtain an approximation of $d_{\mathcal{M}}$
- we numerically differentiate to obtain $\partial d_{\mathcal{M}}(u)$

To update an individual signal, we use a Halpern update

$$u^{k+1} = \gamma_k u^1 + (1 - \gamma_k) g_k(u^k),$$

where $\gamma_k \in (0, 1]$, $\gamma_k \to 0$, and $\sum_{k\in\mathbb{N}} \gamma_k = \infty$.

The distribution is updated as

$$\mathcal{D}^k := \{u^k(\omega) : \omega \in \Omega\}, \quad \text{for all } k \in \mathbb{N},$$

How to choose step size $\lambda_k$?

# Choice of Step Size

Denote the mean distance by

$$\beta_k = \mathbb{E}_{\omega \sim \Omega} \left[ d_{\mathcal{M}}(u^k(\omega)) \right] - \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}} \left[ d_{\mathcal{M}}(u) \right].$$

# Choice of Step Size

Denote the mean distance by

$$\beta_k = \mathbb{E}_{\omega\sim\Omega}\left[d_{\mathcal{M}}(u^k(\omega))\right] - \mathbb{E}_{u\sim\mathcal{D}_{\text{true}}}\left[d_{\mathcal{M}}(u)\right].$$

Three options for choosing step size $\lambda_k$:

- $\lambda_k(u) = \mu d_{\mathcal{M}}(u^k)$ individual distance to manifold
- $\lambda_k(u) = \mu\beta_k$ mean distance to manifold, or
- $\lambda_k(u) = \mu(\beta_k + d_{\mathcal{M}}(u^k))$ combined distance to manifold

for relaxation parameter $\mu \in (0, 1)$

# Projection Illustration

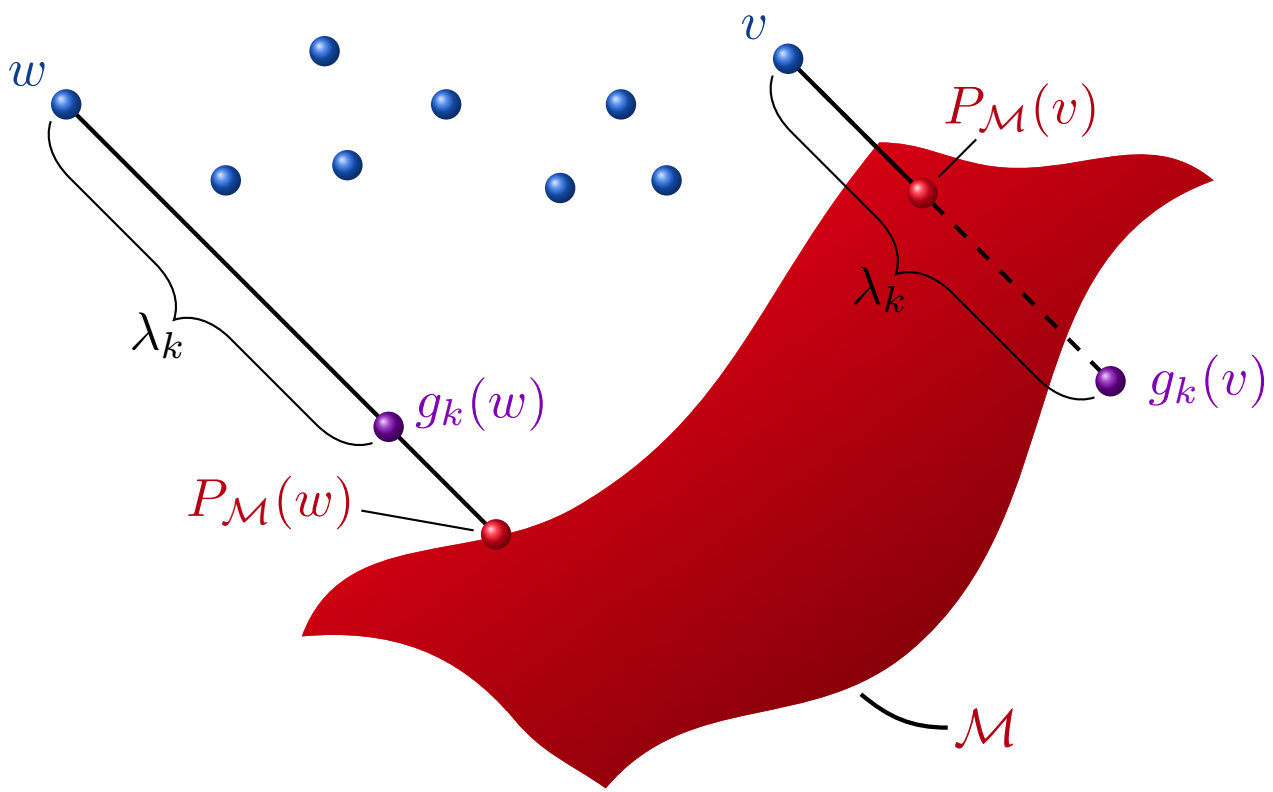



**Figure 1:** Blue points are samples drawn from distribution $\mathcal{D}^k$, which are updated to the relaxed projection $g_k$ of the projection $P_{\mathcal{M}}$ manifold $\mathcal{M}$.

# Learned Projection Algorithm

**Algorithm 1:** Training to generate parameters for distribution sequence $\{\mathcal{D}^k\}$

Choose relaxation parameter $\alpha \in (0, 1)$

Choose anchoring sequence $\{\gamma_k\} \subset (0, 1]$

Choose function parameterization $\mathcal{I}$

Choose initial distribution $\mathcal{D}^1 = \{u^1(\omega) : \omega \in \Omega\}$ ◁ Initial signal estimates

**for** $k = 1, 2, \ldots$ **do**

  $\theta^k \leftarrow \underset{\theta \in \mathcal{I}}{\operatorname{argmin}} \; \mathbb{E}_{u \sim \mathbb{P}_{\mathcal{U}}} \left[ J_\theta(u) + \tau J_\theta(u)^p \right] - \mathbb{E}_{u \sim \mathbb{P}^k} \left[ J_\theta(u^k) \right]$ ◁ Train to find weights

  $\beta_k \leftarrow \mathbb{E}_{u^k \sim \mathbb{P}^k} \left[ J_{\theta^k}(u^k) \right] - \mathbb{E}_{u \sim \mathbb{P}_{\mathcal{U}}} \left[ J_{\theta^k}(u) \right]$ ◁ Assign mean distance

  $u^{k+1}(\omega) \leftarrow \gamma_k u^1(\omega) + (1 - \gamma_k) g_k(u^k(\omega)), \quad \text{for all } \omega \in \Omega$ ◁ Update each sample

**end**

Return $\{\mu, \beta_k, \gamma_k, \theta^k\}$

# Learned Projection Algorithm

**Algorithm 1:** Training to generate parameters for distribution sequence $\{\mathcal{D}^k\}$

Choose relaxation parameter $\alpha \in (0, 1)$

Choose anchoring sequence $\{\gamma_k\} \subset (0, 1]$

Choose function parameterization $\mathcal{I}$

Choose initial distribution $\mathcal{D}^1 = \{u^1(\omega) : \omega \in \Omega\}$ ◁ Initial signal estimates

**for** $k = 1, 2, \ldots$ **do**

  $\theta^k \leftarrow \underset{\theta \in \mathcal{I}}{\operatorname{argmin}} \ \mathbb{E}_{u \sim \mathbb{P}_{\mathcal{U}}} \left[ J_\theta(u) + \tau J_\theta(u)^p \right] - \mathbb{E}_{u \sim \mathbb{P}^k} \left[ J_\theta(u^k) \right]$ ◁ Train to find weights

  $\beta_k \leftarrow \mathbb{E}_{u^k \sim \mathbb{P}^k} \left[ J_{\theta^k}(u^k) \right] - \mathbb{E}_{u \sim \mathbb{P}_{\mathcal{U}}} \left[ J_{\theta^k}(u) \right]$ ◁ Assign mean distance

  $u^{k+1}(\omega) \leftarrow \gamma_k u^1(\omega) + (1 - \gamma_k) g_k(u^k(\omega)), \quad \text{for all } \omega \in \Omega$ ◁ Update each sample

**end**

Return $\{\mu, \beta_k, \gamma_k, \theta^k\}$

## Remarks:

- $J$ is parameterized by a neural network, where $\theta$ are the weights
- we learn a sequence of relaxed projections which amount to $P_{\mathcal{M}}$
- encode information on NN by forcing $J(u) \geq 0 \ \ \forall u \in \mathbb{R}^n$

# Convergence

**Algorithm 1:** Training to generate parameters for distribution sequence $\{\mathcal{D}^k\}$

Choose relaxation parameter $\alpha \in (0, 1)$

Choose anchoring sequence $\{\gamma_k\} \subset (0, 1]$

Choose function parameterization $\mathcal{I}$

Choose initial distribution $\mathcal{D}^1 = \{u^1(\omega) : \omega \in \Omega\}$ ◁ Initial signal estimates

**for** $k = 1, 2, \ldots$ **do**

  $\theta^k \leftarrow \underset{\theta \in \mathcal{I}}{\operatorname{argmin}} \ \mathbb{E}_{u \sim \mathbb{P}_{\mathcal{U}}} \left[ J_\theta(u) + \tau J_\theta(u)^p \right] - \mathbb{E}_{u \sim \mathbb{P}^k} \left[ J_\theta(u^k) \right]$ ◁ Train to find weights

  $\beta_k \leftarrow \mathbb{E}_{u^k \sim \mathbb{P}^k} \left[ J_{\theta^k}(u^k) \right] - \mathbb{E}_{u \sim \mathbb{P}_{\mathcal{U}}} \left[ J_{\theta^k}(u) \right]$ ◁ Assign mean distance

  $u^{k+1}(\omega) \leftarrow \gamma_k u^1(\omega) + (1 - \gamma_k) g_k(u^k(\omega)), \quad \text{for all } \omega \in \Omega$ ◁ Update each sample

**end**

Return $\{\mu, \beta_k, \gamma_k, \theta^k\}$

## Convergence Theorem

Under previously stated assumptions, the sequence $\{u^k\}$ converges to $P_{\mathcal{M}}(u^1)$ in probability.

# Related Works

Three key connections:

- **Wasserstein GANS**[1]: Our learned distance function $J$ can be interpreted as a discriminator function in W-GANs. $J$ discriminates between "true" and approximate signals.

---

Arjovsky et al. Wasserstein Generative Adversarial Networks. ICML 2017

Lunz et al. Adversarial Regularizers in Inverse Problems. NeurIPS 2018

Gilboa. Expert Regularizers for task specific processing. SSVM, 2013

# Related Works

Three key connections:

- **Wasserstein GANS**[1]: Our learned distance function $J$ can be interpreted as a discriminator function in W-GANs. $J$ discriminates between "true" and approximate signals.
- **Adversarial Regularizers**[2]: Our work extends the work from adversarial regularizers and provides convergence guarantees.

---

Arjovsky et al. Wasserstein Generative Adversarial Networks. ICML 2017

Lunz et al. Adversarial Regularizers in Inverse Problems. NeurIPS 2018

Gilboa. Expert Regularizers for task specific processing. SSVM, 2013

# Related Works

Three key connections:

- **Wasserstein GANS**[1]: Our learned distance function $J$ can be interpreted as a discriminator function in W-GANs. $J$ discriminates between "true" and approximate signals.
- **Adversarial Regularizers**[2]: Our work extends the work from adversarial regularizers and provides convergence guarantees.
- **Expert Regularizers**[3]: $J$ can be interpreted as an expert regularizer, which attain small values at signals similar to the distribution of true signals and larger values at signals drawn elsewhere.

---

Arjovsky et al. Wasserstein Generative Adversarial Networks. ICML 2017

Lunz et al. Adversarial Regularizers in Inverse Problems. NeurIPS 2018

Gilboa. Expert Regularizers for task specific processing. SSVM, 2013

# Agenda

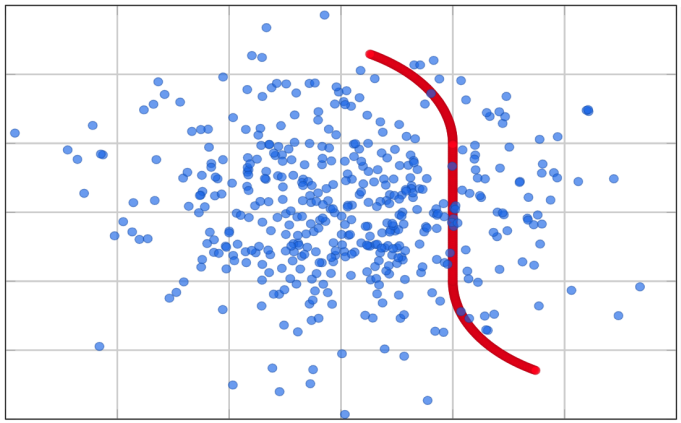
1 Background

2 Adversarial Projections

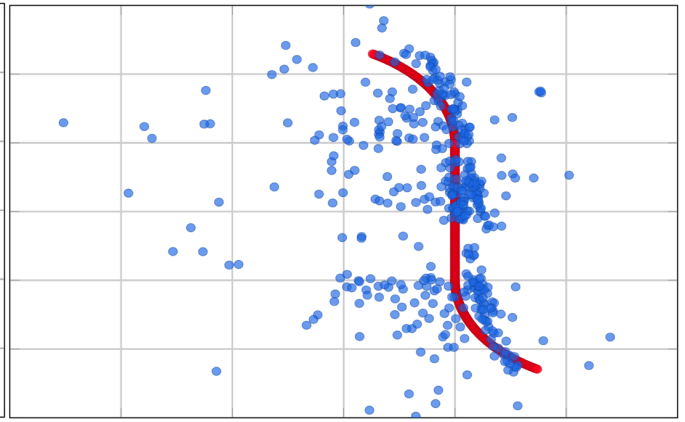
3 Numerical Experiments

4 Conclusions

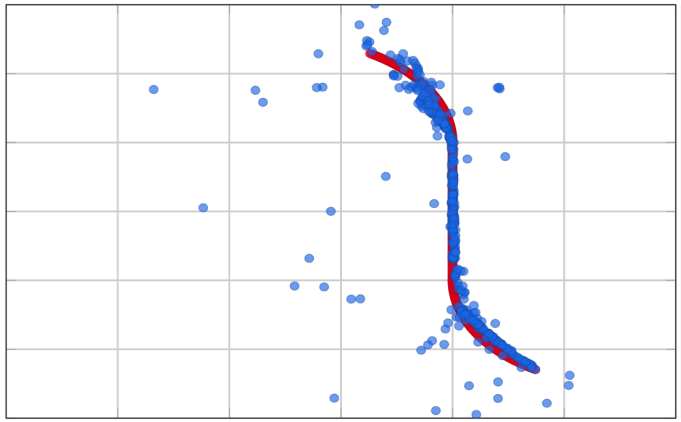
# Toy Example



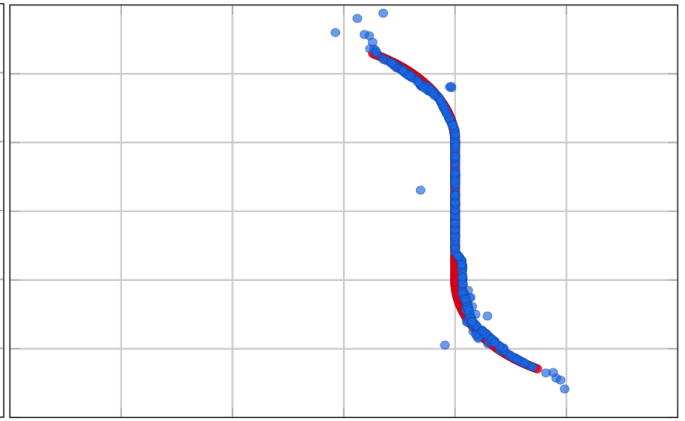
**(a)** $\mathcal{D}^1$



**(b)** $\mathcal{D}^5$

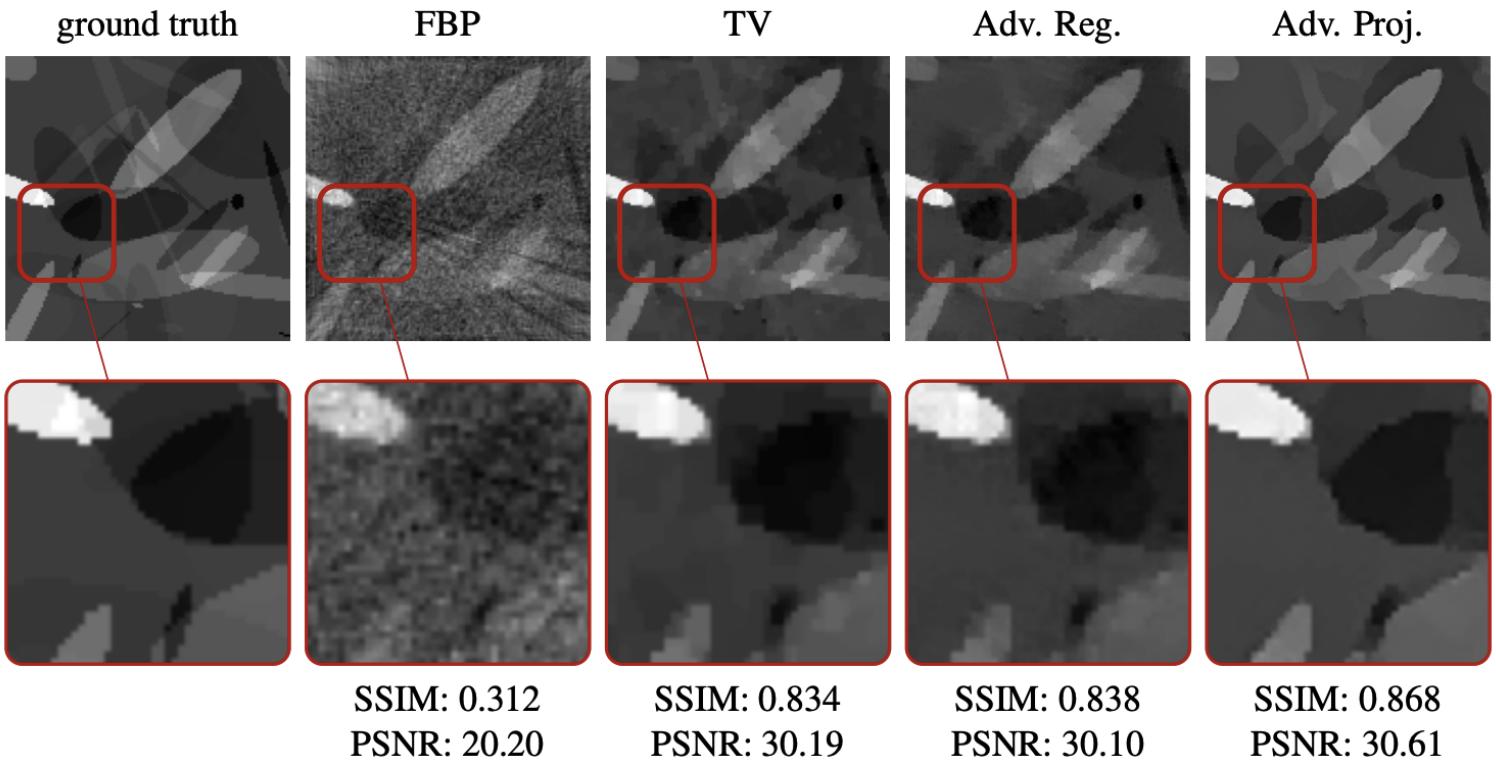


**(c)** $\mathcal{D}^{25}$



**(d)** $\mathcal{D}^{300}$

# Low-Dose CT: Ellipses



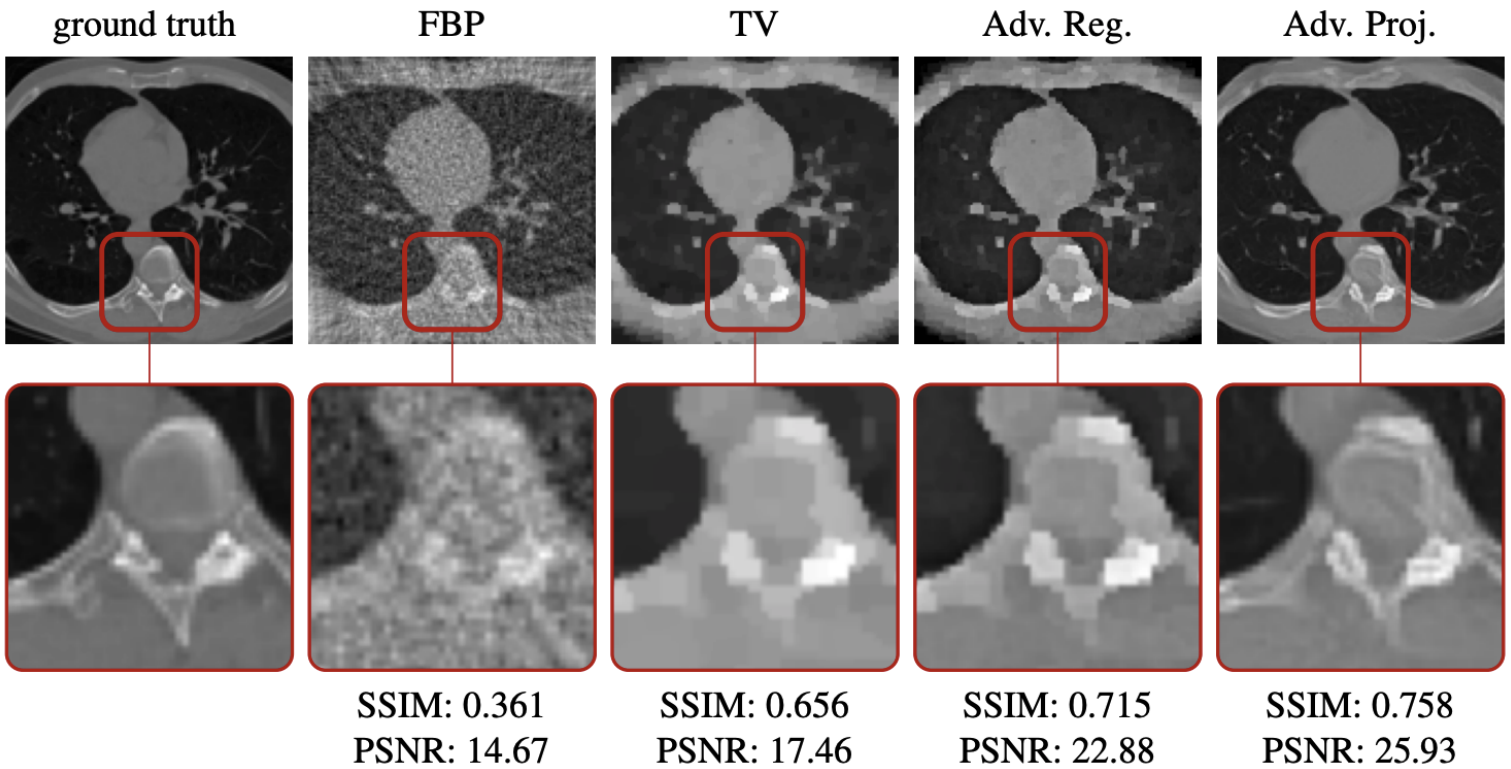
# Low-Dose CT: Ellipses

CT Results on Ellipses Dataset

| Method | Avg. PSNR (dB) | Avg. SSIM |
|---|---|---|
| Filtered Backprojection | 16.53 | 0.179 |
| Total Variation | 26.43 | 0.624 |
| Adversarial Regularizers | 26.95 | 0.680 |
| Adversarial Projections (ours) | **28.11** | **0.783** |

**Table 1:** Average PSNR and SSIM on a validation dataset consisting 1,000 images of random ellipses.

# Low-Dose CT: Human Phantoms



CT Results on Human Phantoms

| Method | Avg. PSNR (dB) | Avg. SSIM |
|---|---|---|
| Filtered Backprojection | 14.77 | 0.314 |
| Total Variation | 19.72 | 0.672 |
| Adversarial Regularizers | 22.58 | 0.691 |
| Adversarial Projections (ours) | **26.07** | **0.750** |

**Table 2:** Average PSNR and SSIM on a validation dataset consisting 2,000 images of human phantoms.

# Agenda

1 Background

2 Adversarial Projections

3 Numerical Experiments

4 Conclusions

# Conclusions

**Summary:** We present a new mechanism for solving inverse problems

- Use projections to find closest "true" point to measurements
- Leverage ideas from Wasserstein GANs and Expert Regularizers
- Focus on distribution, not individual signals

Preprint will be updated soon!

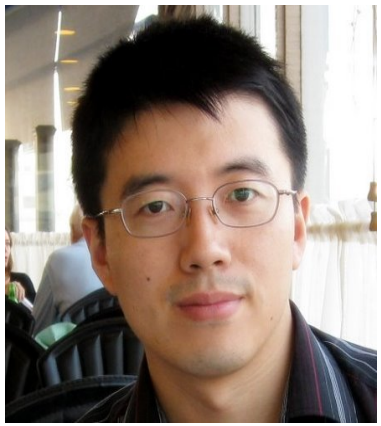
# Team and Acknowledgments

Howard Heaton

Alex Lin

Stan Osher

Wotao Yin



**Funding:** AFOSR MURI FA9550-18-1-0502, AFOSRFA9550-18-1-0167, ONR N00014-18-1-2527, NSF DGE-1650604