Projecting to Manifolds via Unsupervised Learning

Howard Heaton*, Samy Wu Fung*, Alex Tong Lin*, Stanley Osher, Wotao Yin

University of California, Los Angeles†

August 5, 2020

Abstract

We present a new framework, called adversarial projections, for solving inverse problems by learning to project onto manifolds. Our goal is to recover a signal from a collection of noisy measurements. Traditional methods for this task often minimize the addition of a regularization term and an expression that measures compliance with measurements (e.g., least squares). However, it has been shown that convex regularization can introduce bias, preventing recovery of the true signal. Our approach avoids this issue by iteratively projecting signals toward the (possibly non-linear) manifold of true signals. This is accomplished by first solving a sequence of unsupervised learning problems. The solution to each learning problem provides a collection of parameters that enables access to an iteration-dependent step size and access to the direction to project each signal toward the closest true signal. Given a signal estimate (e.g., recovered from a pseudo-inverse), we prove our method generates a sequence that converges in mean square to the projection onto this manifold. Several numerical illustrations are provided.

Keywords— adversarial projection, inverse problems, Wasserstein GANs, generative networks, optimal transport, deep neural networks, regularization, projection, learning to optimize, computed tomography

1 Introduction

Inverse problems arise in numerous applications such as medical imaging [6, 7, 39, 53], phase retrieval [10, 15, 26, 64], geophysics [13, 27, 28, 34, 35, 42], and machine learning [21, 25, 36, 67, 69]. The underlying goal of inverse problems is to recover a signal from a collection of indirect noisy measurements. Formally stated, consider a finite dimensional Hilbert space $\mathcal{X}$ (e.g., $\mathbb{R}^n$) with scalar product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|$ for the domain space, and similarly for $\mathcal{Y}$ (e.g., $\mathbb{R}^m$) for the measurement space. Let $A : \mathcal{X} \to \mathcal{Y}$ be a mapping between $\mathcal{X}$ and $\mathcal{Y}$, and let $b \in \mathcal{Y}$ be the available measurement data given by

$$b = A(u^\star) + \varepsilon, \tag{1.1}$$

where $u^\star \in \mathcal{X}$ denotes the true signal and $\varepsilon \in \mathcal{Y}$ denotes the noise in the measurement. The aim in inverse problems is to

$$\text{Estimate } u^\star \text{ from the noisy measurements } b. \tag{1.2}$$

*Equal contribution.

†hheaton@ucla.edu, swufung@math.ucla.edu, atlin@math.ucla.edu, sjo@math.ucla.edu, wotaoyin@math.ucla.edu

A difficulty in completing the task in (1.2) is that inverse problems are often ill-posed, making their solutions unstable for noise-affected data. To overcome ill-posedness, traditional approaches for solving inverse problems involve a regularized variational approach that estimates the signal $u^\star$ by

$$\tilde{u} \in \operatorname*{argmin}_{u \in \mathcal{X}} \ \ell(A(u), b) + J(u), \tag{1.3}$$

where[1] $\ell \colon \mathcal{Y} \times \mathcal{Y} \to \overline{\mathbb{R}}$ measures the discrepancy between the measurements and the application of the forward operator A to our signal estimate (e.g., least squares). The function $J \colon \mathcal{X} \to \overline{\mathbb{R}}$ serves as a regularizer, which ensures that the solution to (1.3) is unique and that its computation is stable. In addition to ensuring well-posedness, regularizers are constructed in an effort to incorporate prior knowledge of the true signal. Common model-based regularizers include, e.g., sparsity $J(u) = \|u\|_1$ [11, 16, 17, 24], Tikhonov $J(u) = \|u\|^2$ [14, 31], Total Variation $J(u) = \|\nabla u\|_1$ [19, 57], and, more recently, data-driven regularizers [3, 44, 50]. A related approach includes Bregman iteration methods [73]. An underlying theme in regularization is that it is commonly assumed that signals exhibit redundant representations and admit a compact low-dimensional manifold representation. However, directly approximating the manifold is highly nontrivial. Thus, a key question remains:

How do we guarantee that the reconstructed signal lies on the manifold of true signals?

Below we demonstrate that this guarantee can be ensured by using a projection algorithm. In addition, we emphasize that our approach is i) unsupervised and ii) does not require directly representing the manifold. This means that a direct correspondence between noisy signal estimate data and true signal data is *not* needed (e.g., we may even have different amounts of samples from each data set).

Remark 1.1 *Throughout this work we refer to reconstructing signals. This phrase is meant in a general sense to describe an object of interest that can be represented mathematically. This includes, e.g., images, parameters of a differential equation, and points in a Hilbert space.*

1.1 Contribution

We present adversarial projections, a new framework for solving inverse problems. Our core result is to demonstrate how *unsupervised* learning can be used to project signal estimates onto the underlying low-dimensional manifold of true signals. This is accomplished without making a direct representation of the manifold. The training process consists of solving a sequence of minimization problems (related to the inner maximization in general adversarial networks, discussed below). During implementation, our proposed algorithm forms a Halpern-type method with relaxed projections, which we prove converges in mean square to the projection of the initial estimate onto the manifold. At the level of individual signals, this work may also be interpreted as learned gradient descent with a sequence of expert-like regularizers [30]. And at the aggregate level of distributions, it may be viewed as a subgradient method for minimizing the Wasserstein-1 distance between the distribution of initial estimates and the true distribution.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of generative adversarial networks (GANs) and their connections to optimal transport (OT), adversarial regularizers, and expert regularizers. In Section 3, we describe our adversarial projections approach. The convergence analysis is covered in Section 4. In Section 5, we review the related works. In Section 6, we show the potential of adversarial projections on a two-dimensional distribution as well as two low-dose parallel beam computed tomography experiments. We conclude with a brief discussion in Section 7.

[1] Here we use $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$.

2 Background

In this section, we briefly review adversarial regularizers [50], Wasserstein GANs [5, 32] and their connections to optimal transport [46, 63], and expert regularizers [30]. These topics will be useful for interpreting adversarial projections.

2.1 Wasserstein GANs and Optimal Transport

In GANs [5, 32], access is given to a discriminator and generator, and the goal is to train the generator to produce samples from a desired distribution. The generator does this by taking samples from a known distribution $\mathcal{N}$ and transforming them into samples from the desired distribution $\mathcal{D}_{\text{true}}$. Meanwhile, the purpose of the discriminator is to guide the optimization of the generator. Given a generator network G_θ and a discriminator network D_ω, the goal in Wasserstein GANs is to find a saddle point solution to the minimax problem

$$\inf_{G_\theta} \sup_{D_\omega} \mathbb{E}_{u\sim\mathcal{D}_{\text{true}}}\left[D_\omega(u)\right] - \mathbb{E}_{z\sim\mathcal{N}}\left[D_\omega(G_\theta(z))\right], \quad \text{s.t.} \quad \|\nabla D_\omega\| \leqslant 1, \tag{2.1}$$

Here, the discriminator attempts to distinguish real images from fake/generated images, and the generator aims to produce samples that "fool" the discriminator by appearing real. The supremum expression in (2.1) is the Kantorovich-Rubenstein dual formulation [66] of the Wasserstein-1 distance, and the discriminator is required to be 1-Lipschitz. Thus, the discriminator computes the Wasserstein-1 distance between the true distribution $\mathcal{D}_{\text{true}}$ and the distribution of fake images generated by $G_\theta(z)$. Originally, weight-clipping was to enforce the Lipschitz condition of the discriminator network [5], but an improved method using a penalty on the gradient was used in [33].

2.2 Adversarial Regularizers

A good regularizer $J\colon \mathcal{X} \to \overline{\mathbb{R}}$ is able to distinguish between signals drawn from the true distribution $\mathcal{D}_{\text{true}}$ and drawn from an approximate distribution $\tilde{\mathcal{D}}$ – taking low values on signals from $\mathcal{D}_{\text{true}}$ and high values otherwise [12]. Such regularizer plays a similar role as the discriminator described in Section 2.1; however, this setting is different in that D_ω assigns high values to true signals instead. Mathematically, $J = -D_\omega$. These regularizers are called adversarial regularizers [50]. They are trained a priori in a GAN-like fashion, and are then used to solve a classical inverse problem (1.2). These regularizers were shown to have desired distributional properties in that their gradients provide a descent direction for the Wasserstein-1 distance [50, Section 3.2]. Our work takes advantage of this fact to provide convergence guarantees (see Section 4).

2.3 Expert Regularizers

For many inverse problems, well-posed reconstructions can be obtained by incorporating additional knowledge about the signals to be recovered. Expert regularizers [30] are functions used for accomplishing this task that attain small values at signals similar to the distribution of true signals and larger values at signals drawn elsewhere. Inclusion of experts regularizers, thus, should encourage recovery of signals from the true distribution $\mathcal{D}_{\text{true}}$ while not introducing additional artifacts. Formally stated, given constants $\beta, \mu \in (0, \infty)$, desirable properties include

1. $\phi(u) \geqslant 0$ with equality if and only if $u \in \mathcal{D}_{\text{true}}$.

2. For all $\varepsilon > 0$, there exists $\delta > 0$ such that if $v \in \mathcal{D}_{\text{true}}$, then $\|v - u\| \leqslant \delta$ implies $\phi(u) \leqslant \varepsilon$.

3. For all $v \in \mathcal{D}_{\text{true}}$, $\|v - u\| \geqslant \mu$ implies $\phi(u) \geqslant \beta\mu$.

Note that, if $\mathcal{D}_{\text{true}}$ is closed and convex, all of the above properties are satisfied by the function that measures the distance between u and the set $\mathcal{D}_{\text{true}}$. In addition, the second item is automatically satisfied if the first item holds and ϕ is Lipschitz. The primary task in the training process (Algorithm 1) for our proposed method may be viewed as finding a sequence of regularizers $\{\phi_k\}$ that approximate the properties of expert regularizers. Our adversarial projection method (Algorithm 2) then performs a sequence of gradient descent steps successively using each ϕ_k.

3 Adversarial Projections

Herein a projection method is proposed to solve the inverse problem (1.2). Suppose access is provided to a reasonable estimate $\tilde{u}$ of the true signal $u^\star$ and that $u^\star$ is contained in a compact manifold $\mathcal{M}$ (formally stated in Section 4). The key idea is that the point in $\mathcal{M}$ closest to the estimate $\tilde{u}$ forms an improved approximation of $u^\star$. That is, $u^\star \approx P_{\mathcal{M}}(\tilde{u})$ where $P_{\mathcal{M}}$ is the projection operator onto $\mathcal{M}$ defined by[2]

$$P_{\mathcal{M}}(u) := \operatorname*{argmin}_{v \in \mathcal{M}} \|v - u\|. \tag{3.1}$$

In most practical settings we do not have direct access to the manifold $\mathcal{M}$ to determine this projection. However, below we indirectly form projections using the pointwise distance function

$$d_{\mathcal{M}}(u) := \inf_{v \in \mathcal{M}} \|v - u\|. \tag{3.2}$$

Indeed, for $\alpha \in \mathbb{R}$ and $\lambda = \alpha \cdot d_{\mathcal{M}}(u)$, we obtain the inclusion relation[3]

$$u + \alpha \left(P_{\mathcal{M}}(u) - u\right) \in u - \lambda \partial d_{\mathcal{M}}(u), \tag{3.3}$$

and the left hand side is called the α-relaxed projection of u onto $\mathcal{M}$. In particular, we can directly obtain the left hand side from the subgradient expression on the right (described below). With additional assumptions (see Section 4), we find that when the estimate $\tilde{u}$ is drawn from a distribution of estimates $\tilde{\mathcal{D}}$ and the true signal $u^\star$ is drawn from the distribution of true signals $\mathcal{D}_{\text{true}}$,

$$d_{\mathcal{M}} \in \operatorname*{argmax}_{\|f\|_L \leqslant 1} \mathbb{E}_{u \sim \tilde{\mathcal{D}}}\left[f(u)\right] - \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}\left[f(u)\right], \tag{3.4}$$

i.e., the pointwise distance function $d_{\mathcal{M}}$ is a maximizer of the expression on the right hand side. (Here $\|f\|_L \leqslant 1$ denotes the set of all 1-Lipschitz functions f.) Thus, our task is to solve (3.4), which is a form of unsupervised learning, and then use our estimate of $d_{\mathcal{M}}$ to form a relaxed projection. This also illustrates that the name *adversarial projection* derives from the fact that the relaxed projection operation we implement using (3.3) and (3.4) comes from the inner expression in (2.1) used for WGANs.

In practice, the estimate obtained for $d_{\mathcal{M}}$ may be a rough approximation. In light of this, our method uses a small fixed step size common for all $\tilde{u} \in \tilde{\mathcal{D}}$ when performing each update (rather

[2]The projection is well-defined precisely when the minimization problem admits a unique solution.

[3]For completeness, this statement is proven in Lemma 8.1 of the Appendix.

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

Result: Weights $\{\theta^k\}$, step sizes $\{\lambda_k\}$, and anchoring weights $\{\gamma_k\}$

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

Choose anchoring sequence $\{\gamma_k\} \subset (0, 1]$ ◁ See Assumption 4.8

Choose function parameterization $\mathcal{I}$ ◁ See Assumption 4.6

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

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

  $\theta^k \leftarrow \underset{\theta \in \mathcal{I}}{\operatorname{argmin}} \ \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}\left[J_\theta(u)\right] - \mathbb{E}_{\omega \sim \Omega}\left[J_\theta(u^k(\omega))\right]$ ◁ Use ADAM to find weights

  $\lambda_k \leftarrow \alpha \cdot (\mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}\left[J_{\theta^k}(u)\right] - \mathbb{E}_{\omega \sim \Omega}\left[J_{\theta^k}(u^k(\omega))\right])$ ◁ Compute step size

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

end

Return $\{\theta^k, \lambda_k, \gamma_k\}$

than individualized step sizes), mimicking a gradual and (hopefully) stable flow of the distribution $\tilde{D}$ toward $\mathcal{D}_{\text{true}}$. In some cases, this causes the updates to overshoot the manifold in such a way that the projection of the new update onto the manifold is *not* the point $P_{\mathcal{M}}(\tilde{u})$. However, we can still ensure the sequence generated by our method converges to the $P_{\mathcal{M}}(\tilde{u})$ by incorporating a form of anchoring (i.e., pulling updates closer to $\tilde{u}$). Given a sequence of real numbers $\{\gamma_k\} \subset (0, 1)$ and a 1-Lipschitz operator $T : \mathcal{X} \to \mathcal{X}$ (i.e., nonexpansive), Halpern [37] proposed finding the projection of $u^1 = \tilde{u}$ onto the fixed point set of the operator T (i.e., the points such that $u = T(u)$) by generating a sequence $\{u^k\}$ of the form

$$u^{k+1} = \gamma_k u^1 + (1 - \gamma_k) T(u^k), \quad \text{for all } k \in \mathbb{N}, \tag{3.5}$$

where each update is a convex combination of u^1 and $T(u^k)$. Our method takes a related form

$$u^{k+1} = \gamma_k u^1 + (1 - \gamma_k)\Big(u^k + \alpha_k (P_{\mathcal{M}}(u^k) - u^k)\Big), \quad \text{for all } k \in \mathbb{N}, \tag{3.6}$$

where we note the expression replacing T on the right is *not* necessarily 1-Lipschitz for our choice of sequence $\{\alpha_k\}$; however, this expression has the desired fixed point set $\mathcal{M}$ since the terms multiplied by α_k cancel when $u^k \in \mathcal{M}$. We choose step sizes so that typical updates in our method will have $\alpha_k \in (0, 1)$, resulting in an under-relaxed projection and convergence to $P_{\mathcal{M}}(\tilde{u})$ in probability (see Theorem 4.9). Algorithm 1 articulates the training procedure for identifying the parameters needed for our adversarial projection scheme in Algorithm 2.

Remark 3.1 *The pointwise distance function $d_{\mathcal{M}}(u)$ is distinct from the Wasserstein-1 distance Wass$(\tilde{\mathcal{D}}, \mathcal{D}_{\text{true}})$ between the distribution $\tilde{D}$ of estimates and the true signal distribution $\mathcal{D}_{\text{true}}$. The former measures the distance from an individual point to a set while the latter is a metric for distributions of points. The connection between these, in our setting, is that the expected value of the distance to the manifold among all $\tilde{u} \sim \tilde{D}$ is equivalent to the Wasserstein-1 distance, i.e.,*

$$\mathbb{E}_{\tilde{u} \sim \tilde{D}}\left[d_{\mathcal{M}}(\tilde{u})\right] = \textit{Wass}(\tilde{D}, \mathcal{D}_{\text{true}}). \tag{3.7}$$

Algorithm 1 for flowing the training data distribution toward the true distribution may be described as follows. First note that $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space, where Ω, $\mathcal{F}$, and $\mathbb{P}$ are the sample space,

σ-algebra, and probability measure, respectively. Rather than write a composition of operations applied to $\omega \in \Omega$, we adopt the notation convention that $u^1(\omega)$ gives the initial estimate recovered from the measurement data[4] b and, for all $k \in \mathbb{N}$, $u^k(\omega)$ is the k-th iterate of the method. The output of Algorithm 1 is a collection of parameterized functions and step sizes. The relaxation constant α in Step 2 follows its use in (3.3) to determine the step size for relaxed projections. The anchoring sequence $\{\gamma_k\}$ is chosen to pull successive updates closer to the initial iterate (e.g., $\gamma_k = 1/k$). The function parameterization $\mathcal{I}$ defines the collection $\{J_\theta\}_{\theta \in \mathcal{I}}$ of functions over which the optimization occurs in Step 3, which in practice forms an approximation of the set of all 1-Lipschitz functions. The initial collection of signal estimates is denoted by $\mathcal{D}^1$ and for k-th iterate we write $\mathcal{D}^k := \{u^k(\omega) : \omega \in \Omega\}$ so that

$$\mathbb{E}_{\omega \sim \Omega}\left[J_\theta(u^k(\omega))\right] = \mathbb{E}_{u \sim \mathcal{D}^k}\left[J_\theta(u)\right]. \tag{3.8}$$

A **for** loop occurs from Lines 5-9 with each index k giving rise to a distribution $\mathcal{D}^k$ of signal estimates. Flipping the sign of the problem in (3.4) yields the minimization problem in Line 6, which can be solved using ADAM [43]. Line 7 then defines the step size λ_k, which is proportional to the average distance between points in $\mathcal{D}^k$ and $\mathcal{D}_{\text{true}}$. Line 8 defines the updates for each iterate $u^k(\omega)$ using the Halpern-type update described in (3.6). There we use the definition

$$g_k(u) := \begin{cases} u + \lambda_k \cdot \nabla J_{\theta^k}(u) & \text{if } J_{\theta^k} \text{ is differentiable at } u, \\ u & \text{otherwise.} \end{cases} \tag{3.9}$$

so that, upon flipping signs to assume $d_{\mathcal{M}} = -J_{\theta^k}$, we obtain the relaxed projection

$$g_k(u) = u + \underbrace{\frac{\lambda_k}{d_{\mathcal{M}}(u)}}_{=:\alpha_k(u)} \left(P_{\mathcal{M}}(u) - u\right) = u + \alpha_k(u)\left(P_{\mathcal{M}}(u) - u\right) \in u - \lambda_k \partial d_{\mathcal{M}}(u), \tag{3.10}$$

where $\alpha_k(u)$ is defined to be the underbraced term and we adopt the convention of taking $\alpha_k(u) = 0$ when $d_{\mathcal{M}}(u) = 0$. (This is justified since $d_{\mathcal{M}}(u) = 0$ implies $P_{\mathcal{M}}(u) = u$.) This Halpern-type update forms a convex combination of the initial iterate u^1 and the relaxed projection $g_k(u^k)$. Upon completion of this training process, inferences can be performed using the learned quantities $\{\theta^k, \lambda_k\}$ by applying Algorithm 2.

Remark 3.2 *In practice, because we perform numerical differentiation, we abusively write*

$$g_k(u) = u + \lambda_k \cdot \nabla J_{\theta^k}(u), \tag{3.11}$$

which is the expression used in our experiments, with J_{θ^k} analogous to discriminators in GANs.

Explanation of Algorithm 2 is as follows. First the parameters $\{\theta^k\}$, step sizes $\{\lambda_k\}$, and anchoring sequence $\{\gamma_k\}$ are chosen according to Algorithm 1. Then in Line 3 the point u^1 is initialized to an initial estimate of $u^\star$. This estimate can be generated using, for example, a pseudo inverse or a solution to an associated regularized problem. Then, for each k, the Halpern-type update is computed using a relaxed projection with g_k (Line 6). Note g_k is defined using λ_k and θ^k. Upon repeating this process the same number of times as the training iterations, we obtain our estimate u^k in Line 7. The following section proves convergence of adversarial projections.

Algorithm 2: Adversarial Projection (How to Reconstruct Individual Signal)

Result: True Signal Estimate u^k from Data b

Choose weights $\{\theta^k\}$, step sizes $\{\lambda_k\}$, and sequence $\{\gamma_k\}$ from Algorithm 1

AdvProj(b)

  $u^1 \leftarrow \tilde{u}(b)$ ◁ Initialize to pseudo inverse or solution to (1.3)

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

    $u^{k+1} \leftarrow \gamma_k u^1 + (1 - \gamma_k) g_k(u^k)$ ◁ Halpern-type update (see (3.6) and (3.11))

  end

  return u^k

end

4 Convergence Analysis

This section formalizes the assumptions and states the main convergence result for the adversarial projections method. We first articulate a form of the intuitive idea that the true data is contained in a low dimensional manifold $\mathcal{M}$. Then we assume the initial distribution estimate is bounded and each successive distribution $\mathcal{D}^k$ is not "too noisy" (i.e., the observed signal is not missing significant features from the true signal, commonly known as collapsed modes).

Assumption 4.1 *The true distribution $\mathcal{D}_{\text{true}}$ is supported on a convex, compact set $\mathcal{M} \subset \mathcal{X}$.*

Assumption 4.2 *The initial distribution $\mathcal{D}^1$ is uniformly bounded.*

Assumption 4.3 *For all $k \in \mathbb{N}$, the distribution $\mathcal{D}^k$ is such that the push forward of the projection operation onto the manifold $\mathcal{M}$ recovers the true signal distribution $\mathcal{D}_{\text{true}}$ up to a set of measure zero, i.e.,*

$$\mathcal{D}_{\text{true}} = (P_{\mathcal{M}})_{\#}(\mathcal{D}^k) := \{P_{\mathcal{M}}(u) : u \in \mathcal{D}^k\} = \{P_{\mathcal{M}}(u^k(\omega)) : \omega \in \Omega\}. \tag{4.1}$$

Remark 4.4 *This assumption effectively states the noise is not "too large" and that the method used to obtain the initial distribution is sufficiently representative and does not collapse modes. This is a weaker statement than assuming each individual signal can be recovered from its measurements. And, if our method is actually making progress, the assumption holding for $k = 1$ should naturally imply it also holding for all subsequent values of k.*

Using the above assumptions, [50] provides an equivalent variation of the following theorem, which relates the set of 1-Lipschitz functions to the distance function $d_{\mathcal{M}}$ from points to the manifold $\mathcal{M}$.

Theorem 4.5 *Under Assumptions 4.1 and 4.3, for all $k \in \mathbb{N}$, $d_{\mathcal{M}}$ is a solution to*

$$\sup_{\|f\|_L \leqslant 1} \mathbb{E}_{u \sim \mathcal{D}^k}\left[f(u)\right] - \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}\left[f(u)\right]. \tag{4.2}$$

That is,

$$\mathbb{E}_{u \sim \mathcal{D}^k}\left[d_{\mathcal{M}}(u)\right] = \sup_{\|f\|_L \leqslant 1} \mathbb{E}_{u \sim \mathcal{D}^k}\left[f(u)\right] - \mathbb{E}_{u \sim \mathcal{D}_{\text{true}}}\left[f(u)\right]. \tag{4.3}$$

[4] For all practical purposes, we can assume $\omega = b$ and that Ω is the set of all plausible measurement data.

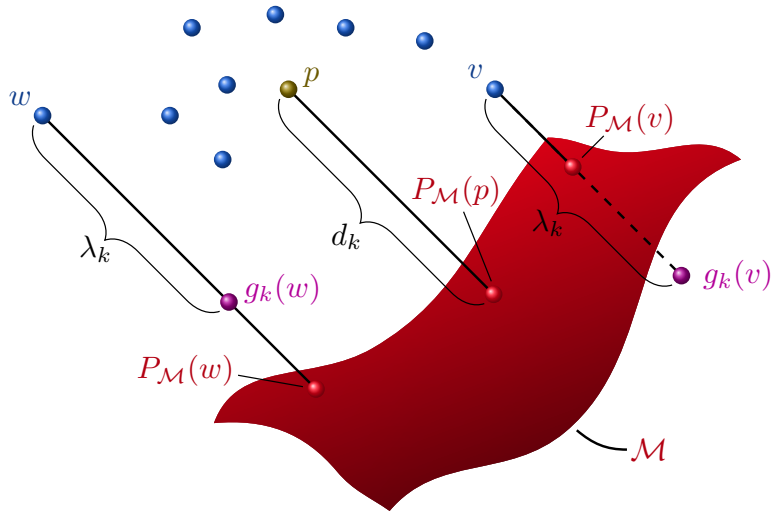


Figure 1: Each point in the distribution $\mathcal{D}^k$ (blue) is updated by the relaxed projection g_k (purple) for the projection $P_{\mathcal{M}}$ onto the manifold $\mathcal{M}$ (red). We label the over-relaxation $g_k(v)$ and under-relaxation $g_k(w)$. The point p (brown) illustrates an approximate average of the points in $\mathcal{D}^k$ with distance $\mathbb{E}_{u\sim\mathcal{D}^k}[d_{\mathcal{M}}(u)]$ to the manifold $\mathcal{M}$ (i.e., the distance between p and $\mathcal{M}$ equals the average distance from each point $u \in \mathcal{D}^k$ to $\mathcal{M}$).

Theorem 4.5 is incredibly useful for our task as it provides a way to possibly approximate the pointwise distance function $d_{\mathcal{M}}$. Note the set of maximizers of (4.2) is *not* unique. For example, if $f(u)$ is a maximizer, then for any $c \in \mathbb{R}$ so also is $g(u) := f(u)+c$. However, in the practical settings we have considered, we find that the gradient of our estimate of a maximizer of (4.2) adequately approximates the gradient of $d_{\mathcal{M}}$ on points in $\mathcal{D}$. In order to apply Theorem 4.5, we utilize the following assumption.

Assumption 4.6 *The parameter set $\mathcal{I}$ is such that $\{J_\theta\}_{\theta\in\mathcal{I}}$ forms the set of 1-Lipschitz functions.*

Remark 4.7 *In practice, Assumption 4.6 can be approximately implemented by, e.g., adding a gradient penalty to the loss function [33] or using sorting [4].*

Finally, together the above assumptions with the following conditions on the anchoring sequence $\{\gamma_k\}$, we state our main convergence result.

Assumption 4.8 *The sequence $\{\gamma_k\}$ satisfies the following properties: i) $\gamma_k \in (0, 1]$ for all $k \in \mathbb{N}$, ii) $\lim_{k\to\infty} \gamma_k = 0$, and iii) $\sum_{k\in\mathbb{N}} \gamma_k = \infty$.*

Theorem 4.9 (Convergence of Adversarial Projections) *Suppose Assumptions 4.1, 4.2, 4.3, 4.6, and 4.8 hold. If the sequence $\{u^k\}$ is generated by Algorithm 2 and the minimizer θ^k in Line 6 is chosen so that $J_{\theta^k} = -d_{\mathcal{M}}$ for all $k \in \mathbb{N}$ (as permitted by Theorem 4.5), then the sequence $\{u^k\}$ converges to $P_{\mathcal{M}}(u^1)$ in mean square.*

Recall that convergence in mean-square implies convergence in probability. And, by the definition of convergence in probability, this theorem implies that, given $\varepsilon > 0$, the probability that $\|u^k -$

$P_{\mathcal{M}}(u^1)\| > \varepsilon$ goes to zero as $k \to \infty$. That is,

$$\lim_{k\to\infty} \mathbb{P}\left[\{\omega \in \Omega : \|u^k(\omega) - P_{\mathcal{M}}(u^1)\| > \varepsilon\}\right] = 0. \tag{4.4}$$

In common language, this may be interpreted as saying the probability of the sequence $\{u^k\}$ "not converging to $P_{\mathcal{M}}(u^1)$" becomes smaller and smaller as the sequence progresses.

Below we present a lemma about the adversarial projections that can find use during training (i.e., in solving the minimization problem in Line 6 of Algorithm 1).

Lemma 4.10 *In the same setting as Theorem 4.9, for $\lambda_k > 0$, choosing*

$$\eta_k := \frac{1}{\lambda_k^2} \cdot \mathbb{E}_{u\sim\mathcal{D}^k}\left[\|g_k(u^k) - u^k\|^2\right] \tag{4.5}$$

yields an upper bound $\eta_k \in [0, 1]$ on the proportion of the distribution $\mathcal{D}^k$ that is not contained in the manifold $\mathcal{M}$, i.e.,

$$|\mathcal{D}^k - \mathcal{M}| \leqslant \eta_k \leqslant 1, \tag{4.6}$$

where $|\cdot|$ denotes the measure of the set.

Remark 4.11 *Note (4.5) can be abusively rewritten as*

$$\eta_k = \mathbb{E}_{u\sim\mathcal{D}^k}\left[\|\nabla J_{\theta^k}(u)\|^2\right], \tag{4.7}$$

where the abuse comes from the fact in practice each differentiation is performed numerically.

Remark 4.12 *During training, Lemma 4.10 can be used to determine when a good estimate of θ^k has been found. Initially, by evaluating (4.7) our estimate of η_k will be small (less than unity). As we get closer to the optimal weights θ^k, this lemma indicates that our estimate of η_k should increase as our estimate of J_{θ^k} is better able to distinguish between points in $\mathcal{D}^k$ and points in the manifold $\mathcal{M}$. We can then use the fact η_k is bounded to the interval $[0, 1]$ to identify a pair of stopping criteria. Namely, if $\{\eta_k^\ell\}$ is a sequence indexed by ℓ corresponding to the sequence of weight estimates $\{\theta_\ell^k\}$ of optimal weights θ^k, and if $\varepsilon_1, \varepsilon_2 \in (0, 1)$, then training can terminate if either*

1. *$\eta_k \geqslant 1 - \varepsilon_1$, or*
2. *$\eta_k^{\ell+1} \leqslant \eta_k^\ell + \varepsilon_2$.*

The first condition holds if there is negligible overlap of $\mathcal{D}^k$ and $\mathcal{M}$ and our estimate of η_k is close to the ideal value (unity). The second condition halts training when progress becomes small and this condition puts an upper bound on the number of epochs used to perform training (i.e., $\ell < 1/\varepsilon_2$).

5 Related Works

Our work bears connections with GANs [5, 32], and its applications to inverse problems [61]. Our approach can be viewed as training a GAN, except that rather than solving a minimax problem, we solve a sequence of minimization problems. In this case, J is the discriminator network that distinguishes between signals coming from the "fake" distribution (i.e., our approximate distribution) and

the true distribution, and g_η is the generator which tries to generate signals that resemble those from the true distribution.

Our work also bears connections with optimal transport [60, 66]. In particular, under certain assumptions (see Section 4), the adversarial projections can be interpreted as a subgradient flow that minimizes the Wasserstein-1 distance, where the function J corresponds to the Kantorovich potential [5, 46, 47, 52, 63], or in the context of mean field games and optimal control, the value function [46, 59]. Analogous to classical physics, the signals flow in a manner that minimize their potential energy. Our approach learns a sequence of these potential functions that project (or "flow") the distribution of signals towards the true distribution of signals.

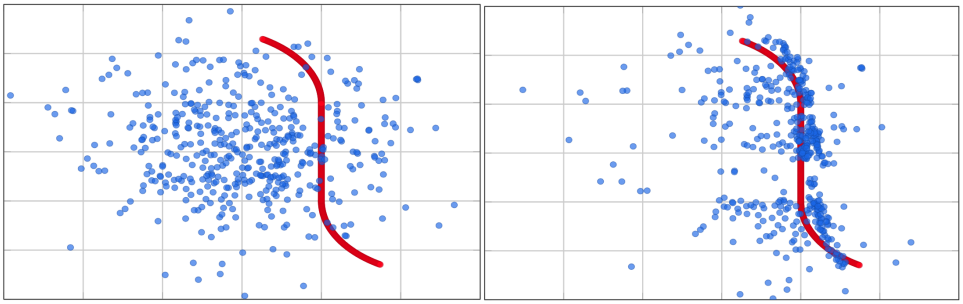
From an inverse problems perspective, our approach falls under the category of using deep learning to solve inverse problems [68]. One approach, known as post-processing, first applies a pseudo-inverse operator to the measurement data (e.g., FBP) and then learns a transformation in the image space. This approach has been investigated and found effective by several authors [20, 33, 41, 56]. Another approach is to learn a regularizer, and then use it in a classical variational reconstruction scheme according to (1.2). Other works investigate using dictionary learning [71], variational auto-encoders [51], and wavelet transforms [23] for these learned regularizers. Perhaps the most popular schemes are learned iterative algorithms such as gradient descent [2, 38, 44], proximal gradient descent or primal-dual algorithms [3, 62]. These iterative schemes are typically unrolled, and an "adaptive" iteration-dependent regularizer is learned. One key difference between adversarial projections and the aforementioned data-driven approaches is that our approach is *unsupervised*. That is, we do not need a correspondence between the measurement b and the true underlying signal $u^\star$. The adversarial projections simply requires a batch of true signals and a batch of measurements, regardless of whether these directly correspond to each other (i.e., an injective map between the two might not be available); this is especially useful in some applications (e.g., medical imaging) where the true image corresponding to the measurement is often not available.

Another set of work uses deep image priors (DIP) [8, 65], which attempt to parameterize the *signal* by a neural network. The weights are optimized by a gradient descent method that minimizes the data discrepancy of the output of the network. The authors in [8] show that combining DIPs with classical regularization techniques are effective in limited-data regimes.

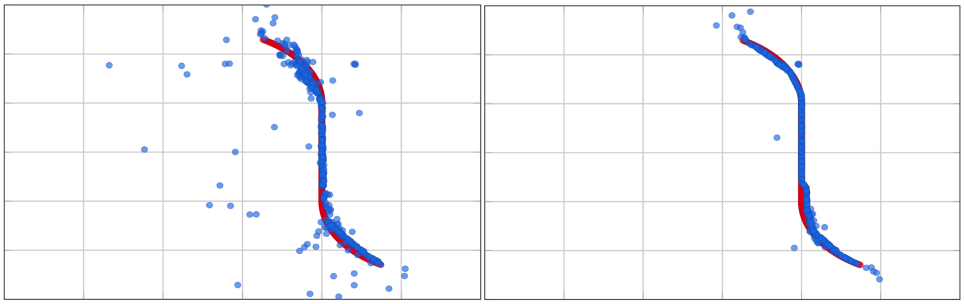
Our work is perhaps most similar to adversarial regularizers [50], where a regularizer J is trained in a GAN-like process (see Section 2.2). The regularizer learns to discriminate between FBP reconstructions and the training data, and it is used to reconstruct an approximate signal by solving the variational problem (1.3) using gradient descent (see Algorithms 1 and 2 in [50]). On the other hand, our approach learns a sequence of regularizers $\{J_{\theta^k}\}$. Each of these regularizers are used project the k^{th} distribution $\mathcal{D}^k$ toward the manifold $\mathcal{M}$ of true signals. Under certain assumptions (see Section 4), these regularizers can be viewed as potential functions that "flow" our current estimate signals to the distribution of true signals by performing a subgradient descent on the Wasserstein-1 distance at each iteration k. In our setting, each J_{θ^k} can be viewed as approximations to expert regularizer [30] (see Section 2.3) for the current distribution $\mathcal{D}^k$.

6 Numerical Experiments

In this section, we outline the potential of adversarial projections. We begin with a distributional illustration showing how adversarial projections project (or "flow") a distribution onto another. We then test our approach on computed tomography (CT) examples using two standard datasets: a synthetic dataset comprised of randomly generated ellipses as well as the Low-Dose Parallel Beam



(a) $\mathcal{D}^1$ (b) $\mathcal{D}^5$



(c) $\mathcal{D}^{25}$ (d) $\mathcal{D}^{300}$

Figure 2: Flow of distribution $\mathcal{D}^k$ toward the manifold $\mathcal{M}$ via Algorithm 1. Progressing from left to right and top to bottom, snapshots are provided at $k = 1, 5, 25, 300$. These plots agree with Theorem 4.9 in verifying that, as $k \to \infty$, the probability that each $u^k \in \mathcal{D}^k$ is also in $\mathcal{M}$ goes to unity.

(LoDoPaB) dataset [45]. We focus on the unsupervised learning setting, where we do not have a correspondence between the distribution of true signals and approximate signals. Therefore, we set adversarial regularizers (an unsupervised learning approach) as our benchmark. The quality of the image reconstructions are determined using the Peak Signal-To-Noise Ratio (PSNR) and structural similarity index measure (SSIM). For all experiments, we use the PyTorch deep learning framework [54] and the ADAM [43] optimizer. We also use the Operator Discretization Library (ODL) python library [1] to compute the TV and filtered backprojection (FBP) solutions. The experiments are run on a single NVIDIA TITAN X GPU with 12GB RAM.

Remark 6.1 *Although for practical reasons we consider linear inverse problems in our experiments, we emphasize that our presented methodology applies even when $u^\star$ is recovered from nonlinear measurements (i.e., when A is a nonlinear operator).*

6.1 Distributional Illustration

In this section, we show a toy example to provide intuition for the flow from an initial distribution estimate $\mathcal{D}^1$ to a true distribution $\mathcal{D}_{\text{true}}$ contained in a manifold $\mathcal{M}$. For simplicity, here we take $\mathcal{D}_{\text{true}} = \mathcal{M}$. To coincide with the assumption that the manifold $\mathcal{M}$ admits a lower dimensional representation, we let it take the form of a curve in 2D. The initial distribution $\mathcal{D}^1$ takes the form

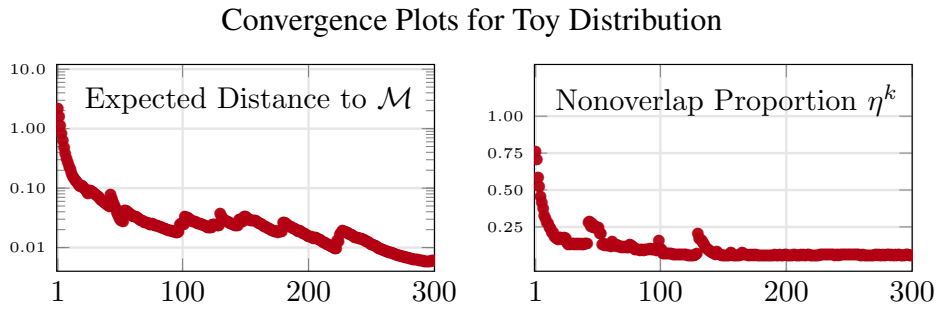


Figure 3: Convergence plots for illustration used to describe the flow of distribution $\mathcal{D}^k$ toward the manifold $\mathcal{M}$ in Figure 2. The expected distance between points and the manifold is given by (4.3), which, in this setting, is effectively equivalent to the Wasserstein-1 distance $\text{Wass}(\mathcal{D}^k, \mathcal{M})$. The nonoverlap proportion $\{\eta_k\}$ provides an upper bound on the measure of the difference between the distributions (i.e., $|\mathcal{D}^k - \mathcal{M}|$), and so its going to zero also provides an insight into convergence.

of a collection of finite samples from a Gaussian distribution in 2D. This is illustrated in Figure 2a. Here 600 samples are drawn from each distribution. Algorithm 1 is then used to successively update each point in $\mathcal{D}^k$ to flow all of the points toward the manifold. Snapshots of this flow are illustrated through the sequence of photos in Figure 2, which demonstrates that the expected distance converges to zero. Figure 3 provides a plot of the distance converging to zero and reveals that the nonoverlap proportion bound $\{\eta_k\}$ also decreases, as expected.

In this example, it is safe to suppose Assumptions 4.1 holds by our choice of $\mathcal{M}$ and roughly uniform sampling there. Although $\mathcal{D}^1$ was sampled from a Gaussian, 4.2 holds because we use finitely many samples. However, Assumption 4.3 does not actually hold, although it is "close" to being true. This is because most of the points in $\mathcal{D}^1$ would project directly onto the vertical portion of $\mathcal{M}$ rather than the curved ends. Despite this, the flow behaves as one would hope in the sense that it spreads $\mathcal{D}^1$ out to the points covering $\mathcal{M}$ while, roughly speaking, being close to their original projections onto the manifold. Assumption 4.6 approximately holds for this example since the network used follows the neural network structure proposed by [4], which possesses the property of being universal 1-Lipschitz function approximators as the number of parameters/layers increase. Lastly, we ensured Assumption 4.8 by choosing $\gamma_k = 1/k$.

6.2 Low-Dose Computed Tomography

We now demonstrate adversarial projections on two low-dose CT examples.

Ellipse Phantoms We use a synthetic dataset consisting of random phantoms of combined ellipses as in [2]. The images have a resolution of 128×128 pixels. Measurements are simulated with a parallel beam geometry with a sparse-angle setup of 30 angles and 183 projection beams. Moreover, we add Gaussian noise with a standard deviation of 2.5% of the mean absolute value of the projection data to the projection data. In total, the training set contains 10,000 pairs, while the validation and test set consist of 1,000 pairs each.

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.

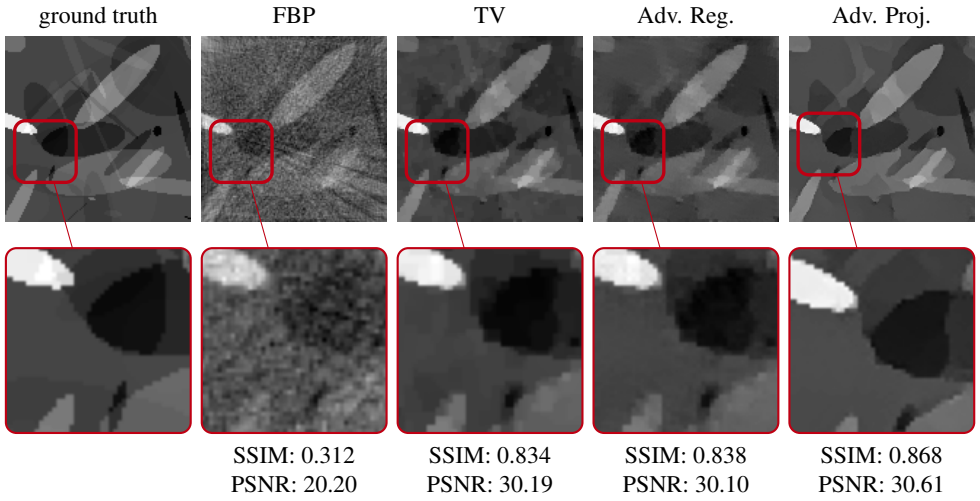


Figure 4: Reconstruction on a validation sample obtained with Filtered Back Projection (FBP) method, TV regularization, Adversarial Regularizer, and Adversarial Projections (left to right). Bottom row shows expanded version of corresponding cropped region indicated by red box.

Human Phantoms As a more realistic dataset, we use human phantoms consisting of chest CT scans from the Low-Dose Parallel Beam dataset (LoDoPaB) [45]. In our setup, we use 20,000 training images and 2,000 validation images of size 128×128. Similar to the ellipse phantoms, we simulate the data using 30 angles and 183 projection beams. We also add Gaussian noise with a standard deviation of 2.5% of the mean absolute value of the projection data to the projection data.

Network Structure We use a simple 5 layer neural network containing 38,534 trainable parameters. The first three being convolution layers with kernel size 4 and stride 2, with output channels $32, 64$, and 1 for layers one, two, and three, respectively. For the last two layers, we use fully connected layers to bring the dimensions back to a scalar. As nonlinear activation function,we choose the Parametric Rectified Linear Units (PReLU) functions

$$\sigma_c(x) = \begin{cases} x & \text{if } x \geqslant 0 \\ -cx & \text{else} \end{cases},$$

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.

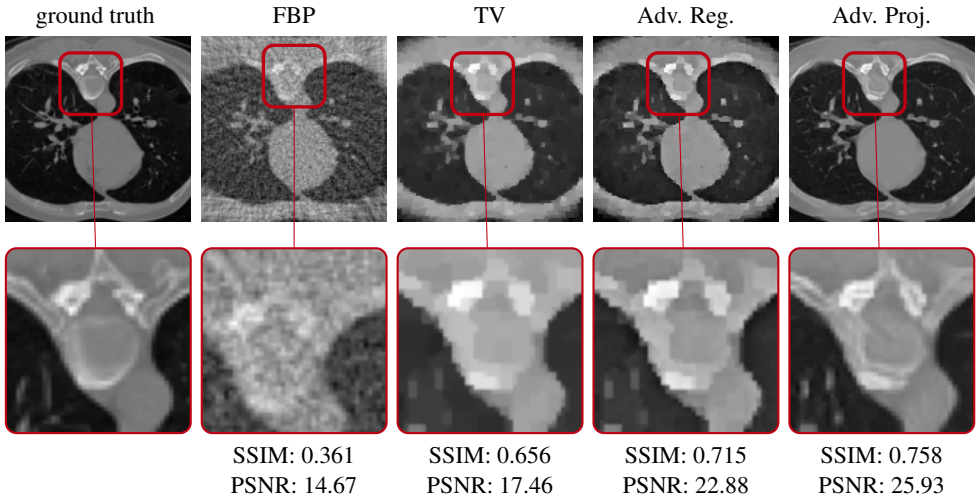


Figure 5: Reconstruction on a validation sample obtained with Filtered Back Projection (FBP) method, TV regularization, Adversarial Regularizer, and Adversarial Projections (left to right). Bottom row shows expanded version of corresponding cropped region indicated by red box.

which was shown to be effective in other applications such as classification [40].

For adversarial regularizers, we use the network structure described in [50], which consists of an 8-layer CNN with Leaky-Relu activation. The network contains 2,495,201 parameters. More details can be found in [50, Appendix B].

Training Setup To train the adversarial projections, we begin with an initial distribution obtained from the TV reconstructions. We update the distribution whenever 200 epochs have passed since the last update for both ellipses and human phantoms datasets. We also update the distributions if the conditions in Remark 4.12 is satisfied for $\eta_k = 10^{-4}$. As stopping criterion, we set a maximum of 50 iterations, i.e., generator updates, in Algorithm 1. We note that in practice, the number of epochs and η_k are hyperparameters that need to be tuned. These are of particular importance since it determines how well we approximate Line 6 in Algorithm 1.

In the ADAM optimizer, we use a learning rate of 10^{-5}, and use a batch size of 16 samples. To

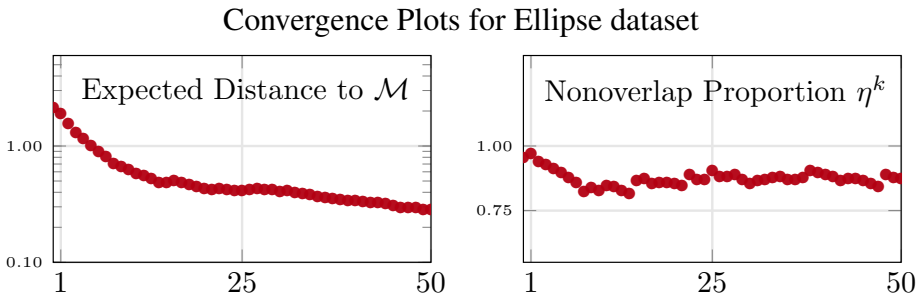


Figure 6: Convergence plots for the ellipse dataset. The expected distance between points and the manifold is given by (4.3), which, in this setting, is effectively equivalent to the Wasserstein-1 distance $\text{Wass}(\mathcal{D}^k, \mathcal{M})$. The nonoverlap proportion $\{\eta_k\}$ provides an upper bound on the measure of the difference between the distributions (i.e., $|\mathcal{D}^k - \mathcal{M}|$)

update the samples/distribution in Algorithm 1, we use step-size constant $\alpha = 0.5$. To ensure Assumption 4.8 is satisfied, we choose $\gamma_k = 10^{-8}/k$. Finally, to approximately satisfy Assumption 4.6, we enforce J to be 1-Lipschitz by adding a gradient penalty [33].

To train the adversarial regularizers, we use the code provided in [49]. Here, for the ellipses we use a learning rate of 10^{-4}, a batchsize of 16, and a gradient-norm-weight of 20, and for the LoDoPaB dataset we use a learning rate of 10^{-3}, a batchsize of 32, and a gradient-norm-weight of 1. We note that the setup for adversarial regularizers in [50] adds white Gaussian noise independent of the data, and is therefore different from our setup. As a result, we re-train the adversarial regularizers and tune the reconstruction parameters to the best of our ability. In particular, after the regularizer is trained, we tune the regularization parameter, stepsize, and number of gradient steps (see Algorithm 2 in [50]) for the highest PSNR. For a fair comparison, the adversarial regularizer is also trained on TV reconstructions as the initial distribution.

Experimental Results In Tables 1 and 2, we compare the average PSNR and SSIM on the validation datasets (1,000 images) for the ellipse dataset and LoDoPaB dataset (2,000 images), respectively. These results compare adversarial projections with FBP, TV, and adversarial regularizers.

We also show an ellipses image in Figure 4 and a LoDoPab image in Figure 5. For the adversarial regularizers, we find that using 25 steps with a stepsize of 0.05 and a regularization parameter of 2 leads to the highest PSNR on the ellipse dataset. Similarly, we find that using 25 steps with a stepsize of 0.01 and a regularization parameter of 2 leads to the highest PSNR on the LoDoPaB dataset.

In Figures 6 and 7, we observe that the approximate expected distance to the manifold (i.e., our approximation of the Wasserstein distance) decreases as we update the distribution. We also show the values of η_k, which provide a bound on the nonoverlap proportion.

While adversarial projections performs the best, we note that, e.g., some ellipses are not reconstructed in adversarial projections (this is also seen in adversarial regularizers). This is due to the fact that the initial TV reconstruction completely erases some ellipses due to the sparse angle setup. In this case, we have that some modes collapse, and Assumption 4.3 is not entirely satisfied. In particular, we obtain that the pushforward is simply a subset of the true manifold $\mathcal{M}$. More image reconstructions can be found in Appendix 8.2.

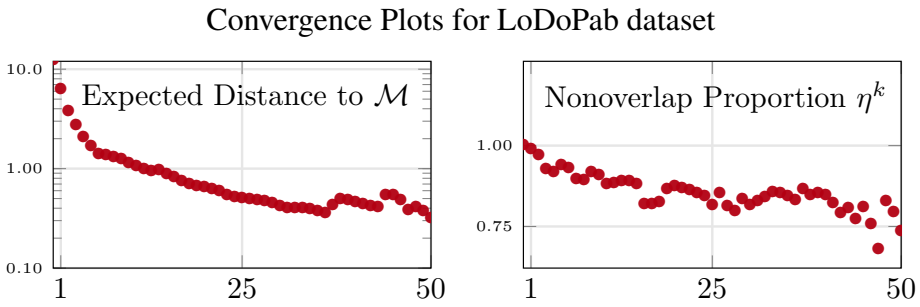


Figure 7: Convergence plots for the LoDoPab dataset. The expected distance between points and the manifold is given by (4.3), which, in this setting, is effectively equivalent to the Wasserstein-1 distance $\text{Wass}(\mathcal{D}^k, \mathcal{M})$. The nonoverlap proportion $\{\eta_k\}$ provides an upper bound on the measure of the difference between the distributions (i.e., $|\mathcal{D}^k - \mathcal{M}|$)

7 Conclusion

We present adversarial projections, a new framework for solving inverse problems. The main idea is that, by solving *unsupervised* learning problems, we can project signal estimates onto the underlying low-dimensional manifold of true signals. The training process consists of solving a sequence of minimization problems, which can be interpreted as training a sequence of discriminator networks that attempt to distinguish between signals in the approximate and true distributions. During implementation, our proposed algorithm forms a Halpern-type method with relaxed projections, which we prove converges in mean square to the projection of the initial estimate onto the manifold. At the level of individual signals, this work may also be interpreted as learned gradient descent with a sequence of expert-like regularizers. At the aggregate level of distributions, adversarial projections may be viewed as a subgradient method for minimizing the Wasserstein-1 distance between the distribution of initial estimates and the true distribution. Our numerical experiments show that adversarial projections outperform adversarial regularizers, a state-of-the-art unsupervised learning method for inverse problems. An extension to our work we intend to investigate the semi-supervised regime, where we have labels for some of the data, and to investigate inclusion of the measurement data into the projection scheme. We also intend to investigate guidelines on the design of more effective network architectures such as PDE-based neural networks [36, 58].

Acknowledgments

Samy Wu Fung, Alex Tong Lin, Stanley Osher, and Wotao Yin are supported by AFOSR MURI FA9550-18-1-0502, AFOSR Grant No. FA9550-18-1-0167, and ONR Grants N00014-18-1-2527 snf N00014-17-1-21. Howard Heaton's work was supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1650604. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Jonas Adler, Holger Kohr, and Ozan Öktem. Operator discretization library (odl), January 2017.

[2] Jonas Adler and Ozan Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.

[3] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

[4] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301, 2019.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[6] Simon R Arridge. Optical tomography in medical imaging. *Inverse problems*, 15(2):R41, 1999.

[7] Simon R Arridge and John C Schotland. Optical tomography: forward and inverse problems. *Inverse problems*, 25(12):123010, 2009.

[8] Daniel Otero Baguer, Johannes Leuschner, and Maximilian Schmidt. Computed tomography reconstruction using deep image prior and learned reconstruction methods. *arXiv preprint arXiv:2003.04989*, 2020.

[9] Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer International Publishing, 2017.

[10] Heinz H Bauschke, Patrick L Combettes, and D Russell Luke. Phase retrieval, error reduction algorithm, and fienup variants: a view from convex optimization. *JOSA A*, 19(7):1334–1345, 2002.

[11] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[12] Martin Benning, Guy Gilboa, Joana Sarah Grah, and Carola-Bibiane Schönlieb. Learning filter functions in regularisers by minimising quotients. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 511–523. Springer, 2017.

[13] Tan Bui-Thanh, Omar Ghattas, James Martin, and Georg Stadler. A computational framework for infinite-dimensional bayesian inverse problems part i: The linearized case, with application to global seismic inversion. *SIAM Journal on Scientific Computing*, 35(6):A2494–A2523, 2013.

[14] Daniela Calvetti and Lothar Reichel. Tikhonov regularization of large linear problems. *BIT Numerical Mathematics*, 43(2):263–283, 2003.

[15] Emmanuel J Candes, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski. Phase retrieval via matrix completion. *SIAM review*, 57(2):225–251, 2015.

[16] Emmanuel J Candes and Justin Romberg. Quantitative robust uncertainty principles and optimally sparse decompositions. *Foundations of Computational Mathematics*, 6(2):227–254, 2006.

[17] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[18] Andrzej Cegielski. *Iterative methods for fixed point problems in Hilbert spaces*, volume 2057. Springer, 2012.

[19] Raymond H Chan, Kelvin K Kan, Mila Nikolova, and Robert J Plemmons. A two-stage method for spectral–spatial classification of hyperspectral images. *Journal of Mathematical Imaging and Vision*, pages 1–18, 2020.

[20] Hu Chen, Yi Zhang, Mannudeep K Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. Low-dose ct with a residual encoder-decoder convolutional neural network. *IEEE transactions on medical imaging*, 36(12):2524–2535, 2017.

[21] Felipe Cucker and Steve Smale. Best choices for regularization parameters in learning theory: on the bias-variance problem. *Foundations of computational Mathematics*, 2(4):413–428, 2002.

[22] Frank Deutsch. *Best Approximation in Inner Product Spaces*, volume 7. Springer Science & Business Media, 2001.

[23] Ivan Dokmanić, Joan Bruna, Stéphane Mallat, and Maarten de Hoop. Inverse problems with invariant multiscale statistics. *arXiv preprint arXiv:1609.05502*, 2016.

[24] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[25] Samy Wu Fung. *Large-Scale Parameter Estimation in Geophysics and Machine Learning*. PhD thesis, Emory University, 2019.

[26] Samy Wu Fung and Zichao Wendy Di. Multigrid optimization for large-scale ptychographic phase retrieval. *SIAM Journal on Imaging Sciences*, 13(1):214–233, 2020.

[27] Samy Wu Fung and Lars Ruthotto. A multiscale method for model order reduction in PDE parameter estimation. *Journal of Computational and Applied Mathematics*, 350:19–34, 2019.

[28] Samy Wu Fung and Lars Ruthotto. An uncertainty-weighted asynchronous ADMM method for parallel PDE parameter estimation. *SIAM Journal on Scientific Computing*, 41(5):S129–S148, 2019.

[29] Aurél Galántai. *Projectors and projection methods*, volume 6. Springer Science & Business Media, 2003.

[30] Guy Gilboa. Expert regularizers for task specific processing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 24–35. Springer, 2013.

[31] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. Tikhonov regularization and total least squares. *SIAM journal on matrix analysis and applications*, 21(1):185–194, 1999.

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[33] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[34] E Haber, UM Ascher, DA Aruliah, and DW Oldenburg. Fast simulation of 3d electromagnetic problems using potentials. *Journal of Computational Physics*, 163(1):150–171, 2000.

[35] Eldad Haber, Uri M Ascher, and Douglas W Oldenburg. Inversion of 3d electromagnetic data in frequency and time domain using an inexact all-at-once approach. *Geophysics*, 69(5):1216–1228, 2004.

[36] Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.

[37] Benjamin Halpern. Fixed points of nonexpanding maps. *Bulletin of the American Mathematical Society*, 73(6):957–961, 1967.

[38] Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P Recht, Daniel K Sodickson, Thomas Pock, and Florian Knoll. Learning a variational network for reconstruction of accelerated MRI data. *Magnetic resonance in medicine*, 79(6):3055–3071, 2018.

[39] Per Christian Hansen, James G Nagy, and Dianne P O'leary. *Deblurring images: matrices, spectra, and filtering*. SIAM, 2006.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[41] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[42] Kelvin Kan, Samy Wu Fung, and Lars Ruthotto. PNKH-B: A projected newton-krylov method for large-scale bound-constrained optimization. *arXiv preprint arXiv:2005.13639*, 2020.

[43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[44] Erich Kobler, Teresa Klatzer, Kerstin Hammernik, and Thomas Pock. Variational networks: connecting variational methods and deep learning. In *German conference on pattern recognition*, pages 281–293. Springer, 2017.

[45] Johannes Leuschner, Maximilian Schmidt, Daniel Otero Baguer, and Peter Maaß. The LoDoPaB-CT dataset: A benchmark dataset for low-dose CT reconstruction methods. *arXiv preprint arXiv:1910.01113*, 2019.

[46] Alex Tong Lin, Samy Wu Fung, Wuchen Li, Levon Nurbekyan, and Stanley J Osher. APAC-Net: Alternating the population and agent control via two neural networks to solve high-dimensional stochastic mean field games. *arXiv preprint arXiv:2002.10113*, 2020.

[47] Jingrong Lin, Keegan Lensink, and Eldad Haber. Fluid flow mass transport for generative networks. *arXiv preprint arXiv:1910.01694*, 2019.

[48] Li-Shan Liu. for nonlinear strongly accretive mappings in banach spaces. *Journal of Mathematical Analysis and Applications*, 194:114–125, 1995.

[49] Sebastian Lunz. https://github.com/lunz-s/DeepAdverserialRegulariser, 2018.

[50] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. Adversarial regularizers in inverse problems. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8507–8516. Curran Associates, Inc., 2018.

[51] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1781–1790, 2017.

[52] Derek Onken, Samy Wu Fung, Xingjian Li, and Lars Ruthotto. OT-Flow: Fast and accurate continuous normalizing flows via optimal transport. *arXiv preprint arXiv:2006.00104*, 2020.

[53] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[54] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[55] Simeon Reich. Constructive techniques for accretive and monotone operators. In *Applied nonlinear analysis*, pages 335–345. Elsevier, 1979.

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[57] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[58] Lars Ruthotto and Eldad Haber. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, pages 1–13, 2019.

[59] Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.

[60] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[61] Viraj Shah and Chinmay Hegde. Solving linear inverse problems using gan priors: An algorithm with provable guarantees. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4609–4613. IEEE, 2018.

[62] Jian Sun, Huibin Li, Zongben Xu, et al. Deep ADMM-Net for compressive sensing MRI. In *Advances in neural information processing systems*, pages 10–18, 2016.

[63] Akinori Tanaka. Discriminator optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS*, pages 6813–6823, 2019.

[64] Nguyen Hieu Thao, David Russell Luke, Oleg Soloviev, and Michel Verhaegen. Phase retrieval with sparse phase constraint. *SIAM Journal on Mathematics of Data Science*, 2(1):246–263, 2020.

[65] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[66] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[67] Ernesto De Vito, Lorenzo Rosasco, Andrea Caponnetto, Umberto De Giovannini, and Francesca Odone. Learning from examples as an inverse problem. *Journal of Machine Learning Research*, 6(May):883–904, 2005.

[68] Ge Wang. A perspective on deep imaging. *IEEE access*, 4:8914–8924, 2016.

[69] Samy Wu Fung, Sanna Tyrväinen, Lars Ruthotto, and Eldad Haber. ADMM-softmax: An ADMM approach for multinomial logistic regression. *Electronic Transactions on Numerical Analysis*, 52:214–229, 2020.

[70] Hong-Kun Xu. Iterative algorithms for nonlinear operators. *Journal of the London Mathematical Society*, 66(1):240–256, 2002.

[71] Qiong Xu, Hengyong Yu, Xuanqin Mou, Lei Zhang, Jiang Hsieh, and Ge Wang. Low-dose x-ray CT reconstruction via dictionary learning. *IEEE transactions on medical imaging*, 31(9):1682–1697, 2012.

[72] Yonghong Yao, Mihai Postolache, and Shahzad Naseer. Strong convergence of halpern method for firmly type nonexpansive mappings. *Journal of Nonlinear Science and Applications*, 10:5932–5938, 2017.

[73] Wotao Yin, Stanley Osher, Donald Goldfarb, and Jerome Darbon. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM Journal on Imaging sciences*, 1(1):143–168, 2008.

8 Appendix

8.1 Proofs

We begin with the elementary result stated in Section 3.

Lemma 8.1 *Let $u \in \mathcal{X}$ and $\alpha \in \mathbb{R}$. If $\lambda = \alpha \cdot d_{\mathcal{M}}(u)$, then the inclusion relation in (3.3) holds.*

Proof: First consider the case where $u \in \mathcal{M}$. Since $d_{\mathcal{M}}$ is a metric, it is nonnegative. And,

$$d_{\mathcal{M}}(u) = \inf_{v \in \mathcal{M}} \|v - u\| \leqslant \|u - u\| = 0 \quad \Longrightarrow \quad d_{\mathcal{M}}(u) = 0. \tag{8.1}$$

Thus, u is a minimizer of $d_{\mathcal{M}}$. Because $d_{\mathcal{M}}$ is convex and u is a minimizer, it follows that $0 \in \partial d_{\mathcal{M}}(u)$. Additionally, $P_{\mathcal{M}}(u) = u$. Combining these results reveals

$$u + \alpha \left(P_{\mathcal{M}}(u) - u\right) = u + \alpha \left(u - u\right) = u + 0 \in u - \lambda \partial d_{\mathcal{M}}(u). \tag{8.2}$$

Now suppose $u \notin \mathcal{M}$. Then $d_{\mathcal{M}}(u) > 0$ and, by Lemma 2.2.28 in [18],

$$\nabla d_{\mathcal{M}}(u) = \frac{u - P_{\mathcal{M}}(u)}{\|u - P_{\mathcal{M}}(u)\|} = \frac{u - P_{\mathcal{M}}(u)}{d_{\mathcal{M}}(u)}. \tag{8.3}$$

Thus, direct substitution reveals

$$u - \lambda \nabla d_{\mathcal{M}}(u) = u + \alpha \left(P_{\mathcal{M}}(u) - u\right), \tag{8.4}$$

and the proof is complete. ■

Below we restate and prove the lemma about the sequence $\{\eta_k\}$.

Lemma 4.10 *In the same setting as Theorem 4.9, for $\lambda_k > 0$, choosing*

$$\eta_k := \frac{1}{\lambda_k^2} \cdot \mathbb{E}_{u \sim \mathcal{D}^k} \left[\|g_k(u^k) - u^k\|^2\right] \tag{8.5}$$

yields an upper bound $\eta_k \in [0, 1]$. This η_k represents the proportion of the distribution $\mathcal{D}^k$ that is not contained in the manifold $\mathcal{M}$, i.e.,

$$|\mathcal{D}^k - \mathcal{M}| \leqslant \eta_k \leqslant 1, \tag{8.6}$$

where $|\cdot|$ denotes the measure of the set.
Proof: For notational brevity, below we write $u^k = u^k(\omega)$ and use $\alpha_k = \alpha_k(u)$, as defined in (3.10). For each $u^k \notin \mathcal{M}$, observe that

$$\begin{aligned}
\|g_k(u^k) - u^k\|^2 &= \|\alpha_k(u^k)(P_{\mathcal{M}}(u^k) - u^k)\|^2 && (8.7)\\
&= \alpha_k^2(u^k) d_{\mathcal{M}}(u^k)^2 && (8.8)\\
&= \frac{\lambda_k^2}{d_{\mathcal{M}}(u^k)^2} \cdot d_{\mathcal{M}}(u^k)^2 && (8.9)\\
&= \lambda_k^2. && (8.10)
\end{aligned}$$

And, if $u^k \in \mathcal{M}$, then $g_k(u^k) = u^k$. Thus, for $\lambda_k > 0$,

$$\frac{1}{\lambda_k^2} \cdot \|g_k(u^k) - u^k\|^2 = \begin{cases} 0 & \text{if } u^k \in \mathcal{M}, \\ 1 & \text{if } u^k \notin \mathcal{M}, \end{cases} \tag{8.11}$$

Hence taking the expectation yields

$$\mathbb{E}_{\omega\sim\Omega}\left[\frac{1}{\lambda_k^2} \cdot \|g_k(u^k) - u^k\|^2\right] = \mathbb{P}\left[\{\omega \in \Omega : u^k(\omega) \in \mathcal{M}\}\right] \cdot 0 + \mathbb{P}\left[\{\omega \in \Omega : u^k(\omega) \notin \mathcal{M}\}\right] \cdot 1 \tag{8.12}$$

$$= \mathbb{P}\left[\{\omega \in \Omega : u^k(\omega) \notin \mathcal{M}\}\right] \tag{8.13}$$

$$= |\mathcal{D}^k - \mathcal{M}|, \tag{8.14}$$

which gives the first equality for η_k. Since $|\mathcal{D}^k| = 1$, the final inequality also holds, and the proof is complete. ■

The following lemma can be found in various forms in the literature (e.g., see [48, 55, 70]).

Lemma 8.2 *If $\{\delta_n\}$ is a sequence of nonnegative real numbers such that*

$$\delta_{k+1} \leqslant (1 - \gamma_k)\delta_k + \gamma_k\sigma_k, \quad \textit{for all } k \in \mathbb{N}, \tag{8.15}$$

where $\{\gamma_k\}$ is a sequence in $(0, 1]$ and $\{\sigma_k\}$ is a sequence in $\mathbb{R}$ such that

$$\sum_{k\in\mathbb{N}} \gamma_k = \infty \tag{8.16}$$

and

$$\limsup_{k\to\infty} \sigma_k \leqslant 0, \tag{8.17}$$

then

$$\lim_{k\to\infty} \delta_k = 0. \tag{8.18}$$

Below is a proof of the main result, Theorem 4.9. The analysis for the Halpern iteration closely follows the approach in [72]. For completeness, we first restate the theorem.

Theorem 4.9 (Convergence of Adversarial Projections) *Suppose Assumptions 4.1, 4.2, 4.3, 4.6, and 4.8 hold. If the sequence $\{u^k\}$ is generated by Algorithm 2 and the minimizer θ^k in Line 6 is chosen so that $J_{\theta^k} = -d_{\mathcal{M}}$ for all $k \in \mathbb{N}$ (as permitted by Theorem 4.5), then the sequence $\{u^k\}$ converges to $P_{\mathcal{M}}(u^1)$ in mean square.*

Proof: Before beginning the proof, we define the following quantities that are used throughout. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, where Ω, $\mathcal{F}$, and $\mathbb{P}$ are the sample space, σ-algebra, and probability measure, respectively. We take $\mathcal{D}^1 = \{u^1(\omega) : \omega \in \Omega\}$, $z(\omega) := P_{\mathcal{M}}(u^1)$, and, for all $k \in \mathbb{N}$,

$$\delta_k := \mathbb{E}_{\omega\sim\Omega}\left[\|u^k - z\|^2\right], \tag{8.19}$$

$$d_k := \mathbb{E}_{\omega\sim\Omega}\left[\|u^k - z\|\right], \tag{8.20}$$

$$\sigma_k := \mathbb{E}_{\omega\sim\Omega}\left[2\langle u^{k+1} - z, u^1 - z\rangle\right] - \frac{1 - \gamma_k}{\gamma_k} \cdot \alpha(2 - \alpha)d_k^2. \tag{8.21}$$

For notational brevity, below we write $u^k = u^k(\omega)$, $z = z(\omega)$, and similarly for other quantities defined in terms of these expressions. We also use $\alpha_k = \alpha_k(u)$, as defined in (3.10). Note α is a fixed constant whereas $\alpha_k(u)$ varies by iteration k and iterate u.

We proceed in the following manner. First an inequality is derived bounding the expectation of $\|g_k(u^k) - z\|^2$ (Step 1). This is used to show the sequence $\{\delta_k\}$ is bounded (Step 2) and then obtain an inequality relating δ_{k+1}, δ_k, and σ_k as in (8.15) (Step 3). We next verify the limit supremum of the sequence $\{\sigma_k\}$ is finite (Step 4), which enables us to deduce that a subsequence of $\{d_k\}$ converges to zero (Step 5). This implies $\limsup_{k\to\infty} \sigma_k \leqslant 0$, and so $\delta_k \to 0$ (Step 6), completing the proof.

Step 1. We first derive a descent inequality for relaxed projections. Define the residual operator

$$S(u) := u - P_{\mathcal{M}}(u). \tag{8.22}$$

Fix any $k \in \mathbb{N}$. Using (3.10) with $\alpha_k = \alpha_k(u)$, observe

$$\begin{aligned}
\|g_k(u^k) - z\|^2 &= \|u^k + \alpha_k(P_{\mathcal{M}}(u^k) - u^k) - z\|^2 && (8.23)\\
&= \|u^k - z\|^2 - 2\alpha_k \langle u^k - z, u^k - P_{\mathcal{M}}(u^k)\rangle + \alpha_k^2 \|u^k - P_{\mathcal{M}}(u^k)\|^2 && (8.24)\\
&= \|u^k - z\|^2 - 2\alpha_k \langle u^k - z, S(u^k) - S(z)\rangle + \alpha_k^2 \|S(u^k)\|^2, && (8.25)
\end{aligned}$$

where we note $S(z) = 0$ since $z = P_{\mathcal{M}}(u^1) \in \mathcal{M}$. Furthermore, S is firmly nonexpansive (e.g., see Prop. 4.16 in [9]), which implies

$$\langle u^k - z, S(u^k) - S(z)\rangle \geqslant \|S(u^k) - S(z)\|^2 = \|S(u^k)\|^2. \tag{8.26}$$

Combining (8.25) and (8.26) yields

$$\begin{aligned}
\|g_k(u^k) - z\|^2 &\leqslant \|u^k - z\|^2 - \alpha_k(2 - \alpha_k)\|S(u^k)\|^2 && (8.27)\\
&= \|u^k - z\|^2 - \alpha_k(2 - \alpha_k) d_{\mathcal{M}}(u^k)^2 && (8.28)\\
&= \|u^k - z\|^2 - \lambda_k(2 d_{\mathcal{M}}(u^k) - \lambda_k). && (8.29)
\end{aligned}$$

Thus, taking the expectation of (8.29),

$$\begin{aligned}
\mathbb{E}_{\omega\sim\Omega}\left[\|g_k(u^k) - z\|^2\right] &\leqslant \mathbb{E}_{\omega\sim\Omega}\left[\|u^k - z\|^2\right] - \mathbb{E}_{\omega\sim\Omega}\left[\lambda_k(2 d_{\mathcal{M}}(u^k) - \lambda_k)\right] && (8.30)\\
&\leqslant \mathbb{E}_{\omega\sim\Omega}\left[\|u^k - z\|^2\right] - \lambda_k(2 d_k - \lambda_k) && (8.31)\\
&= \mathbb{E}_{\omega\sim\Omega}\left[\|u^k - z\|^2\right] - \alpha(2 - \alpha) d_k^2 && (8.32)\\
&= \delta_k - \alpha(2 - \alpha) d_k^2. && (8.33)
\end{aligned}$$

The first equality above holds by the definition of λ_k in Line 7 of Algorithm 1 and d_k in (8.20), noting that $J_{\theta^k} = -d_{\mathcal{M}}$ and recalling (4.3). Also, the rightmost term in the final line (8.33) is nonnpositive since $\alpha \in (0, 2)$.

Step 2. Expanding the expression for δ_{k+1}, we deduce

$$\begin{aligned}
\delta_{k+1} &= \mathbb{E}_{\omega\sim\Omega}\left[\|u^{k+1}-z\|^2\right] && (8.34)\\
&= \mathbb{E}_{\omega\sim\Omega}\left[\|\gamma_k u^1+(1-\gamma_k)g_k(u^k)-z\|^2\right] && (8.35)\\
&\leqslant \mathbb{E}_{\omega\sim\Omega}\left[\gamma_k\|u^1-z\|^2+(1-\gamma_k)\|g_k(u^k)-z\|^2\right] && (8.36)\\
&= \gamma_k\cdot\mathbb{E}_{\omega\sim\Omega}\left[\|u^1-z\|^2\right]+(1-\gamma_k)\mathbb{E}_{\omega\sim\Omega}\left[\|g_k(u^k)-z\|^2\right] && (8.37)\\
&\leqslant \gamma_k\delta_1+(1-\gamma_k)\delta_k && (8.38)\\
&\leqslant \max\left(\delta_1,\delta_k\right), && (8.39)
\end{aligned}$$

where (8.36) follows from (8.35) by Jensen's inequality, (8.38) holds by applying (8.33), and the final inequality holds by Assumption 4.8i. Through induction, it follows that $\{\delta_k\}$ is bounded since

$$\delta_{k+1}\leqslant\delta_1<\infty,\quad\text{for all } k\in\mathbb{N}. \tag{8.40}$$

Step 3. To establish a useful inequality bounding δ_{k+1}, we expand this expression once again to obtain, for all $k\in\mathbb{N}$,

$$\begin{aligned}
\delta_{k+1} &= \mathbb{E}_{\omega\sim\Omega}\left[\|u^{k+1}-z\|^2\right] && (8.41)\\
&= \mathbb{E}_{\omega\sim\Omega}\left[\|\gamma_k u^1+(1-\gamma_k)g_k(u^k)-z\|^2\right] && (8.42)\\
&= \mathbb{E}_{\omega\sim\Omega}\left[\gamma_k^2\|u^1-z\|^2+(1-\gamma_k)^2\|g_k(u^k)-z\|^2+2\gamma_k(1-\gamma_k)\langle u^1-z,g_k(u^k)-z\rangle\right] && (8.43)\\
&\leqslant (1-\gamma_k)\cdot\mathbb{E}_{\omega\sim\Omega}\left[\|g_k(u^k)-z\|^2\right]+2\gamma_k\cdot\mathbb{E}_{\omega\sim\Omega}\left[\langle u^{k+1}-z,u^1-z\rangle\right] && (8.44)\\
&\leqslant (1-\gamma_k)\left(\delta_k-\alpha(2-\alpha)d_k^2\right)+2\gamma_k\cdot\mathbb{E}_{\omega\sim\Omega}\left[\langle u^{k+1}-z,u^1-z\rangle\right] && (8.45)\\
&= (1-\gamma_k)\delta_k+\gamma_k\left[\mathbb{E}_{\omega\sim\Omega}\left[2\langle u^{k+1}-z,u^1-z\rangle\right]-\frac{1-\gamma_k}{\gamma_k}\cdot\alpha(2-\alpha)d_k^2\right], && (8.46)
\end{aligned}$$

where we leverage the definition of u^{k+1} and the inclusions $\gamma_k,(1-\gamma_k)\in[0,1]$. Substituting the definition of σ_k from (8.21) into (8.46) yields the inequality

$$\delta_{k+1}\leqslant(1-\gamma_k)\delta_k+\gamma_k\sigma_k,\quad\text{for all } k\in\mathbb{N}. \tag{8.47}$$

Step 4. We now show the limit supremum of $\{\sigma_k\}$ is finite. Indeed, the fact that, for all $k\in\mathbb{N}$,

$$\begin{aligned}
\sigma_k &\leqslant \mathbb{E}_{\omega\sim\Omega}\left[2\langle u^{k+1}-z,u^1-z\rangle\right] && (8.48)\\
&\leqslant \mathbb{E}_{\omega\sim\Omega}\left[\|u^{k+1}-z\|^2+\|u^1-z\|^2\right] && (8.49)\\
&= \delta_{k+1}+\delta_1 && (8.50)\\
&\leqslant 2\delta_1 && (8.51)\\
&< \infty && (8.52)
\end{aligned}$$

implies

$$\limsup_{k\to\infty}\sigma_k<\infty. \tag{8.53}$$

Next, by way of contradiction, suppose

$$\limsup_{k\to\infty}\sigma_k<-1. \tag{8.54}$$

This implies there exists $N_1 \in \mathbb{N}$ such that

$$\sigma_k \leqslant -1, \quad \text{for all } k \geqslant N_1, \tag{8.55}$$

and so

$$\delta_{k+1} \leqslant (1-\gamma_k)\delta_k - \gamma_k \leqslant \delta_k - \gamma_k, \quad \text{for all } k \geqslant N_1. \tag{8.56}$$

By induction, it follows that

$$\delta_{k+1} \leqslant \delta_{N_1} - \sum_{\ell=N_1}^{k} \gamma_\ell. \tag{8.57}$$

Applying Assumption 4.8iii and letting $k \to \infty$ reveals

$$\limsup_{k\to\infty} \delta_k \leqslant \delta_{N_1} - \sum_{\ell=N_1}^{\infty} \gamma_\ell = -\infty, \tag{8.58}$$

which induces a contradiction since the sequence $\{\delta_k\}$ is nonnegative. This proves (8.54) is false, and so

$$-1 \leqslant \limsup_{k\to\infty} \sigma_k < \infty. \tag{8.59}$$

Step 5. Because (8.59) shows the limit supremum of the sequence $\{\sigma_k\}$ is finite, there is a convergent subsequence $\{\sigma_{n_k}\} \subseteq \{\sigma_k\}$ satisfying

$$\limsup_{k\to\infty} \sigma_k = \lim_{k\to\infty} \sigma_{n_k} \tag{8.60}$$

$$= \lim_{k\to\infty} \left[\mathbb{E}_{\omega\sim\Omega} \left[2\langle u^{n_k+1} - z, u^1 - z\rangle \right] - \frac{1-\gamma_{n_k}}{\gamma_{n_k}} \cdot \alpha(2-\alpha) d_{n_k}^2 \right]. \tag{8.61}$$

By the result (8.40) in Step 1,

$$\mathbb{E}_{\omega\sim\Omega} \left[\left| \langle u^{n_k+1} - z, u^1 - z\rangle \right| \right] \leqslant \mathbb{E}_{\omega\sim\Omega} \left[\frac{1}{2} \left(\|u^{n_k+1} - z\|^2 + \|u^1 - z\|^2 \right) \right] \tag{8.62}$$

$$= \frac{1}{2} \left(\delta_{n_k+1} + \delta_1 \right) \tag{8.63}$$

$$\leqslant \delta_1, \tag{8.64}$$

and so $\{\mathbb{E}_{\omega\sim\Omega}[\langle u^{n_k+1} - z, u^1 - z\rangle]\}$ is a bounded sequence of real numbers. Thus, it contains a convergent subsequence $\{\langle u^{m_k+1} - z, u^1 - z\rangle\}$ (i.e., $\{m_k\} \subseteq \{n_k\}$). This implies, when combined with the convergence of $\{\sigma_{m_k}\}$ and (8.61), existence of the limit

$$\lim_{k\to\infty} \frac{1-\gamma_{m_k}}{\gamma_{m_k}} \cdot \alpha(2-\alpha) d_{m_k}^2. \tag{8.65}$$

Since Assumption 4.8ii asserts $\gamma_k \to 0$, it follows that

$$\lim_{k\to\infty} (1-\gamma_{m_k})\alpha(2-\alpha) d_{m_k}^2 = 0 \quad \Longrightarrow \quad \lim_{k\to\infty} d_{m_k} = 0, \tag{8.66}$$

i.e., a subsequence $\{d_{m_k}\}$ of $\{d_k\}$ converges to zero.

Step 6. Observe, for all $k \in \mathbb{N}$,

$$\begin{aligned}
\mathbb{E}_{\omega\sim\Omega}\left[\|u^{k+1}-u^k\|\right] &\leqslant \gamma_k \mathbb{E}_{\omega\sim\Omega}\left[\|u^k-u^1\|\right] + (1-\gamma_k)\mathbb{E}_{\omega\sim\Omega}\left[\|g_k(u^k)-u^k\|\right] && (8.67)\\
&\leqslant \gamma_k d_1 + (1-\gamma_k)\,\mathbb{E}_{\omega\sim\Omega}\left[\lambda_k\right] && (8.68)\\
&= \gamma_k d_1 + (1-\gamma_k)\lambda_k && (8.69)\\
&= \gamma_k d_1 + (1-\gamma_k)\alpha d_k, && (8.70)
\end{aligned}$$

where (8.68) holds since, by the choice of g_k in (3.10),

$$g_k(u) - u \in \lambda_k \partial d_{\mathcal{M}}(u) \quad\Longrightarrow\quad \|g_k(u)-u\| \leqslant \lambda_k, \tag{8.71}$$

with the implication following from the fact that $\partial d_{\mathcal{M}}(u)$ is a subset of the unit ball centered at the origin since $d_{\mathcal{M}}$ is 1-Lipschitz.
Utilizing (8.66) and the fact $\gamma_{m_k} \to 0$, we deduce

$$\lim_{k\to\infty} \mathbb{E}_{\omega\sim\Omega}\left[\|u^{m_k+1}-u^{m_k}\|\right] \leqslant \lim_{k\to\infty} \gamma_{m_k} d_1 + (1-\gamma_{m_k})\alpha d_{m_k} = 0. \tag{8.72}$$

Because the left hand side is nonnegative, the squeeze lemma implies

$$\lim_{k\to\infty} \mathbb{E}_{\omega\sim\Omega}\left[\|u^{m_k+1}-u^{m_k}\|\right] = 0. \tag{8.73}$$

Also, noting that the boundedness of $\mathcal{D}^1$ and $\mathcal{M}$ implies there exists a constant $C > 0$ such that

$$2\|u^1(\omega)-z(\omega)\| \leqslant 2(\|u^1(\omega)\| + \|z(\omega)\|) \leqslant C, \quad \text{for all } \omega \in \Omega, \tag{8.74}$$

we deduce

$$\begin{aligned}
\limsup_{k\to\infty} \sigma_k &= \lim_{k\to\infty} \sigma_{m_k} && (8.75)\\
&= \lim_{k\to\infty} \left[\mathbb{E}_{\omega\sim\Omega}\left[2\langle u^{m_k+1}-z, u^1-z\rangle\right] - \frac{1-\gamma_{m_k}}{\gamma_{m_k}}\cdot\alpha(2-\alpha)d_{m_k}^2\right] && (8.76)\\
&\leqslant \lim_{k\to\infty} \mathbb{E}_{\omega\sim\Omega}\left[2\langle u^{m_k+1}-z, u^1-z\rangle\right] && (8.77)\\
&= \lim_{k\to\infty} \mathbb{E}_{\omega\sim\Omega}\left[2\langle P_{\mathcal{M}}(u^{m_k+1})-z, u^1-z\rangle\right] && (8.78)\\
&+ \mathbb{E}_{\omega\sim\Omega}\left[2\langle u^{m_k+1}-P_{\mathcal{M}}(u^{m_k+1}), u^1-z\rangle\right] && (8.79)\\
&\leqslant \lim_{k\to\infty} C\cdot\mathbb{E}_{\omega\sim\Omega}\left[\|u^{m_k+1}-P_{\mathcal{M}}(u^{m_k+1})\|\right], && (8.80)
\end{aligned}$$

where the final inequality holds by application of the Cauchy Schwarz inequality, and utilizing the fact that $z = P(u^1)$ and, by the projection identity (e.g., see Thm. 3.16 in [9], Thm 4.1 [22], and Thm 7.45 in [29]),

$$\langle v - P_{\mathcal{M}}(u^1), u^1 - P_{\mathcal{M}}(u^1)\rangle \leqslant 0, \quad \text{for all } v \in \mathcal{M}. \tag{8.81}$$

Then applying the triangle inequality and using the fact the projection $P_{\mathcal{M}}$ is 1-Lipschitz yields

$$\begin{aligned}
\limsup_{k\to\infty} \sigma_k &\leqslant \lim_{k\to\infty} C\cdot\mathbb{E}_{\omega\sim\Omega}\left[\|u^{m_k+1}-u^{m_k}\| + \|u^{m_k}-P_{\mathcal{M}}(u^{m_k})\|\right] && (8.82)\\
&+ \mathbb{E}_{\omega\sim\Omega}\left[\|P_{\mathcal{M}}(u^{m_k})-P_{\mathcal{M}}(u^{m_k+1})\|\right] && (8.83)\\
&\leqslant \lim_{k\to\infty} C\cdot\mathbb{E}_{\omega\sim\Omega}\left[2\|u^{m_k+1}-u^{m_k}\| + \|u^{m_k}-P_{\mathcal{M}}(u^{m_k})\|\right] && (8.84)\\
&= \lim_{k\to\infty} C\cdot\left(\mathbb{E}_{\omega\sim\Omega}\left[2\|u^{m_k+1}-u^{m_k}\|\right] + d_{m_k}\right) && (8.85)\\
&= 0, && (8.86)
\end{aligned}$$

where (8.86) follows from (8.85) by (8.66) and (8.73). Now, since the limit supremum of $\{\sigma_k\}$ is nonpositive, we may apply Lemma 8.2 to (8.47) to deduce

$$\delta_k \to 0 \quad \Longrightarrow \quad \lim_{k\to\infty} \mathbb{E}_{\omega\sim\Omega}\left[\|u^k - z\|^2\right] = 0, \tag{8.87}$$

completing the proof. ■

8.2 More Reconstructions

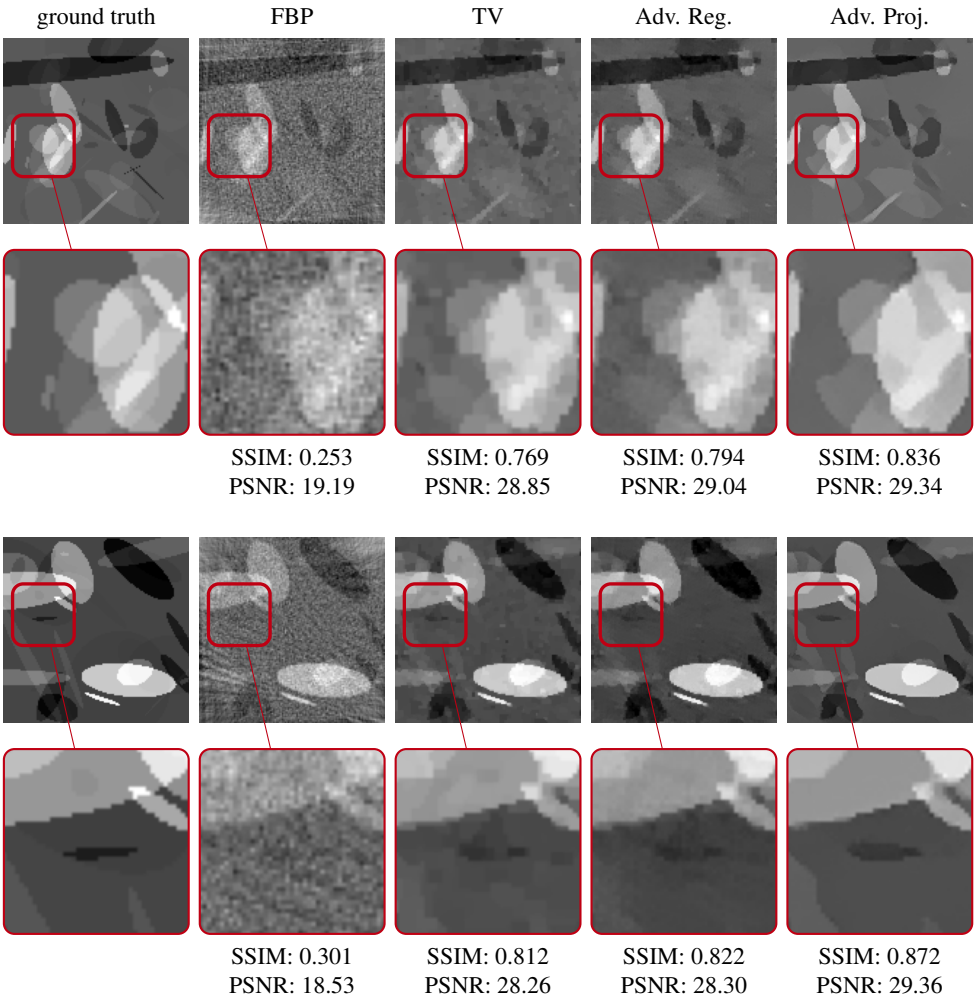


Figure 8: Additional ellipse reconstructions on a validation sample obtained with Filtered Back Projection (FBP) method, TV regularization, Adversarial Regularizer, and Adversarial Projections (left to right). Bottom row shows corresponding cropped region indicated by red box.

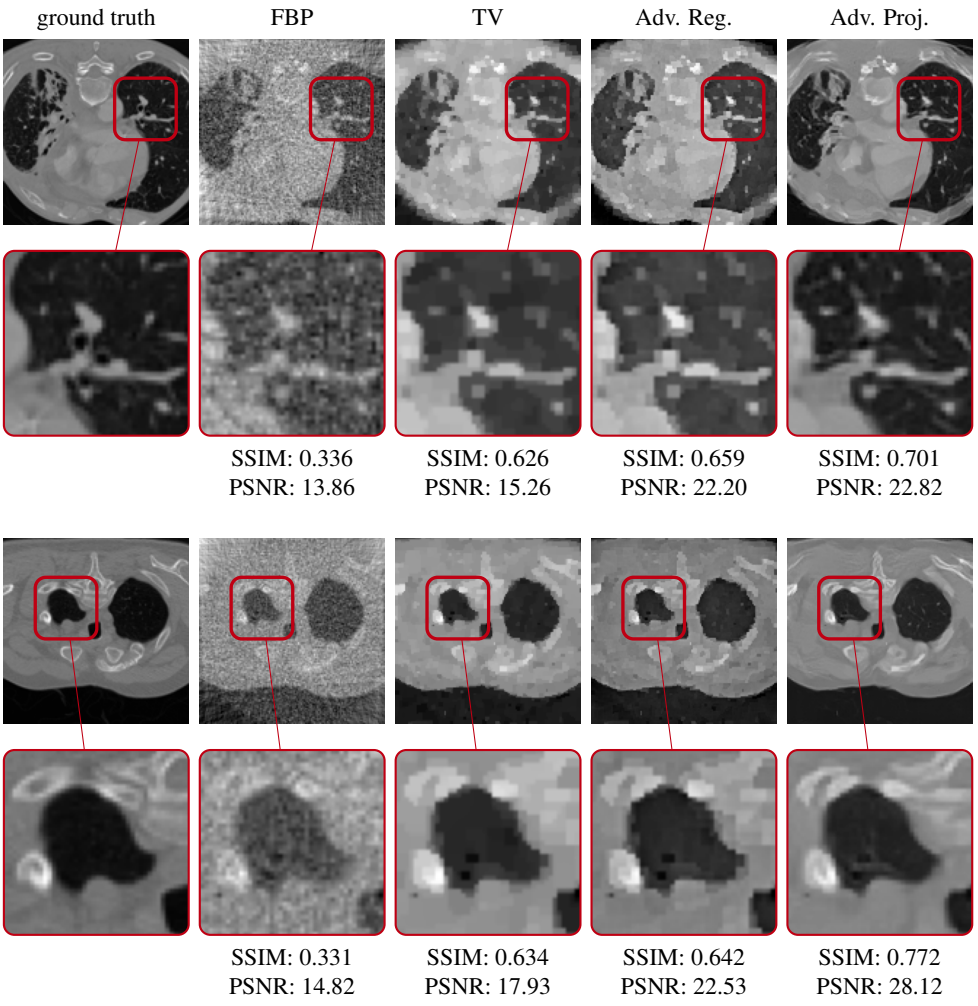


Figure 9: Additional human phantom reconstructions on a validation sample obtained with Filtered Back Projection (FBP) method, TV regularization, Adversarial Regularizer, and Adversarial Projections (left to right). Bottom row shows corresponding cropped region indicated by red box.