UNDERSTANDING ADVERSARIAL ATTACKS ON AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial vulnerability is a fundamental limitation of deep neural networks which remains poorly understood. Recent work suggests that adversarial attacks on deep neural network classifiers exploit the fact that non-robust models rely on superficial statistics to form predictions. While such features are semantically meaningless, they are strongly predictive of the input's label, allowing non-robust networks to achieve good generalization on unperturbed test inputs. However, this hypothesis fails to explain why autoencoders are also vulnerable to adversarial attacks, despite achieving low reconstruction error on clean inputs. We show that training an autoencoder on adversarial input-target pairs leads to low reconstruction error on the standard test set, suggesting that adversarial attacks on autoencoders are predictive. In this work, we study the predictive power of adversarial examples on autoencoders through the lens of compressive sensing. We characterize the relationship between adversarial perturbations and target inputs and reveal that training autoencoders on adversarial input-target pairs is a form of knowledge distillation, achieved by learning to attenuate structured noise.

1 INTRODUCTION

Szegedy et al. (2013) observed that small imperceptible perturbations can cause accurate image classifiers to confidently change their prediction. The pernicious difficulty of defending neural networks from adversarial attacks (Athalye et al., 2018; Madry et al., 2018; Carlini et al., 2019) motivates the study of the root causes and properties of adversarial vulnerability (Goodfellow et al., 2014; Tsipras et al., 2019; Simon-Gabriel et al., 2018; Moosavi-Dezfooli et al., 2017; Gilmer et al., 2018). Ilyas et al. (2019) hypothesize that *non-robust yet predictive features* are to blame for adversarial vulnerability. More precisely, the authors claim that adversarial attacks exploit the tendency of neural networks to rely on semantically meaningless patterns in training images which are highly predictive of the input's label. These patterns may become correlated with a target label when an adversarial perturbation is added to the input. As a consequence, follow-up work describes adversarial robustness as a special case of robustness to distributional shift (Gilmer & Hendrycks, 2019). Yin et al. (2019) show that imperceptible features in the high frequency spectrum of natural images are strongly predictive of class. Jacobsen et al. (2019) use an invertible neural network to study the features used by a downstream classifier and conclude that a classifier can be excessively invariant to class-specific content in the input, relying on a few highly predictive features. Zhang & Zhu (2019) find that adversarially trained classifiers are more robust to changes in texture and rely more on *global* features such as shape.

So far the discussion around adversarial vulnerability has mostly focused on classification, overlooking the fact that autoencoders are also vulnerable to adversarial attack (Kos et al., 2018; Cemgil et al., 2020; Willetts et al., 2019; Gondim-Ribeiro et al., 2018). While classifiers might rely on a possibly non-robust subset of class-predictive features, autoencoders are required to effectively compress an input image to then reconstruct it with low reconstruction error. As a consequence we cannot argue that encoders rely on semantically meaningless yet useful features to explain their adversarial vulnerability.

Cemgil et al. (2020) attribute the adversarial vulnerability of Variational Autoencoders to the limited support of the training set, but their analysis makes no distinction between adversarial perturbations and random input noise. We show instead, that targeted attacks on autoencoders are structured. Ilyas et al. (2019) make the striking observation that a classifier trained on adversarial input-target pairs generalizes to the standard test set, in support of the hypothesis that adversarial attacks exploit *non-robust yet predictive* features of the input image. We make similar observations with autoencoders. Unlike Ilyas et al. (2019), our discussion does not attribute adversarial vulnerability to the presence of non-robust features within the data, instead we focus on the relationship between worst-case noise and the low intrinsic dimension of the data. Additionally, we uncover a mechanism with which an encoder trained on adversarial input-target pairs learns to represent the adversarial image and the clean target image similarly. We describe this behaviour as knowledge distillation achieved by learning to attenuate structured noise.

In this work, we study targeted attacks on auto-encoders, where a norm-bounded perturbation $\boldsymbol{\delta}$ is added to a source image $\mathbf{x}_s$ so as to produce a similar representation to that of a randomly selected target image $\mathbf{x}_t$. Denoting the encoder by $E(.)$ and the decoder by $D(.)$ our attack objective is formulated as shown in 1.2. Similar objectives have been used by Kos et al. (2018); Cemgil et al. (2020); Gondim-Ribeiro et al. (2018); Sabour et al. (2016). The success of the attack is determined by the squared error between the target $\mathbf{x}_t$ and the reconstruction $D \circ E(\mathbf{x}_s + \delta)$.

$$\boldsymbol{\delta}^* = \text{argmin} \|E(\mathbf{x}_t) - E(\mathbf{x}_s + \boldsymbol{\delta})\|_2^2 \tag{1.1}$$

$$\|\boldsymbol{\delta}\|_2^2 \leq \epsilon \tag{1.2}$$

With the attack objective shown above we generate a training set of adversarial input-target pairs $(\mathbf{x}_s + \boldsymbol{\delta}, \mathbf{x}_t)$. We observe that a newly initialized autoencoder trained on the adversarial training set learns to reconstruct unperturbed images from the standard test set. We conclude that adversarial perturbations which fool encoders are *predictive* of the target image $\mathbf{x}_t$. More precisely, adversarial perturbations are predictably related to the low-dimensional representation of $\mathbf{x}_t$, allowing a decoding procedure to reconstruct a target $\mathbf{x}_t$ with bounded error from any adversarial input $\mathbf{x}_s + \boldsymbol{\delta}$.

The goal of our paper is to provide insight, explaining why adversarial attacks on autoencoders are *predictive*. We first study adversarial attacks on a linear encoder, assuming the data admits a sparse representation with respect to a dictionary of atoms or features. We show how adversarial perturbations are closely related to the sparse representations of the source $\mathbf{x}_s$ and target input $\mathbf{x}_t$. We use our analysis to show that training an autoencoder on adversarial input-target pairs is a form of knowledge distillation, achieved by learning to attenuate structured noise. We support our findings with experiments on MNIST and CelebA.

2 Preliminaries

To shed light on the structure of targeted attacks, we assume that the data admits a sparse representation with respect to a dictionary of atoms or features D. Recent work has also leveraged assumptions regarding the generative model of data, in particular Gilmer et al. (2018); Fawzi et al. (2018) have appealed to the manifold hypothesis to understand adversarial attacks. However, an arbitrary manifold may not be a faithful model for natural images, unlike the sparse coding model which is behind the success of many image denoising and compression algorithms (Candés & Wakin, 2008). Additionally, this assumption allows us to investigate targeted attacks on a linear encoder via the compressive sensing framework.

Compressive sensing is an effective method for simultaneous sparse signal acquisition and compression Candès et al. (2005). It's aim is to answer the following question, given a sparse vector $\mathbf{x}$, how must one project x onto a lower dimensional vector $\mathbf{y}$ such that $\mathbf{x}$ can be recovered from $\mathbf{y}$? Recovery with low error is made possible by exploiting the signal sparsity. A signal $\mathbf{x}$ in $\mathbb{R}^N$ is said to be s-sparse if at most s components of $\mathbf{x}$ are non-zero. A measurement matrix $\Phi \in \mathbb{R}^{M \times N}$ acquires M measurements of x to form a lower dimensional representation $\mathbf{y} \in \mathbb{R}^M$. Since $N >> M$, without further assumptions, the problem of recovering $\mathbf{x}$ from $\mathbf{y}$ is ill-posed since it involves solving an under-determined system of equations. However, the sparsity of $\mathbf{x}$ proves useful by providing uniqueness and stability guarantees.

In the compressive sensing literature, the recovery of x from y is expressed as the P_0 problem shown in 2.1. P_0 is guaranteed to have a unique solution $\mathbf{x}$ if the sparsity of x is bounded as shown in equation 2.2

$$P_0 : \min_{x} \|\mathbf{x}\|_0 \text{ s.t. } \Phi\mathbf{x} = \mathbf{y} \tag{2.1}$$

$$s = \|\mathbf{x}\|_0 \leq \frac{1}{2}\left(1 + \frac{1}{\mu(\Phi)}\right) \tag{2.2}$$

Where $\mu(\Phi)$ is the *mutual coherence of* Φ defined as

$$\mu(\Phi) := \max_{i \neq j} \frac{\langle \Phi_i, \Phi_j \rangle}{\|\Phi_i\|_2 \|\Phi_j\|_2} \tag{2.3}$$

This implies that, in the worst case, a signal $\mathbf{x}$ which does not satisfy the sparsity condition in 2.2 may have a counterpart signal with similar sparsity and measurement vector. In this case, the P_0 problem does not admit a unique solution. Even when the signal $\mathbf{x}$ satisfies the sparsity condition, we may obtain a counterpart signal for $\mathbf{x}$ under an $l2$-norm constraint. In section 3 we describe adversarial attacks on a linear encoder Φ as norm-bounded, dense counterparts of sparse vectors which exploit the redundancy of Φ.

3 Adversarial vulnerability of a Linear Encoder

We begin by studying a motivating toy example using a synthetic dataset of structured sparse signals. Our constructed dataset consists of 28 x 28 images made up of at most 5 Discrete Fourier Transform (DFT) components. A 28 x 28 image can consist of at most $(28/2 + 1)^2$ = 225 discrete Fourier frequencies Bracewell (1965) , however to keep our synthetic images aesthetically pleasing, we restrict our dataset to only contain 28 frequencies corresponding to periodic signals along either the horizontal or vertical axis of the image but not both. Furthermore, only 5 frequencies may be present in a single image, the combination of which is selected from a set of 200 possible configurations. We denote the sparse representation in the DFT domain of a 28x28 dimensional image as **x**.

For simplicity, we assume the encoder acts on the 225-dimensional sparse representation of the input. The reason for this assumption is to illustrate the relationship between adversarial perturbations and the sparse representation of the input. Note that for a dense vector $\gamma = D\mathbf{x}$ which admits a sparse representation $\mathbf{x}$ with respect to a dictionary of atoms D, one can obtain a measurement vector $\mathbf{y} = \Phi D\mathbf{x}$ where ΦD is effectively the measurement matrix.

Since our data is K-sparse with K=5, we use a randomly initialized measurement matrix $\Phi \in \mathbb{R}^{M\times N}$ as the encoder where $N = 225$ and $M = 50$. To construct Φ we sample i.i.d entries from $\mathcal{N}(0, \frac{1}{M})$. This results in a measurement matrix which satisfies the structured K-RIP with high probability Baraniuk et al. (2008). We can therefore obtain a dense measurement vector $\mathbf{y} = \Phi\mathbf{x}$ from which the signal $\mathbf{x}$ can be recovered. We use a deconvolution network to reconstruct the 28 x 28 image from the measurement vector **y**.

3.1 Attack Formulation

We describe an adversarial attack on the encoder Φ as a norm-bounded perturbation $\boldsymbol{\delta}$ added to the sparse representation of a source image $\mathbf{x}_s$ such that the representation of $\mathbf{x}_s + \boldsymbol{\delta}$ is similar to that of a target input $\mathbf{x}_t$. Our objective is written as a constrained optimization problem in equation 3.2.

$$\min_{\delta} \|\Phi(\mathbf{x_s} + \boldsymbol{\delta}) - \Phi\mathbf{x_t}\|_2^2 \tag{3.1}$$

$$\|\boldsymbol{\delta}\|_2^2 \leq \epsilon^2 \tag{3.2}$$

Since the above optimization problem is convex, we obtain a closed-form solution for $\boldsymbol{\delta}$ by minimizing the Lagrangian 3.3 with penalty coefficient λ. We decompose the adversarial perturbation $\boldsymbol{\delta}$ into two components $\boldsymbol{\delta_s}$ and $\boldsymbol{\delta_t}$. Where $\boldsymbol{\delta_s}$ is such that $\|\Phi(\mathbf{x_s} + \boldsymbol{\delta_s})\|_2^2$ is minimized, while $\boldsymbol{\delta_t}$ is such that $\|\Phi(\boldsymbol{\delta_t} - \mathbf{x_t})\|_2^2$ is minimized. That is, $\boldsymbol{\delta_s}$ is crafted so as to obfuscate $\mathbf{x}_s$ while $\boldsymbol{\delta_t}$ is crafted so as to pass as $\mathbf{x}_t$. We can therefore express $\boldsymbol{\delta_s}$ and $\boldsymbol{\delta_t}$ in terms of the source input $\mathbf{x}_s$ 3.4 and the target input $\mathbf{x}_t$ 3.5 respectively.

Computing $\boldsymbol{\delta_s}$ and $\boldsymbol{\delta_t}$:

$$L(\boldsymbol{\delta}, \lambda) = (\mathbf{x}_s - \mathbf{x}_t + \boldsymbol{\delta})^T \Phi^T \Phi (\mathbf{x}_s - \mathbf{x}_t + \boldsymbol{\delta}) + \lambda(\boldsymbol{\delta}^T \boldsymbol{\delta} - \epsilon) \tag{3.3}$$

$$\nabla_\delta L(\boldsymbol{\delta}, \lambda) = 2\Phi^T \Phi (\mathbf{x}_s - \mathbf{x}_t + \boldsymbol{\delta}) + 2\lambda \boldsymbol{\delta} = 0$$

$$\boldsymbol{\delta} = \left(\Phi^T \Phi + \lambda I\right)^{-1} \Phi^T \Phi (\mathbf{x}_t - \mathbf{x}_s)$$

$$\boldsymbol{\delta_s} = -\left(\Phi^T \Phi + \lambda I\right)^{-1} \Phi^T \Phi \mathbf{x}_s \tag{3.4}$$

$$\boldsymbol{\delta_t} = \left(\Phi^T \Phi + \lambda I\right)^{-1} \Phi^T \Phi \mathbf{x}_t \tag{3.5}$$

We denote the transformation $\left(\Phi^T \Phi + \lambda I\right)^{-1} \Phi^T \Phi$ by the matrix $\mathbf{M}_\Phi$. The final expression for $\boldsymbol{\delta}$ which we use to attack Φ is shown in equation 3.6.

$$\boldsymbol{\delta} = \mathbf{M}_\Phi \mathbf{x}_t - \mathbf{M}_\Phi \mathbf{x}_s \tag{3.6}$$

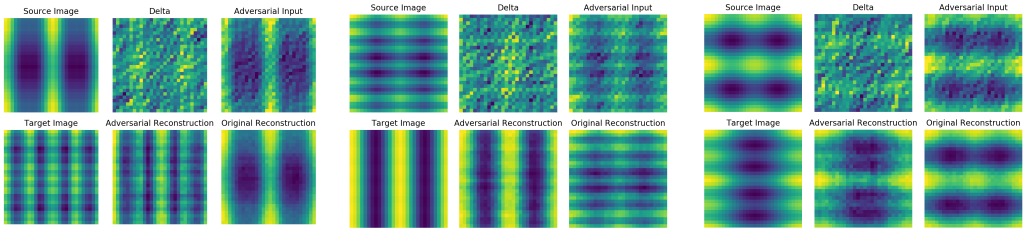

Figure 1: Closed form targeted attack on linear encoder using synthetic dataset.

4 Adversarial Perturbations are Predictive

We see that the adversarial perturbation δ exploits the redundancy of Φ. That is, while the mutual coherence of Φ might be sufficiently small to guarantee the recovery of K-sparse vectors, a sufficiently large number of column vectors of Φ may be combined so as to produce a representation that is highly correlated with that of $\mathbf{x}_t$. Equations 3.4 and 3.5 express adversarial perturbations as dense counterparts of sparse representations $\mathbf{x}_s$ and $\mathbf{x}_t$, using the dictionary $\mathbf{M}_\Phi$. We therefore obtain a precise description of how $\boldsymbol{\delta_t}$ is *predictive* of $\mathbf{x}_t$ in the case of a linear encoder.

We verify whether a newly initialized autoencoder can be trained to reconstruct the target images $\mathbf{x}_t$ from adversarial inputs $\mathbf{x}_{adv}$. We perform a similar experiment to that performed by Ilyas et al. (2019). Using the random measurement matrix Φ as our first encoder, we generate adversarial input-target pairs using the closed form expression for $\boldsymbol{\delta}$ shown in equation 3.6, where $\mathbf{x}_{adv} = \mathbf{x}_s + \boldsymbol{\delta_t} + \boldsymbol{\delta_s}$ and the target to be reconstructed is $\mathbf{x}_t$. We use the adversarial input-target pairs $\{(\mathbf{x}_{adv}, \mathbf{x}_t)\}$ to train a new autoencoder. This time, the new measurement matrix Ψ is updated along with the decoder. Table 1 shows the average $l2$-distance (over 128000 samples) between measurement vectors of different inputs. The distance between the measurement vectors of two random samples $\|\Psi \mathbf{x}_s - \Psi \mathbf{x}_t\|_2$ is included for comparison. We find that Ψ learns to represent $\mathbf{x}_{adv}$ and $\mathbf{x}_t$ with a similar measurement vector, despite the fact that while training, Ψ only encounters adversarial inputs $\mathbf{x}_{adv}$. Additionally, when the input is composed of $\boldsymbol{\delta_t}$ alone, Ψ also produces a measurement vector similar to $\Psi \mathbf{x}_t$.

Table 1: Average $l2$ distance between measurement vectors

$\|\Psi(\mathbf{x}_s + \boldsymbol{\delta_s})\|_2$	$\|\Psi \boldsymbol{\delta}_t\|_2$	$\|\Psi \mathbf{x}_t - \Psi \boldsymbol{\delta_t}\|_2$	$\|\Psi \mathbf{x}_{adv} - \Psi \mathbf{x}_t\|_2$	$\|\Psi \mathbf{x}_s - \Psi \mathbf{x}_t\|_2$
3.08 ±0.05	5.33 ±0.10	3.10 ±0.05	1.27 ±0.08	7.0 ±0.2

The newly trained autoencoder computes similar representations for $\mathbf{x}_t$, $\boldsymbol{\delta}_t$ and $\mathbf{x}_{adv}$, and successfully reconstructs $\mathbf{x}_t$ from $\Psi\mathbf{x}_t$, $\Psi\boldsymbol{\delta}_t$ and $\Psi\mathbf{x}_{adv}$. We also notice that the $l2$-norm of the extraneous component $\|\Psi(\mathbf{x}_s + \boldsymbol{\delta_s})\|_2$ is small compared to the $l2$-norm of the informative component $\|\Psi\boldsymbol{\delta}_t\|_2$. While $\boldsymbol{\delta}$ was crafted to induce such behaviour in Φ, the encoder Ψ has inherited such behaviour by training on adversarial input-target pairs $(\mathbf{x}_{adv}, \mathbf{x}_t)$.

Since the source and target images are selected at random, $\boldsymbol{\delta}_t$ is the only component of the adversarial input informative of the target $\mathbf{x}_t$. Yet, when the autoencoder is trained on $(\boldsymbol{\delta}_t, \mathbf{x}_t)$ pairs it fails to reconstruct standard inputs from our synthetic dataset. This is because while training, Ψ learns to represent $\Psi\mathbf{x}_t$ similarly to $\Psi\boldsymbol{\delta}_t$ by learning to attenuate $\mathbf{x}_s - \boldsymbol{\delta}_s$ in comparison to the informative component $\boldsymbol{\delta}_t$. In short, the encoder trained on a dataset of $(\mathbf{x}_{adv}, \mathbf{x}_t)$ pairs learns to emulate the encoder Φ by effectively denoising $\mathbf{x}_{adv}$. More concretely Ψ is such that for all vectors $\mathbf{x}_s$ sampled from the structured sparsity model, the norm of the nuisance term $(\Psi - \Psi\mathbf{M}_\Phi)\mathbf{x}_s$ is reduced compared to $\|\Psi\mathbf{M}_{\boldsymbol{\Phi}}\mathbf{x_t}\|_2$. Training an autoencoder on adversarial input-target pairs is therefore a kind of knowledge distillation, achieved by learning to attenuate structured noise.

We observe similar results with Variational Autoencoders (Kingma & Welling, 2014; Rezende et al., 2014) with Gaussian priors trained on CelebA and MNIST. Our experiments are similar to those presented by Ilyas et al. (2019) and can be summarized as follows:

- We first train a VAE on the standard training set, which we denote by VAE_{std}.
- For each source sample $\mathbf{x}_s$ in the training set, we randomly select a target $\mathbf{x}_t$ from the training set. We perform a targeted attack on VAE_{std} by minimizing $\|\mu(\mathbf{x}_s + \boldsymbol{\delta}) - \mu(\mathbf{x}_t)\|_2$, where $\mu(x)$ denotes the mean of the posterior distribution $q(z|x)$ given by the encoder $E(.)$.
- We then train a newly initialized VAE on the training set of adversarial input-target pairs $(\mathbf{x}_s + \boldsymbol{\delta}, \mathbf{x}_t)$.

We generate targeted adversarial attacks with bounded $l2$ norm using Projected Gradient Descent (PGD) Madry et al. (2018) with random starts. We set $\epsilon = 10.0$, step size $= 0.05$ and number of steps $= 1000$ for CelebA, and $\epsilon = 3.0$ stepsize $= 0.1$ and number of steps $= 200$ for MNIST. To generate a training set of adversarial input-target pairs, we randomly sample a source image $\mathbf{x}_s$ and a target image $\mathbf{x}_t$, we find a perturbation $\boldsymbol{\delta}$ such that $\|\boldsymbol{\delta}\|_2 \leq \epsilon$ and $\|\mu(\mathbf{x}_s + \boldsymbol{\delta}) - \mu(\mathbf{x}_t)\|_2$ is minimized, where $\mu(x)$ denotes the mean of the posterior distribution $q(z|x)$ given by the encoder $E(.)$. Sample attacks are shown in figure 2.

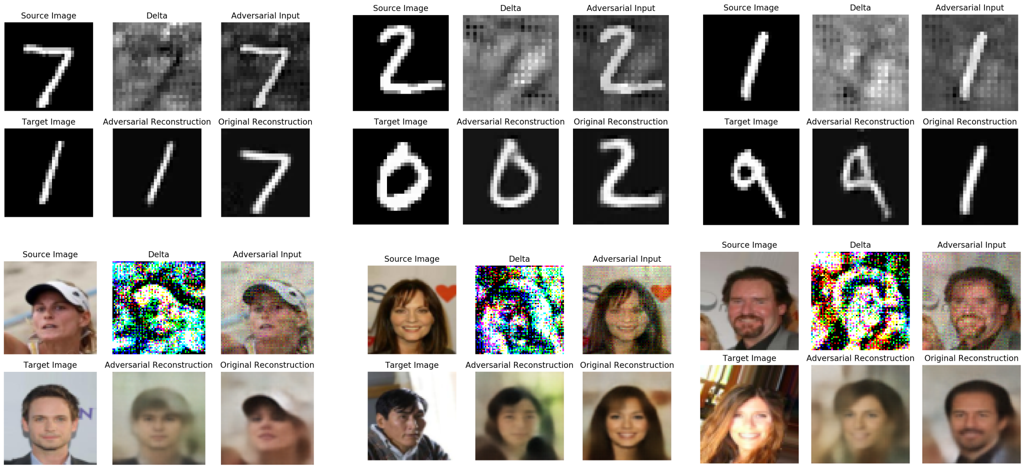

Figure 2: Targeted adversarial attacks on encoder trained on MNIST (top) and CelebA (bottom).

We then train a newly initialized VAE on the training set of adversarial input-target pairs $(\mathbf{x}_s + \boldsymbol{\delta}, \mathbf{x}_t)$. We find that the newly trained VAE learns to reconstruct samples from the standard test set, we obtain reasonable reconstructions for both MNIST and CelebA as shown in figure 3.

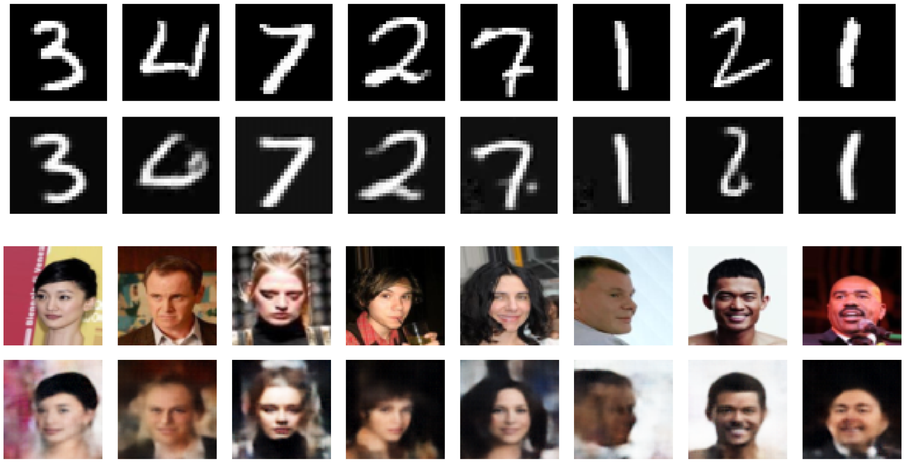

Figure 3: Reconstruction of images from the standard test set from a model trained on $(\mathbf{x}_s + \delta, \mathbf{x}_t)$ pairs.

We repeat the above procedure, this time with $\mathbf{x}_{adv}$ constructed without a source input $\mathbf{x}_s$. We construct an adversarial input $\mathbf{x}_{adv}$ starting with a zero image and performing PGD to obtain an $l2$-norm bounded input whose representation matches that of the target image from the training set. We find that while a model trained on such adversarial input-target pairs can learn to reconstruct the target from adversarial inputs, it fails to reconstruct standard inputs as shown in figure 4.

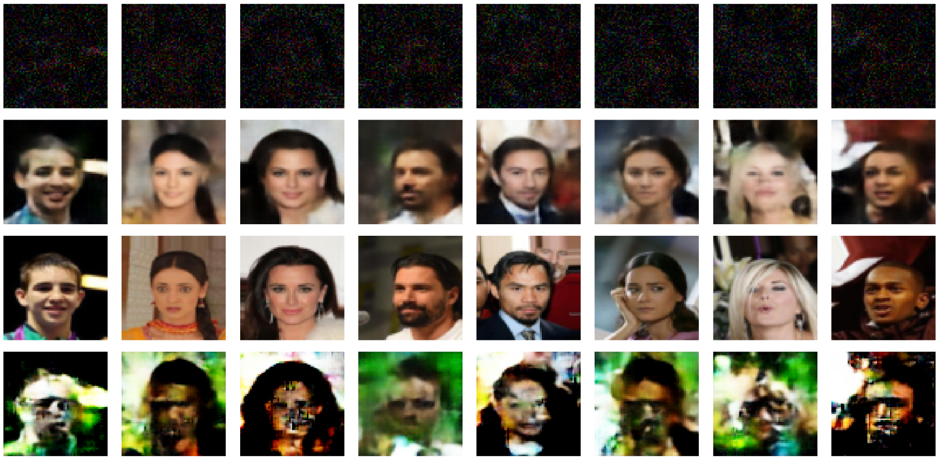

Figure 4: Sample input and reconstructions for autoencoder trained on $(\boldsymbol{\delta}_t, \mathbf{x}_t)$ pairs. Targeted perturbation computed using samples from the test set as targets $\mathbf{x}_t$ (top row) and corresponding decoder output (second row). Original target image (third row) with corresponding decoder output (last row). An autoencoder trained on $(\boldsymbol{\delta}_t, \mathbf{x}_t)$ pairs learns to reconstruct target images $\mathbf{x}_t$ from corresponding perturbations $\boldsymbol{\delta}_t$ yet fails to decode images from the standard test set.

It is worth mentioning that this is not observed in the case of classification. That is, constructing an adversarial training set without source inputs $\mathbf{x}_s$ leads to generalization to the standard test set as demonstrated by (Nayak et al., 2019), who term this phenomenon *zero-shot knowledge distillation*. Additionally, it has been shown by Krishna et al. (2020) that models can be distilled or *stolen* using a different training set than that used to train the teacher network. Another interesting and related phenomenon is that of dataset distillation (Wang et al., 2018), which demonstrates that classifiers can learn to generalize to the standard test set by training on a small dataset constructed using a teacher network. We believe that these observations point to future research directions aimed at understanding model extraction and generalization in neural networks.

5 Adversarial Attacks on a Two-Layer Encoder

In section 4 we illustrate how training a linear encoder on adversarial input-target pairs is a form of knowledge distillation, achieved by learning to attenuate a structured nuisance term $\mathbf{x}_s - \mathbf{M}_\Phi \mathbf{x}_s$. That is, $(\Psi - \Psi\mathbf{M}_\Phi)\mathbf{x} \leq \xi$ for all sparse vectors $\mathbf{x}$ sampled from a distribution of structured sparse vectors. In the case of a non-linear encoder, it is not immediately clear that the perturbation consists of a nuisance component that is uninformative of $\mathbf{x}_t$. In this section we illustrate that adversarial attacks on a two-layer ReLU-based encoder also admit a decomposition into a nuisance term and an informative component. We include details of our analysis in A.

We derive the form of the perturbations crafted for a two-layer encoder shown in 5. The intermediate activation vector Γ is produced by a 1D-convolution layer followed by a ReLU. We assume zero-bias for simplicity. The matrix Φ projects Γ onto a lower dimensional representation vector Y. We consider an adversarial perturbation δ_0 applied to the input vector X_s so as to induce a representation vector Y_{adv} similar to Y_t, the representation of a target X_t. We consider δ_1 to be the difference in intermediate activation vector Γ induced by δ_0.

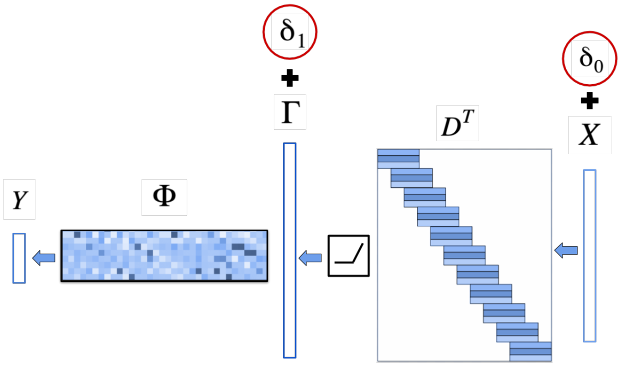

Figure 5: The perturbation δ_0 added to the input X induces a perturbation δ_1 on the hidden representation Γ.

We assume that a successful perturbation δ_0 is already found which results in $\left\|\Phi ReLU(D^T X_s + \delta_0) - Y_t\right\|_2 \leq E$ for small E. Our analysis begins by observing that given the set of indices $\mathcal{I}$ of $\Gamma + \delta_1$ where $\Gamma[i] + \delta_1[i] > 0$, and the $l2$ norm of the perturbation δ_1, we can obtain an expression for δ_1 as shown in equation 5.1.

$$\delta_1 = -\mathbf{M}_{(\Phi,\mathcal{I},\lambda)}\Phi_\mathcal{I}\Gamma_s + \mathbf{M}_{(\Phi,\mathcal{I},\lambda)}\Phi\Gamma_t \tag{5.1}$$

We consider the effect of δ_0 on Γ_s under the different modes of operation of the ReLU function. In each case we describe a condition on δ_0 in terms of the target $\Gamma_s + \delta_1$.

$$d_i^T\delta_0 = \begin{cases} \delta_1[i] + \Gamma_s[i] - d_i^T X_s - \xi_i & \text{if } \Gamma_s[i] + \delta_1[i] > 0 \text{ then } \xi_i = 0 \\ \delta_1[i] + \Gamma_s[i] - d_i^T X_s - \xi_i & \text{if } \Gamma_s[i] + \delta_1[i] = 0 \text{ then } \xi_i > 0 \end{cases}$$

The closed form solution of δ_0 expressed in terms of the pre-ReLU target γ is shown in 5.3.

$$\delta_0 = (DD^T + \lambda I)^{-1} D\gamma \tag{5.3}$$

To simplify, we denote $(DD^T + \lambda I)^{-1}D$ as $\mathbf{M}_D$.

$$\delta_0 = \mathbf{M}_D \left[\delta_1 + \Gamma_s - D^T X_s \right] - \mathbf{M}_D \xi \tag{5.4}$$

We substitute 5.1 to obtain:

$$X_s + \delta_0 = X_s - \mathbf{M}_D(D^T X_s) + \mathbf{M}_D \Gamma_s - \mathbf{M}_D \mathbf{M}_{(\Phi,\mathcal{I},\lambda)} \Phi_{\mathcal{I}} \Gamma_s + \mathbf{M}_D \mathbf{M}_{(\Phi,\mathcal{I},\lambda)} \Phi \Gamma_t - \mathbf{M}_D \xi$$

The informative component of the target X_t is $\mathbf{M}_D \mathbf{M}_{(\Phi,\mathcal{I},\lambda)} \Phi \Gamma_t$ and the remaining terms form the extraneous component $X_s - \mathbf{M}_D(D^T X_s) + \mathbf{M}_D \Gamma_s - \mathbf{M}_D \mathbf{M}_{(\Phi,\mathcal{I},\lambda)} \Phi_{\mathcal{I}} \Gamma_s$ which depends on the source input X_s.

6 Discussion

In this work we examine how adversarial perturbations are predictive of a target sample $\mathbf{x}_t$. We show that autoencoders trained on adversarial input-target pairs achieve low reconstruction error on the standard test set, similar to a phenomenon observed with classifiers by Ilyas et al. (2019). Using a structured sparse toy dataset and a linear encoder with random weights, we illustrate how adversarial perturbations are closely related to the sparse representation of the input, motivating the decomposition of a targeted perturbation $\boldsymbol{\delta}$ into two components $\boldsymbol{\delta_s}$ and $\boldsymbol{\delta_t}$. While the role of $\boldsymbol{\delta_s}$ is to attenuate the representation pertaining to the source input $\mathbf{x}_s$, the role of $\boldsymbol{\delta_t}$ is to induce a similar representation to that of the target input $\mathbf{x}_t$. Using the randomly initialized encoder Φ, we construct a training set of adversarial input-target pairs $(\mathbf{x}_s + \boldsymbol{\delta}, \mathbf{x}_t)$ to train a new encoder Ψ. We find that Ψ learns to represent $\mathbf{x}_t$ and $\boldsymbol{\delta_t} = \mathbf{M}_\Phi \mathbf{x}_t$ similarly, but this is not the case when Ψ is trained on $(\boldsymbol{\delta_t}, \mathbf{x}_t)$ pairs. We conclude that training encoders on $(\mathbf{x}_s + \delta, \mathbf{x}_t)$ pairs is a form of knowledge distillation. More concretely, Ψ learns to represent $\mathbf{x}_t$ similarly to δ_t by learning to attenuate the nuisance term $\mathbf{x}_s - \mathbf{M}_\Phi \mathbf{x}_s$.

We perform similar experiments with CelebA and MNIST and observe similar behaviour. A VAE trained on $(\mathbf{x}_s + \delta, \mathbf{x}_t)$ pairs learns to reconstruct inputs from the standard test set. However, while a VAE trained on $(\delta_t, \mathbf{x}_t)$ pairs can reconstruct $\mathbf{x}_t$ from newly sampled δ_t, it fails to reconstruct samples from the standard test set. We conclude that targeted attacks on a non-linear encoder can also be decomposed into a nuisance term and an informative component. We obtain expressions for such components for a 2-layer ReLU encoder in section 5.

Our results regarding attacks on encoders suggest a different interpretation of the observations of Ilyas et al. (2019); that an adversarial example can be predictive of the target image itself rather than only its class label. Rather than viewing adversarial examples as consisting of the non-robust features within the training data, we view adversarial examples as an alternate form of the data. In the future, we plan to extend our analysis to explain related observations regarding classifiers.

We note that our work is aimed at understanding why adversarial examples can be used as a well-generalizing training set rather than providing a complete characterization of the causes of adversarial vulnerability. Recent efforts have focused on answering different questions regarding adversarial attacks. The work of Romano et al. (2019) shares our assumption that the data admits a sparse representation. The authors compare the stability of two layered pursuit algorithms and find that layered Basis Pursuit is more stable than layered soft-thresholding, which is closely related to the forward pass of a neural network. Allen-Zhu & Li (2020) also assume that the data has an underlying sparse representation. The authors introduce the concept of *feature purification* and prove that adversarial training removes components from a classifier's weights which are weakly correlated with multiple class labels. We note that our use of the sparsity assumptions has allowed us to uncover how the encoder learns to represent samples from the standard test set by learning to be invariant to a structured nuisance component. An interesting direction for future work is to further characterize how auxiliary objectives are tied to the behaviour of a neural network.

References

Zeyuan Allen-Zhu and Y. Li. Feature purification: How adversarial training performs robust deep learning. *ArXiv*, abs/2005.10190, 2020.

A. Athalye, Nicholas Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *ArXiv*, abs/1802.00420, 2018.

Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

R. Bracewell. The fourier transform and its applications. 1965.

E. Candés and M. Wakin. An introduction to compressive sampling [a sensing/sampling paradigm that goes against the common knowledge in data acquisition]. 2008.

E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59:1207–1223, 2005.

Nicholas Carlini, A. Athalye, Nicolas Papernot, W. Brendel, Jonas Rauber, D. Tsipras, Ian J. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *ArXiv*, abs/1902.06705, 2019.

T. Cemgil, S. Ghaisas, Krishnamurthy Dvijotham, and Pushmeet Kohli. Adversarially robust representations with smooth encoders. In *ICLR*, 2020.

Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. In *Advances in neural information processing systems*, pp. 1178–1187, 2018.

J. Gilmer and Dan Hendrycks. A discussion of 'adversarial examples are not bugs, they are features': Adversarial example researchers need to expand what is meant by 'robustness'. 2019.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

George Gondim-Ribeiro, Pedro Tabacof, and Eduardo Valle. Adversarial attacks on variational autoencoders. *arXiv preprint arXiv:1806.04646*, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Andrew Ilyas, Shibani Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

J. Jacobsen, Jens Behrmann, R. Zemel, and M. Bethge. Excessive invariance causes adversarial vulnerability. 2019.

Diederik P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models. In *2018 ieee security and privacy workshops (spw)*, pp. 36–42. IEEE, 2018.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of bert-based apis. *ArXiv*, abs/1910.12366, 2020.

A. Madry, Aleksandar Makelov, L. Schmidt, D. Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and P. Frossard. Universal adversarial perturbations. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 86–94, 2017.

G. Nayak, Konda Reddy Mopuri, Vaisakh Shaj, R. Venkatesh Babu, and A. Chakraborty. Zero-shot knowledge distillation in deep networks. In *ICML*, 2019.

Danilo Jimenez Rezende, S. Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

Yaniv Romano, Aviad Aberdam, Jeremias Sulam, and Michael Elad. Adversarial noise attacks of deep learning architectures: Stability analysis via sparse-modeled signals. *Journal of Mathematical Imaging and Vision*, 62:313–327, 2019.

Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J. Fleet. Adversarial manipulation of deep representations. *CoRR*, abs/1511.05122, 2016.

Carl-Johann Simon-Gabriel, Y. Ollivier, B. Schölkopf, L. Bottou, and David Lopez-Paz. Adversarial vulnerability of neural networks increases with input dimension. *ArXiv*, abs/1802.01421, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

D. Tsipras, Shibani Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv: Machine Learning*, 2019.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

M. Willetts, Alexander Camuto, Tom Rainforth, S. Roberts, and C. Holmes. Improving vaes' robustness to adversarial attack. *arXiv: Machine Learning*, 2019.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *NeurIPS*, 2019.

Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. *ArXiv*, abs/1905.09797, 2019.

A Adversarial Attacks on a Two-Layer Encoder

We provide a detailed description of the ideas presented in section 5. We first consider the expression for the perturbation δ_1. The additional challenge in this case is due to the non-convexity of the set of activations which are permissible under an $l2$ norm constraint on δ_0. We note that our goal is not to find the worst-case perturbation, which would require solving an integer program, rather it is to obtain a general form for δ_1 and δ_0 in terms of the source input X_s, and hidden representations Γ_s and Γ_t. We therefore assume that a successful perturbation δ_0 is already found which results in $\left\|\Phi ReLU(D^T X_s + \delta_0) - Y_t\right\|_2 \leq E$ for small E. The perturbation δ_0 induces a perturbation δ_1 on the post-ReLU activation Γ_s. More precisely, the indices of δ_1 which are of interest are those where $\Gamma_s + \delta_1 > 0$. We denote the i^{th} element of δ_1 as $\delta_1[i]$ and the elements of δ_1 whose indices are in $\mathcal{I}$ as $\delta_1[\mathcal{I}]$. Suppose that we are given the indices where $\Gamma_s[i] + \delta_1[i] > 0$, denoted by the set $\mathcal{I}$. Suppose also that we have knowledge that $\|\delta_1\|_2 = \epsilon_1$. What remains to be found are the values for $\delta_1[i]$ for $i \in \mathcal{I}$. We can express $\delta_1[\mathcal{I}]$ as the solution to the following optimization problem shown in equation A.3.

$$\min \frac{1}{2}\|\Phi_{\mathcal{I}}(\Gamma_s[\mathcal{I}] + \delta_1[\mathcal{I}]) - \Phi\Gamma_t\|_2^2 \tag{A.1}$$

$$\text{s.t } \Gamma_s[\mathcal{I}] + \delta_1[\mathcal{I}] \geq 0 \tag{A.2}$$

$$\frac{1}{2}\|\delta_1[\mathcal{I}]\|_2^2 = \epsilon_1^2 \tag{A.3}$$

We obtain the following expression for $\delta_1[\mathcal{I}]$.

$$\mathcal{L}(\delta_1, \lambda, \nu) = \|\Phi_{\mathcal{I}}(\Gamma_s[\mathcal{I}] + \delta_1[\mathcal{I}]) - \Phi\Gamma_t\|_2^2 - \nu^T(\Gamma_s[\mathcal{I}] + \delta_1[\mathcal{I}]) + \lambda(\|\delta_1\|_2^2 - \epsilon_1) \tag{A.4}$$

$$\delta_1[\mathcal{I}] = (\Phi_{\mathcal{I}}^T\Phi_{\mathcal{I}} + \lambda I)^{-1}\Phi_{\mathcal{I}}^T(\Phi_{\mathcal{I}}\Gamma_s[\mathcal{I}] + \Phi\Gamma_t + \nu) \tag{A.5}$$

Since we only consider entries such that $\delta_1[i] > 0$ given by the index set $\mathcal{I}$ the inequality constraints $\Gamma_s[\mathcal{I}] + \delta_1[\mathcal{I}] \geq 0$ are inactive and $\nu = 0$.

We denote $(\Phi_{\mathcal{I}}^T \Phi_{\mathcal{I}} + \lambda I)^{-1} \Phi_{\mathcal{I}}^T$ by $\mathbf{M}_{(\Phi, \mathcal{I}, \lambda)}$.

$$\delta_1 = -\mathbf{M}_{(\Phi,\mathcal{I},\lambda)} \Phi_{\mathcal{I}} \Gamma_s + \mathbf{M}_{(\Phi,\mathcal{I},\lambda)} \Phi \Gamma_t \tag{A.6}$$

We can therefore frame the attack objective as finding a perturbation δ_0 to be added to the dense input X_s such that $ReLU(D^T(X_s + \delta_0)) = \Gamma_s + \delta_1$.

We consider the effect of δ_0 on Γ_s under the different modes of operation of the ReLU function. In each case we describe a condition on δ_0 in terms of the target $\Gamma_s + \delta_1$.

If $\Gamma_s[i] + \delta_1[i] > 0$:

$$d_i^T X_s + d_i^T \delta_0 = \Gamma_s[i] + \delta_1[i] \tag{A.7}$$

If $\Gamma_s[i] + \delta_1[i] = 0$:

$$d_i^T X_s + d_i^T \delta_0 < 0 \tag{A.8}$$

$$\text{or equivalently, } \exists 0 < \xi_i \text{ s.t.} \tag{A.9}$$

$$d_i^T X_s + d_i^T \delta_0 = -\xi_i \tag{A.10}$$

Summarizing the above, we obtain the following piecewise target for $D^T \delta_0$.

$$d_i^T \delta_0 = \begin{cases} \delta_1[i] + \Gamma_s[i] - d_i^T X_s - \xi_i & \text{if } \Gamma_s[i] + \delta_1[i] > 0 \text{ then } \xi_i = 0 \\ \delta_1[i] + \Gamma_s[i] - d_i^T X_s - \xi_i & \text{if } \Gamma_s[i] + \delta_1[i] = 0 \text{ then } \xi_i > 0 \end{cases}$$

We add $\Gamma_s[i] + \delta_1[i]$ to $-d_i^T X_s - \xi_i$ which effectively does not change the target when $\Gamma_s[i] + \delta_1[i] = 0$.

We express the perturbation δ_0 as the solution to the optimization problem shown in A.13, where γ is the pre-ReLU target is shown in A.14. By construction (we assumed δ_1 is induced by δ_0) the minimum value of the objective is 0, achieved when $D^T \delta_0 = \gamma$.

$$\min_{\delta} \left\| D^T \delta_0 - \gamma \right\|_2^2 \tag{A.12}$$

$$\text{s.t. } \|\delta\|_2^2 \leq \epsilon^2 \tag{A.13}$$

$$\delta_0 = (DD^T + \lambda I)^{-1} D \gamma \tag{A.14}$$

To simplify, we denote $(DD^T + \lambda I)^{-1} D$ as $\mathbf{M}_D$. Using A.6 to obtain

$$X_s + \delta_0 = X_s + \mathbf{M}_D \left[-\mathbf{M}_{(\Phi,\mathcal{I},\lambda)} \Phi_{\mathcal{I}} \Gamma_s + \mathbf{M}_{(\Phi,\mathcal{I},\lambda)} \Phi \Gamma_t + \Gamma_s \right] - \mathbf{M}_D D^T X_s - \mathbf{M}_D \xi \tag{A.15}$$

$$= X_s - \mathbf{M}_D D^T X_s + \mathbf{M}_D \Gamma_s - \mathbf{M}_D \mathbf{M}_{(\Phi,\mathcal{I},\lambda)} \Phi_{\mathcal{I}} \Gamma_s + \mathbf{M}_D \mathbf{M}_{(\Phi,\mathcal{I},\lambda)} \Phi \Gamma_t - \mathbf{M}_D \xi \tag{A.16}$$