

Learning Provably Robust Estimators for Inverse Problems via Jittering

Anselm Krainovic*, Mahdi Soltanolkotabi†, and Reinhard Heckel*

*Department of Computer Engineering, Technical University of Munich

†Ming Hsieh Department of Electrical Engineering, University of Southern California

July 25, 2023

Abstract

Deep neural networks provide excellent performance for inverse problems such as denoising. However, neural networks can be sensitive to adversarial or worst-case perturbations. This raises the question of whether such networks can be trained efficiently to be worst-case robust. In this paper, we investigate whether jittering, a simple regularization technique that adds isotropic Gaussian noise during training, is effective for learning worst-case robust estimators for inverse problems. While well studied for prediction in classification tasks, the effectiveness of jittering for inverse problems has not been systematically investigated. In this paper, we present a novel analytical characterization of the optimal ℓ_2 -worst-case robust estimator for linear denoising and show that jittering yields optimal robust denoisers. Furthermore, we examine jittering empirically via training deep neural networks (U-nets) for natural image denoising, deconvolution, and accelerated magnetic resonance imaging (MRI). The results show that jittering significantly enhances the worst-case robustness, but can be suboptimal for inverse problems beyond denoising. Moreover, our results imply that training on real data which often contains slight noise is somewhat robustness enhancing.

1 Introduction

Deep neural networks achieve state-of-the-art performance for image reconstruction tasks including compressive sensing, super-resolution, and denoising. Due to their excellent performance, deep networks are now used in a variety of imaging technologies, for example in MRI and CT. However, concerns have been voiced that neural networks can be sensitive to worst-case or adversarial perturbations. Those concerns are fuelled by neural networks being sensitive to small, adversarially selected perturbations for prediction tasks such as image classification [Sze+14].

Recent empirical work for image reconstruction tasks [Hua+18; Ant+20; GMM22; DCH21] found that worst-case perturbations can have a significantly larger effect on the image quality than random perturbations. This sensitivity to worst-case perturbations is not unique to neural networks, classical imaging methods are similarly sensitive [DCH21].

This raises the question of whether networks can be designed or trained to be worst-case robust. A successful method proposed in the context of classification is adversarial training, which optimizes a robust or adversarial loss during training [Mad+18]. However, the robust loss requires finding worst-case perturbations during training which is difficult and computationally expensive.

In this work, we study jittering, a simple regularization technique that adds noise during training as a robustness-enhancing technique for inverse problems. It is long known that adding noise during training has regularizing effect and can be beneficial for generalization [Bis95; HK92; RMI99]. Prior work also studied adding noise for enhancing adversarial robustness for classification [ZNR17;

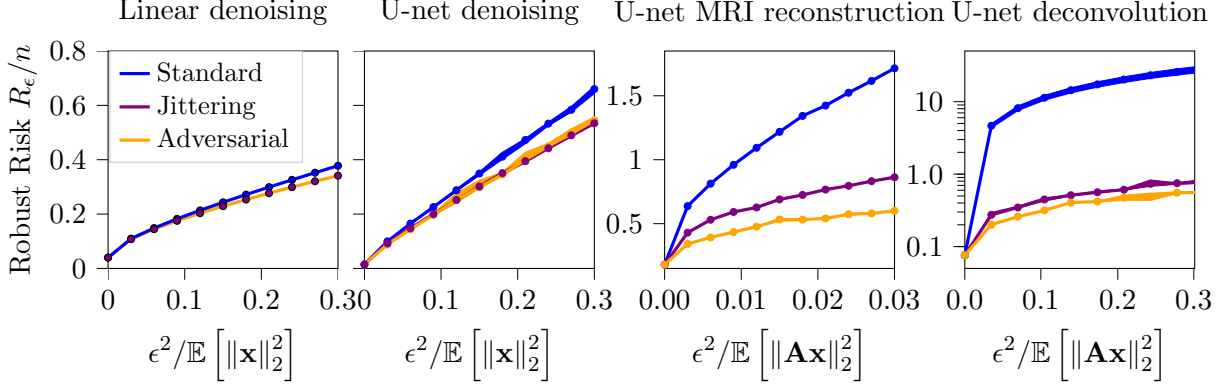


Figure 1: **Jittering yields worst-case robust reconstruction methods.** The plots show the pixel-wise robust risks R_ϵ/n of models trained to minimize the robust risk R_ϵ and jittering risk J_{σ_w} , respectively, with suitable choices of jittering levels $\sigma_w(\epsilon)$. The shaded areas are 66% confidence intervals. Left panel is for subspace denoising where jittering and robust training are provably equivalent, the panels from second left to right are for image reconstruction problems with the U-net, where jittering is particularly effective for denoising.

KKG18; Gil+19]. However, jittering has not been systematically studied as a robustness enhancing technique for training robust networks for inverse problems.

We consider the following signal reconstruction problem. Let $f: \mathbb{R}^m \rightarrow \mathbb{R}^n$ be an estimator (a neural network in practice) for a signal (often an image) $\mathbf{x} \in \mathbb{R}^n$ based on the measurement $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z} \in \mathbb{R}^m$, where $\mathbf{A}$ is a measurement matrix and $\mathbf{z}$ is random noise. We want to learn an estimator that has small robust risk defined as

$$R_\epsilon(f) = \mathbb{E}_{(\mathbf{x}, \mathbf{y})} \left[\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|f(\mathbf{y} + \mathbf{e}) - \mathbf{x}\|_2^2 \right]. \quad (1)$$

The robust risk is the expected worst-case error with respect to a ℓ_2 -perturbation of norm at most ϵ of f measured with the mean-squared error.

Theoretical results. We start with Gaussian denoising of a signal lying in a subspace, and first characterize the optimal linear robust denoiser, i.e., the estimator that minimizes the robust risk $R_\epsilon(f)$. While the resulting estimator is quite intuitive, proving optimality is fairly involved and relies on interesting applications of Jensen’s inequality.

Second, we show that the optimal linear robust estimator minimizes the Jittering-risk

$$J_{\sigma_w}(f) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}), \mathbf{w}} \left[\|f(\mathbf{y} + \mathbf{w}) - \mathbf{x}\|_2^2 \right], \quad (2)$$

where $\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I})$ is Gaussian jittering noise with noise variance σ_w^2 that depends on the desired robustness level ϵ . This finding implies that instead of performing robust training via minimizing an empirical version of robust risk, we can train a denoiser via jittering, i.e., injecting Gaussian noise during training, at least for denoising a signal lying in a subspace. Figure 1, left panel, demonstrates

the equivalence of training via minimizing a jittering risk and robust training numerically for the subspace model. It is evident that both methods of training yield an equally robust estimator.

Moreover, we discuss extensions of our theory for linear inverse problems $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$ and find that jittering can result in slightly suboptimal worst-case estimators for some classes of forward operators.

Empirical results for real-world denoising, deconvolution and compressive sensing.

Jittering is also effective for learning robust neural network estimators for solving inverse problems in practice. Figure 1, second from left to right, depicts the worst-case risk achieved by training a U-net model for denoising, compressive sensing, and deconvolution, with standard training (blue), with jittering (purple), and with adversarial training (orange). For denoising, we see that jittering is as effective for obtaining a worst-case robust estimator as adversarial training, as suggested by theory. For compressive sensing and deconvolution, we find that jittering can be suboptimal beyond denoising, but is still effective for enhancing robustness.

Those findings make jittering a potentially attractive method for learning robust estimators in the context of inverse problems, since jittering can also be implemented easily and needs far less computational resources than adversarial training. Moreover, those findings imply that training on real data which often contains slight noise is somewhat robustness enhancing.

2 Related work

Empirical investigation of worst-case robustness for imaging. Several works investigated the sensitivity of neural networks for image reconstruction tasks to adversarial perturbations, for limited angle tomography [Hua+18], MRI and CT [Ant+20; GMM22; DH21], and image-to-image tasks (colorization, deblurring, denoising, and super-resolution) [Cho+22; Yan+22; Cho+19]. Collectively, those works show that neural networks for imaging problems are significantly more sensitive to adversarial perturbations than to random perturbations, as expected. The effect of adversarial ℓ_2 -perturbations measured in mean-squared-error is roughly proportional to the energy of the perturbations in most of those problems, demonstrating that up to a constant (that might be large) neural networks can be relatively stable for imaging tasks. Classical reconstruction methods, in particular ℓ_1 -regularized least-squares, are similarly sensitive to adversarial perturbations [DH21].

Learning robust methods with robust optimization. To learn robust classifiers, Madry et al. [Mad+18] proposed to minimize a robust loss and to find worst-case perturbation during training with projected gradient descent. Adversarial training can be effective for learning robust methods, but is computationally expensive due to the cost of finding adversarial perturbations. A variety of heuristics exist to lower the computational cost of robust training for neural networks. For example, Raj et al. [RBL20] consider a compressive sensing reconstruction problem and propose to generate adversarial perturbations for training with an auxiliary network instead of solving a maximization problem. As another example, Wong et al. [WRK20] considers adversarial training of classifiers and propose to calculate adversarial perturbations during training by first randomly perturbing the initial point and then applying a single step of projected gradient descent.

Jittering for enhancing robustness in inverse problems. The literature is somewhat split on whether jittering is effective for enhancing worst-case robustness for imaging. Genzel et al. [GMM22] suggested that jittering can enhance worst-case robustness. Contrary, Gandikota et al. [GCM22] consider the robustness to ℓ_∞ -perturbations for neural-network based deblurring and observed that the DeepWiener architecture, trained with Jittering at constant noise levels, is sensitive to adversarial perturbations.

Robustness for inverse problems versus robustness for classification problems. Robustness in general and adding noise during training in particular, has been intensively studied in the classification setting. However, inverse problems and classification/prediction problems are very different. Adversarial robustness for classifiers is defined as the (average) minimal distance to the decision boundary, and random noise robustness as the minimal noise strength (for example the radius of Gaussian noise sphere) such that one likely crosses the decision boundary [FMDF16]. For inverse problems, there is no notion of a decision boundary. Therefore, results and intuitions from classification, which are often based on geometric insights on distances to surfaces (see for example Fawzi et al. [FMDF16] and Shafahi et al. [Sha+19]) do not apply to inverse problems.

Jittering for enhancing robustness in classification. Prior work in classification considered Gaussian data augmentation or adding Gaussian noise during training (which is conceptually very similar to jittering) as an robustness-enhancing technique and found that adding noise enhances adversarial robustness, but reported mixed results on its effectiveness. Fawzi et al. [FMDF16] proved for linear classifiers that adding Gaussian noise during training increases adversarial robustness, and Gilmer et al. [Gil+19] demonstrated that empirically adding Gaussian noise during training also increases adversarial robustness for neural networks in the context of classification. Rusak et al. [Rus+20] also found Gaussian noise addition beneficial for corruption robustness (including noise, compression and weather artifacts). Furthermore, Kannan et al. [KKG18] and Zantedeschi et al. [ZNR17] considered adding Gaussian noise during training together with other regularization methods and report that adding noise at a fixed noise level alone yields a noticeable increase in robustness. Contrary, Carlini and Wagner [CW17] reevaluated the methods proposed by Zantedeschi et al. [ZNR17] and reported that the robustness gains are small compared to adversarial training.

Randomized smoothing. Randomized smoothing is a very successful technique for obtaining robust classifiers [CRK19; Car+23], and is based on constructing a smoothed classifier from a base classifier by averaging the base classifier’s outputs under Gaussian noise perturbation. The smoothed classifier is provably robust within a specified radii, without making any restrictions on the base classifier. Despite similarities at first sight, randomized smoothing considers surrogate smoothed models, whereas jittering is a training technique (see the appendix on a detailed discussion).

3 Theory for robust reconstruction of a signal lying in a subspace

In this section, we characterize the optimal robust estimator for denoising a signal in a subspace. While the resulting estimator is quite intuitive, proving optimality is fairly involved and relies on interesting applications of Jensen’s inequality. We then show that the optimal robust estimator is also the unique minimizer of the jittering loss. Finally, we conjecture a precise characterization of

optimal estimators for linear inverse problems beyond denoising, and show that jittering can result in suboptimal estimators for linear inverse problems beyond denoising.

3.1 Problem setup

We consider a signal reconstruction problem, where the goal is to estimate a signal $\mathbf{x}$ based on a noisy measurement $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(0, \sigma_z^2 \mathbf{I}/m)$ is Gaussian noise and $\mathbf{A} \in \mathbb{R}^{m \times n}$ a measurement or forward operator. The random noise is scaled so that the expected noise energy is $\mathbb{E}[\|\mathbf{z}\|_2^2] = \sigma_z^2$. The random noise is denoted by $\mathbf{z}$, to distinguish it from the adversarial noise or worst-case error, denoted by $\mathbf{e}$. We assume that the signal is (approximately) chosen uniformly from the intersection of a sphere and a subspace. Specifically, the signal is generated as $\mathbf{x} = \mathbf{U}\mathbf{c}$, where $\mathbf{c} \sim \mathcal{N}(0, \sigma_c^2 \mathbf{I}/d)$ is Gaussian and $\mathbf{U} \in \mathbb{R}^{n \times d}$ is an orthonormal basis for a d -dimensional subspace of $\mathbb{R}^n$. The expected signal energy is $\mathbb{E}[\|\mathbf{x}\|_2^2] = \sigma_c^2$.

We consider a linear estimator of the form $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ for estimating the signal from the measurement. For the standard reconstruction problem of estimating the signal $\mathbf{x}$ from the measurement $\mathbf{y}$, performance is often measured in terms of the expected least-squared error. We are interested in robust reconstruction and consider the expected worst-case reconstruction error with respect to an ℓ_2 -perturbation, defined in equation (1), and given by

$$R_\epsilon(f) = \mathbb{E}_{(\mathbf{x}, \mathbf{y})} \left[\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{H}(\mathbf{y} + \mathbf{e}) - \mathbf{x}\|_2^2 \right].$$

For $\epsilon = 0$, the robust risk reduces to the standard expected mean-squared error.

3.2 Denoising

We start with denoising where the forward map is the identity, i.e., $\mathbf{A} = \mathbf{I}$. The following theorem characterizes the optimal worst-case robust denoiser.

Theorem 1. *For $d \rightarrow \infty$, the optimal worst-case estimator, i.e., the estimator minimizing the worst-case risk $R_\epsilon(f)$ amongst all estimators of the form $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ with $\mathbf{H}$ symmetric is $\mathbf{H} = \alpha \mathbf{U}\mathbf{U}^T$, where*

$$\alpha = \begin{cases} \frac{\sigma_c^2 - \frac{\epsilon \sigma_c \sigma_z \sqrt{\frac{d}{n}}}{\sqrt{\sigma_c^2 + \sigma_z^2 \frac{d}{n} - \epsilon^2}}}{\sigma_c^2 + \sigma_z^2 \frac{d}{n}} & \text{if } \sigma_c^2 > \epsilon^2 \\ 0 & \text{else.} \end{cases}$$

The worst-case optimal estimator projects onto the signal subspace, and then shrinks towards zero, by a factor determined by the noise variance σ_z^2 and the worst-case noise energy ϵ^2 . We consider the asymptotic setup where $d \rightarrow \infty$ only for expositional convenience; our proof shows that the estimator in the theorem is also near optimal for finite d .

To understand the implications of the theorem, let us first consider the case where the worst-case perturbation is zero. Then, the optimal estimator simply projects on the signal-subspace and shrinks towards zero, by a factor of $\alpha = \frac{\sigma_c^2}{\sigma_c^2 + \sigma_z^2 \frac{d}{n}}$. The larger the noise, the more shrinkage.

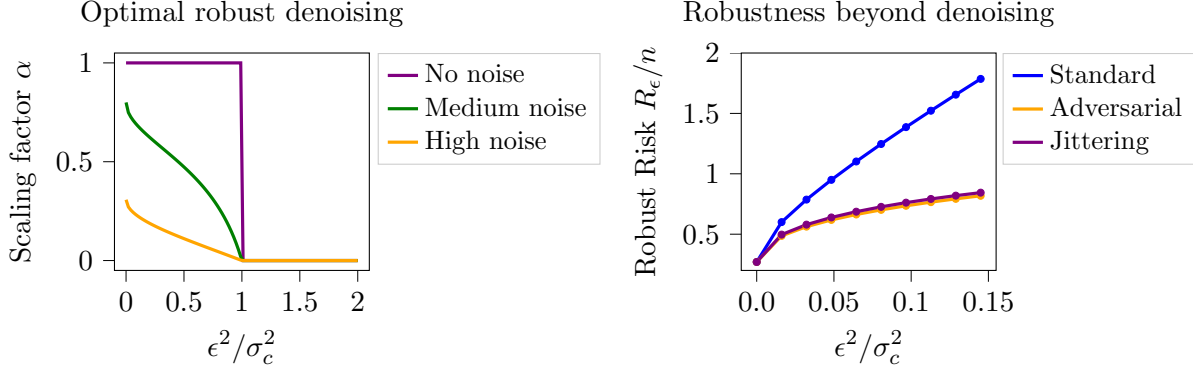


Figure 2: **Robust reconstruction of signals lying in a subspace.** The left panel shows the scaling factor α of the optimal robust denoiser according to Theorem 1 at noise levels $\sigma_z \sqrt{d/n} \in \{0, 0.4, 1.2\}$ and signal energy $\sigma_c^2 = 1$. The right panel depicts the robust risks of standard training and jittering as well as the optimal robust risk for an inverse problem beyond denoising, with the details stated in subsection 3.3.

Next, consider the most interesting regime, where non-zero adversarial noise is present. If the adversarial noise energy is larger than the signal energy, the estimator projects onto zero. However, this is an extreme regime since the adversarial noise can cancel the signal, and no good estimate of the signal can be achieved.

For the more practical regime where the adversarial noise energy is smaller than the signal energy, the theorem states that the optimal estimator projects onto the signal-subspace and shrinks towards zero—just like the optimal estimator for the noise-free case—but this time by a factor α , that decreases in the adversarial noise energy ϵ^2 .

The proof of Theorem 1 is in the appendix. Note that for estimator $\mathbf{H} = \alpha \mathbf{U} \mathbf{U}^T$, a worst-case perturbation can be computed in closed form for a fixed $\mathbf{y}$ and $\mathbf{x}$: a worst-case perturbation is the vector that points into the direction of the signal plus noise lying in the signal subspace, i.e., $\mathbf{e} = \mathbf{U} \epsilon \frac{(1-\alpha)\mathbf{c} + \alpha \mathbf{U}^T \mathbf{z}}{\|(1-\alpha)\mathbf{c} + \alpha \mathbf{U}^T \mathbf{z}\|_2}$. However, for a general estimator $\mathbf{H}$, the perturbation can not be written in closed form, which makes proving optimality quite challenging. Our proof relies on a characterization of the inner maximization problem as the solution to an optimization problem in one variable, and several unusual applications of Jensen’s inequality.

3.2.1 Robust denoisers via jittering

An important consequence of the characterization of the worst-case optimal estimator in Theorem 1 is that, at least for the linear setup considered in this section, a worst-case optimal estimator can be obtained by regularization with jittering.

Recall from the introduction that regularization via jittering simply adds Gaussian noise $\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I})$ to the measurement during training. The jittering risk (2) of the estimator $f(\mathbf{y}) = \mathbf{H} \mathbf{y}$ for denoising is $J_{\sigma_w}(f) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}), \mathbf{w}} [\|f(\mathbf{y} + \mathbf{w}) - \mathbf{x}\|_2^2]$. Choosing the variance of the jittering noise level accordingly as a function of the desired robustness level ϵ yields an optimal worst-case robust estimator by minimizing the jittering loss, as formalized by the following corollary of Theorem 1.

Corollary 1. For $\epsilon^2 < \sigma_c^2$, the symmetric linear estimator $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ that minimizes the jittering risk J_{σ_w} with noise level chosen as a function of the desired noise level ϵ as $\sigma_w(\epsilon) = \sqrt{\frac{\epsilon^2 \sigma_z^2 \frac{d}{n} + \sigma_z \sqrt{\frac{d}{n}} \sigma_c \epsilon \sqrt{\sigma_c^2 - \epsilon^2 + \sigma_z^2 \frac{d}{n}}}{d(\sigma_c^2 - \epsilon^2)}}$ also minimizes the worst-case risk R_ϵ .

Hence, if we aim for a robustness level $\epsilon < \sigma_c$, we can simply apply training via Jittering instead of adversarial training by choosing the Jittering noise level using the explicit formula for the jittering noise level $\sigma_w(\epsilon)$ in corollary 1.

Figure 1, left panel, shows the results of numerical simulation for adversarial training and jittering. In the implementation we treat the linear reconstructions as neural networks with a single layer without bias and perform adversarial training and jittering for each perturbation level. The simulations show that the robust risk performance of the models are identical, as predicted by the theory. Details on how adversarial training is performed are in Section 4.

3.2.2 Robustness accuracy trade-off

Another consequence of Theorem 1 is an explicit robustness-accuracy trade-off: increased worst-case robustness comes at a loss of accuracy. In the practically relevant regime of $0 \leq \epsilon^2 < \sigma_c^2$ the standard risk of the optimal worst-case estimator $f_{\alpha(\epsilon)}$ is $R_0(f_{\alpha(\epsilon)}) = \sigma_c^2 \cdot \frac{\sigma_z^2 \frac{d}{n}}{\sigma_c^2 + \sigma_z^2 \frac{d}{n} - \epsilon^2}$. This expression yields the optimal standard error for $\epsilon = 0$ and is strictly monotonically increasing in ϵ , hence showing the loss of accuracy when increasing robustness. Robustness-accuracy tradeoffs can also be observed in other machine learning settings, for classification and regression settings, see for example [Tsi+19]. For linear inverse problems with applications in control, robustness accuracy-tradeoffs were recently characterized by Lee et al. [Lee+21] and Javanmard et al. [JSH20].

3.3 General linear inverse problems

In the previous section, we characterized the optimal worst-case robust estimator and found that jittering yields optimal robust denoisers. In this section, we derive a conjecture for the worst-case optimal robust estimator for more general linear inverse problems of reconstructing a signal $\mathbf{x}$ from a measurement $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$, with a forward operator $\mathbf{A}$ (with $\mathbf{A} \neq \mathbf{I}$ in general), and show that this estimator is in general not equal to the estimator obtained with jittering, thus jittering is in general sub-optimal.

Optimal robust estimator. Let $\mathbf{A}\mathbf{U} = \mathbf{W}^T \mathbf{\Lambda} \mathbf{V}$ be the singular value decomposition of the matrix $\mathbf{A}\mathbf{U}$ with singular values λ_i . As formalized by Lemma 1 of the appendix, the robust-risk (1) of the estimator f can be written as an expectation involving a minimization problem over a single variable (instead of a maximization over an n -dimensional variable, as in the original definition):

$$R_\epsilon(\mathbf{H}) = \mathbb{E}_{\mathbf{v}} \left[\min_{\lambda \geq \sigma_i^2} \lambda \epsilon^2 + \mathbf{v}^T \left(\mathbf{I} - \frac{1}{\lambda} \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{v} \right], \quad \mathbf{v} = (\mathbf{H}\mathbf{A} - \mathbf{I})\mathbf{x} + \mathbf{H}\mathbf{z}. \quad (3)$$

Here, σ_i are the singular values of the matrix $\mathbf{H}$. In order to find the optimal robust estimator we wish to solve the optimization problem $\arg \min_{\mathbf{H}} R_\epsilon(\mathbf{H})$. The difficulty in solving this optimization problem is that we can't solve the minimization problem within the expectation (3) in closed form. In order to prove Theorem 1 for denoising (i.e., for $\mathbf{A} = \mathbf{I}$) we derived an upper and a matching

lower bound of the risks using several unusual applications of Jensen's inequality. The proof does not generalize in a straightforward manner to the more general case where $\mathbf{A} \neq \mathbf{I}$. However, for large d , the random variable $\mathbf{v}^T(\mathbf{I} - \frac{1}{\lambda}\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{v}$ concentrates around its expectation, and thus we conjecture that for large d , we can exchange expectation and minimization, which yields:

$$R_\epsilon(\mathbf{H}) = \min_{\lambda \geq \sigma_i^2} \lambda \epsilon^2 + \mathbb{E}_{\mathbf{v}} \left[\mathbf{v}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{v} \right]. \quad (4)$$

The expectation in the risk expression (4) can be explicitly computed, which yields the following conjecture for the worst-case optimal estimator:

Conjecture 1. *For $d \rightarrow \infty$ the optimal worst-case estimator, i.e., the estimator minimizing the worst-case risk $R_\epsilon(f)$ amongst all estimators of the form $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ is $\mathbf{H} = \mathbf{U}\mathbf{V}\text{diag}(\sigma_i)\mathbf{W}^T$ with*

$$\sigma_i = \frac{1 + \lambda \lambda_i^2}{2\lambda_i} + \frac{d}{m} \frac{\lambda}{2\lambda_i} \frac{\sigma_z^2}{\sigma_c^2} - \sqrt{\left(\frac{1 + \lambda \lambda_i^2}{2\lambda_i} + \frac{d}{m} \frac{\lambda}{2\lambda_i} \frac{\sigma_z^2}{\sigma_c^2} \right)^2 - \lambda},$$

if $\lambda_i \neq 0$ and $\sigma_i = 0$ otherwise. Here, the parameter λ is a solution of:

$$\operatorname{argmin}_{\lambda \geq 0} \lambda \epsilon^2 + \sum_{i=1}^d \frac{1 - \lambda \lambda_i^2}{2} \frac{\sigma_c^2}{d} - \frac{\lambda}{2} \frac{\sigma_z^2}{m} + \sqrt{\left(\frac{1 + \lambda \lambda_i^2}{2} \frac{\sigma_c^2}{d} + \frac{\lambda}{2} \frac{\sigma_z^2}{m} \right)^2 - \lambda \lambda_i^2 \frac{\sigma_c^4}{d^2}}.$$

The optimization problem involved is convex and box-constrained and can thus be solved numerically. Besides the argument above, we confirmed our conjecture with numerical simulations.

Optimal jittering estimator. Unlike for denoising, for general inverse problems, the jittering-risk minimizing estimator is in general not equal to the optimal worst-case estimator, but the two estimators are often close. The optimal estimator minimizing the jittering risk is given as (see Appendix C):

$$\mathbf{H}_J(\sigma_w) = \mathbf{U}\mathbf{V}\text{diag} \left(\frac{\sigma_c^2 \lambda_i}{\sigma_c^2 \lambda_i^2 + \sigma_z^2 \frac{d}{m} + \sigma_w^2 d} \right) \mathbf{W}^T, \quad (5)$$

where as before $\mathbf{A}\mathbf{U} = \mathbf{W}^T \mathbf{\Lambda} \mathbf{V}$ is the singular value decomposition with singular values λ_i . While the estimator (5) has the same form as the worst-case optimal estimator in Conjecture 1, the diagonal matrix in the two estimators is in general slightly different.

Numerical Simulation. The worst-case sub-optimality of the jittering-risk optimal estimator (5) depends on the singular values of the matrix $\mathbf{A}\mathbf{U}$; if they are equal the jittering-risk estimator is optimal, and if they are not equal there is typically a small gap. To illustrate the gap, we consider a forward operator $\mathbf{A}$ with linearly decaying singular values $\frac{i}{n}$, for $1 \leq i \leq n$ with signal energy $\sigma_c^2 = 1$ and noise level $\sigma_z = 0.2$. We compare the (conjectured) optimal robust estimator specified by Conjecture 1 with the optimal Jittering estimator (5) at noise level σ_w , where σ_w is optimized such that one obtains minimal robust risk R_ϵ at a given perturbation level ϵ . The results in Figure 2, right panel, show a small gap in robust risk, which implies that Jittering is suboptimal for this case. However, simulations with varying forward operators and noise levels indicate that the gap is small relative to the robust risk of the standard estimator. Experiments on image deconvolution using U-Net presented in Section 4 show similar results.

4 Experiments

In this section, we train standard convolutional neural-networks with standard training, adversarial training, and jittering for three inverse problems: denoising images, image deconvolution, and compressive sensing, and study their robustness. We find that Jittering yields well-performing robust denoisers at a computational cost similar to standard training, which is significantly cheaper than adversarial training. We also find that jittering yields robust estimators for deconvolution and compressive sensing. This indicates that training on real data which often contains slight measurement noise is robustness enhancing.

4.1 Problem setup

We start by describing the datasets, networks, and methodology.

Natural images. We consider denoising and deconvolution of natural images, where our goal is to reconstruct an image $\mathbf{x} \in \mathbb{R}^n$ from a noisy measurement $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(0, \sigma_z^2 \mathbf{1}/n\mathbf{I})$ is Gaussian noise and $\mathbf{A}$ a measurement matrix, which is equal to identity for denoising, and implements a convolution for deconvolution. For deconvolution we use a 8×8 -sized discretization of the 2-dimensional Gaussian normal distribution with standard deviation 2. The kernel is visualized in Figure 3 in the appendix. We obtain train and validation datasets $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$ of sizes 34k and 4k, respectively, from colorized images of size $n = 128 \cdot 128 \cdot 3$ generated by randomly cropping and flipping ImageNet images. The methods are tested on 2k original-sized images.

Medical data. We also perform experiments on accelerated singlecoil magnetic resonance imaging (MRI) data, where the goal is to reconstruct an image $\mathbf{x} \in \mathbb{R}^n$ from a noisy and subsampled measurement in the frequency domain $\mathbf{y} = \mathbf{M}\mathbf{F}\mathbf{x} + \mathbf{z} \in \mathbb{R}^{2m}$. We use the fastMRI singlecoil knee dataset [Zbo+18], which contains the images $\mathbf{x}$ and fully sampled measurements ($\mathbf{M} = \mathbf{I}$). We process it by random subsampling at acceleration factor 4 and obtain train, validation and test datasets with approximately 31k, 3.5k and 7k slices, respectively. While perturbations are sought in frequency domain, the inverse Fourier transform is applied to the measurements before feeding the 320×320 cropped and normalized images into the network.

Network architecture. We use the U-net architecture [RFB15] since it gives excellent performance for denoising [Bro+19] and medical image reconstruction tasks, such as computed tomography [Jin+17] and is used as a building block for state-of-the-art methods for magnetic resonance imaging [Zbo+18; Sri+20]. For natural images, we use a U-net with 3×3 padded convolutions with ReLU activation functions, 2×2 max-pooling layers for downscaling and transposed convolutions for upscaling. The network has 120k parameters. For MRI reconstruction we use a U-Net architecture similar to Zbontar et al. [Zbo+18] with 3×3 padded convolutions, leaky ReLU activation function, 2×2 average pooling and transposed convolutions (480k learnable parameters). We denote the U-Net by the parameterized mapping $f_{\theta}: \mathbb{R}^m \rightarrow \mathbb{R}^n$ in the following.

Evaluation. We evaluate networks by measuring its robustness via the empirical robust risk defined as $\hat{R}_{\epsilon}(\theta) = \sum_{i=1}^N \max_{\|\mathbf{e}_i\|_2 \leq \epsilon} \|f_{\theta}(\mathbf{y}_i + \mathbf{e}_i) - \mathbf{x}_i\|_2^2$. For evaluation, the robust empirical risk is computed over the test set. We assess the accuracy by computing the standard empirical risk $\hat{R}_0(\theta)$.

Computing the robust empirical risk is non-trivial since it requires finding adversarial perturbations for solving the inner maximization problem. This is explained next. We also study the computational cost of the different methods, which we measure in terms of GPU cost and time.

Finding adversarial perturbations. To evaluate the empirical risk and for robust training, we need to compute adversarial perturbations $\mathbf{e} = \arg \max_{\|\mathbf{e}\|_2 \leq \epsilon} \|f_\theta(\mathbf{y} + \mathbf{e}) - \mathbf{x}\|_2^2$. We find the perturbations by running N_a projected gradient ascent steps, starting with initial perturbation $\mathbf{e}^0 = 0$ and iterate

$$\mathbf{e}^{j+1} = \mathcal{P}_{B(0, \epsilon)} \left(\mathbf{e}^j + 2.5 \frac{\epsilon}{N_a} \frac{\Delta \mathbf{e}^j}{\|\Delta \mathbf{e}^j\|_2} \right), \quad \text{where} \quad \Delta \mathbf{e}^j = \nabla_{\mathbf{e}^j} \|f_\theta(\mathbf{e}^j + \mathbf{y}) - \mathbf{x}\|_2^2.$$

Here, $\mathcal{P}_{B(0, \epsilon)}$ is the projection into the ℓ_2 -ball $B(0; \epsilon)$ of radius ϵ around the origin. The gradient is normalized to facilitate step size optimization with multiplier 2.5 such that the iteration can reach and move along the boundary, as suggested by Madry et al. [Mad+18].

Training methods. **Standard training** minimizes the standard empirical risk $\hat{R}_0$. **Adversarial training** minimizes the empirical robust risk $\hat{R}_\epsilon$. To minimize the empirical robust risk, we approximate the inner maximization, $\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|f_\theta(\mathbf{y}_i + \mathbf{e}) - \mathbf{x}_i\|_2^2$, with $\|f_\theta(\tilde{\mathbf{y}}_i) - \mathbf{x}_i\|_2^2$, where $\tilde{\mathbf{y}}_i$ is the adversarially perturbed measurement computed as described above. Training via jittering minimizes

$$\hat{J}_{\sigma_w}(\boldsymbol{\theta}) = \sum_{i=1}^N \mathbb{E}_{\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I})} \left[\|f_\theta(\mathbf{y}_i + \mathbf{w}) - \mathbf{x}_i\|_2^2 \right],$$

where the jittering level σ_w is chosen depending on the desired robustness level. To approximate the expectation we draw independent jittering noise samples $\mathbf{w}$ in each iteration of SGD. We treat the jittering noise level as a hyperparameter optimized using the validation dataset (shown in the appendix).

Throughout, we use PyTorch’s Adam optimizer with learning rate 10^{-3} and batch size 50 for natural images, and 10^{-2} and 1 for MRI data. As perturbation levels, we consider values within the practically interesting regime of $\epsilon^2 / \mathbb{E} [\|\mathbf{Ax}\|_2^2] < 0.3$ for natural images and 0.03 for MRI data. Note that for $\epsilon^2 > \mathbb{E} [\|\mathbf{x}\|_2^2]$, Theorem 1 predicts for denoising ($\mathbf{A} = \mathbf{I}$) that the optimal robust estimator is zero everywhere. Figure 7 in the appendix shows that for large perturbations ϵ the trained U-net denoiser also maps to zero.

4.2 Results

We now discuss the results of the denoising, deconvolution, and compressive sensing experiments.

Robust and standard performance. Figure 1, shows that the standard estimator is relatively robust for Gaussian denoising and increasingly sensitive for more ill-posed problems (deconvolution and compressive sensing). The experiments further show that jittering is effective for enhancing robustness, in particular relative to the sensitivity of the standard estimator. Nevertheless, as suggested by theory, we see a gap between the robust risk of adversarial training and jittering

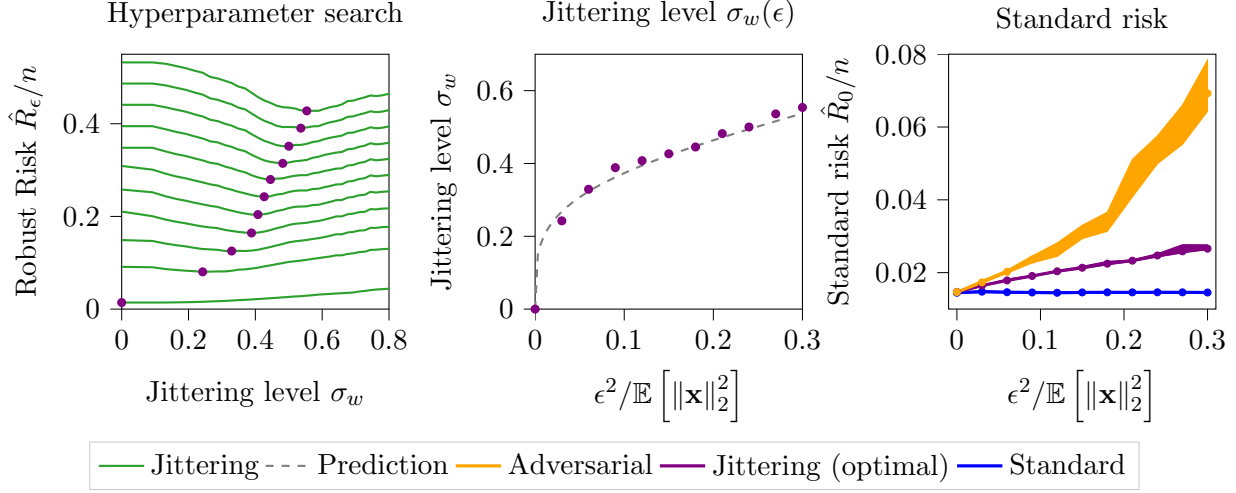


Figure 3: **Estimating the optimal jittering noise levels for the denoising task.** The left panel shows the results of training networks via Jittering, at noise levels σ_w , and calculating the empirical robust risk $\hat{R}_\epsilon$ of each model. Each green line corresponds to robust risks at one perturbation level. The optimal jittering noise levels are shown in the middle panel and follow well the prediction from theory (Cor. 1, details in appendix). The jittering estimators are similarly robust as adversarial training (Figure 1), but attain lower standard risks (right panel).

for image deconvolution and compressive sensing. For Gaussian denoising, however, Jittering is particularly effective and yields increasingly better performing networks in terms of standard risks for larger perturbations.

Choice of the jittering level. The results are based on choosing the jittering noise levels via hyperparameter search for each task. Figure 3 shows the results for Gaussian denoising: It can be seen that the choice of noise level is important for minimizing the robust risk. The estimated noise levels also aligns well the theoretical prediction. Details on this and the parameter choices for deconvolution and the compressive sensing experiments are in the appendix.

Computational complexity. We measured the GPU time until convergence and memory utilization of the methods and present the results in the Table 1 of the appendix. Performing adversarial training is by a factor of the projected gradient ascent steps more expensive than standard training. Moreover, training via jittering has similar computational cost as standard training, since it solely consists of drawing and adding Gaussian noise on the training data.

Visual reconstructions. For the linear subspace setting adversarial training and jittering are equivalent. For Gaussian denoising with a neural network, however, they perform differently. For larger perturbations jittering tends to yield smoother images than networks trained adversarially, as can be seen in the example reconstructions shown in Figure 4. This effect is particularly noticeable for the Gaussian deconvolution task. In the appendix, we show examples using smaller perturbation

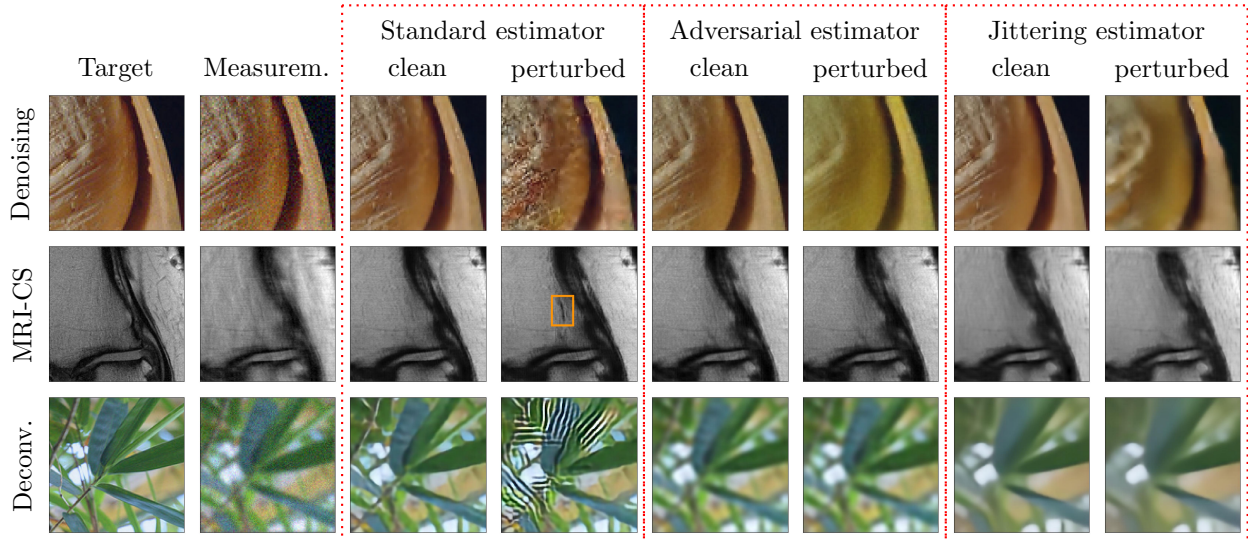


Figure 4: Example reconstructions using measurements (second column) and separately calculated perturbed measurements (shown in appendix). The reconstructions are denoted as clean and perturbed, respectively. The perturbation levels are $\epsilon^2/\mathbb{E}[\|\mathbf{Ax}\|_2^2] = 0.03$ for denoising, 0.003 for compressive sensing and 0.001 for deconvolution. We can see that the standard estimator is visibly sensitive to perturbations. Jittering yields robust estimators, but at the same time yields smoother reconstructions.

levels. Moreover, we discuss an approximation to jittering, Jacobian regularization, which similarly enhances robustness. It is computationally more expensive, but yields less smooth reconstructions.

5 Conclusion

In this paper, we characterized the optimal worst-case robust estimator for Gaussian subspace denoising and found that the optimal estimator can be provably learned with jittering. Our results for training neural networks for Gaussian denoising of images show that jittering enables the training of neural networks that are as robust as neural networks trained adversarially, but at a fraction of the computational cost, and without the hassle of having to find adversarial perturbations. While we demonstrated that jittering can yield suboptimal robust estimators in general, in practice, jittering is effective at improving the robustness for compressive sensing and image deconvolution. Moreover, our results imply that training on real data that contains slight measurements noise is robustness enhancing.

Reproducibility The repository at https://github.com/MLI-lab/robust_reconstructors_via_jittering contains the code to reproduce all results in the main body of this paper.

Acknowledgments A.K. and R.H. are supported by the Institute of Advanced Studies at the Technical University of Munich, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 456465471, 464123524, the DAAD, and the German Federal Ministry of Education and Research, and the Bavarian State Ministry for Science and the Arts. M.S. is supported by

the Packard Fellowship in Science and Engineering, a Sloan Research Fellowship in Mathematics, an NSF-CAREER under award #1846369, DARPA FastNICS programs, and NSF-CIF awards #1813877 and #2008443.

References

- [Ant+20] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen. “On Instabilities of Deep Learning in Image Reconstruction and the Potential Costs of AI”. In: *Proceedings of the National Academy of Sciences*. 2020.
- [Bis95] C. M. Bishop. “Training with Noise Is Equivalent to Tikhonov Regularization”. In: *Neural Computation* 7.1 (1995).
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. 2004.
- [Bro+19] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron. “Unprocessing Images for Learned Raw Denoising”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
- [Car+23] N. Carlini, F. Tramer, K. D. Dvijotham, L. Rice, M. Sun, and J. Z. Kolter. “(Certified!!) Adversarial Robustness for Free!” In: *International Conference on Learning Representations*. 2023.
- [CW17] N. Carlini and D. Wagner. “MagNet and ”Efficient Defenses Against Adversarial Attacks” Are Not Robust to Adversarial Examples”. In: *arXiv:1711.08478* (2017).
- [Cho+22] J.-H. Choi, H. Zhang, J.-H. Kim, C.-J. Hsieh, and J.-S. Lee. “Deep Image Destruction: Vulnerability of Deep Image-to-Image Models against Adversarial Attacks”. In: *International Conference on Pattern Recognition*. 2022.
- [Cho+19] J.-H. Choi, H. Zhang, J.-H. Kim, C.-J. Hsieh, and J.-S. Lee. “Evaluating Robustness of Deep Image Super-Resolution Against Adversarial Attacks”. In: *IEEE International Conference on Computer Vision*. 2019.
- [CRK19] J. Cohen, E. Rosenfeld, and Z. Kolter. “Certified Adversarial Robustness via Randomized Smoothing”. In: *International Conference on Machine Learning*. 2019.
- [DCH21] M. Z. Darestani, A. S. Chaudhari, and R. Heckel. “Measuring Robustness in Deep Learning Based Compressive Sensing”. In: *International Conference on Machine Learning*. 2021.
- [DH21] M. Z. Darestani and R. Heckel. “Accelerated MRI With Un-Trained Neural Networks”. In: *IEEE Transactions on Computational Imaging* 7 (2021).
- [FMDF16] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard. “Robustness of Classifiers: From Adversarial to Random Noise”. In: *Advances in Neural Information Processing Systems*. 2016.
- [GCM22] K. V. Gandikota, P. Chandramouli, and M. Moeller. “On Adversarial Robustness of Deep Image Deblurring”. In: *IEEE International Conference on Image Processing*. 2022.

- [GMM22] M. Genzel, J. Macdonald, and M. Marz. “Solving Inverse Problems With Deep Neural Networks - Robustness Included”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (1 2022).
- [Gil+19] J. Gilmer, N. Ford, N. Carlini, and E. Cubuk. “Adversarial Examples Are a Natural Consequence of Test Error in Noise”. In: *International Conference on Machine Learning*. 2019.
- [HK92] L. Holmstrom and P. Koistinen. “Using Additive Noise in Back-Propagation Training”. In: *IEEE Transactions on Neural Networks* 3.1 (1992).
- [Hua+18] Y. Huang, T. Würfl, K. Breininger, L. Liu, G. Lauritsch, and A. Maier. “Some Investigations on Robustness of Deep Learning in Limited Angle Tomography”. In: *Medical Image Computing and Computer Assisted Intervention*. 2018.
- [JSH20] A. Javanmard, M. Soltanolkotabi, and H. Hassani. “Precise Tradeoffs in Adversarial Training for Linear Regression”. In: *Conference on Learning Theory*. 2020.
- [Jin+17] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. “Deep Convolutional Neural Network for Inverse Problems in Imaging”. In: *IEEE Transactions on Image Processing* 26.9 (2017).
- [KKG18] H. Kannan, A. Kurakin, and I. Goodfellow. “Adversarial Logit Pairing”. In: *arXiv:1803.06373 [cs]* (2018).
- [Lee+21] B. D. Lee, T. T. C. K. Zhang, H. Hassani, and N. Matni. “Adversarial Tradeoffs in Linear Inverse Problems and Robust State Estimation”. In: *arXiv:2111.08864 [cs, eess]* (2021).
- [Mad+18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. “Towards Deep Learning Models Resistant to Adversarial Attacks”. In: *International Conference on Learning Representations*. 2018.
- [RBL20] A. Raj, Y. Bresler, and B. Li. “Improving Robustness of Deep-Learning-Based Image Reconstruction”. In: *International Conference on Machine Learning*. 2020.
- [RMO95] R. Reed, R. Marks, and S. Oh. “Similarities of Error Regularization, Sigmoid Gain Scaling, Target Smoothing, and Training with Jitter”. In: *IEEE Transactions on Neural Networks* 6.3 (1995).
- [RMI99] R. Reed and R. J. Marks II. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. 1999.
- [RFB15] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention*. 2015.
- [Rus+20] E. Rusak, L. Schott, R. S. Zimmermann, J. Bitterwolf, O. Bringmann, M. Bethge, and W. Brendel. “A Simple Way to Make Neural Networks Robust Against Diverse Image Corruptions”. In: *European Conference on Computer Vision*. 2020.
- [Sal+20] H. Salman, M. Sun, G. Yang, A. Kapoor, and J. Z. Kolter. “Denoised Smoothing: A Provable Defense for Pretrained Classifiers”. In: *Advances in Neural Information Processing Systems*. 2020.

- [Sal+19] H. Salman, G. Yang, J. Li, P. Zhang, H. Zhang, I. Razenshteyn, and S. Bubeck. “Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers”. In: *Advances in Neural Information Processing Systems*. 2019.
- [Sha+19] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein. “Are Adversarial Examples Inevitable?” In: *International Conference on Learning Representations*. 2019.
- [Sri+20] A. Sriram, J. Zbontar, T. Murrell, A. Defazio, C. L. Zitnick, N. Yakubova, F. Knoll, and P. Johnson. “End-to-End Variational Networks for Accelerated MRI Reconstruction”. In: *Medical Image Computing and Computer Assisted Intervention*. 2020.
- [Sze+14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. “Intriguing properties of neural networks”. In: *International Conference on Learning Representations*. 2014.
- [Tsi+19] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. “Robustness May Be at Odds with Accuracy”. In: *International Conference on Learning Representations*. 2019.
- [WRK20] E. Wong, L. Rice, and J. Z. Kolter. “Fast Is Better than Free: Revisiting Adversarial Training”. In: *International Conference on Learning Representations*. 2020.
- [Yan+22] H. Yan, J. Zhang, J. Feng, M. Sugiyama, and V. Y. F. Tan. “Towards Adversarially Robust Deep Image Denoising”. In: *International Joint Conference on Artificial Intelligence*. 2022.
- [ZNR17] V. Zantedeschi, M.-I. Nicolae, and A. Rawat. “Efficient Defenses Against Adversarial Attacks”. In: *ACM Workshop on Artificial Intelligence and Security*. 2017.
- [Zbo+18] J. Zbontar et al. “fastMRI: An Open Dataset and Benchmarks for Accelerated MRI”. In: *arXiv:1811.08839 [cs.CV]* (2018).

A Proof of Theorem 1

In the main body we stated an analytical characterization of the optimal worst-case robust denoiser. We present the proof in the following and show that the risk

$$\min_{\mathbf{H}} R_{\epsilon}(\mathbf{H}) = \min_{\mathbf{H}} \mathbb{E}_{\mathbf{x}, \mathbf{z}} \left[\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{H}(\mathbf{x} + \mathbf{e} + \mathbf{z}) - \mathbf{x}\|_2^2 \right] \quad (6)$$

is minimized by $\mathbf{H} = \sigma \mathbf{U} \mathbf{U}^T$ with

$$\sigma = \begin{cases} \sigma_c^2 - \frac{\epsilon \sigma_c \sigma_z \sqrt{\frac{d}{n}}}{\sqrt{\sigma_c^2 + \sigma_z^2 \frac{d}{n} - \epsilon^2}} & \text{if } \sigma_c^2 > \epsilon^2 \\ \sigma_c^2 + \sigma_z^2 \frac{d}{n} & \text{else} \\ 0 & \text{else} \end{cases}$$

The scaling factor σ of optimal worst-case estimator is visualized in Figure 2.

We start by proving a lower bound of the risk, which relies on a characterization of the maximization and many unexpected applications of Jensens inequality.

We then compute the risk for $\mathbf{H} = \sigma \mathbf{U} \mathbf{U}^T$ and show that it is equivalent to the lower bound on the risk.

A.1 Lower bounding the risk

Towards lower bounding the risk we define, for notational convenience

$$\begin{bmatrix} \mathbf{H}_{\parallel} & \mathbf{H}_0 \\ \mathbf{H}_0^T & \mathbf{H}_{\perp} \end{bmatrix} = \begin{bmatrix} \mathbf{U}^T \mathbf{H} \mathbf{U} & \mathbf{U}^T \mathbf{H} \mathbf{U}_{\perp} \\ \mathbf{U}_{\perp}^T \mathbf{H} \mathbf{U} & \mathbf{U}_{\perp}^T \mathbf{H} \mathbf{U}_{\perp} \end{bmatrix}.$$

With

$$\mathbf{H} = [\mathbf{U} \quad \mathbf{U}_{\perp}] \begin{bmatrix} \mathbf{H}_{\parallel} & \mathbf{H}_0 \\ \mathbf{H}_0^T & \mathbf{H}_{\perp} \end{bmatrix} \begin{bmatrix} \mathbf{U}^T \\ \mathbf{U}_{\perp}^T \end{bmatrix}$$

we get

$$\begin{aligned} \max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{H}(\mathbf{x} + \mathbf{e} + \mathbf{z}) - \mathbf{x}\|_2^2 &= \max_{\|\mathbf{e}\|_2 \leq \epsilon} \left\| [\mathbf{U} \quad \mathbf{U}_{\perp}] \begin{bmatrix} \mathbf{H}_{\parallel} & \mathbf{H}_0 \\ \mathbf{H}_0^T & \mathbf{H}_{\perp} \end{bmatrix} \begin{bmatrix} \mathbf{U}^T \\ \mathbf{U}_{\perp}^T \end{bmatrix} (\mathbf{U} \mathbf{c} + \mathbf{e} + \mathbf{z}) - \mathbf{U} \mathbf{c} \right\|_2^2 \\ &= \max_{\|\mathbf{e}\|_2 \leq \epsilon} \left\| \begin{bmatrix} \mathbf{H}_{\parallel} & \mathbf{H}_0 \\ \mathbf{H}_0^T & \mathbf{H}_{\perp} \end{bmatrix} \begin{bmatrix} \mathbf{c} + \mathbf{U}^T \mathbf{z} + \mathbf{U}^T \mathbf{e} \\ \mathbf{U}_{\perp}^T \mathbf{z} + \mathbf{U}_{\perp}^T \mathbf{e} \end{bmatrix} - \begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix} \right\|_2^2 \\ &\geq \max_{\|\mathbf{e}\|_2 \leq \epsilon} \left\| \begin{bmatrix} \mathbf{H}_{\parallel} & \mathbf{H}_0 \end{bmatrix} \begin{bmatrix} \mathbf{c} + \mathbf{U}^T \mathbf{z} + \mathbf{U}^T \mathbf{e} \\ \mathbf{U}_{\perp}^T \mathbf{z} + \mathbf{U}_{\perp}^T \mathbf{e} \end{bmatrix} - \mathbf{c} \right\|_2^2 \\ &\geq \max_{\|\mathbf{e}_{\parallel}\|_2 \leq \epsilon} \left\| \mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z}_{\parallel} + \mathbf{e}_{\parallel}) - \mathbf{c} + \mathbf{H}_0 \mathbf{z}_{\perp} \right\|_2^2. \end{aligned}$$

Here, we defined $\mathbf{e}_{\parallel} = \mathbf{U}^T \mathbf{e}$ and $\mathbf{z}_{\parallel} = \mathbf{U}^T \mathbf{z}$ for notational convenience, and the last inequality follows by adding $\mathbf{U}_{\perp}^T \mathbf{e} = 0$ as a constraint to the maximization, which gives a lower bound.

Next, note that since $\mathbf{z} = \begin{bmatrix} \mathbf{z}_{\parallel} \\ \mathbf{z}_{\perp} \end{bmatrix} \sim \mathcal{N}(0, \sigma_z^2/n\mathbf{I})$ is Gaussian, the vector $\begin{bmatrix} \mathbf{z}_{\parallel} \\ -\mathbf{z}_{\perp} \end{bmatrix}$ is equally Gaussian distributed $\mathcal{N}(0, \sigma_z^2/n\mathbf{I})$. This implies that

$$\mathbb{E} \left[\max_{\|\mathbf{e}_{\parallel}\|_2 \leq \epsilon} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z}_{\parallel} + \mathbf{e}_{\parallel}) - \mathbf{c} + \mathbf{H}_0\mathbf{z}_{\perp}\|_2^2 \right] = \mathbb{E} \left[\max_{\|\mathbf{e}_{\parallel}\|_2 \leq \epsilon} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z}_{\parallel} + \mathbf{e}_{\parallel}) - \mathbf{c} - \mathbf{H}_0\mathbf{z}_{\perp}\|_2^2 \right].$$

Then

$$\begin{aligned} & \mathbb{E} \left[\max_{\|\mathbf{e}_{\parallel}\|_2 \leq \epsilon} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z}_{\parallel} + \mathbf{e}_{\parallel}) - \mathbf{c} + \mathbf{H}_0\mathbf{z}_{\perp}\|_2^2 \right] \\ &= \frac{1}{2} \mathbb{E} \left[\max_{\|\mathbf{e}_1\|_2 \leq \epsilon} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z}_{\parallel} + \mathbf{e}_1) - \mathbf{c} + \mathbf{H}_0\mathbf{z}_{\perp}\|_2^2 + \max_{\|\mathbf{e}_2\|_2 \leq \epsilon} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z}_{\parallel} + \mathbf{e}_2) - \mathbf{c} - \mathbf{H}_0\mathbf{z}_{\perp}\|_2^2 \right] \\ &\geq \mathbb{E} \left[\max_{\|\mathbf{e}_1\|_2 \leq \epsilon} \frac{1}{2} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z}_{\parallel} + \mathbf{e}_1) - \mathbf{c} + \mathbf{H}_0\mathbf{z}_{\perp}\|_2^2 + \frac{1}{2} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z}_{\parallel} + \mathbf{e}_1) - \mathbf{c} - \mathbf{H}_0\mathbf{z}_{\perp}\|_2^2 \right] \\ &\geq \mathbb{E} \left[\max_{\|\mathbf{e}_1\|_2 \leq \epsilon} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z}_{\parallel} + \mathbf{e}_1) - \mathbf{c}\|_2^2 \right], \end{aligned}$$

where the last step follows from Jensens inequality. Thus, we have shown that

$$\min_{\mathbf{H}} \mathbb{E} \left[\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{H}(\mathbf{x} + \mathbf{e} + \mathbf{z}) - \mathbf{x}\|_2^2 \right] \geq \min_{\mathbf{H}_{\parallel}} \mathbb{E} \left[\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z} + \mathbf{e}) - \mathbf{c}\|_2^2 \right]. \quad (7)$$

For simplicity of exposition, we drop the $\|\cdot\|$ -notation. Thus, with a slight abuse of notation, the vectors on the left and right hand side have different dimensions. On the left hand side, $\mathbf{z}, \mathbf{e} \in \mathbb{R}^n$, while on the right hand side $\mathbf{z}, \mathbf{e} \in \mathbb{R}^d$ and $\mathbf{z}$ throughout has iid $\mathcal{N}(0, \sigma_z^2/n)$ entries.

Next, we'll apply the lemma below for characterizing the maximization inside of the expectation. The proof is in Section A.3. Similar computations as used to prove the lemma are on page 19-20 in the paper Lee et al. [Lee+21] for deriving robustness-accuracy trade-off bounds.

Lemma 1. *For any $\mathbf{z}$,*

$$\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{z} - \mathbf{H}\mathbf{e}\|_2^2 = \min_{\lambda: \lambda \geq \lambda_{\max}(\mathbf{H}^T\mathbf{H})} \lambda \epsilon^2 + \mathbf{z}^T \left(\mathbf{I} - \frac{1}{\lambda} \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{z}. \quad (8)$$

Using Lemma 1 the term within the expectation is

$$\begin{aligned} & \max_{\|\mathbf{e}\|_2 \leq \epsilon} \|(\mathbf{H} - \mathbf{I})\mathbf{c} + \mathbf{H}\mathbf{z} + \mathbf{H}\mathbf{e}\|_2^2 \\ &= \min_{\lambda: \lambda \geq \lambda_{\max}(\mathbf{H}^T\mathbf{H})} \lambda \epsilon^2 + ((\mathbf{H}^T - \mathbf{I})\mathbf{c} + \mathbf{H}\mathbf{z})^T \left(\mathbf{I} - \frac{1}{\lambda} \mathbf{H}\mathbf{H}^T \right)^{-1} ((\mathbf{H}^T - \mathbf{I})\mathbf{c} + \mathbf{H}\mathbf{z}) \\ &= \min_{\lambda: \lambda \geq \lambda_{\max}(\mathbf{H}^T\mathbf{H})} \lambda \epsilon^2 + \mathbf{c}^T (\mathbf{H} - \mathbf{I}) \left(\mathbf{I} - \frac{1}{\lambda} \mathbf{H}\mathbf{H}^T \right)^{-1} (\mathbf{H} - \mathbf{I})\mathbf{c} \\ & \quad + 2\mathbf{z}^T \mathbf{H}^T \left(\mathbf{I} - \frac{1}{\lambda} \mathbf{H}\mathbf{H}^T \right)^{-1} (\mathbf{H} - \mathbf{I})\mathbf{c} \\ & \quad + \mathbf{z}^T \mathbf{H}^T \left(\mathbf{I} - \frac{1}{\lambda} \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{H}\mathbf{z}. \end{aligned}$$

Now let's rewrite using the eigenvalue decomposition of the symmetric matrix $\mathbf{H}_{\parallel} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$. With this notation, the optimization problem on the right hand side of inequality (7) is

$$\begin{aligned}
& \min_{\mathbf{H}_{\parallel}} \mathbb{E} \left[\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z} + \mathbf{e}) - \mathbf{c}\|_2^2 \right] \\
&= \min_{\mathbf{V}, \sigma_i} \mathbb{E}_{\mathbf{c}, \mathbf{z}} \left[\min_{\lambda: \lambda \geq \sigma_i^2} \lambda \epsilon^2 + \sum_{i=1}^d \frac{(\mathbf{v}_i^T \mathbf{c})^2 (\sigma_i - 1)^2 + (\mathbf{v}_i^T \mathbf{z})^2 \sigma_i^2 + 2\sigma_i(\sigma_i - 1)(\mathbf{v}_i^T \mathbf{z})(\mathbf{v}_i^T \mathbf{c})}{1 - \frac{\sigma_i^2}{\lambda}} \right] \\
&= \min_{\mathbf{V}, \sigma_i} \mathbb{E}_{\mathbf{c}, \mathbf{z}} \left[\min_{\lambda: \lambda \geq \sigma_i^2} \lambda \epsilon^2 + \sum_{i=1}^n \frac{((\mathbf{v}_i^T \mathbf{c})(\sigma_i - 1) + \mathbf{v}_i^T \mathbf{z} \sigma_i)^2}{1 - \frac{\sigma_i^2}{\lambda}} \right] \\
&\stackrel{\text{i}}{=} \min_{\sigma_i} \mathbb{E}_{\mathbf{c}, \mathbf{z}} \left[\min_{\lambda: \lambda \geq \sigma_i^2} \lambda \epsilon^2 + \sum_{i=1}^d \frac{(c_i(\sigma_i - 1) + z_i \sigma_i)^2}{1 - \frac{\sigma_i^2}{\lambda}} \right] \\
&\stackrel{\text{ii}}{=} \min_{\sigma_i} \mathbb{E}_{\mathbf{g}} \left[\min_{\lambda: \lambda \geq \sigma_i^2} \lambda \epsilon^2 + \sum_{i=1}^d \frac{g_i^2 \left(\frac{\sigma_c^2}{d} (\sigma_i - 1)^2 + \frac{\sigma_z^2}{n} \sigma_i^2 \right)}{1 - \frac{\sigma_i^2}{\lambda}} \right],
\end{aligned}$$

where minimization above is over an orthonormal $\mathbf{V} \in \mathbb{R}^{n \times n}$ and over the singular values σ_i . Inequality (i) follows from $\mathbf{c}$ and $\mathbf{z}$ having iid Gaussian entries and inequality (ii) holds since the random variables $c_i(\sigma_i - 1) + z_i \sigma_i$ are iid zero-mean Gaussian with variance $\frac{\sigma_c^2}{d}(\sigma_i - 1)^2 + \frac{\sigma_z^2}{n}\sigma_i^2$, and $g_i \sim \mathcal{N}(0, 1)$.

First note that the function $\frac{x^2}{y}$ is convex in (x, y) when $y > 0$. Also the extended value function is increasing in the first input and decreasing in the second input. Furthermore the mappings $(x, z) \mapsto x - 1$ and $(x, z) \mapsto 1 - \frac{x^2}{z}$ are convex and concave. Thus by the composition rule of convex functions we conclude that the functions

$$\frac{x^2}{1 - \frac{(x-1)^2}{z}} \quad \text{and} \quad \frac{(x-1)^2}{1 - \frac{(x-1)^2}{z}}$$

are jointly convex in (x, z) .

Jensen's inequality states that for a convex function ψ we have $\frac{\sum_i z_i^2 \psi(x_i)}{\sum_i z_i^2} \geq \psi\left(\frac{\sum_i z_i^2 x_i}{\sum_i z_i^2}\right)$. Thus by Jensen's inequality the sum in the expectation in the right-hand-side of the equation above can be lower-bounded as

$$\begin{aligned}
\sum_{i=1}^d \frac{g_i^2 \frac{\sigma_c^2}{d} (\sigma_i - 1)^2 + g_i^2 \frac{\sigma_z^2}{n} \sigma_i^2}{1 - \frac{\sigma_i^2}{\lambda}} &= \|\mathbf{g}\|_2^2 \sum_{i=1}^d \frac{\frac{g_i^2}{\|\mathbf{g}\|_2^2} \frac{\sigma_c^2}{d} (\sigma_i - 1)^2 + \frac{g_i^2}{\|\mathbf{g}\|_2^2} \frac{\sigma_z^2}{n} \sigma_i^2}{1 - \frac{\sigma_i^2}{\lambda}} \\
&\geq \|\mathbf{g}\|_2^2 \frac{\frac{\sigma_c^2}{d} \left(\sum_{i=1}^d \frac{g_i^2}{\|\mathbf{g}\|_2^2} \sigma_i - 1 \right)^2 + \frac{\sigma_z^2}{n} \left(\sum_{i=1}^d \frac{g_i^2}{\|\mathbf{g}\|_2^2} \sigma_i \right)^2}{1 - \frac{\left(\sum_{i=1}^d \frac{g_i^2}{\|\mathbf{g}\|_2^2} \sigma_i \right)^2}{\lambda}} \\
&= \|\mathbf{g}\|_2^2 \frac{\frac{\sigma_c^2}{d} (\bar{\sigma}(\mathbf{g}) - 1)^2 + \frac{\sigma_z^2}{n} (\bar{\sigma}(\mathbf{g}))^2}{1 - \frac{(\bar{\sigma}(\mathbf{g}))^2}{\lambda}}
\end{aligned}$$

where $\bar{\sigma}(\mathbf{g}) = \sum_{i=1}^d \frac{g_i^2}{\|\mathbf{g}\|_2^2} \sigma_i$.

Now consider the event $\mathcal{E} = \{\mathbf{g}: \|\mathbf{g}\|_2^2 \geq (1-\delta)d\}$ which holds with probability at least $1 - e^{-\frac{\delta^2}{2}d}$. On this event, we have

$$\lambda\epsilon^2 + \sum_{i=1}^d \frac{g_i^2 \frac{\sigma_c^2}{d} (\sigma_i - 1)^2 + g_i^2 \frac{\sigma_z^2}{n} \sigma_i^2}{1 - \frac{\sigma_i^2}{\lambda}} \geq \lambda\epsilon^2 + (1-\delta) \frac{\sigma_c^2 (\bar{\sigma}(\mathbf{g}) - 1)^2 + \sigma_z^2 \frac{d}{n} (\bar{\sigma}(\mathbf{g}))^2}{1 - \frac{(\bar{\sigma}(\mathbf{g}))^2}{\lambda}}.$$

Using the same argument as before the right hand side of the above inequality is jointly convex in $(\lambda, \bar{\sigma}(\mathbf{c}))$. Since partial minimization of a jointly convex function preserves convexity we conclude that the function

$$\psi(\bar{\sigma}) = \min_{\lambda: \lambda \geq \sigma_i^2} \mathbb{1}_{\mathcal{E}} \lambda\epsilon^2 + \mathbb{1}_{\mathcal{E}} (1-\delta) \frac{\sigma_c^2 (\bar{\sigma}(\mathbf{g}) - 1)^2 + \sigma_z^2 \frac{d}{n} (\bar{\sigma}(\mathbf{g}))^2}{1 - \frac{(\bar{\sigma}(\mathbf{g}))^2}{\lambda}}$$

is convex in $\bar{\sigma}$. Thus by using convexity in terms of $\bar{\sigma}$, applying Jensen's inequality (i.e., $\mathbb{E}[\psi(\bar{\sigma})] \geq \psi(\mathbb{E}[\bar{\sigma}])$) we have

$$\begin{aligned} & \min_{\sigma_i} \mathbb{E}_{\mathbf{g}} \left[\min_{\lambda: \lambda \geq \sigma_i^2} \lambda\epsilon^2 + \|\mathbf{g}\|_2^2 \frac{1}{d} \frac{\sigma_c^2 (\bar{\sigma}(\mathbf{g}) - 1)^2 + \sigma_z^2 \frac{d}{n} (\bar{\sigma}(\mathbf{g}))^2}{1 - \frac{(\bar{\sigma}(\mathbf{g}))^2}{\lambda}} \right] \\ & \geq \min_{\sigma_i} \mathbb{E}_{\mathbf{g}} \left[\min_{\lambda: \lambda \geq \sigma_i^2} \mathbb{1}_{\mathcal{E}} \lambda\epsilon^2 + \mathbb{1}_{\mathcal{E}} (1-\delta) \frac{\sigma_c^2 (\bar{\sigma}(\mathbf{g}) - 1)^2 + \sigma_z^2 \frac{d}{n} (\bar{\sigma}(\mathbf{g}))^2}{1 - \frac{(\bar{\sigma}(\mathbf{g}))^2}{\lambda}} \right] \\ & \geq \min_{\sigma_i} \mathbb{E}_{\mathbf{g}|\mathcal{E}} \left[\min_{\lambda: \lambda \geq \sigma_i^2} \lambda\epsilon^2 + (1-\delta) \frac{\sigma_c^2 (\bar{\sigma}(\mathbf{g}) - 1)^2 + \sigma_z^2 \frac{d}{n} (\bar{\sigma}(\mathbf{g}))^2}{1 - \frac{(\bar{\sigma}(\mathbf{g}))^2}{\lambda}} \right] \\ & \geq \min_{\sigma_i} \min_{\lambda: \lambda \geq \sigma_i^2} \lambda\epsilon^2 + (1-\delta) \frac{\sigma_c^2 (\mathbb{E}[\bar{\sigma}(\mathbf{g})] - 1)^2 + \sigma_z^2 \frac{d}{n} (\mathbb{E}[\bar{\sigma}(\mathbf{g})])^2}{1 - \frac{(\mathbb{E}[\bar{\sigma}(\mathbf{g}))^2]}{\lambda}} \\ & \geq \min_{\sigma_i} \min_{\lambda: \lambda \geq \sigma_i^2} \lambda\epsilon^2 + (1-\delta) \frac{\sigma_c^2 (\bar{\sigma} - 1)^2 + \sigma_z^2 \frac{d}{n} (\bar{\sigma})^2}{1 - \frac{(\bar{\sigma})^2}{\lambda}} \end{aligned}$$

where we defined $\bar{\sigma} = \sum_{i=1}^d \mathbb{E}_{\mathbf{g}|\mathcal{E}} \left[\frac{g_i^2}{\|\mathbf{g}\|_2^2} \right] \sigma_i$. Putting things together, we have shown that

$$\begin{aligned} \min_{\mathbf{H}} R_{\epsilon}(\mathbf{H}) & \geq \min_{\mathbf{H}_{\parallel}} \mathbb{E} \left[\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{H}_{\parallel}(\mathbf{c} + \mathbf{z} + \mathbf{e}) - \mathbf{c}\|_2^2 \right] \\ & = \min_{\sigma_i} \mathbb{E}_{\mathbf{g}} \left[\min_{\lambda: \lambda \geq \sigma_i^2} \lambda\epsilon^2 + \sum_{i=1}^d \frac{g_i^2 \frac{\sigma_c^2}{d} (\sigma_i - 1)^2 + g_i^2 \frac{\sigma_z^2}{n} \sigma_i^2}{1 - \frac{\sigma_i^2}{\lambda}} \right] \\ & \geq \min_{\bar{\sigma}} \min_{\lambda: \lambda \geq \sigma_i^2} \lambda\epsilon^2 + (1-\delta) \frac{\sigma_c^2 (\bar{\sigma} - 1)^2 + \sigma_z^2 \frac{d}{n} (\bar{\sigma})^2}{1 - \frac{(\bar{\sigma})^2}{\lambda}} \\ & \geq \min_{\bar{\sigma}} \min_{\lambda: \lambda \geq \bar{\sigma}^2} \lambda\epsilon^2 + (1-\delta) \frac{\sigma_c^2 (\bar{\sigma} - 1)^2 + \sigma_z^2 \frac{d}{n} (\bar{\sigma})^2}{1 - \frac{(\bar{\sigma})^2}{\lambda}}, \end{aligned}$$

where the last inequality follows from $\sigma_{\max} \geq \bar{\sigma}$. For $d \rightarrow \infty$, we can choose δ arbitrarily small, which yields

$$\min_{\mathbf{H}} R_{\epsilon}(\mathbf{H}) \geq \min_{\sigma} \min_{\lambda: \lambda \geq \sigma^2} \lambda \epsilon^2 + \frac{\sigma_c^2 (\sigma - 1)^2 + \sigma_z^2 \frac{d}{n} \sigma^2}{1 - \frac{\sigma^2}{\lambda}}. \quad (9)$$

Solving the optimization problem (9): Consider the inner minimization problem in equation (9), i.e.,

$$\min_{\lambda: \lambda \geq \sigma^2} f(\lambda) \quad \text{with } f(\lambda) = \lambda \epsilon^2 + \frac{c(\sigma)}{1 - \frac{\sigma^2}{\lambda}},$$

where $c(\sigma) = \sigma_c^2 (\sigma - 1)^2 + \sigma_z^2 \frac{d}{n} \sigma^2$ for notational convenience. Since f is differentiable on (σ^2, ∞) we can calculate its critical point λ^* by setting the derivative with respect to λ to zero, which yields

$$\lambda^* = \frac{\sqrt{c(\sigma)}}{\epsilon} \sigma + \sigma^2.$$

From this expression, we see that the constraint $\lambda^* > \sigma^2$ is satisfied and by convexity of f we know that λ^* is the unique minimizer. Hence, we have:

$$\min_{\lambda: \lambda \geq \sigma^2} f(\lambda) = f(\lambda^*) = \left(\epsilon \sigma + \sqrt{c(\sigma)} \right)^2. \quad (10)$$

It follows that

$$\min_{\mathbf{H}} R_{\epsilon}(\mathbf{H}) \geq \min_{\sigma} g(\sigma), \quad g(\sigma) = \left(\epsilon \sigma + \sqrt{\sigma_c^2 (\sigma - 1)^2 + \sigma_z^2 \frac{d}{n} \sigma^2} \right)^2. \quad (11)$$

Calculating the derivative of $g(\sigma)$ and setting it to zero, we get:

$$2 \left(\epsilon \sigma + \sqrt{\sigma_c^2 (\sigma - 1)^2 + \sigma_z^2 \frac{d}{n} \sigma^2} \right) \left(\epsilon + \frac{2\sigma_c^2 (\sigma - 1) + \sigma_z^2 \frac{d}{n} 2\sigma}{2\sqrt{\sigma_c^2 (\sigma - 1)^2 + \sigma_z^2 \frac{d}{n} \sigma^2}} \right) = 0$$

The left factor is non-negative, and the right factor is zero if $\sigma_c^2 > \epsilon^2$ and if

$$\sigma^* = \frac{\sigma_c^2 - \frac{\epsilon \sigma_c \sigma_z \sqrt{\frac{d}{n}}}{\sqrt{\sigma_c^2 + \sigma_z^2 \frac{d}{n} - \epsilon^2}}}{\sigma_c^2 + \sigma_z^2 \frac{d}{n}}. \quad (12)$$

If $\sigma_c^2 \leq \epsilon^2$, then the function $g(\sigma)$ is monotonically increasing on $[0, \infty)$ and hence $\sigma_* = 0$ is the minimizer.

A.2 Upper bound for the risk of the estimator $\mathbf{H} = \sigma \mathbf{U} \mathbf{U}^T$

We upper bound the risk of the estimator $\mathbf{H} = \sigma \mathbf{U} \mathbf{U}^T$. For $\mathbf{H} = \sigma \mathbf{U} \mathbf{U}^T$ the risk (6) becomes

$$\begin{aligned} R_\epsilon(\sigma \mathbf{U} \mathbf{U}^T) &= \mathbb{E}_{\mathbf{x}, \mathbf{z}} \left[\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{H}(\mathbf{x} + \mathbf{e} + \mathbf{z}) - \mathbf{x}\|_2^2 \right] \\ &= \mathbb{E}_{\mathbf{x}, \mathbf{z}} \left[\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\sigma \mathbf{U} \mathbf{U}^T (\mathbf{U} \mathbf{c} + \mathbf{e} + \mathbf{z}) - \mathbf{U} \mathbf{c}\|_2^2 \right] \\ &= \mathbb{E}_{\mathbf{c}, \mathbf{z}} \left[\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|(\sigma - 1)\mathbf{c} + \sigma \mathbf{e}_\parallel + \sigma \mathbf{z}_\parallel\|_2^2 \right]. \end{aligned}$$

With Lemma 1,

$$\begin{aligned} R_\epsilon(\sigma \mathbf{U} \mathbf{U}^T) &= \min_{\mathbf{H}} \mathbb{E}_{\mathbf{c}, \mathbf{z}_\parallel} \left[\min_{\lambda: \lambda \geq \sigma^2} \lambda \epsilon^2 + ((\sigma - 1)\mathbf{c} + \sigma \mathbf{z}_\parallel)^T \left(\mathbf{I} - \frac{1}{\lambda} \sigma^2 \mathbf{I} \right)^{-1} ((\sigma - 1)\mathbf{c} + \sigma \mathbf{z}_\parallel) \right] \\ &= \min_{\mathbf{H}} \mathbb{E}_{\mathbf{c}, \mathbf{z}} \left[\min_{\lambda: \lambda \geq \sigma^2} \lambda \epsilon^2 + \sum_{i=1}^d \frac{((\sigma - 1)c_i + \sigma z_i)^2}{1 - \frac{\sigma^2}{\lambda}} \right] \\ &= \min_{\mathbf{H}} \mathbb{E}_{\mathbf{g}} \left[\min_{\lambda: \lambda \geq \sigma^2} \lambda \epsilon^2 + \sum_{i=1}^d g_i^2 \frac{(\sigma - 1)^2 \frac{\sigma_c^2}{d} + \sigma \frac{\sigma_z^2}{n}}{1 - \frac{\sigma^2}{\lambda}} \right]. \end{aligned}$$

Using the optimal λ^* , by equation 10, we get

$$\begin{aligned} R_\epsilon(\sigma \mathbf{U} \mathbf{U}^T) &= \mathbb{E} \left[\left(\epsilon \sigma + \|\mathbf{g}\|_2 \frac{1}{\sqrt{d}} \sqrt{(\sigma - 1)^2 \sigma_c^2 + \sigma \frac{d}{n} \sigma_z^2} \right)^2 \right] \\ &= \mathbb{E} \left[(\epsilon \sigma)^2 + 2\epsilon \sigma \|\mathbf{g}\|_2 \frac{1}{\sqrt{d}} \sqrt{(\sigma - 1)^2 \sigma_c^2 + \sigma \frac{d}{n} \sigma_z^2} + \|\mathbf{g}\|_2^2 \frac{1}{d} \left((\sigma - 1)^2 \sigma_c^2 + \sigma \frac{d}{n} \sigma_z^2 \right) \right] \\ &\stackrel{\text{i}}{\leq} (\epsilon \sigma)^2 + 2\epsilon \sigma \sqrt{(\sigma - 1)^2 \sigma_c^2 + \sigma \frac{d}{n} \sigma_z^2} + \left((\sigma - 1)^2 \sigma_c^2 + \sigma \frac{d}{n} \sigma_z^2 \right) \\ &\stackrel{\text{ii}}{=} \left(\epsilon \sigma + \sqrt{(\sigma - 1)^2 \sigma_c^2 + \sigma \frac{d}{n} \sigma_z^2} \right)^2, \end{aligned}$$

where equation (i) follows by using Jensen's inequality once again (specifically, using $(\mathbb{E}[\|\mathbf{g}\|_2])^2 \leq \mathbb{E}[\|\mathbf{g}\|_2^2] = d$). Noting that (ii) is equal to the lower bound of the risk for any symmetric $\mathbf{H}$ in equation (11) shows that $\mathbf{H} = \sigma \mathbf{U} \mathbf{U}^T$ with the optimal parameter σ^* derived above is optimal.

A.3 Proof of Lemma 1

The optimization problem

$$\max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{z} - \mathbf{H} \mathbf{e}\|_2^2 \tag{13}$$

can be written as

$$\min_{\mathbf{e}} -\|\mathbf{z} - \mathbf{H}\mathbf{e}\|_2^2 \text{ subject to } \epsilon^2 - \mathbf{e}^T \mathbf{e} \geq 0.$$

The corresponding Lagrangian is, for $\lambda \geq 0$

$$\begin{aligned} L(\mathbf{e}, \lambda) &= -(\mathbf{z} - \mathbf{H}\mathbf{e})^T (\mathbf{z} - \mathbf{H}\mathbf{e}) - \lambda(\epsilon^2 - \mathbf{e}^T \mathbf{e}) \\ &= -\mathbf{z}^T \mathbf{z} + 2\mathbf{z}^T \mathbf{H}\mathbf{e} - \mathbf{e}^T \mathbf{H}^T \mathbf{H}\mathbf{e} - \lambda\epsilon^2 + \lambda\mathbf{e}^T \mathbf{e} \\ &= \mathbf{e}^T (\lambda \mathbf{I} - \mathbf{H}^T \mathbf{H}) \mathbf{e} + 2\mathbf{z}^T \mathbf{H}\mathbf{e} - \lambda\epsilon^2 - \mathbf{z}^T \mathbf{z} \end{aligned}$$

The Lagrange Dual Function is

$$q(\lambda) = \inf_{\mathbf{e}} L(\mathbf{e}, \lambda).$$

Using that $\min_{\mathbf{e}} 2\mathbf{c}^T \mathbf{e} + \mathbf{e}^T \mathbf{B}\mathbf{e} = -\mathbf{c}^T \mathbf{B}^{-1} \mathbf{c}$, we get, provided that $\lambda \geq \lambda_{\max}(\mathbf{H}^T \mathbf{H})$,

$$\begin{aligned} q(\lambda) &= -\mathbf{z}^T \mathbf{H}^T (\lambda \mathbf{I} - \mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{z} - \lambda\epsilon^2 - \mathbf{z}^T \mathbf{z} \\ &= -\mathbf{z}^T \left(\mathbf{I} + \mathbf{H}(\lambda \mathbf{I} - \mathbf{H}^T \mathbf{H})^{-1} \mathbf{H} \right) \mathbf{z} - \lambda\epsilon^2 \\ &= -\mathbf{z}^T \left(\mathbf{I} - \frac{1}{\lambda} \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{z} - \lambda\epsilon^2 \end{aligned}$$

where the last equality follows from the Woodbury Identity. Thus, the dual problem is

$$\begin{aligned} \max_{\lambda \geq 0} q(\lambda) &= \min_{\lambda \geq 0} -q(\lambda) \\ &= \min_{\lambda \geq \lambda_{\max}(\mathbf{H}^T \mathbf{H})} \lambda\epsilon^2 + \mathbf{z}^T \left(\mathbf{I} - \frac{1}{\lambda} \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{z}. \end{aligned}$$

Even though the primal problem (13) is non-convex, strong duality holds, since the primal program is a quadratic program that is strictly feasible, see Boyd and Vandenberghe [BV04, Appendix B.1]. The primal problem is strictly feasible if there exists a vector $\mathbf{e}$ such that $\epsilon^2 - \mathbf{e}^T \mathbf{e} > 0$. This is trivially satisfied as long as $\epsilon > 0$.

B Additional proofs for denoising

We state two more proofs on optimal worst-case denoisers and jittering.

B.1 Proof of Corollary 1

In the main text it was stated that symmetric linear estimators $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ that minimize the jittering risk J_{σ_w} with noise level chosen as a function of the desired noise level ϵ as

$$\sigma_w(\epsilon) = \sqrt{\frac{\epsilon^2 \sigma_z^2 \frac{d}{n} + \sigma_z \sqrt{\frac{d}{n}} \sigma_c \epsilon \sqrt{\sigma_c^2 - \epsilon^2} + \sigma_z^2 \frac{d}{n}}{d(\sigma_c^2 - \epsilon^2)}}$$

also minimizes the worst-case risk R_ϵ .

The result follows from Theorem 1. For that let f_r and f_j be linear estimators minimizing the robust risk R_ϵ and the jittering risk J_{σ_w} , respectively. By Theorem 1, the two estimators are scaled projections onto the subspace, i.e., $f_r(\mathbf{y}) = \alpha_r \mathbf{U}\mathbf{U}^T$ and $f_j(\mathbf{y}) = \alpha_j \mathbf{U}\mathbf{U}^T$ with

$$\alpha_r = \frac{\sigma_c^2 - \frac{\epsilon \sigma_c \sigma_z \sqrt{\frac{d}{n}}}{\sqrt{\sigma_c^2 - \epsilon^2 + \sigma_z^2 \frac{d}{n}}}}{\sigma_c^2 + \sigma_z^2 \frac{d}{n}} \quad \text{and} \quad \alpha_j = \frac{\sigma_c^2}{\sigma_c^2 + \sigma_z^2 \frac{d}{n} + \sigma_w^2 d}.$$

Setting $\alpha_r = \alpha_j$ and solving for the standard deviation σ_w yields the result.

B.2 Form of perturbations for estimators $\mathbf{H} = \alpha \mathbf{U}\mathbf{U}^T$

We noted in the main body that for estimators $\mathbf{H} = \alpha \mathbf{U}\mathbf{U}^T$ worst-case perturbations can be computed in closed form for fixed $\mathbf{y}$ and $\mathbf{x}$. We calculate:

$$\begin{aligned} \hat{\mathbf{e}} &= \arg \max_{\|\mathbf{e}\|_2 \leq \epsilon} \|\mathbf{H}(\mathbf{x} + \mathbf{z} + \mathbf{e}) - \mathbf{x}\|_2^2 \\ &= \arg \max_{\|\mathbf{e}\|_2 \leq \epsilon} \|(\mathbf{H} - \mathbf{I})\mathbf{U}\mathbf{c} + \mathbf{U}\mathbf{z} + \mathbf{H}\mathbf{e}\|_2^2 \\ &= \arg \max_{\|\mathbf{e}\|_2 \leq \epsilon} \|(\alpha \mathbf{U}\mathbf{U}^T - \mathbf{I})\mathbf{U}\mathbf{c} + \alpha \mathbf{U}\mathbf{U}^T \mathbf{z} + \alpha \mathbf{U}\mathbf{U}^T \mathbf{e}\|_2^2 \\ &= \arg \max_{\|\mathbf{e}'\|_2 \leq \epsilon} \|(\alpha - 1)\mathbf{U}\mathbf{c} + \alpha \mathbf{U}\mathbf{U}^T \mathbf{z} + \alpha \mathbf{U}\mathbf{U}^T \mathbf{e}'\|_2^2 \\ &= \mathbf{U} \arg \max_{\|\mathbf{e}'\|_2 \leq \epsilon} \|(\alpha - 1)\mathbf{c} + \mathbf{U}^T \mathbf{z} + \sigma \mathbf{e}'\|_2^2 \\ &= \mathbf{U} \epsilon \frac{(1 - \alpha)\mathbf{c} + \alpha \mathbf{U}^T \mathbf{z}}{\|(1 - \alpha)\mathbf{c} + \alpha \mathbf{U}^T \mathbf{z}\|_2}. \end{aligned}$$

Thus the perturbation points into the direction of the signal plus noise lying in the signal subspace.

C Theory for general linear inverse problems

In the following, we consider linear inverse problems $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$, with a linear forward operator $\mathbf{A} \in \mathbb{R}^{m \times n}$ and noise $\mathbf{z} \sim \mathcal{N}(0, \sigma_z^2/m\mathbf{I})$. For denoising ($\mathbf{A} = \mathbf{I}$) we stated an explicit analytical characterization of the optimal worst-case estimator and presented a proof in the appendix. The proof, however, does not generalize in a straightforward manner to more general inverse problems. We conjecture the worst-case optimal linear estimator for large dimensions d and present numerical simulation results. Moreover, we state the proof for the optimal jittering estimator, and demonstrate that it can yield sub-optimal worst-case estimators in general.

C.1 Optimal worst-case robust estimator

As formalized by Lemma 1, the robust-risk (1) of the estimator f can be written as an expectation involving a minimization problem over a single variable (instead of a maximization over an n -

dimensional variable, as in the original definition):

$$R_\epsilon(\mathbf{H}) = \mathbb{E}_{\mathbf{v}} \left[\min_{\lambda \geq \sigma_i^2} \lambda \epsilon^2 + \mathbf{v}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{v} \right], \quad \mathbf{v} = (\mathbf{H} \mathbf{A} - \mathbf{I}) \mathbf{x} + \mathbf{H} \mathbf{z}. \quad (14)$$

Here, σ_i are the singular values of the matrix $\mathbf{H}$. In order to find the optimal robust estimator we wish to solve the optimization problem $\arg \min_{\mathbf{H}} R_\epsilon(\mathbf{H})$. The difficulty in solving this optimization problem is that we can't solve the minimization problem within the expectation (14) in closed form. In order to prove Theorem 1 for denoising (i.e., for $\mathbf{A} = \mathbf{I}$) we derived an upper and a matching lower bound of the risks using several unusual applications of Jensen's inequality. The proof does not generalize in a straightforward manner to the more general case where $\mathbf{A} \neq \mathbf{I}$. However, for large d , the random variable $\mathbf{v}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{v}$ concentrates around it's expectation, and thus we conjecture that for large d , we can exchange expectation and minimization, which yields:

$$R_\epsilon(\mathbf{H}) = \min_{\lambda \geq \sigma_i^2} \lambda \epsilon^2 + \mathbb{E}_{\mathbf{v}} \left[\mathbf{v}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{v} \right]. \quad (15)$$

Based on equation (15) we derive a characterization of the optimal worst-case estimator. We proceed similar to the denoising case and start with rearranging the terms in expectation:

$$\begin{aligned} \mathbf{v}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{v} &= ((\mathbf{H} \mathbf{A} - \mathbf{I}) \mathbf{x} + \mathbf{H} \mathbf{z})^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} ((\mathbf{H} \mathbf{A} - \mathbf{I}) \mathbf{x} + \mathbf{H} \mathbf{z}) \\ &= \mathbf{c}^T ((\mathbf{H} \mathbf{A} - \mathbf{I}) \mathbf{U})^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} (\mathbf{H} \mathbf{A} - \mathbf{I}) \mathbf{U} \mathbf{c} \\ &\quad + 2 \mathbf{z}^T \mathbf{H}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} (\mathbf{H} \mathbf{A} - \mathbf{I}) \mathbf{U} \mathbf{c} + \mathbf{z}^T \mathbf{H}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{z}. \end{aligned}$$

Now, let $\mathbf{A} \mathbf{U} = \mathbf{W}^T \mathbf{\Lambda} \mathbf{V}$ be the singular value decomposition of the matrix $\mathbf{A} \mathbf{U}$ and $\mathbf{H} = \mathbf{U} \mathbf{V}^T \mathbf{\Sigma} \mathbf{W}$ the singular value decomposition of $\mathbf{H}$. We then have:

$$\begin{aligned} \mathbf{H} \mathbf{H}^T &= \mathbf{U} \mathbf{V}^T \mathbf{\Sigma} \mathbf{W} (\mathbf{U} \mathbf{V}^T \mathbf{\Sigma} \mathbf{W})^T = \mathbf{U} \mathbf{V}^T \mathbf{\Sigma} \mathbf{W} \mathbf{W}^T \mathbf{\Sigma} \mathbf{V} \mathbf{U}^T = \mathbf{U} \mathbf{V}^T \mathbf{\Sigma}^2 \mathbf{V} \mathbf{U}^T \\ \mathbf{H} \mathbf{A} \mathbf{U} &= \mathbf{U} \mathbf{V}^T \mathbf{\Sigma} \mathbf{W} \mathbf{W}^T \mathbf{\Lambda} \mathbf{V} = \mathbf{U} \mathbf{V}^T \mathbf{\Sigma} \mathbf{\Lambda} \mathbf{V}. \end{aligned}$$

For the individual parts in the summation it follows:

$$\begin{aligned} &\mathbf{c}^T (\mathbf{H} \mathbf{A} \mathbf{U} - \mathbf{U})^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} (\mathbf{H} \mathbf{A} \mathbf{U} - \mathbf{U}) \mathbf{c} \\ &= \mathbf{c}^T (\mathbf{U} \mathbf{V}^T \mathbf{\Sigma} \mathbf{\Lambda} \mathbf{V} - \mathbf{U})^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{U} \mathbf{V}^T \mathbf{\Sigma}^2 \mathbf{V} \mathbf{U}^T)^{-1} (\mathbf{U} \mathbf{V}^T \mathbf{\Sigma} \mathbf{\Lambda} \mathbf{V} - \mathbf{U}) \mathbf{c} \\ &= \mathbf{c}^T (\mathbf{U} \mathbf{V}^T \mathbf{\Sigma} \mathbf{\Lambda} \mathbf{V} - \mathbf{U})^T \mathbf{U} \mathbf{V}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{\Sigma}^2)^{-1} \mathbf{V} \mathbf{U}^T (\mathbf{U} \mathbf{V}^T \mathbf{\Sigma} \mathbf{\Lambda} \mathbf{V} - \mathbf{U}) \mathbf{c} \\ &= (\mathbf{V} \mathbf{c})^T (\mathbf{\Sigma} \mathbf{\Lambda} - \mathbf{I}) (\mathbf{I} - \frac{1}{\lambda} \mathbf{\Sigma}^2)^{-1} (\mathbf{\Sigma} \mathbf{\Lambda} - \mathbf{I}) \mathbf{V} \mathbf{c} = \sum_{i=1}^d \frac{(\sigma_i \lambda_i - 1)^2 c_i}{1 - \frac{\sigma_i^2}{\lambda}} \end{aligned}$$

where we define $c_i := \mathbf{v}_i^T \mathbf{c}$ using the i -th row vector $\mathbf{v}_i$ of the matrix $\mathbf{V}$. Similarly, we set $z_i := \mathbf{w}_i^T \mathbf{z}$, with $\mathbf{w}_i$ the i -th row vector of $\mathbf{W}$, and get for the other parts:

$$2\mathbf{z}^T \mathbf{H}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} (\mathbf{H} \mathbf{A} \mathbf{U} - \mathbf{U}) \mathbf{c} = \sum_{i=1}^d 2 \frac{z_i \sigma_i (\sigma_i \lambda_i - 1) c_i}{1 - \frac{\sigma_i^2}{\lambda}}$$

$$\mathbf{z}^T \mathbf{H}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{z} = \sum_{i=1}^d \frac{z_i^2 \sigma_i^2}{1 - \frac{\sigma_i^2}{\lambda}}.$$

Hence, for the term in expectation in Eq. (15) we get:

$$\begin{aligned} \mathbb{E}_{\mathbf{v}} \left[\mathbf{v}^T (\mathbf{I} - \frac{1}{\lambda} \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{v} \right] &= \mathbb{E}_{\mathbf{c}, \mathbf{z}} \left[\sum_{i=1}^d \frac{(c_i (\sigma_i \lambda_i - 1) + z_i \sigma_i)^2}{1 - \frac{\sigma_i^2}{\lambda}} \right] \\ &= \mathbb{E}_{\mathbf{g}} \left[\sum_{i=1}^d \frac{g_i^2 \left(\frac{\sigma_i^2}{d} (\sigma_i \lambda_i - 1)^2 + \frac{\sigma_z^2}{m} \sigma_i^2 \right)}{1 - \frac{\sigma_i^2}{\lambda}} \right], \end{aligned}$$

where we note that $c_i (\sigma_i \lambda_i - 1) + z_i \sigma_i$ are iid zero-mean gaussian with variance $\frac{\sigma_c^2}{d} (\lambda_i \sigma_i - 1)^2 + \frac{\sigma_z^2}{m} \sigma_i^2$ and $g_i \sim \mathcal{N}(0, 1)$. From this it follows, assuming the robust risk conjecture (15), that the optimal robust estimator minimizes:

$$\min_{\mathbf{H}} R_{\epsilon}(\mathbf{H}) = \min_{\sigma_i, \lambda \geq \sigma_i^2} f(\lambda, \sigma_1, \dots, \sigma_n), \quad (16)$$

$$f(\lambda, \sigma_1, \dots, \sigma_n) = \lambda \epsilon^2 + \sum_{i=1}^d \frac{\frac{\sigma_z^2}{m} \sigma_i^2 + \frac{\sigma_c^2}{d} (\sigma_i \lambda_i - 1)^2}{1 - \frac{\sigma_i^2}{\lambda}}. \quad (17)$$

We first calculate the unconstrained minimizer of the function f and get:

$$\frac{\partial f}{\partial \sigma_i} = \lambda \frac{\lambda \lambda_i^2 \sigma_i \frac{\sigma_c^2}{d} - \lambda_i (\lambda + \sigma_i^2) \frac{\sigma_c^2}{d} + \sigma_i (\frac{\sigma_c^2}{d} + \lambda \frac{\sigma_z^2}{m})}{\lambda - \sigma_i^2} \quad (18)$$

For $\lambda_i = 0$ we obtain $\frac{\partial f}{\partial \sigma_i} = 0 \Rightarrow \sigma_i = 0$ and for $\lambda_i \neq 0$:

$$\frac{\partial f}{\partial \sigma_i} |_{\sigma_i = \sigma_{i,\pm}^*} = 0 \Rightarrow \sigma_{i,\pm}^* = \frac{1 + \lambda_i^2 \lambda}{2 \lambda_i} + \frac{d}{m} \frac{\lambda}{2 \lambda_i} \frac{\sigma_z^2}{\sigma_c^2} \pm \sqrt{\left(\frac{1 + \lambda_i^2 \lambda}{2 \lambda_i} + \frac{d}{m} \frac{\lambda}{2 \lambda_i} \frac{\sigma_z^2}{\sigma_c^2} \right)^2 - \lambda}.$$

Note, for suitable $a(\lambda)$ the solutions $\sigma_{i,\pm}^*$ are of the form

$$\sigma_{i,\pm} = a(\lambda) \pm \sqrt{a(\lambda)^2 - \lambda}.$$

On further examining the constraint $\sigma_i^2 \leq \lambda$ in the problem (16) note that for real numbers $a > b$ it holds $\sqrt{a^2 - b^2} \geq a - b$. Hence, it follows:

$$\sigma_{i,\pm} - \sqrt{\lambda} = a(\lambda) - \sqrt{\lambda} \pm \sqrt{a(\lambda)^2 - \lambda} \geq 0.$$

Hence, we can rule out $\sigma_{i,+}$ and set $\sigma_i^* := \sigma_{i,-}^*$. Finally, inserting σ_i^* for $1 \leq i \leq d$ into the robust risk (17) yields:

$$\min_{\mathbf{H}} R_\epsilon(\mathbf{H}) = \min_{\lambda \geq 0} \lambda \epsilon^2 + \sum_{i=1}^d \frac{1 - \lambda \lambda_i^2}{2} \frac{\sigma_c^2}{d} - \frac{\lambda}{2} \frac{\sigma_z^2}{m} + \sqrt{\left(\frac{1 + \lambda \lambda_i^2}{2} \frac{\sigma_c^2}{d} + \frac{\lambda}{2} \frac{\sigma_z^2}{m} \right)^2 - \lambda \lambda_i^2 \frac{\sigma_c^4}{d^2}}. \quad (19)$$

The optimization problem involved is convex and box-constrained and can thus be solved numerically. Besides the argument above, we confirmed our conjecture with numerical simulations.

C.2 Optimal jittering estimator

We now derive the optimal jittering estimator, i.e. the estimator $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ that minimizes the jittering risk

$$J_{\sigma_w}(f) = \mathbb{E}_{(\mathbf{x}, \mathbf{z}, \mathbf{w})} \left[\|\mathbf{H}(\mathbf{A}\mathbf{x} + \mathbf{w} + \mathbf{z}) - \mathbf{x}\|_2^2 \right],$$

where as before the signal is assumed to lie within a subspace $\mathbf{x} = \mathbf{U}\mathbf{c}$. We first calculate the expectation by using that $\mathbf{x} = \mathbf{U}\mathbf{c}$ with $\mathbf{c} \sim \mathcal{N}(0, \sigma_c^2/d\mathbf{I})$, $\mathbf{z} \sim \mathcal{N}(0, \sigma_z^2/m\mathbf{I})$ and $\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2\mathbf{I})$ are Gaussian distributed.

$$\begin{aligned} J_{\sigma_w}(f) &= \mathbb{E}_{(\mathbf{x}, \mathbf{z}, \mathbf{w})} \left[\|\mathbf{H}(\mathbf{A}\mathbf{x} + \mathbf{z} + \mathbf{w}) - \mathbf{x}\|_2^2 \right] \\ &= \mathbb{E}_{\mathbf{x}} \left[\|\mathbf{H}(\mathbf{A} - \mathbf{I})\mathbf{U}\mathbf{c}\|_2^2 \right] + \mathbb{E}_{\mathbf{z}} \left[\|\mathbf{H}\mathbf{z}\|_2^2 \right] + \mathbb{E}_{\mathbf{w}} \left[\|\mathbf{H}\mathbf{w}\|_2^2 \right] \\ &= \text{tr}((\mathbf{H}\mathbf{A} - \mathbf{I})\mathbf{U}\mathbf{U}^T(\mathbf{H}\mathbf{A} - \mathbf{I})^T) \frac{\sigma_c^2}{d} + \text{tr}(\mathbf{H}\mathbf{H}^T) \frac{\sigma_z^2}{m} + \text{tr}(\mathbf{H}\mathbf{H}^T) \sigma_w^2, \end{aligned}$$

Hence, the optimal jittering estimator minimizes

$$J_{\sigma_w}(f) = \text{tr}(\mathbf{H}\mathbf{X}\mathbf{H}^T) - 2\text{tr}(\mathbf{H}\mathbf{Y}),$$

where $\mathbf{X} := \frac{\sigma_c^2}{d} \mathbf{A}\mathbf{U}(\mathbf{A}\mathbf{U})^T + (\sigma_w^2 + \sigma_z^2/m) \mathbf{I}$ and $\mathbf{Y} := \frac{\sigma_c^2}{d} \mathbf{A}\mathbf{U}\mathbf{U}^T$. Using matrix calculus we calculate the optimal estimator as

$$\begin{aligned} \nabla_{\mathbf{H}} J_{\sigma_w}(f) |_{\mathbf{H}=\mathbf{H}^*} &= 2\mathbf{H}^*\mathbf{X} - 2\mathbf{Y}^T = 0 \\ \Rightarrow \mathbf{H}^* &= \mathbf{Y}^T \mathbf{X}^{-1}. \end{aligned}$$

Now, let $\mathbf{A}\mathbf{U} = \mathbf{W}^T \mathbf{\Lambda} \mathbf{V}$ be the singular value decomposition of $\mathbf{A}\mathbf{U}$. Then:

$$\begin{aligned} \mathbf{X} &:= \frac{\sigma_c^2}{d} \mathbf{W}^T \mathbf{\Lambda}^2 \mathbf{W} + (\sigma_w^2 + \sigma_z^2/m) \mathbf{I} \\ \mathbf{Y} &:= \frac{\sigma_c^2}{d} \mathbf{W}^T \mathbf{\Lambda} \mathbf{V} \mathbf{U}^T. \end{aligned}$$

We get the optimal jittering estimator as:

$$\text{argmin}_{\mathbf{H}} J_{\sigma_w}(f) = \mathbf{Y}^T \mathbf{X}^{-1} = \frac{\sigma_c^2}{d} \mathbf{U} \mathbf{V}^T \mathbf{\Lambda} \mathbf{W} \left(\frac{\sigma_c^2}{d} \mathbf{W}^T \mathbf{\Lambda}^2 \mathbf{W} + (\sigma_w^2 + \sigma_z^2/m) \mathbf{I} \right)^{-1} \quad (20)$$

$$= \frac{\sigma_c^2}{d} \mathbf{U} \mathbf{V}^T \mathbf{\Lambda} \left(\frac{\sigma_c^2}{d} \mathbf{\Lambda}^2 + (\sigma_w^2 + \sigma_z^2/m) \mathbf{I} \right)^{-1} \mathbf{W} \quad (21)$$

$$= \mathbf{U} \mathbf{V}^T \text{diag} \left(\frac{\sigma_c^2 \lambda_i}{\sigma_c^2 \lambda_i^2 + d\sigma_w^2 + \sigma_z^2 \frac{d}{m}} \right) \mathbf{W}. \quad (22)$$

C.3 Numerical simulations supporting Conjecture 1

In the following we present numerical simulations for general linear inverse problems in the subspace model to support the Conjecture 1 further. The results show that the (empirical) optimal robust risk, obtained via adversarial training, is the same as the robust risk of the conjectured optimal estimator. Moreover, the results show that jittering can yield suboptimal robust estimators in some cases.

We consider linear inverse problems $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$ with the signal $\mathbf{x}$ lying in a subspace $\mathbf{x} = \mathbf{U}\mathbf{c}$ with $\mathbf{c} \sim \mathcal{N}(0, \sigma_c^2/d\mathbf{I})$ and Gaussian noise $\mathbf{z} \sim \mathcal{N}(0, \sigma_z^2/m\mathbf{I})$. For the simulations, we choose $\sigma_c/\sqrt{d} = 1$ and $\sigma_z/\sqrt{n} = 0.1$ with dimensions $d = 50$ and $n = 100$. Moreover, we consider two diagonal forward operators $\mathbf{A} = \text{diag}(\lambda_i)$: an operator with linear decaying singular values ($\lambda_i = i/n$) and one with geometrically decaying singular values $\lambda_i = 0.7^i$.

Optimal worst-case estimator. We first estimate the optimal robust risks by performing adversarial training and compare it with the robust risk of the conjectured optimal estimator. Adversarial training is performed as described in Section 4, where we generate data $\{(\mathbf{y}_i, \mathbf{x}_i)\}$ using the respective forward model in the subspace. The estimator $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ can be viewed as a neural network with one layer and without bias. Figure 5 shows that the robust risk of the conjectured estimator is essentially the same as the (empirical) optimal robust risk.

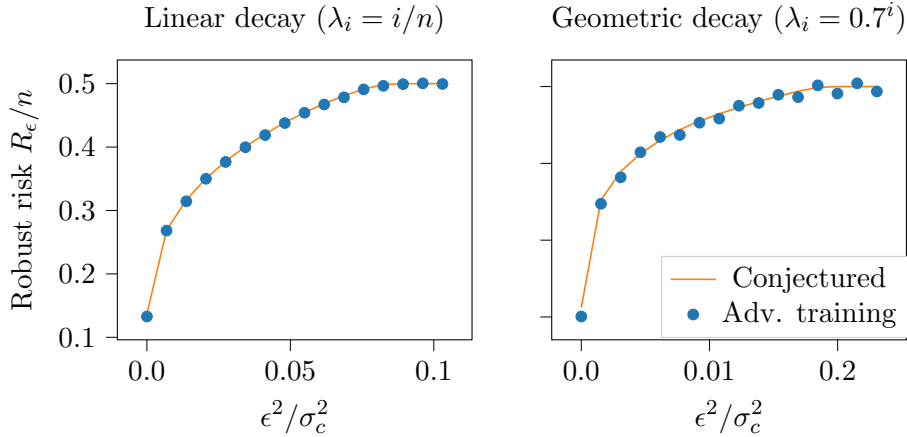


Figure 5: Robust risks of adversarial training (dots) and the conjectured optimal estimator (lines) for two forward operators within the subspace model. It can be seen that the conjectured estimators yield essentially the same robust risks as those of adversarial training.

Suboptimality of jittering. We further investigate whether optimal robust estimators can be obtained via jittering. To that end, we calculate the minimal robust risk attainable via jittering

$$\min_{\sigma_w} R_\epsilon(\text{argmin}_f J_{\sigma_w}(f)),$$

where $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ is a linear reconstruction operator as before. We make use of the analytic expression of the jittering estimator Eq. (22) and calculate the attainable robust risk by minimizing

the robust risk of jittering with respect to the jittering noise level. The results are compared to the (conjectured) optimal robust risks via Eq. (19) and the robust risk of the standard estimator. Figure 6 shows the results of the calculations for the forward operator with linear and geometrically decaying singular values described above. It can be seen that the standard estimator is noticeably less robust compared to denoising setups. Moreover, a gap can be observed between the robust risk obtained via jittering and the optimal robust risks.

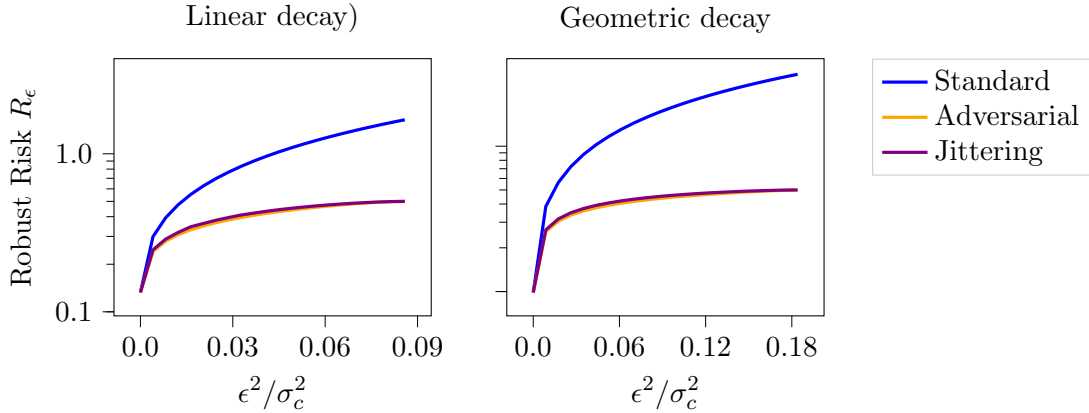


Figure 6: Pixel-wise robust risk of the optimal worst-case estimator, the jittering estimator with optimal jittering noise level, and the standard estimators. A small gap can be observed, showing that (isotropic) jittering does not yield the optimal linear worst-case estimator in general. Moreover, the standard estimator is more sensitive to adversarial perturbations as in the denoising setup.

D Details on the experimental results and further experimental results

In this section, we present details on the experimental results in Section 4, and present further experimental results on U-nets trained with robustness-enhancing methods for image denoising, deconvolution, and compressive sensing. The setup and methods considered are as described in the main body.

D.1 Optimal robust denoiser for large perturbation levels

In the main body, we presented empirical results for perturbation levels in the range $\epsilon^2/\mathbb{E}[\|\mathbf{x}\|_2^2] \in [0, 0.3]$, since this is the practically most relevant regime. Figure 7 shows the risk of linear estimators and U-nets trained adversarially for perturbation levels in the range $\epsilon^2/\mathbb{E}[\|\mathbf{x}\|_2^2] \in [0, 1.5]$. From those plots, we see that the transition at $\epsilon^2/\mathbb{E}[\|\mathbf{x}\|_2^2] = 1$ predicted by Theorem 1 for the estimator to map to zero occurs for the subspace model (as predicted by the theory) as well as for the U-net for image denoising.

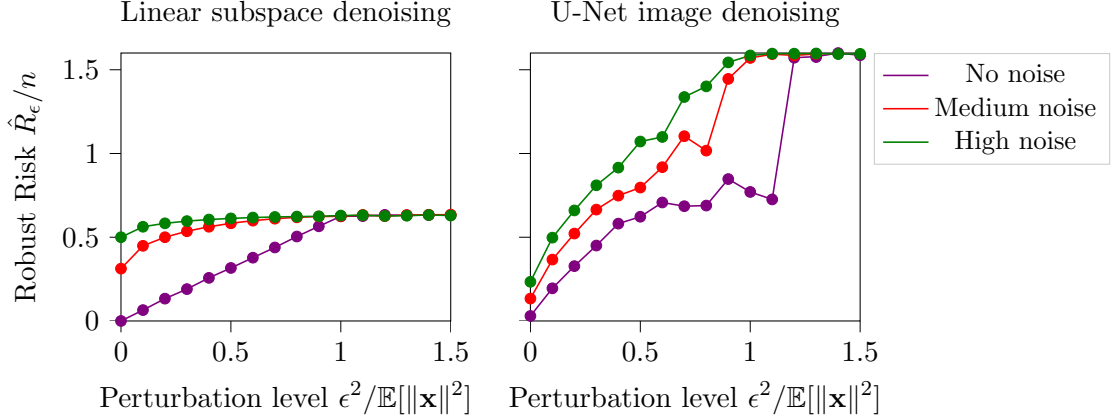


Figure 7: The empirical robust risk for models trained with adversarial training for noise levels $\sigma_z = 0$ (no noise), $\sigma_z/\sqrt{n} = 0.5$ (medium noise) and $\sigma_z/\sqrt{n} = 1.5$ (high noise). The transition predicted by Theorem 1 at $\epsilon^2 \approx \mathbb{E}[\|\mathbf{x}\|_2^2]$ for the estimator to map to zero occurs for the subspace and image denoising settings.

D.2 Convolution kernel for deconvolution experiments

In addition to experiments on denoising image data, we consider a deconvolution setup $\mathbf{y} = \mathbf{k} \star \mathbf{x} + \mathbf{z}$ in this work. The kernel $\mathbf{k}$ is Gaussian, applied channel-wise and visualized in Figure 8.

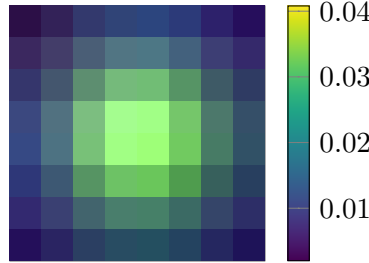


Figure 8: **Gaussian kernel for the deconvolution experiment.** The 8×8 -pixel kernel is calculated as discretization of the $2d$ Gaussian density with standard deviation 2.

D.3 Hyperparameter selection

In the experiments we treat the Jittering noise level σ_w as a hyperparameter, which we optimize over a validation dataset to obtain robust estimators at the desired perturbation levels. The hyperparameter search is performed by choosing a grid of jittering noise levels for each task. For each noise level neural networks (U-Nets) are trained via Jittering, and subsequently evaluated on the considered perturbation levels. Figures 9, 10 and 11 show the robust risks of jittering for the considered tasks, as well as the derived jittering choice rule. The smooth curves on the left panels represent the robust risk at a particular robustness level and are obtained by applying uniform

filters on the evaluation results. It can be seen that for image denoising the empirical jittering choice is close to the prediction from theory.

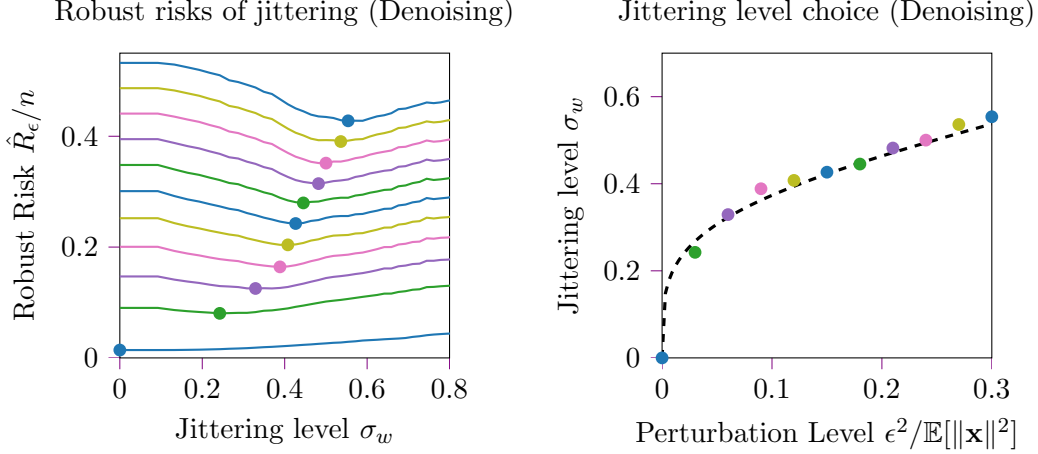


Figure 9: **Jittering hyperparameter results for image denoising.** Neural networks (U-nets) are trained via jittering on a grid of jittering noise levels σ_w (with fixed noise level $\sigma_z/\sqrt{n} = 0.25$). The models are subsequently evaluated on the robust risk R_ϵ , which is calculated for each of the perturbation levels ϵ . Each line in the left panel corresponds to the robust risk at one perturbation level. The derived choice rule is displayed in the right panel. It can be seen that for denoising the empirical jittering choice matches the prediction (dashed line) from theory well (Corollary 1, with $n = 128 \cdot 128 \cdot 3 = 49152$ and subspace dimension $d = 32000$).

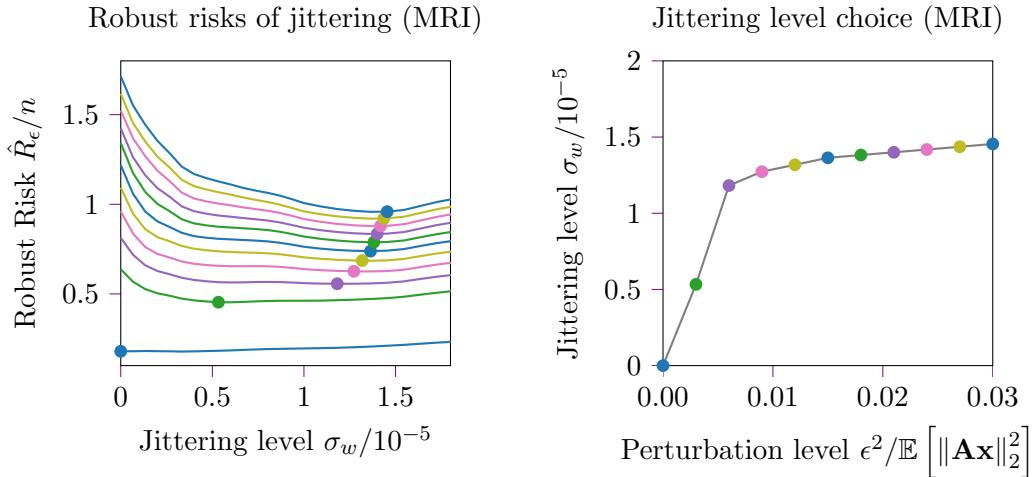


Figure 10: **Jittering hyperparameter search for compressive sensing.** Neural networks (U-nets) are trained via jittering on a grid of noise levels σ_w . The models are subsequently evaluated on the robust risk R_ϵ , which is calculated for each of the perturbation levels ϵ . Each line in the left panel corresponds to the robust risk at one perturbation level. The derived choice rule is displayed in the right panel.

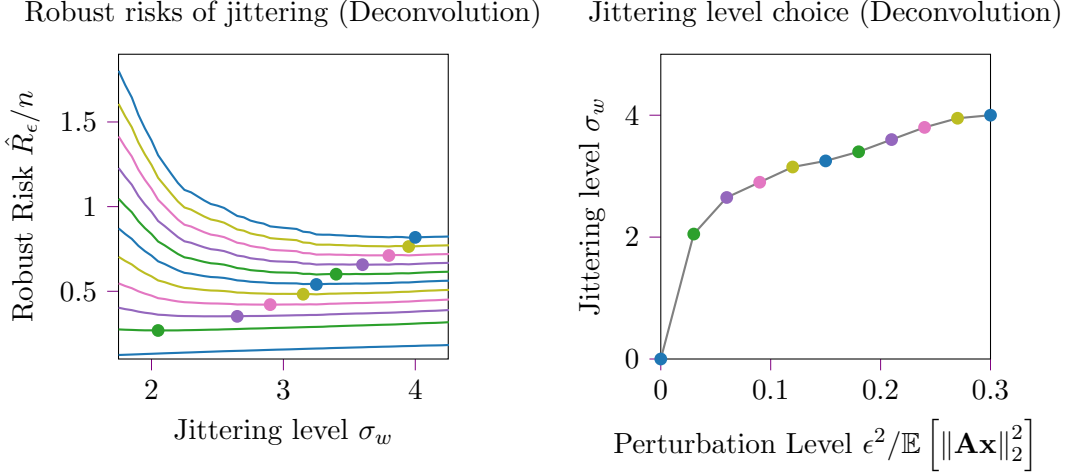


Figure 11: **Jittering hyperparameter results for image deconvolution.** Neural networks (U-nets) are trained via jittering on a grid of jittering noise levels σ_w (with fixed noise level $\sigma_z/\sqrt{n} = 0.25$). The models are subsequently evaluated on the robust risk R_ϵ , which is calculated for each of the perturbation levels ϵ . Each line in the left panel corresponds to the robust risk at one perturbation level. The derived choice rule is displayed in the right panel.

Table 1: GPU time until convergence and memory utilization of adversarial training, jittering and standard training for the task of denoising colorized images. The required number of epochs is estimated from the data visualized in Figure 12. GPU time and memory consumption are measured on a Nvidia RTX A6000 GPU.

Method	Total GPU hours	Memory
Adversarial training	9.7 h	9971 MiB
Training via jittering	3.0 h	9523 MiB
Standard training	3.0 h	9523 MiB

D.4 Computational complexity

We measured the GPU time until convergence and memory utilization of the robustness-enhancing schemes on the task of Gaussian denoising of colorized images. Figure 12 shows the training error of adversarial training, training via jittering and standard training as a function of the number of epochs. It shows networks trained at two perturbation levels for adversarial training and jittering (parameter choice taken from Figure 9). We find that all methods require a similar number of epochs for convergence (roughly 600 epochs). Table 1 presents the measured GPU time until convergence and average memory consumption. It can be seen that adversarial training is by a factor of the projected gradient ascent steps (3 in this plot) more expensive than jittering. Moreover, training via jittering has similar computational cost as standard training in terms of GPU time. All three methods require a similar amount of GPU memory to train.

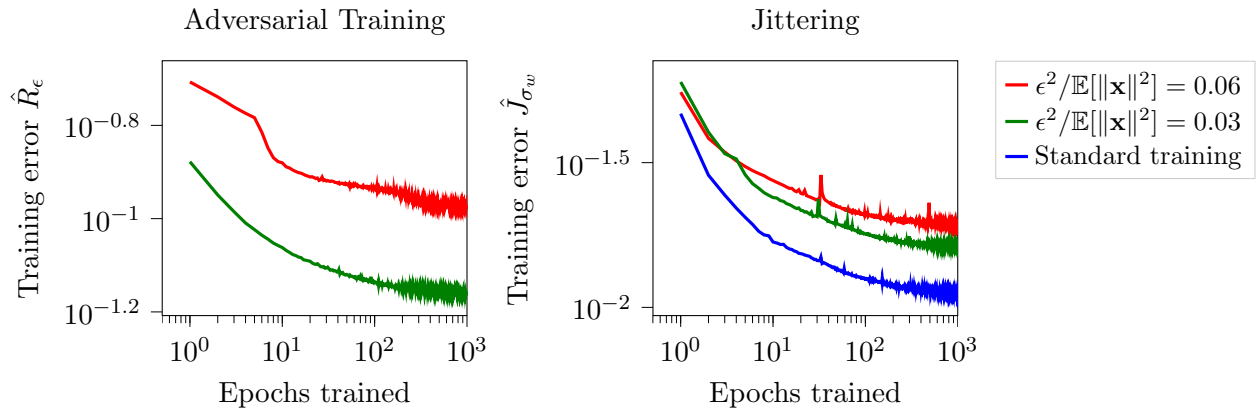


Figure 12: Training metrics of U-nets for adversarial training, jittering and standard training. The plot shows that the convergence rates are very similar for the considered setup. The curves show the training of networks for perturbation levels $\epsilon^2/\mathbb{E}[\|\mathbf{x}\|^2] = 0.03$ (red) and 0.06 (green). Training the methods takes roughly 600 epochs for convergence.

E Discussion of the related work on randomized smoothing

Randomized smoothing is a very successful technique for obtaining robust classifiers [CRK19; Car+23]. Randomized smoothing constructs a smoothed classifier based on a base classifier by averaging the base classifier’s outputs under Gaussian noise perturbation. The smoothed classifier allows for certified radii in which it is provably robust, without making any restrictions on the base classifier. However, Salman et al. [Sal+20] demonstrated that it can give loose bounds, since the base classifier is not trained to be robust to Gaussian noise. For that reason Salman et al. [Sal+20] propose denoised smoothing, which considers a composition of the base classifier with a denoising method. At first sight, randomized smoothing might sound similar to the Jittering approach investigated here. However, as we argue below, randomized smoothing it is conceptually very different from Jittering.

Given a classifier $f: \mathbb{R}^d \rightarrow \{1, \dots, K\}$ randomized smoothing constructs a smoothed classifier g from the classifier f as:

$$g(\mathbf{x}) = \arg \min_{k \in \{1, \dots, K\}} \mathbb{E}_{\mathbf{e} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})} [\mathbb{1}\{f(\mathbf{x} + \mathbf{e}) \neq k\}],$$

where the parameter σ^2 controls the robustness-accuracy tradeoff.

For an inverse problem, where we aim to reconstruct a signal $\mathbf{x} \in \mathbb{R}^n$ from a measurement $\mathbf{y} \in \mathbb{R}^m$ using a given reconstruction method f , replacing the 0/1 by the ℓ_2 loss yields:

$$\begin{aligned} g(\mathbf{y}) &= \arg \min_{\mathbf{x}} \mathbb{E}_{\mathbf{e} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})} [\|\mathbf{f}(\mathbf{y} + \mathbf{e}) - \mathbf{x}\|_2^2] \\ &= \mathbb{E}_{\mathbf{e} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})} [\mathbf{f}(\mathbf{y} + \mathbf{e})]. \end{aligned}$$

For a linear estimator $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ we see that $g(\mathbf{y}) = f(\mathbf{y})$, so for the linear setting considered in the theory part of this paper randomized smoothing would not change the original estimator. If one considers $f: \mathbb{R}^n \rightarrow [0, 1]^n$ the smoothed estimator $g(\mathbf{y})$ differs from $f(\mathbf{y})$ and robustness gains can be expected, which follows from Salman et al. [Sal+19], Lemma 1. In summary, randomized smoothing is very different to jittering in that it constructs a surrogate smoothed model $g(\mathbf{y})$ based on a given fixed estimator $f(\mathbf{y})$, whereas jittering is a training technique.

F Regularizing beyond jittering for enhancing robustness

Training neural networks via jittering, with noise levels chosen for larger perturbations, yields smoother reconstructions compared to adversarial training (see results in Section 4). In this section, we investigate two related regularization methods, ℓ_2 -regularization and Jacobian regularization, discuss the connection to regularization with jittering, and present experimental results for denoising grayscale images.

F.1 ℓ_2 - and Jacobian regularization in the subspace model

In the subspace model we established that the optimal jittering estimator is also worst-case optimal, when using a suitable choice of noise level $\sigma_w(\epsilon)$. It turns out, that jittering can further be approximated with an explicit regularizer, Jacobian regularization. For the linear setup considered here, this approximation becomes exact and therefore Jacobian regularization also enables training

a worst-case robust estimator. Specifically, using the linear approximation of the function f around the point $\mathbf{y}$, we get

$$\mathbb{E}_{\mathbf{w}} \left[\|f(\mathbf{y} + \mathbf{w}) - \mathbf{x}\|_2^2 \right] \approx \mathbb{E}_{\mathbf{w}} \left[\|f(\mathbf{y}) + \mathbf{J}_{\mathbf{y}}\mathbf{w} - \mathbf{x}\|_2^2 \right] = \|f(\mathbf{y}) - \mathbf{x}\|_2^2 + \sigma_w^2 \|\mathbf{J}_{\mathbf{y}}\|_F^2. \quad (23)$$

Here, $\mathbf{J}_{\mathbf{y}}$ is the Jacobian of the function f at $\mathbf{y}$. The approximation is good for small values of the noise variance σ_w^2 , and is exact for the linear estimator $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ we consider in this section. The approximate relation (23) motivates the Jacobian regularized risk, defined as

$$\text{Jac}_{\lambda}(f) = \mathbb{E}_{(\mathbf{x}, \mathbf{y})} \left[\|f(\mathbf{y}) - \mathbf{x}\|_2^2 + \lambda \|\mathbf{J}_{\mathbf{y}}\|_F^2 \right] \quad (24)$$

The connection between jittering and Jacobian regularization is well known in the literature and discussed by Reed et al. [RMO95]. Recall that for the linear estimator considered in this section the approximation in equation (23) is exact, and therefore Jacobian regularization is equivalent to jittering. Thus, Jacobian regularization yields a provably robust estimator, if the regularization parameter is chosen as $\lambda = \sigma_w^2(\epsilon)$ according to corollary 1.

For the linear case, Jacobian regularization is even equivalent to ℓ_2 -regularization, since the Jacobian of the function $f(\mathbf{y}) = \mathbf{H}\mathbf{y}$ is $\mathbf{J}_{\mathbf{y}} = \mathbf{H}$, and thus even ℓ_2 -regularization yields a robust estimator.

F.2 Experimental results on grayscale image denoising

In the following we present results on Gaussian denoising of grayscale images. While ℓ_2 regularization is equivalent to jittering in the subspace model, we find that the parameter choice $\lambda = \sigma_w^2(\epsilon)$ does not yield robust neural networks using ℓ_2 regularization. In contrast, Jacobian regularization turns out to be quite effective for learning neural network denoisers, but is computationally demanding compared to jittering.

F.2.1 Problem setup

We consider once again the dataset of natural image and convert the images to grayscale. We perform Gaussian denoising, i.e. the problem is to reconstruct the image $\mathbf{x}$ from a measurement $\mathbf{y} = \mathbf{x} + \mathbf{z}$, with $\mathbf{z} \sim \mathcal{N}(0, \sigma_z^2/n\mathbf{I})$ and $\sigma_z/\sqrt{n} = 0.2$.

The estimators are chosen as neural networks (U-nets) with the same architecture as for colorized images. We consider adversarial training, jittering and standard training as baseline and compare against ℓ_2 and Jacobian regularization:

ℓ_2 regularization. Implemented as weight-decay in PyTorch’s SGD optimizer to minimize

$$\hat{W}_{\lambda}(\boldsymbol{\theta}) = \sum_{i=1}^N \|f_{\boldsymbol{\theta}}(\mathbf{y}_i) - \mathbf{x}_i\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2.$$

Jacobian regularization. We train networks with the Jacobian regularized empirical risk

$$\widehat{\text{Jac}}_{\lambda}(\boldsymbol{\theta}) = \sum_{i=1}^N \|f_{\boldsymbol{\theta}}(\mathbf{y}_i) - \mathbf{x}_i\|_2^2 + \lambda \|\mathbf{J}_{\mathbf{y}_i}\|_F^2,$$

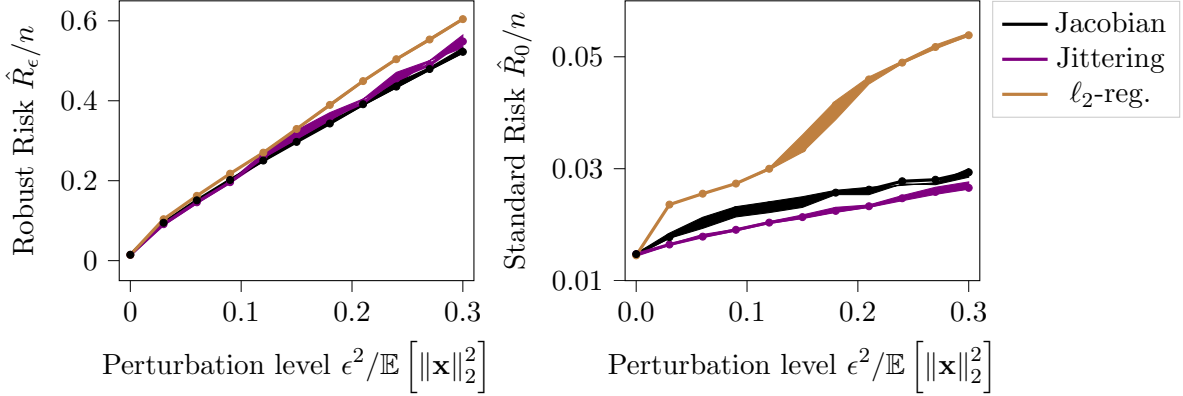


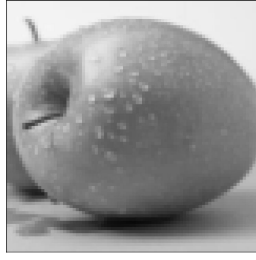
Figure 13: Pixel-wise robust (left) and standard risk (right) of U-nets trained via jittering, ℓ_2 and Jacobian regularization on the task of denoising grayscale images (at noise level $\sigma_z/\sqrt{n} = 0.2$). The plot shows that ℓ_2 regularization yields less robust and accurate estimators compared to jittering. In contrast, Jacobian regularization obtains similarly robust estimators, but with slightly weaker performance in standard risk.

where $\mathbf{J}_{\mathbf{y}_i}$ is the Jacobian of the network f_θ with respect to its input (not its parameters) at $\mathbf{y}_i$. This regularization can be viewed as an approximation of the jittering risk, as described in Section 3.2.1. Calculating the full Jacobian $\mathbf{J}_{\mathbf{y}_i}$ with PyTorch requires n -many calls of the backward function, which is very expensive, since n is large. To mitigate this cost, we approximate the norm of the Jacobian, $\|\mathbf{J}_{\mathbf{y}_i}\|_F^2$ with $\|\mathbf{J}_{\mathbf{y}_i}^T \mathbf{w}\|_2^2$, where $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I})$. This approximation of the norm concentrates around the actual squared norm of the Jacobian, and only costs one call of the PyTorch-backward function.

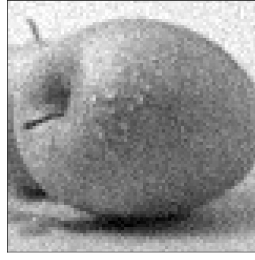
For the experiments we use stochastic gradient descent (SGD) with learning rate 10^{-2} , momentum 0.9 and batch size 100. We evaluate using the empirical pixel-wise robust risk $\hat{R}_\epsilon/n$.

F.2.2 Results

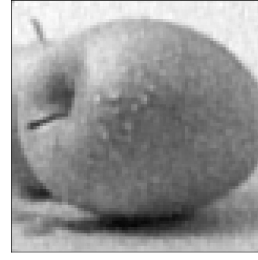
The experimental results, plotted in Figure 13, show that the networks trained with jittering and Jacobian regularization have similar robust risks compared to the adversarial trained one. Weight-decay or ℓ_2 regularization yields worse performing estimators than jittering and Jacobian regularization. While for the linear subspace setting, adversarial training, Jacobian and ℓ_2 regularization are equivalent, for Gaussian denoising they perform differently. Figure 14 shows that Jacobian regularization, unlike Jittering, does not yield smoothed images for larger perturbations. However, Jacobian regularization requires approximately 1.4 as much GPU memory and 4 times more time per epoch.



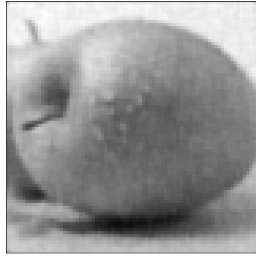
(a) Original



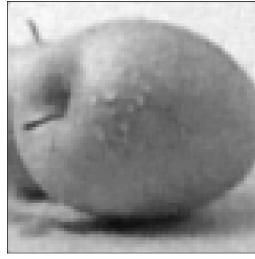
(b) Noisy Image



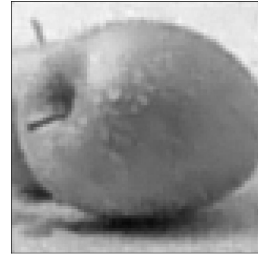
(c) Standard



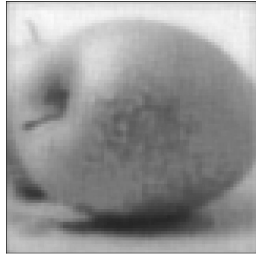
(d) Robust Tr. (*medium* ϵ)



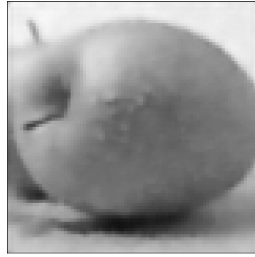
(e) Jittering (*medium* ϵ)



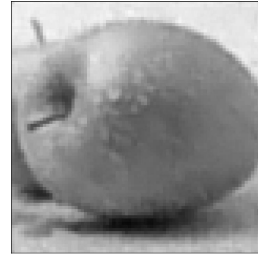
(f) Jacobian (*medium* ϵ)



(g) Robust Tr. (*large* ϵ)



(h) Jittering (*large* ϵ)



(i) Jacobian (*large* ϵ)

Figure 14: Example reconstructions using U-Nets, trained with different robustness-enhancing schemes at noise level $\sigma_z/\sqrt{n} = 0.2$. The second row depicts the results of using neural networks trained with methods tuned on a *medium* perturbation level of $\epsilon^2/\sigma_c^2 = 0.03$, whereas the third row shows results for a *large* perturbation level $\epsilon^2/\sigma_c^2 = 0.3$. The plot shows that jittering yields smooth reconstructions, whereas adversarial training and Jacobian regularization yield less smooth reconstructions.