CONVEX REGULARIZATION BEHIND NEURAL RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks have shown tremendous potential for reconstructing high-resolution images in inverse problems. The non-convex and opaque nature of neural networks, however, hinders their utility in sensitive applications such as medical imaging. To cope with this challenge, this paper advocates a convex duality framework that makes a two-layer fully-convolutional ReLU denoising network amenable to convex optimization. The convex dual network not only offers the optimum training with convex solvers, but also facilitates interpreting training and prediction. In particular, it implies training neural networks with weight decay regularization induces path sparsity while the prediction is piecewise linear filtering. A range of experiments with MNIST and fastMRI datasets confirm the efficacy of the dual network optimization problem.

1 INTRODUCTION

In the age of AI, image reconstruction has witnessed a paradigm shift that impacts several applications ranging from natural image super-resolution to medical imaging. Compared with the traditional iterative algorithms, AI has delivered significant improvements in speed and image quality, making learned reconstruction based on neural networks widely adopted in clinical scanners and personal devices. The non-convex and opaque nature of deep neural networks however raises serious concerns about the authenticity of the predicted pixels in domains as sensitive as medical imaging. It is thus crucial to understand what the trained neural networks represent, and interpret their reconstruction per pixel for unseen images.

Reconstruction is typically cast as an inverse problem, where neural networks are used in different ways to create effective priors; see e.g., (Ongie et al., 2020; Mardani et al., 2018b) and references therein. An important class of methods are *denoising networks*, which given natural data corrupted by some noisy process $\boldsymbol{Y}$, aim to regress the ground-truth, noise-free data $\boldsymbol{X}_*$ (Gondara, 2016; Vincent et al., 2010). These networks are generally learned in a supervised fashion, such that a mapping $f : \mathcal{Y} \to \mathcal{X}$ is learned from inputs $\{\boldsymbol{y}_i\}_{i=1}^n$ to outputs $\{\boldsymbol{x}_{*i}\}_{i=1}^n$, and then can be used in the inference phase on new samples $\hat{\boldsymbol{y}}$ to generate the prediction $\hat{\boldsymbol{x}}_* = f(\hat{\boldsymbol{y}})$.

The scope of denoising networks is so general that it can cover more structured inverse problems appearing, for example, in compressed sensing. In this case one can easily form a poor (linear) estimate of the ground-truth image that is noisy and then reconstruct via end-to-end denoising networks (Mardani et al., 2018b; Mousavi et al., 2015). This method has been proven quite effective on tasks such as medical image reconstruction (Mardani et al., 2018b;a; Sandino et al., 2020; Hammernik et al., 2018), and significantly outperforms sparsity-inducing convex denoising methods, such as total-variation (TV) and wavelet regularization (Candès et al., 2006; Lustig et al., 2008; Donoho, 2006) in terms of both quality and speed.

Despite their encouraging results and growing use in clinical settings (Sandino et al., 2020; Hammernik et al., 2018; Mousavi et al., 2015), little work has explored the interpretation of supervised training of over-parameterized neural networks for inverse problems. Whereas robustness guarantees exist for inverse problems with minimization of convex sparsity-inducing objectives (Oymak & Hassibi, 2016; Chandrasekaran et al., 2012), there exist no such guarantees for predictions of non-convex denoising neural networks based on supervised training. In fact, it has been demonstrated empirically that deep neural networks for image reconstruction can be

unstable; i.e., small perturbations in the input can cause severe artifacts in the reconstruction, which can mask relevant structural features, which are important for medical image interpretation (Antun et al., 2020).

The main challenge in explaining these effects emanate from the non-linear and non-convex structure of deep neural networks that are heuristically optimized via first-order stochastic gradient descent (SGD) based solvers such as Adam (Kingma & Ba, 2014). As a result, it is hard to interpret the inference phase, and the training samples can alter the predictions for unseen images. To cope with these challenges, we present a convex-duality framework for two-layer denoising networks with fully convolutional (conv.) layers with ReLU activation and the representation shared among all output pixels. In essence, inspired by the analysis by Pilanci & Ergen (2020), the zero-duality gap offers a convex bi-dual formulation for the original non-convex objective, that demands only polynomial variable count.

The benefits of the convex dual are three-fold. First, with the convex dual, one can leverage off-the-shelf convex solvers to guarantee convergence to the global optimum in polynomial time and provides robustness guarantees for reconstruction. Second, it provides an interpretation of the training with weight decay regularization as implicit regularization with path-sparsity, a form of capacity control of neural networks (Neyshabur et al., 2015). Third, the convex dual interprets CNN-based denoising as first dividing the input image patches into clusters, based on their latent representation, and then linear filtering is applied for patches in the same cluster. A range of experiments are performed with MNIST and fastMRI reconstruction that confirm the zero-duality gap, interpretability, and practicality of the convex formulation.

All in all, the main contributions of this paper are summarized as follows:

- We, for the first time, formulate a convex program with polynomial complexity for neural image reconstruction, which is provably identical to a two-layer fully-conv. ReLU network.
- We provide novel interpretations of the training objective with weight decay as path-sparsity regularization, and prediction as patch-based clustering and linear filtering.
- We present extensive experiments for MNIST and fastMRI reconstruction that our convex dual coincides with the non-convex neural network, and interpret the learned dual networks.

2 Related work

This paper is at the intersection of two lines of work, namely, convex neural networks, and deep learning for inverse problems. Convex neural networks were introduced in (Bach, 2017; Bengio et al., 2006), and later in (Pilanci & Ergen, 2020; Ergen & Pilanci, 2020a;b). The most relevant to our work are (Pilanci & Ergen, 2020; Ergen & Pilanci, 2020b) which put forth a convex duality framework for two-layer ReLU networks with a single output. It presents a convex alternative in a higher dimensional space for the non-convex and finite-dimensional neural network. It is however restricted to *scalar-output* networks, and considers either fully-connected networks (Pilanci & Ergen, 2020), or, CNNs with average pooling (Ergen & Pilanci, 2020b). Our work however focuses on fully convolutional denoising with an output dimension as large as the number of image pixels, where these pixels share the same hidden representation. This is indeed quite different from the setting considered in (Pilanci & Ergen, 2020) and demands a different treatment.

In recent years, deep learning has been widely deployed in inverse problems to either learn effective priors for iterative algorithms (Bora et al., 2017), or to directly learn the inversion map using feed-forward networks (Jin et al., 2017; Zhang et al., 2017). This work belongs to the second group, which is of utmost interest in real-time applications, and thus widely adopted in medical image reconstruction. Compressed sensing (CS) MRI has been a successful fit, where knowing the forward acquisition model, one forms an initial linear estimate, and trains a non-linear CNNs to de-alias the input (Mardani et al., 2018a). Further, unrolled architectures inspired by convex optimization have been developed for robust de-aliasing (Sun et al., 2016; Mardani et al., 2018b; Hammernik et al., 2018; Sandino et al., 2020; Diamond et al., 2017). Past work however are all based on non-convex training of network filters, and interpretability is not their focus. Note that stability of iterative neural reconstructions has also been recently analyzed in Li et al. (2020); Mukherjee et al. (2020).

3 PRELIMINARIES AND PROBLEM STATEMENT

Consider the problem of denoising, i.e. reconstructing clean signals from ones which have been corrupted by noise. In particular, we are given a dataset of 2D [1] images $\boldsymbol{X}_* \in \mathbb{R}^{N\times h\times w}$, along with their corrupted counterparts $\boldsymbol{Y} = \boldsymbol{X}_* + \boldsymbol{E}$, where noise $\boldsymbol{E}$ has entries drawn from some probability distribution, such as $\mathcal{N}(0, \sigma^2)$ in the case of i.i.d. Gaussian noise. This is a fundamental problem, with a wide range of applications including medical imaging, image restoration, and image encryption problems (Jiang et al., 2018; Dong et al., 2018; Lan et al., 2019).

To solve the denoising problem, we deploy a two-layer CNN, where the first layer has an arbitrary kernel size k and appropriately chosen padding, followed by an element-wise ReLU operation denoted by $(\cdot)_+$. The second and final layer of the network performs a conv. by a 1×1 kernel to generate the predictions of the network. The predictions generated by this neural network with m first-layer conv. filters $\{\boldsymbol{u}_j\}_{j=1}^m$ and second-layer conv. filters $\{\boldsymbol{v}_j\}_{j=1}^m$ can be expressed as

$$f(\boldsymbol{Y}) = \sum_{j=1}^{m} (\boldsymbol{Y} \circledast \boldsymbol{u}_j)_+ \circledast \boldsymbol{v}_j \tag{1}$$

where $\circledast$ represents the 2D conv. operation.

3.1 TRAINING

We then seek to minimize the squared loss of the predictions of the network, along with an ℓ_2-norm penalty (weight decay) on the network weights, to obtain the training problem

$$p^* = \min_{\substack{\boldsymbol{u}_j \in \mathbb{R}^{k\times k} \\ v_j \in \mathbb{R}}} \frac{1}{2} \| \sum_{j=1}^{m} (\boldsymbol{Y} \circledast \boldsymbol{u}_j)_+ \circledast v_j - \boldsymbol{X}_* \|_F^2 + \frac{\beta}{2} \sum_{j=1}^{m} \Big(\|\boldsymbol{u}_j\|_F^2 + |v_j|^2 \Big) \tag{2}$$

The network's output can also be understood in terms of matrix-vector products, when the input is appropriately expressed in terms of patch matrices $\{\boldsymbol{Y}_p \in \mathbb{R}^{k^2}\}_{p=1}^{Nhw}$, where each patch matrix corresponds to a patch of the image upon which a convolutional kernel will operate upon. Then, we can form the two-dimensional matrix input to the network as $\boldsymbol{Y}' = [\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_{Nhw}]^\top \in \mathbb{R}^{Nhw\times k^2}$, and attempt to regress labels $\boldsymbol{X}'_* \in \mathbb{R}^{Nhw}$, which is a flattened vector of the clean images $\boldsymbol{X}_*$. An equivalent form of the two-layer CNN training problem is thus given by

$$p^* = \min_{\substack{\boldsymbol{u}_j \in \mathbb{R}^{k^2} \\ v_j \in \mathbb{R}}} \frac{1}{2} \| \sum_{j=1}^{m} (\boldsymbol{Y}'\boldsymbol{u}_j)_+ v_j - \boldsymbol{X}'_* \|_2^2 + \frac{\beta}{2} \sum_{j=1}^{m} \Big(\|\boldsymbol{u}_j\|_2^2 + |v_j|^2 \Big) \tag{3}$$

In this form, the neural network training problem is equivalent to a 2-layer fully connected scalar-output ReLU network with Nhw samples of dimension k^2, which has previously been theoretically analyzed (Pilanci & Ergen, 2020). We also note that for a fixed kernel-size k, the patch data matrix $\boldsymbol{Y}'$ has a fixed rank, since the rank of $\boldsymbol{Y}'$ cannot exceed the number of columns k^2.

3.2 RELU HYPER-PLANE ARRANGEMENTS

To fully understand the convex formulation of the neural network proposed in (2), we must provide notation for understanding the hyper-plane arrangements of the network. We consider the set of diagonal matrices

$$\mathcal{D} := \{\text{Diag}(\mathbf{1}_{\boldsymbol{Y}'\boldsymbol{u}\geq 0}) : \|\boldsymbol{u}\|_2 \leq 1\}$$

This set, which depends on $\boldsymbol{Y}'$, stores the set of activation patterns corresponding to the ReLU non-linearity, where a value of 1 indicates that the neuron is active, while 0 indicates that the neuron is inactive. In particular, we can enumerate the set of sign patterns as $\mathcal{D} = \{\boldsymbol{D}_i\}_{i=1}^{\ell}$, where ℓ depends on $\boldsymbol{Y}'$ and is bounded by

$$\ell \leq 2r\Big(\frac{e(Nhw-1)}{r}\Big)^r$$

[1] The results contained here are general to all types of conv., but we refer to the 2D case for simplicity.

for $r := \mathbf{rank}(\boldsymbol{Y}')$ (Pilanci & Ergen, 2020). Thus, P is polynomial in Nhw for matrices with a fixed rank r, which occurs for convolutions with a fixed kernel size k. Using these sign patterns, we can completely characterize the range space of the first layer after the ReLU:

$$\{(\boldsymbol{Y}'\boldsymbol{u})_+ : \|\boldsymbol{u}\|_2 \leq 1\} = \{\boldsymbol{D}_i\boldsymbol{Y}'\boldsymbol{u} : \|u\|_2 \leq 1,\ (2\boldsymbol{D}_i - \boldsymbol{I})\boldsymbol{Y}'\boldsymbol{u} \geq 0,\ i \in [\ell]\}$$

With this notation established, we are ready to present our main theoretical result.

4 Convex duality

Theorem 1. *There exists an $m^* \leq Nhw$ such that if the number of conv. filters $m \geq m^* + 1$, the two-layer conv. network with ReLU activation (2) has a strong dual. This dual is a finite-dimensional convex program, given by*

$$d^* = \min_{\substack{(2\boldsymbol{D}_i - \boldsymbol{I})\boldsymbol{Y}'\boldsymbol{w}_i \geq 0 \\ (2\boldsymbol{D}_i - \boldsymbol{I})\boldsymbol{Y}'\boldsymbol{z}_i \geq 0}} \frac{1}{2}\|\sum_{i=1}^{\ell} \boldsymbol{D}_i\boldsymbol{Y}'(\boldsymbol{w}_i - \boldsymbol{z}_i) - \boldsymbol{X}_*'\|_2^2 + \beta \sum_{i=1}^{\ell} \Big(\|\boldsymbol{w}_i\|_2 + \|\boldsymbol{z}_i\|_2\Big) \tag{4}$$

where ℓ refers to the number of sign patterns associated with $\boldsymbol{Y}'$. Furthermore, given a set of optimal dual weights $(\boldsymbol{w}_i^*, \boldsymbol{z}_i^*)_{i=1}^{\ell}$, we can reconstruct the optimal primal weights as follows

$$(\boldsymbol{u}_i^*, \boldsymbol{v}_i^*) = \begin{cases} (\frac{\boldsymbol{w}_i^*}{\sqrt{\|\boldsymbol{w}_i^*\|_2}}, \sqrt{\|\boldsymbol{w}_i^*\|_2}) & \boldsymbol{w}_i^* \neq 0 \\ (\frac{\boldsymbol{z}_i^*}{\sqrt{\|\boldsymbol{z}_i^*\|_2}}, \sqrt{\|\boldsymbol{z}_i^*\|_2}) & \boldsymbol{z}_i^* \neq 0 \end{cases} \tag{5}$$

It is useful to recognize that the convex program has $2\ell Nhw$ constraints and $2\ell k^2$ variables, which can be solved in polynomial time with respect to N, h and w using standard convex optimizers. Further, this result can easily be extended to a residual architecture as stated next.

Corollary 1.1. *Consider a residual two-layer network given by*

$$f_{res}(\boldsymbol{Y}) = \boldsymbol{Y} + \sum_{j=1}^{m} (\boldsymbol{Y} \circledast \boldsymbol{u}_j)_+ \circledast \boldsymbol{v}_j \tag{6}$$

We can also pose the convex dual network (4) in a similar fashion, where now we simply regress upon the residual labels $\boldsymbol{X}_* - \boldsymbol{Y}$.

4.1 Implicit regularization

In this section, we discuss the implicit regularization induced by the weight decay in the primal model (3). In particular, each dual variable $\boldsymbol{w}_i$ or $\boldsymbol{z}_i$ represents a *path* from the input to the output, since the the product of corresponding primal weights is given by

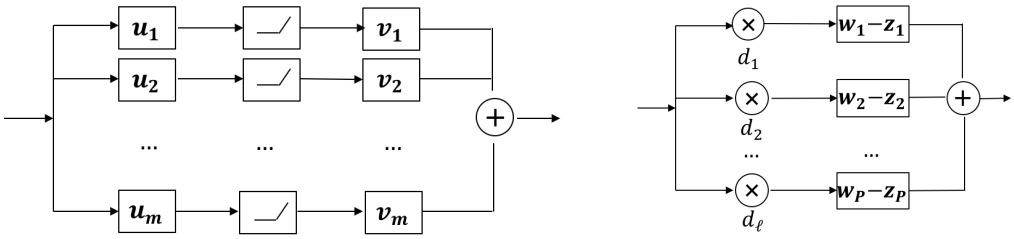
$$\boldsymbol{u}_i^*\boldsymbol{v}_i^* = \begin{cases} \boldsymbol{w}_i^* & \boldsymbol{w}_i^* \neq 0 \\ \boldsymbol{z}_i^* & \boldsymbol{z}_i^* \neq 0 \end{cases} \tag{7}$$

Thus, the sparsity-inducing group-lasso penalty on the dual weights $\boldsymbol{w}_i$ and $\boldsymbol{z}_i$ induces sparsity in the paths of the primal model. In particular, a penalty is ascribed to $\|\boldsymbol{w}_i\|_2 + \|\boldsymbol{z}_i\|_2$, which in terms of primal weights corresponds to a penalty on $|v_i|\|\boldsymbol{u}_i\|_2$. This sort of penalty has been explored previously in (Neyshabur et al., 2015), and refers to the path-based regularizer from their work.

4.2 Interpretable reconstruction

The convex dual model (4) allows us to understand how an output pixel is predicted from a particular patch. Note that in this formulation, each input patch is regressed upon the center pixel of the output. In particular, for an input patch $\boldsymbol{y}_p'$, the prediction of the network corresponding to the p-th output pixel is given by

$$f(\boldsymbol{y}_p') = \sum_{i=1}^{\ell} d_i^{(p)} \boldsymbol{y}_p'^{\top}(\boldsymbol{w}_i - \boldsymbol{z}_i) \tag{8}$$



(a) Primal Network (b) Dual Network

Figure 1: Primal and dual network interpretation. In the primal network, m refers to the number of conv. filters, while in the dual network ℓ refers to the number of sign patterns.

where $d_i^{(p)} \in \{0, 1\}$ refers to the pth diagonal element of D_i. Thus, for an individual patch y'_p, the network can be interpreted as first selecting relevant sets of linear filters for that individual patch, and then taking a sum of the inner product of the patch with those filters–a piece-wise linear filtering operation. Thus, once it is identified which filters are active for a particular patch, the network's predictions are given as linear. This interpretation of the dual network contrasts with the opaque understanding of the primal network, in which due to the non-linear ReLU operation it is unclear how to interpret its predictions, as shown in Fig. 1.

Furthermore, because of the group-lasso penalty (Yuan & Lin, 2006) on $\boldsymbol{w}_i$ and $\boldsymbol{z}_i$ in the dual objective, these weights are sparse. Thus, for particular patch $\boldsymbol{y}'_p$, only a few sign patterns $d_i^{(p)}$ influence its prediction. Therefore, different patches are implicitly clustered by the network according to the linear weights $\boldsymbol{w}_i - \boldsymbol{z}_i$ which are active for their predictions. A forward pass of the network can thus be considered as first a clustering operation, followed by a linear filtering operation for each individual cluster. As the neural network becomes deeper, we expect that this clustering becomes hierarchical–at each layer, the clusters become more complex, and capture more contextual information from surrounding patches.

4.3 Deep Networks

While the result of Theorem 1 holds only for two-layer fully conv. networks, these networks are essential for interpreting the implicit regularization and reconstruction of deeper neural networks. For one, these two-layer networks can be greedily trained to build a successively richer representation of the input. This allows for increased field of view for the piecewise linear filters to operate upon, along with allowing for more complex clustering of input patches. This approach is not dissimilar to the end-to-end denoising networks described by Mardani et al. (2018b), though it is more interpretable due to the simplicity of the convex dual of each successive trained layer.

This layer-wise training has been found to be successful in a variety of contexts. Greedily pre-training denoising autoencoders layer-wise has been shown to improve classification performance in deep networks (Vincent et al., 2010). Greedy layer-wise supervised training has also been shown to perform competitively with much deeper end-to-end trained networks on image classification tasks (Belilovsky et al., 2019). Although analyzing the behavior of end-to-end trained deep networks is outside the scope of this work, we expect that end-to-end models are similar to networks trained greedily layerwise, which can fully be interpreted with our convex dual model.

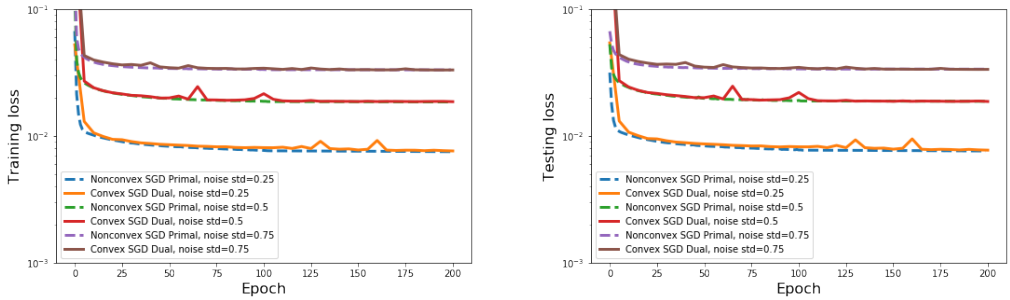
5 Experiments

5.1 MNIST Denoising

Dataset. We use a subset of the MNIST handwritten digits (LeCun et al., 1998). In particular, for training, we select 600 gray-scale 28×28 images, pixel-wise normalized by the mean and standard deviation over the entire dataset. The full test dataset of 10,000 images is used for evaluation.

Training. We seek to solve the task of denoising the images from the MNIST dataset. In particular, we add i.i.d. noise from the distribution $\mathcal{N}(0, \sigma^2)$ for various noise levels, $\sigma \in \{0.25, 0.5, 0.75\}$. The resulting noisy images, $\boldsymbol{Y}$, are the inputs to our network, and we attempt to learn the clean images $\boldsymbol{X}_*$. We train both the primal network (2) and the dual network (4) using Adam (Kingma & Ba, 2014). For the primal network, we use 512 filters, whereas for the dual network, we randomly sample 8,000 sign patterns $\boldsymbol{D}_i$ as an approximation to the full set of sign patterns ℓ. Further experimental details can be found in the appendix.

Zero duality gap. We find that for this denoising problem, there is no gap between the primal and dual objective values across all values of σ tested, verifying the theoretical result of Theorem 1, as demonstrated in Fig. 2. This is irrespective of the sign-pattern approximation, wherein we select only 8,000 sign patterns for the dual network, rather than enumerating the entire set of ℓ patterns. The illustrations of reconstructed images in Fig.3 also makes it clear that the primal and dual reconstructions are of similar quality.



(a) Train loss (b) Test loss

Figure 2: Train and test curves for MNIST denoising problem, for various noise levels σ.

(a) $\sigma = 0.25$ (b) $\sigma = 0.75$

Figure 3: Test examples from MNIST denoising problem for two values of σ from primal (top) and dual (bottom) networks. From left to right, images are: (a) noisy network input, (b) ground truth, (c) network output.

Interpretable reconstruction. Further, we can interpret what these networks have learned using our knowledge of the dual network. In particular, we can visualize both the sparsity of the learned filters, and the network's clustering of input patches. Because of the piece-wise linear nature of the dual network, we can visualize the dual filters $\boldsymbol{w}_i$ or $\boldsymbol{z}_i$ as linear filters for the selected sign patterns. Thus, the frequency response of these filters explains the filtering behavior of the end-to-end network, where depending on the input patch, different filters are activated. We visualize this frequency response of the dual weights $\boldsymbol{w}_i$ in Fig. 4, where we randomly select 600 representative filters of size 28×28. We note that because of the path sparsity induced by the group-Lasso penalty on the dual weights, some of these dual filters are essentially null filters.

The clustering of input patches can be detected via the set of sign patterns $d_i^{(p)}$ which correspond to non-zero filters for each output pixel p. Each output pixel p can thus be represented by a binary vector $\boldsymbol{d}^{(p)} \in \{0, 1\}^\ell$. We thus feed the trained network clean test images and interpret how they are clustered, using k-means clustering with $k = 12$ to interpret the similarity among these binary

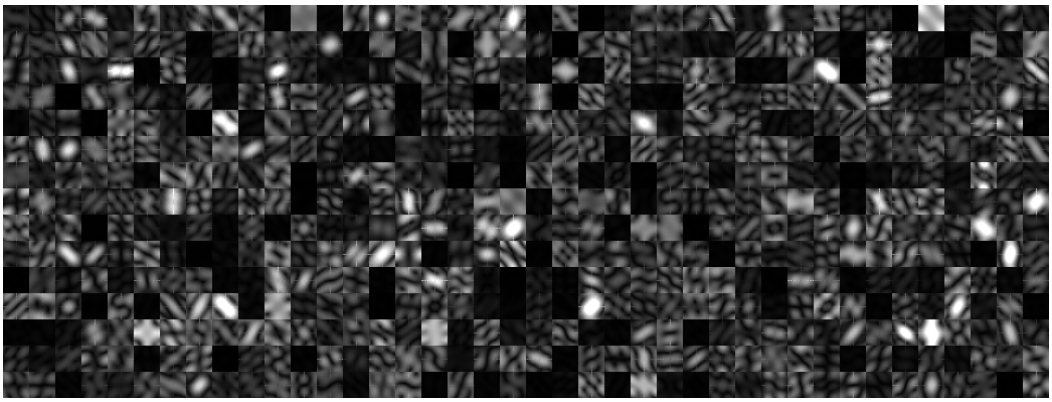
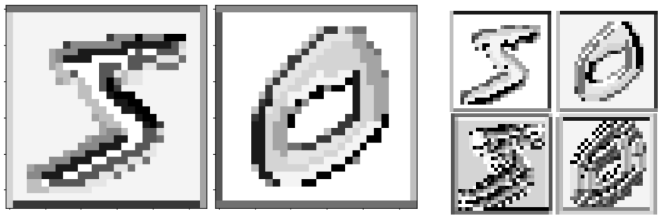


Figure 4: Visualization of the frequency response for the learned dual filters $\{\boldsymbol{w}_i\}$ for denoising MNIST. Representative filters (600) are randomly selected for visualization when $\sigma = 0.5$.

vectors for each output pixel of an image. Visualizations of these clusters can be found in Fig. 5(a).

We can also use this clustering interpretation for deeper networks, even those trained end-to-end. We consider a four-layer architecture, which consists of two unrolled iterations of the two-layer architecture from the previous experiment, trained end-to-end. We can perform the same k-means clustering on the implicit representation obtained from each unrolled iteration, using the interpretation from the dual network. This result is demonstrated in Fig. 5(b), where we find that the clustering is more complex in the second iteration than the first, as expected. We note that while this network was trained end-to-end, the clusters from the first iteration are nearly identical to those of the single unrolled iteration, indicating that the early layers of end-to-end trained deeper denoising networks learn similar clusters to those of two-layer denoising networks.



(a) One unrolled iteration

(b) Two unrolled iterations (trained end-to-end)

Figure 5: Visualization of k-means clustering of latent representations from trained networks for denoising MNIST when $\sigma = 0.75$, with $k = 12$. Top row is the output of the first unrolled iteration, bottom is the output of the second unrolled iteration.

5.2 MRI Reconstruction

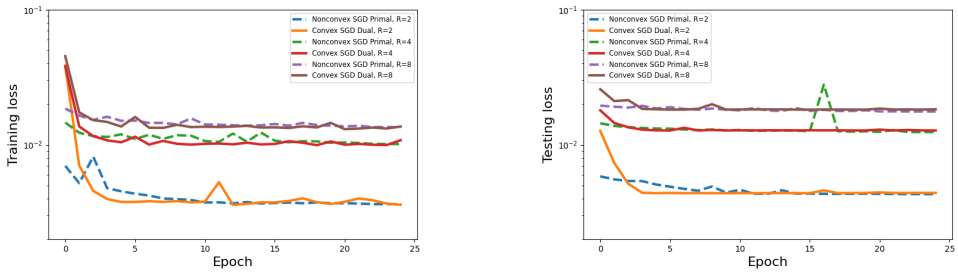
MRI acquisition. In multi-coil MRI, the forward problem for each patient admits $\boldsymbol{y}_i = \Omega \boldsymbol{F} \boldsymbol{S}_i \boldsymbol{x} + \boldsymbol{e}_i, i = 1, \cdots, n_c$ where $\boldsymbol{F}$ is the 2D discrete Fourier transform, $\{\boldsymbol{S}_i\}_{i=1}^{n_c}$ are the sensitivity maps of the receiver coils, and Ω is the undersampling mask that indexes the sampled Fourier coefficients.

Dataset. To assess the effectiveness of our method, we use the fastMRI dataset (Zbontar et al., 2018), a benchmark dataset for evaluating deep-learning based MRI reconstruction methods. We use a subset of the multi-coil knee measurements of the fastMRI training set that consists of 49 patients (1,741 slices) for training, and 10 patients (370 slices) for testing, where each slice is of size 80×80. We select Ω by generating Poisson-disc sampling masks using undersampling factors

$R = 2, 4, 8$ with a calibration region of 16×16 using the SigPy python package (Ong & Lustig, 2019). Sensitivity maps $\boldsymbol{S}_i$ are estimated using JSENSE (Ying & Sheng, 2007a).

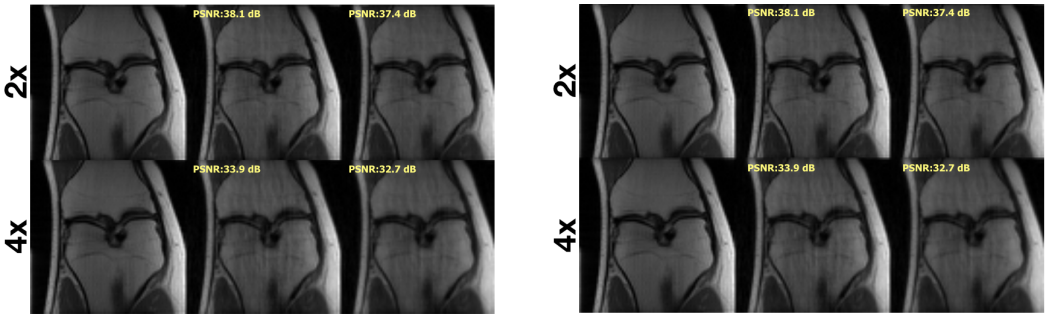
Training. The multi-coil complex data are first undersampled, then reduced to a single-coil complex image using the SENSE model (Ying & Sheng, 2007b). The input of the networks are the real and imaginary components of this complex-valued Zero-Filled (ZF) image, where we wish to recover the fully-sampled ground-truth image. The real and imaginary components of each image are treated as separate examples during training. For the primal network, we use 1,024 filters, whereas for the dual network, we randomly sample 5,000 sign patterns.

Zero duality gap. We observe zero duality gap for CS-MRI, verifying Theorem 1. For different R, both the train and test loss of the primal and dual networks converge to the same optimal value, as depicted in Fig. 6. Furthermore, we show a representative axial slice from a random test patient in Fig. 7 reconstructed by the dual and primal networks, both achieving the same PSNR.



(a) Train loss (b) Test loss

Figure 6: Train and test curves for MRI reconstruction under various undersampling rates R.



(a) Primal Network (b) Dual Network

Figure 7: Representative test knee MRI slice reconstructed via dual and primal network for undersampling $R = 2, 4$. From left to right: ground truth, output, and noisy ZF input.

6 Conclusions

This paper puts forth a convex duality framework for CNN-based denoising networks. Focusing on a two-layer CNN network with ReLU activation, a convex dual program is formulated that offers optimal training using convex solvers, and gains more interpretability. It reveals that the weight decay regularization of CNNs induces path sparsity regularization for training, while the prediction is piece-wise linear filtering. The utility of the convex formulation for deeper networks is also discussed using greedy unrolling. There are other important next directions to pursue. One such direction pertains to stability analysis of the convex neural network for denoising, and more extensive evaluations with pathological medical images to highlight the crucial role of convexity for robustness.

REFERENCES

Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C Hansen. On instabilities of deep learning in image reconstruction and the potential costs of ai. *Proceedings of the National Academy of Sciences*, 2020.

Francis Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to imagenet. In *International conference on machine learning*, pp. 583–593. PMLR, 2019.

Yoshua Bengio, Nicolas L Roux, Pascal Vincent, Olivier Delalleau, and Patrice Marcotte. Convex neural networks. In *Advances in neural information processing systems*, pp. 123–130, 2006.

Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. *arXiv preprint arXiv:1703.03208*, 2017.

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

Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849, 2012.

Steven Diamond, Vincent Sitzmann, Felix Heide, and Gordon Wetzstein. Unrolled optimization with deep priors. *arXiv preprint arXiv:1705.08041*, 2017.

Weisheng Dong, Peiyao Wang, Wotao Yin, Guangming Shi, Fangfang Wu, and Xiaotong Lu. Denoising prior driven deep neural network for image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2305–2318, 2018.

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

Tolga Ergen and Mert Pilanci. Convex duality of deep neural networks. *arXiv preprint arXiv:2002.09773*, 2020a.

Tolga Ergen and Mert Pilanci. Training convolutional relu neural networks in polynomial time: Exact convex optimization formulations. *arXiv preprint arXiv:2006.14798*, 2020b.

Lovedeep Gondara. Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 241–246. IEEE, 2016.

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

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

Dongsheng Jiang, Weiqiang Dou, Luc Vosters, Xiayu Xu, Yue Sun, and Tao Tan. Denoising of 3d magnetic resonance images with multi-channel residual learning of convolutional neural network. *Japanese journal of radiology*, 36(9):566–574, 2018.

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

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

Rushi Lan, Haizhang Zou, Cheng Pang, Yanru Zhong, Zhenbing Liu, and Xiaonan Luo. Image denoising via deep residual convolutional neural networks. *Signal, Image and Video Processing*, pp. 1–8, 2019.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Housen Li, Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. Nett: Solving inverse problems with deep neural networks. *Inverse Problems*, 2020.

Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing mri. *IEEE signal processing magazine*, 25(2):72–82, 2008.

Morteza Mardani, Enhao Gong, Joseph Y Cheng, Shreyas S Vasanawala, Greg Zaharchuk, Lei Xing, and John M Pauly. Deep generative adversarial neural networks for compressive sensing mri. *IEEE transactions on medical imaging*, 38(1):167–179, 2018a.

Morteza Mardani, Qingyun Sun, Vardan Papyan, Hatef Monajemi, Shreyas Vasanawala, John Pauly, and David Donoho. Neural proximal gradient descent for compressive imaging. In *Advances in Neural Information Processing Systems*, pp. 9573–9583, 2018b.

Ali Mousavi, Ankit B Patel, and Richard G Baraniuk. A deep learning approach to structured signal recovery. In *2015 53rd annual allerton conference on communication, control, and computing (Allerton)*, pp. 1336–1343. IEEE, 2015.

Subhadip Mukherjee, Sören Dittmer, Zakhar Shumaylov, Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. Learned convex regularizers for inverse problems. *arXiv preprint arXiv:2008.02839*, 2020.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pp. 1376–1401, 2015.

F Ong and M Lustig. Sigpy: a python package for high performance iterative reconstruction. In *Proceedings of the ISMRM 27th Annual Meeting, Montreal, Quebec, Canada*, volume 4819, 2019.

Gregory Ongie, Ajil Jalal, Christopher A Metzler Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 2020.

Samet Oymak and Babak Hassibi. Sharp mse bounds for proximal denoising. *Foundations of Computational Mathematics*, 16(4):965–1029, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. *arXiv preprint arXiv:2002.10553*, 2020.

Christopher M Sandino, Joseph Y Cheng, Feiyu Chen, Morteza Mardani, John M Pauly, and Shreyas S Vasanawala. Compressed sensing: From research to clinical practice with deep neural networks: Shortening scan times for magnetic resonance imaging. *IEEE Signal Processing Magazine*, 37(1):117–127, 2020.

Jian Sun, Huibin Li, Zongben Xu, et al. Deep admm-net for compressive sensing mri. In *Advances in neural information processing systems*, pp. 10–18, 2016.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Leslie Ying and Jinhua Sheng. Joint image reconstruction and sensitivity estimation in sense (jsense). *Magnetic Resonance in Medicine*, 57(6):1196–1202, 2007a. doi: 10.1002/mrm.21245. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.21245.

Leslie Ying and Jinhua Sheng. Joint image reconstruction and sensitivity estimation in sense (jsense). *Magnetic Resonance in Medicine*, 57(6):1196–1202, 2007b. doi: 10.1002/mrm.21245. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.21245.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Jure Zbontar, Florian Knoll, Anuroop Sriram, Matthew J. Muckley, Mary Bruno, Aaron Defazio, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, James Pinkerton, Duo Wang, Nafissa Yakubova, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastmri: An open dataset and benchmarks for accelerated MRI. *CoRR*, abs/1811.08839, 2018. URL http://arxiv.org/abs/1811.08839.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26 (7):3142–3155, 2017.

A Appendix

B Additional Experimental Details

All experiments were run using the Pytorch deep learning library (Paszke et al., 2019). All primal networks are initialized using Kaiming uniform initialization (He et al., 2015). The losses presented are the sample and dimension-averaged, meaning that they differ by a constant factor of Nhw from the formulas presented in (2) and (4). All networks were trained with an Adam optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. All networks had ReLU activation and had a final layer consisting of a 1×1 convolution kernel with stride 1 and no padding. For all dual networks, in order to enforce the feasibility constraints of (4), we penalize the constraint violation via hinge loss.

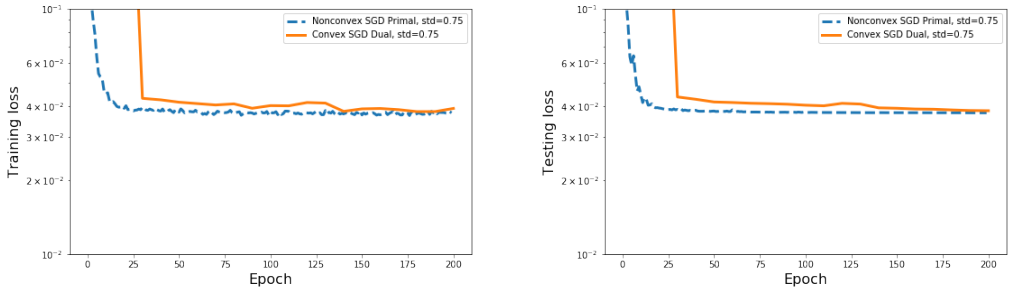
B.1 Additional Experiment: Robustness for MNIST Denoising

While for i.i.d. zero-mean Gaussian noise, training the primal and dual networks coincide, this may not necessarily be the case for all noise distributions. Depending on the distribution of noise, the loss landscape for the non-convex primal optimization problem may have more or fewer valleys–even if SGD does not get stuck in the an exact local minimum, it may converge extremely slowly, mimicking the effect of being stuck in a local minimum. In contrast, because the dual problem is convex, solving the dual problem is guaranteed to converge to the global minimum.

To see this effect, we compare the effect of denoising with i.i.d. zero-mean Gaussian noise, $\sigma = 0.75$, with that of i.i.d. exponentially distributed noise, with $\lambda = \frac{1}{\sqrt{2}}$. The exponential distribution has a higher noise variance, and also a heavier tailed distribution, and thus is more difficult for the non-convex problem to solve. For both cases, we employ 25 primal filters, and 8,000 sign patterns. The results of these experiments are shown in Figs 8 and 9. As we can see, for the Gaussian noise distribution, the primal and dual objective values coincide, but for the exponential noise distribution, the dual program performs better, suggesting the primal problem is stuck in a local minimum or valley. This suggests that when the data distribution is heavy-tailed, the convex dual network may be more robust to train than the non-convex primal network, due to a lack of local minima.

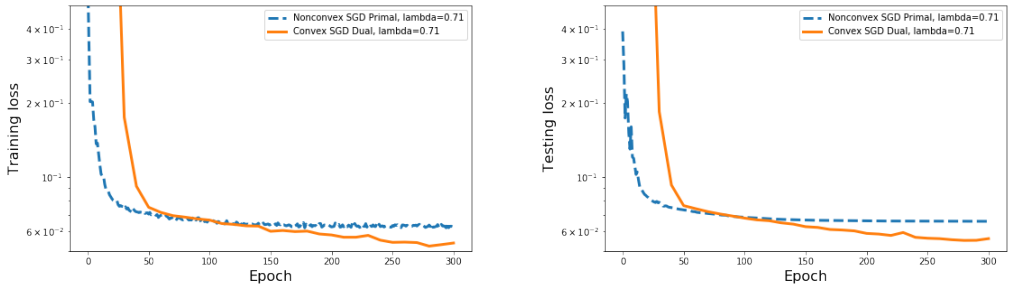
We train both the primal and the dual network in a distributed fashion on a NVIDIA GeForce GTX 1080 Ti GPU and NVIDIA Titan X GPU. For both cases, we use a kernel size of 3×3 with a unity stride and padding for the first layer. For the primal network, we train with a learning rate of $\mu = 10^{-1}$, whereas for the dual network we use a learning rate of $\mu = 10^{-3}$. We use a batch

size of 25 for all cases. For the weight-decay parameter we use a value of $\beta = 10^{-5}$, which is not sample-averaged.



(a) Train loss (b) Test loss

Figure 8: Train and test curves for Gaussian-distributed noise with $\sigma = 0.75$. The primal and dual optimization problems perform similarly well.



(a) Train loss (b) Test loss

Figure 9: Train and test curves for exponentially distributed noise with $\lambda = \frac{1}{\sqrt{2}}$. The primal fails to learn as well as the dual.

B.2 MNIST Denoising

We train both the primal and the dual network in a distributed fashion on a NVIDIA GeForce GTX 1080 Ti GPU and NVIDIA Titan X GPU. For both cases, we use a kernel size of 3×3 with a unity stride and padding for the first layer. For the primal network, we train with a learning rate of $\mu = 10^{-3}$, whereas for the dual network we use a learning rate of $\mu = 10^{-5}$. We use a batch size of 25 for all cases. For the weight-decay parameter we use a value of $\beta = 10^{-5}$, which is not sample-averaged, and which is sufficiently large so as to prevent overfitting on the training set, as demonstrated in Fig. 2, which indicates that test performance is similar to training performance.

B.3 MRI Reconstruction

The fastMRI dataset has data acquired with $n_c = 15$ receiver coils. We train both the primal and the dual network on a single NVIDIA GeForce GTX 1080 Ti GPU with a batch size of 2. As our primal network architecture, we use a two-layer CNN with the first layer kernel size 7×7, unity stride and a padding of 3 with a skip connection. For the training of the primal and dual networks, networks are trained for 25 epochs. A learning rate of $\mu = 10^{-3}$ is used for the primal network, and a learning rate of $\mu = 5 \times 10^{-7}$ is used for the dual network with a weight-decay parameter of $\beta = 10^{-5}$.

C PROOF OF THEOREM 1

We note that, as mentioned in the main paper, (2) is equivalent to (3). Thus, to prove the theorem, it is sufficient to show that (3) is equivalent to (4). This proof follows directly from Theorem 1 from Pilanci & Ergen (2020). An outline of the proof is repeated here for the reader's convenience.

We begin by noting that (3) is equivalent to the following optimization problem:

$$p^* = \min_{\substack{\|\boldsymbol{u}_j\|_2 \leq 1 \\ v_j \in \mathbb{R}}} \|\sum_{j=1}^{m} (\boldsymbol{Y}'\boldsymbol{u}_j)_+ \boldsymbol{v}_j - \boldsymbol{X}_*'\|_2^2 + \beta \sum_{j=1}^{m} |v_j| \tag{9}$$

This is because, without changing the predictions of the network, we can re-scale $\boldsymbol{u}_j$ by some scalar $\gamma_j > 0$, granted that we also scale v_j by its reciprocal $1/\gamma_j$. Then, we can simply minimize the regularization term over γ_j, noting that

$$\min_{\gamma_j} \|\gamma_j \boldsymbol{u}_j\|_2^2 + |v_j/\gamma_j|^2 = \|\boldsymbol{u}_j\|_2 |v_j| \tag{10}$$

Then, we can simply restrict $\|\boldsymbol{u}_j\|_2 \leq 1$ to obtain (9). Then, we can form the equivalent problem

$$p^* = \min_{\|\boldsymbol{u}_j\|_2 \leq 1} \max_{|\boldsymbol{z}^\top (\boldsymbol{Y}'\boldsymbol{u}_j)_+| \leq \beta} -\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{X}_*'\|_2^2 + \frac{1}{2}\|\boldsymbol{X}_*'\|_2^2 \tag{11}$$

As long as $\beta > 0$, this problem is strictly feasible (simply set $\boldsymbol{z} = 0$), so by Slater's condition strong duality holds, so we can form the dual problem

$$p^* = d^* := \max_{|\boldsymbol{z}^\top (\boldsymbol{Y}'\boldsymbol{u}_j)_+| \leq \beta \ \forall \|\boldsymbol{u}\|_2 \leq 1} -\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{X}_*'\|_2^2 + \frac{1}{2}\|\boldsymbol{X}_*'\|_2^2 \tag{12}$$

Now, this dual problem can be finitely parameterized using the sign patterns $\{\boldsymbol{D}_i\}_{i=1}^{\ell}$. We thus have

$$\begin{aligned} d^* := \max_{\boldsymbol{z}} & -\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{X}_*'\|_2^2 + \frac{1}{2}\|\boldsymbol{X}_*'\|_2^2 \\ \text{s.t.} & \max_{\substack{i \in [\ell] \\ \|\boldsymbol{u}\|_2 \leq 1 \\ (2\boldsymbol{D}_i - \boldsymbol{I})\boldsymbol{X}\boldsymbol{u} \geq 0}} |\boldsymbol{z}^\top \boldsymbol{D}_i \boldsymbol{Y}'\boldsymbol{u}| \leq \beta \end{aligned} \tag{13}$$

We can split this absolute value constraint into two constraints, and maximize in closed form over $\boldsymbol{u}$ by introducing Lagrangian variables α_i:

$$\begin{aligned} d^* := \max_{\substack{\boldsymbol{z} \\ \alpha_i \geq 0}} & -\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{X}_*'\|_2^2 + \frac{1}{2}\|\boldsymbol{X}_*'\|_2^2 \\ \text{s.t.} & \|\boldsymbol{Y}'^\top (2\boldsymbol{D}_i - \boldsymbol{I})\alpha_i + \boldsymbol{Y}'^\top \boldsymbol{D}_i \boldsymbol{z}\| \leq \beta \ \forall i \\ & \|\boldsymbol{Y}'^\top (2\boldsymbol{D}_i - \boldsymbol{I})\alpha_i - \boldsymbol{Y}'^\top \boldsymbol{D}_i \boldsymbol{z}\| \leq \beta \ \forall i \end{aligned} \tag{14}$$

We then form the Lagrangian

$$\begin{aligned} d^* := \max_{\substack{\boldsymbol{z} \\ \alpha_i \geq 0}} \min_{\lambda \geq 0, \lambda' \geq 0} & -\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{X}_*'\|_2^2 + \frac{1}{2}\|\boldsymbol{X}_*'\|_2^2 \\ & + \sum_{i=1}^{\ell} \lambda_i \Big(\beta - \|\boldsymbol{Y}'^\top (2\boldsymbol{D}_i - \boldsymbol{I})\alpha_i + \boldsymbol{Y}'^\top \boldsymbol{D}_i \boldsymbol{z}\|_2\Big) \\ & + \sum_{i=1}^{\ell} \lambda_i' \Big(\beta - \|\boldsymbol{Y}'^\top (2\boldsymbol{D}_i - \boldsymbol{I})\alpha_i - \boldsymbol{Y}'^\top \boldsymbol{D}_i \boldsymbol{z}\|_2\Big) \end{aligned} \tag{15}$$

Noting by Sion's minimax theorem that strong duality holds, we can take the strong dual of the Lagrangian and not take the objective value. Flipping max and min, and maximizing with respect to $\boldsymbol{z}$ and α_i yields

$$d^* := \min_{\substack{\lambda \geq 0 \\ \lambda' \geq 0 \\ \|w_i\|_2 \leq 1 \\ \|\boldsymbol{z}_i\|_2 \leq 1 \\ (2\boldsymbol{D}_i - \boldsymbol{I})\boldsymbol{Y}'\boldsymbol{w}_i \geq 0 \\ (2\boldsymbol{D}_i - \boldsymbol{I})\boldsymbol{Y}'\boldsymbol{z}_i \geq 0}} \frac{1}{2}\|\Big(\sum_{i=1}^{\ell} \lambda_i \boldsymbol{D}_i \boldsymbol{Y}'\boldsymbol{w}_i - \lambda_i' \boldsymbol{D}_i \boldsymbol{Y}'\boldsymbol{z}_i\Big) - \boldsymbol{X}_*'\|_2^2 + \beta \sum_{i=1}^{\ell} \lambda_i + \lambda_i' \tag{16}$$

We lastly note that we can perform a change of variables $\boldsymbol{w}_i := \lambda_i \boldsymbol{w}_i$ and $\boldsymbol{z}_i := \lambda_i' \boldsymbol{z}_i$, and then minimize over λ_i and λ_i' to obtain the final convex, finite-dimensional strong dual.

$$d^* := \min_{\substack{(2\boldsymbol{D}_i - \boldsymbol{I})\boldsymbol{Y}'\boldsymbol{w}_i \geq 0 \\ (2\boldsymbol{D}_i - \boldsymbol{I})\boldsymbol{Y}'\boldsymbol{z}_i \geq 0}} \frac{1}{2} \| \sum_{i=1}^{\ell} \boldsymbol{D}_i \boldsymbol{Y}'(\boldsymbol{w}_i - \boldsymbol{z}_i) - \boldsymbol{X}_*' \|_2^2 + \beta \sum_{i=1}^{\ell} \|\boldsymbol{w}_i\|_2 + \|\boldsymbol{z}_i\|_2 \tag{17}$$

D Proof of Corollary 1

In this circumstance, we simply need to re-substitute the same objective as (4), with our new labels as $\boldsymbol{X}_*' - \boldsymbol{Y}_u$, where $\boldsymbol{Y}_u \in \mathbb{R}^{Nhw}$ is a flattened vector of the input image. Then, we simply have the problem given by

$$d^* = \min_{\substack{(2\boldsymbol{D}_i - \boldsymbol{I})\boldsymbol{Y}'\boldsymbol{w}_i \geq 0 \\ (2\boldsymbol{D}_i - \boldsymbol{I})\boldsymbol{Y}'\boldsymbol{z}_i \geq 0}} \| \sum_{i=1}^{P} \boldsymbol{D}_i \boldsymbol{Y}'(\boldsymbol{w}_i - \boldsymbol{z}_i) - (\boldsymbol{X}_*' - \boldsymbol{Y}_u) \|_2^2 + \beta \sum_{i=1}^{P} \Big(\|\boldsymbol{w}_i\|_2 + \|\boldsymbol{z}_i\|_2 \Big) \tag{18}$$

Thus, the general form of the convex dual formulation still holds with a simple residual network.