Computing Multiple Image Reconstructions
with a Single Hypernetwork

Alan Q. Wang   aw847@cornell.edu
School of Electrical and Computer Engineering, Cornell Tech, New York, NY, USA
Department of Radiology, Weill Cornell Medical School, New York, NY, USA

Adrian V. Dalca   adalca@mit.edu
Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA
A.A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Cambridge, MA, USA

Mert R. Sabuncu   msabuncu@cornell.edu
School of Electrical and Computer Engineering, Cornell Tech, New York, NY, USA
Department of Radiology, Weill Cornell Medical School, New York, NY, USA

Abstract

Deep learning based techniques achieve state-of-the-art results in a wide range of image reconstruction tasks like compressed sensing. These methods almost always have hyperparameters, such as the weight coefficients that balance the different terms in the optimized loss function. The typical approach is to train the model for a hyperparameter setting determined with some empirical or theoretical justification. Thus, at inference time, the model can only compute reconstructions corresponding to the pre-determined hyperparameter values. In this work, we present a hypernetwork-based approach, called HyperRecon, to train reconstruction models that are agnostic to hyperparameter settings. At inference time, HyperRecon can efficiently produce diverse reconstructions, which would each correspond to different hyperparameter values. In this framework, the user is empowered to select the most useful output(s) based on their own judgement. We demonstrate our method in compressed sensing, super-resolution and denoising tasks, using two large-scale and publicly-available MRI datasets. Our code is available at `https://github.com/alanqrwang/hyperrecon`.

Keywords: Image Reconstruction, Deep Learning, Hypernetworks, Hyperparameter tuning, Amortization

1. Introduction

The task of recovering high quality images from noisy or under-sampled measurements, often referred to as image reconstruction, is of crucial importance in many imaging applications. Classically, image reconstruction is formulated as an ill-posed inverse problem and solved by optimizing an instance-based regularized regression loss function (Fessler, 2019). More recently, deep learning (DL), and in particular supervised learning, has shown promise in improving over classical methods due to its ability to learn from data and to perform fast inference (Pal and Rathi, 2021).

The quality of solutions, for both the classical and DL-based techniques, depend heavily on the loss function that is minimized (Ghodrati et al., 2019; Zhao et al., 2018). Different loss functions can highlight or suppress varying features or textures in the reconstructions. A common approach is to minimize a composite loss that is a weighted sum of multiple
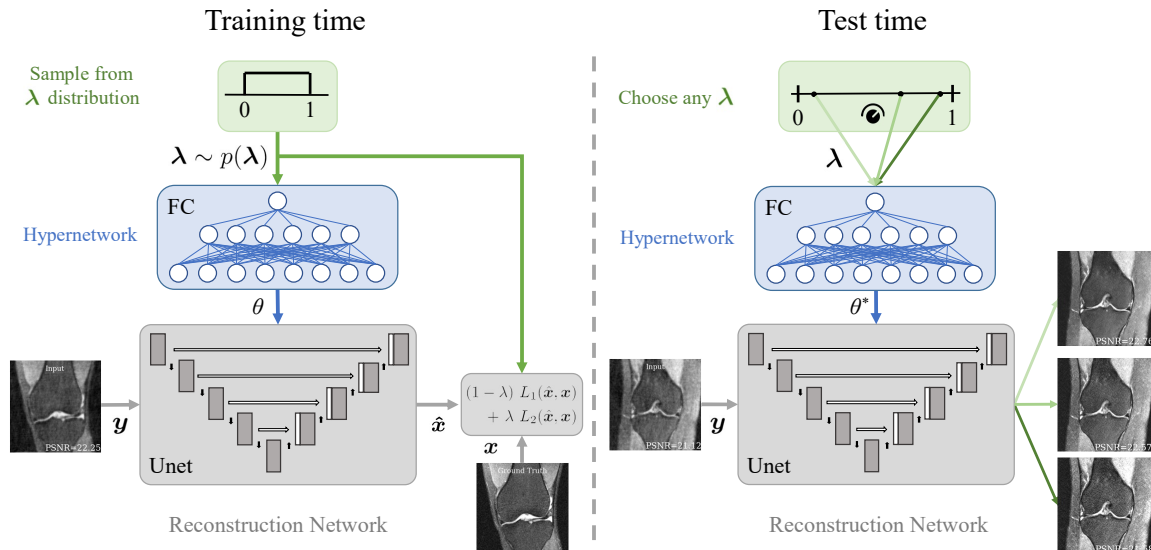
Figure 1: Proposed model. Main reconstruction network (Unet) takes as input the under-sampled measurement $\boldsymbol{y}$ and outputs reconstruction $\hat{\boldsymbol{x}}$, while the fully-connected (FC) hypernetwork takes as input the hyperparameter $\boldsymbol{\lambda}$ and outputs weights θ for the reconstruction network. During training, $\boldsymbol{\lambda}$ is sampled from a uniform distribution. At test-time, $\boldsymbol{\lambda}$ can be interactively selected to efficiently compute reconstructions associated with that $\boldsymbol{\lambda}$, yielding diverse reconstructions.

terms. However, the tuning of the weight hyperparameters is a non-trivial problem, requiring methods such as grid-search, random search, or Bayesian optimization (Frazier, 2018). Another weakness of existing methods is that once they are tuned and/or trained, they often produce a single best estimate of the reconstruction that is consistent with the measurements. Thus, any deviation from the conditions that the tool was optimized for means that the reconstructions can be sub-optimal.

More broadly, there is a lack of *interactive* and *controllable* tools that would enable human users to efficiently consider many reconstructions that are consistent with the measurements (Holzinger, 2016; Xin et al., 2018). Consider the results of the fastMRI Image Reconstruction Challenge[1], which revealed that commonly-used supervised metrics (e.g. mean-squared-error) do not correlate with the quality of images as judged by radiologists. We argue that giving users control and choice over possible reconstructions is a human-in-the-loop approach to mitigate this discrepancy.

In this work, to these ends, we propose a reconstruction strategy that is agnostic to the hyperparameter values and can efficiently produce reconstructions for a range of settings. Specifically, our model uses a hypernetwork that takes as input hyperparameter value(s) and outputs the weights of the reconstruction network. At test-time, arbitrary hyperparameter values can be provided by the user and corresponding reconstructions will be efficiently computed via a forward-pass through the reconstruction network. Thus, our method is capable of producing a range of reconstructions corresponding to different hyperparameter

---

1. `https://ai.facebook.com/blog/results-of-the-first-fastmri-image-reconstruction-challenge`

values, instead of a single reconstruction associated with a single hyperparameter setting. The user can, in turn, interactively view and select from this set of possible solutions, for use in downstream tasks.

Our method is applicable to any loss function where multiple terms need to be weighed. To demonstrate this, we perform experiments on two classes of loss functions in the reconstruction setting: supervised learning and amortized optimization. We experiment on three reconstruction tasks (CS-MRI, additive white Gaussian noise (AWGN) image denoising, and image superresolution) using two large-scale publicly available datasets (brain and knee MRI scans). These experiments highlight our proposed model's ability to match the performance of multiple baseline models trained on single settings of the hyperparameters.

This paper builds on our prior conference publication (Wang et al., 2021a), which dealt with the unsupervised setting where only under-sampled data were available for training. The present paper further considers the supervised setting, where we assume that we have access to fully-sampled training data and the primary design choice we are trying to solve is the weights in a composite loss function. Our contributions are as follows:

- We introduce a method for efficiently generating a range of image reconstructions using hypernetworks. Our method enables the end user to interactively view and choose the most useful reconstruction at inference time.

- We analyze several issues and propose solutions to hypernetwork training in the reconstruction setting, including hyperparameter distribution sampling and loss output scaling. Empirically, these solutions lead to improvement in model performance.

- Using multiple datasets and a variety of reconstruction tasks, we empirically demonstrate the performance of our proposed model and various improvements on two classes of loss functions: supervised learning (SL) and amortized optimization (AO).

2. Background

2.1 Inverse Problems for Image Reconstruction

We assume an unobserved (vectorized) image $\boldsymbol{x} \in \mathbb{C}^N$ is transformed by a forward model A into a measurement vector $\boldsymbol{y}$:

$$\boldsymbol{y} = A\boldsymbol{x} + \epsilon. \tag{1}$$

Here, N is the number of pixels of the full-resolution grid and ϵ encapsulates unmodeled effects such as noise. In image denoising, $A = I$. In image superresolution, $A = D$, where D is a down-sampling operator. In single-coil CS-MRI, $A = \mathcal{F}_u$, where $\mathcal{F}_u \in \mathbb{C}^{M \times N}$ is the under-sampled discrete Fourier operator and $M < N$.

Classically, inverse problems for image reconstruction are reduced to iteratively minimizing a regularized regression cost function on each collected measurement set (Beck and Teboulle, 2009; Boyd et al., 2011; Chambolle and Pock, 2011; Combettes and Pesquet, 2009; Daubechies et al., 2003; Ye et al., 2019).

$$\underset{\boldsymbol{x}}{\arg\min}\, J(\boldsymbol{x}, \boldsymbol{y}) + \sum_{i=1}^{K-1} \lambda_i \mathcal{R}_i(\boldsymbol{x}) \tag{2}$$

The first term, called the data-consistency loss, quantifies the agreement between the measurement vector $\boldsymbol{y}$ and reconstruction. One common choice for J is:

$$J(\boldsymbol{x}, \boldsymbol{y}) = \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2. \tag{3}$$

The $K - 1$ regularization terms $\mathcal{R}_i(\cdot)$ are hand-crafted priors which restrict the space of permissible solutions. A set of hyperparameters $\{\lambda_1, ..., \lambda_{K-1}\}$ weights the competing contributions of the K terms. Common choices for these priors include sparsity-inducing norms of wavelet coefficients (Figueiredo and Nowak, 2003) and total variation (TV) (Liu et al., 2018; Hu and Jacob, 2012; Rudin et al., 1992). While considerable research has been dedicated to designing suitable priors, such methods are invariably limited by lack of flexibility and adaptivity to the data. In addition, solving the minimization involves expensive iterative procedures that can take minutes per instance.

2.2 Supervised Learning and Amortized Optimization

Recent advances in data-driven algorithms and deep learning seek to address both aforementioned shortcomings of instance-based algorithms. Using large datasets and highly-parametrized and nonlinear neural network models, these algorithms learn priors directly from data and enable efficient inference via nearly-instantaneous forward passes.

Given a training dataset $\mathcal{D}_{tr}$ and a neural network model M_θ with parameters θ, the objective to minimize is:

$$\arg\min_\theta \mathbb{E}_{\mathcal{D}_{tr}} \mathcal{L}(M_\theta; \boldsymbol{\lambda}), \tag{4}$$

where $\mathcal{L}$ is a loss function parameterized by θ, $\mathbb{E}_{\mathcal{D}_{tr}}$ denotes the empirical average, and $\boldsymbol{\lambda}$ encapsulates hyperparameters of interest.

In supervised learning (SL), $\mathcal{D}_{tr}$ is assumed to contain pairs of observations and corresponding ground-truth images $(\boldsymbol{y}, \boldsymbol{x}) \sim \mathcal{D}_{tr}$, and $\mathcal{L}$ is some combination of K supervised losses $L(\cdot, \cdot)$:

$$\mathcal{L}(M_\theta; \boldsymbol{\lambda}) = L_K(M_\theta(\boldsymbol{y}), \boldsymbol{x}) + \sum_{k=1}^{K-1} \lambda_i L_i(M_\theta(\boldsymbol{y}), \boldsymbol{x}). \tag{5}$$

Common choices for $L(\cdot, \cdot)$ include mean-squared error (MSE), mean-absolute error (MAE), and structural similarity index measure (SSIM).

In addition to the SL set-up, we can consider the amortized optimization (AO) scenario, where $\mathcal{D}_{tr}$ is assumed to only contain observations $\boldsymbol{y} \sim \mathcal{D}_{tr}$. In AO, $\mathcal{L}$ takes a similar form to Eq. (2):

$$\mathcal{L}(M_\theta; \boldsymbol{\lambda}) = J(M_\theta(\boldsymbol{y}), \boldsymbol{y}) + \sum_{i=1}^{K-1} \lambda_i \mathcal{R}_i(M_\theta(\boldsymbol{y})). \tag{6}$$

Thus, M_θ is trained to minimize Eq. (2) for all observations in the dataset. Although it minimizes the same cost function, AO provides several advantages over classical solutions. First, at test-time, it replaces an expensive iterative optimization procedure with a simple forward pass of a neural network. Second, since the model M_θ estimates the reconstruction for any viable input measurement vector $\boldsymbol{y}$ and not just a single instance, AO acts as a natural regularizer for the optimization problem (Balakrishnan et al., 2019; Shu et al., 2018;

Wang et al., 2020). Third, recent work has shown that inductive biases of convolutional neural networks provide favorable implicit priors for image reconstruction (Heckel and Hand, 2019; Liu et al., 2018; Ulyanov et al., 2020). Since no ground-truth images are required, AO is sometimes referred to as unsupervised learning.

Importantly, similar to the classical cost function, both the SL and AO class of loss functions involve the tuning of a set of hyperparameters $\{\lambda_1, ..., \lambda_{K-1}\}$, which can significantly affect the end reconstructions. Therefore, hyperparameter tuning is typically carefully done using expensive methods like cross-validation. In this work, we propose a strategy *that can replace the expensive hyperparameter tuning step, such that reconstructions with different $\boldsymbol{\lambda}$ values can be efficiently generated at inference time.* To achieve this, we employ hypernetworks.

2.3 Hypernetworks

A hypernetwork is a neural network that generates the weights of a network which solves the main task (e.g. classification, segmentation, reconstruction). In this framework, the parameters of the hypernetwork, and not the main network, are learned. While originally introduced for achieving weight-sharing and model compression (Ha et al., 2016), this idea has found numerous applications including neural architecture search (Brock et al., 2018; Zhang et al., 2019), Bayesian neural networks (Krueger et al., 2018; Ukai et al., 2018), multi-task learning (Lin et al., 2021; Mahabadi et al., 2021; Pan et al., 2018; Shen et al., 2017; Klocek et al., 2019), and hyperparameter optimization (Lorraine and Duvenaud, 2018; Hoopes et al., 2021).

Notably, hypernetworks have been used to replace expensive hyperparameter tuning procedures like cross-validation (Lorraine and Duvenaud, 2018). Hyperparameter tuning can be formulated as a nested optimization. The inner optimization minimizes some loss with respect to the weights of the main network $\theta \in \Theta$ over a training dataset $\mathcal{D}_{tr}$, while the outer optimization minimizes that loss with respect to the hyperparameters $\boldsymbol{\lambda} \in \Lambda$ over a held-out validation dataset $\mathcal{D}_{val}$:

$$\underset{\boldsymbol{\lambda}}{\arg\min} \ \mathbb{E}_{\mathcal{D}_{val}} \mathcal{L} \left(\underset{\theta}{\arg\min} \mathbb{E}_{\mathcal{D}_{tr}} \mathcal{L}(M_\theta; \boldsymbol{\lambda}) \right) := \underset{\boldsymbol{\lambda}}{\arg\min} \ \mathbb{E}_{\mathcal{D}_{val}} \mathcal{L} \left(\theta^*(\boldsymbol{\lambda}) \right). \tag{7}$$

Hypernetworks can be trained to approximate the solution to the inner optimization $\theta^*(\boldsymbol{\lambda})$. In this setting, a hypernetwork maps from hyperparameters to main network weights $H_\phi : \Lambda \to \Theta$. The model is trained via a stochastic optimization scheme whereby hyperparameters are sampled from a pre-defined distribution over Λ during training:

$$\phi^* = \underset{\phi}{\arg\min} \mathbb{E}_\Lambda \mathbb{E}_{\mathcal{D}_{tr}} \mathcal{L}(M_{H_\phi(\boldsymbol{\lambda})}; \boldsymbol{\lambda}). \tag{8}$$

The approximate solution to the inner optimization is then $\theta^*(\boldsymbol{\lambda}) = H_{\phi^*}(\boldsymbol{\lambda})$. Note that only the parameters of the hypernetwork ϕ are learned.

The upshot of this method is that rapid hyperparameter tuning is possible, since the inner optimization can be performed with a forward pass of the hypernetwork. While using this model enables us to address the issue of costly hyperparameter tuning in the context of image reconstruction (as discussed in the previous section), we further build on this idea

in this work by *leveraging the vast set of reconstructions that are capable of being generated for a given measurement as a result of changing* $\boldsymbol{\lambda}$. That is, we propose this method as a means of creating an interactive and controllable image reconstruction tool.

3. Related Works

Hypernetworks have recently shown promise in rendering agnosticm to hyperparameters with minimal increase in training time. Hoopes et. al. use this method to enable test-time rapid tunability in image registration for medical imaging (Hoopes et al., 2021). Our previous conference work applied this idea to the AO setting for CS-MRI (Wang et al., 2021a). In this work, we extend this idea broadly to other image reconstruction tasks in both the SL and AO settings, and validate on additional datasets.

Sahu et. al. propose an interactive method for optimizing smoothing parameters in Digital Breast Tomosynthesis (Sahu et al., 2021). The authors' method produce reconstructions conditioned on a user-specified smoothing parameter by multiplying the intermediate activations of a reconstruction network with the parameter; the network is trained to minimize mean absolute error. However, this method requires ground-truth reconstructions obtained by solving many instance-based iterative procedures, which is computationally expensive.

Our work is similar in spirit to the work of Lin et. al., which uses hypernetworks to enable controllability in the multi-task learning setting (Lin et al., 2021). In their work, they pose multi-task learning as a multi-objective minimization problem, and show that hypernetworks can enable test-time trade-off control among different tasks (e.g. depth prediction and semantic segmentation) with a single model.

4. Methods

We propose to use hypernetworks for image reconstruction, where our goal is to produce a model that can efficiently compute a reconstruction that approximates the solution to Eqs. (5) and (6) for arbitrary hyperparameter coefficient values. We call this model a *controllable* reconstruction network, since the resulting model can produce a diverse set of reconstructions that can be controlled interactively at test-time. We illustrate the model in Fig. 1.

4.1 Controllable Reconstruction Network

Let M_θ denote a main network with parameters $\theta \in \Theta$ which maps an observation $\boldsymbol{y}$ to a reconstruction $\hat{\boldsymbol{x}}$. We define a hypernetwork $H_\phi : \mathbb{R}_+^{K-1} \to \Theta$ that maps a hyperparameter weight vector $\boldsymbol{\lambda}$ to the parameters θ of the main network M_θ. A reconstruction for a given observation $\boldsymbol{y}$ and hyperparameter vector $\boldsymbol{\lambda}$ is then $\hat{\boldsymbol{x}} = M_{H_\phi(\boldsymbol{\lambda})}(\boldsymbol{y})$. Effectively, this makes $\boldsymbol{\lambda}$ an input to the model.

The objective is given in Eq. (8), where $\Lambda = \mathbb{R}_+^{K-1}$.

4.2 Training

We restrict our attention to the case of two and three loss terms (one and two hyperparameters, respectively), although we emphasize that our method is applicable to an arbitrary

number of loss terms. In general, Eqs. (5) and (6) can be manipulated such that the hyperparameter support is bounded to $\boldsymbol{\lambda} \in \Lambda = [0,1]^{K-1}$. For example, for one and two hyperparameter weights and arbitrary loss terms $L_1, L_2,$ and L_3 (omitting θ, $\boldsymbol{x}$, and $\boldsymbol{y}$ for brevity):

$$\mathcal{L}_2 = (1 - \lambda)L_1 + \lambda L_2, \tag{9}$$

$$\mathcal{L}_3 = \lambda_1 L_1 + (1 - \lambda_1)\lambda_2 \ L_2 + (1 - \lambda_1)(1 - \lambda_2) \ L_3. \tag{10}$$

A straightforward strategy for training the hypernetwork involves sampling the coefficients from a uniform distribution $p(\boldsymbol{\lambda}) = U[0,1]^{K-1}$ and sampling an undersampled measurement vector $\boldsymbol{y}$ for each forward pass during training. The gradients are then computed with respect to the loss evaluated at the sampled $\boldsymbol{\lambda}$ via a backward pass. This corresponds to minimizing Eq. (8) with a uniform distribution for $\boldsymbol{\lambda}$. We denote this sampling strategy as uniform hyperparameter sampling (UHS).

4.3 Data-driven Hyperparameter Sampling for AO

In the hypothetical scenario of infinite hypernetwork capacity, the hypernetwork can capture a mapping of any input $\boldsymbol{\lambda}$ to the optimal $\theta^* = H_{\phi^*}(\boldsymbol{\lambda})$ that minimizes Eqs. (5) and (6) with the corresponding $\boldsymbol{\lambda}$. However, in practice, the finite model capacity of the hypernetwork constrains the ability to achieve optimal loss for every hyperparameter value. In this case, the model performance will depend on the adopted distribution for $\boldsymbol{\lambda}$.

Algorithm 1 UHS and DHS Hypernetwork Training

Input: Dataset $\mathcal{D}_{tr}$, model M_{H_ϕ}, K, B, $b < B$, $\gamma > 0$
Output: Model weights ϕ

1: **repeat**
2:      Sample $\{\boldsymbol{y}_1, ..., \boldsymbol{y}_B\} \sim \mathcal{D}_{tr}$
3:      Sample $\{\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_B\} \sim U[0,1]^{K-1}$
4:      $\hat{\boldsymbol{x}}_i \leftarrow M_{H_\phi(\boldsymbol{\lambda}_i)}(\boldsymbol{y}_i)$ for $i = 1, ..., B$
5:      **if** UHS **then**
6:          $\phi \leftarrow \phi - \gamma \nabla_\phi \sum_{i=1}^{B} \mathcal{L}_K \left(\hat{\boldsymbol{x}}_i; \boldsymbol{\lambda}_i\right)$
7:      **else if** DHS **then**
8:          $\{\tilde{\boldsymbol{\lambda}}_1, ..., \tilde{\boldsymbol{\lambda}}_B\} \leftarrow \text{sort}\{\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_B\}$
9:          $\phi \leftarrow \phi - \gamma \nabla_\phi \sum_{i=1}^{b} \mathcal{L}_K \left(\hat{\boldsymbol{x}}_i; \tilde{\boldsymbol{\lambda}}_i\right)$
10: **until** convergence
11: **return** ϕ

Figure 2: Algorithm for UHS and DHS training. Subscripts index over the mini-batch. Sorting (Line 8) is done in ascending order w.r.t. $J(\hat{\boldsymbol{x}}_i, \boldsymbol{y}_i)$. Red highlights differences between UHS and DHS.

In SL loss functions, we are typically interested in the reconstructions associated with all possible settings of the coefficients. Thus, UHS is a necessary sampling strategy. In contrast, in AO loss functions, some coefficients will not produce acceptable reconstructions, even if solved optimally. For example, a reconstruction with a large coefficient for TV tends to be overly smooth with little to no structural content. Thus, sampling hyperparameters from the entire support $[0,1]^{K-1}$ "wastes" model capacity on undesirable regions of hyperparameter space.

In most real-world reconstruction scenarios, we don't have a good prior distribution from which to sample desirable regions. Instead, we propose a data-driven sampling scheme (DHS) which learns the prior over the course of training. We leverage the data-consistency

loss induced by a setting of the coefficients to assess whether the reconstruction will be useful or not. Intuitively, values of $\boldsymbol{\lambda}$ which lead to high data-consistency loss $J\left(M_{H_{\phi}(\boldsymbol{\lambda})}(\boldsymbol{y}), \boldsymbol{y}\right)$ will produce reconstructions that deviate too much from the underlying anatomy, and which therefore can be ignored during training.

One can enforce this idea by computing gradients only on values of $\boldsymbol{\lambda}$ which induce a data-consistency loss below a pre-determined threshold during training. However, this presents two problems. First, calibrating and tuning this threshold can be difficult in practice. Second, at the beginning of training, since the main network will not produce good reconstructions, this threshold will likely not be satisfied.

In lieu of a complex training/threshold scheduler, we adapt the threshold with the quality of reconstructions directly within the training loop. The proposed DHS strategy works by using the best b samples with the lowest data-consistency loss within a mini-batch of size B to compute the gradients. In effect, this induces a variable threshold b/B of the landscape percentage which is optimized, where this threshold adapts dynamically over training. Algorithm 1 details the training loop for both UHS and DHS.

We demonstrate in the Experiments section that DHS leads to improved model performance on the most promising hyperparameter space regions compared to UHS, given a fixed hypernetwork capacity.

4.4 Matched Loss Output Scales

In general, a given loss function L_i can be arbitrarily scaled by a constant $\alpha_i L_i$ and still yield the same reconstruction results (ignoring the influence on learning rate, for example). In multi-term loss functions, the relative scales of the K loss outputs should be approximately matched during training to allow for equal contribution of gradients during backpropagation.

In the context of our proposed method, these loss output scales have an added effect: different scales will change the resulting reconstructions as a function of $\boldsymbol{\lambda}$. If one loss dominates another, then most of the landscape will be dedicated to minimizing the dominant loss and very little variation will exist for different $\boldsymbol{\lambda}$. Indeed, we ideally would like a landscape which varies maximally over $\boldsymbol{\lambda}$, which would (hopefully) capture as wide a range of diverse reconstructions as possible.

For SL, we account for this by normalizing the loss functions by its "best-case" loss averaged over a validation set. Let s_i denote the average validation loss on L_i for a model only optimized for this specific loss function, and we set $\alpha_i = 1/s_i$. During training, validation losses associated with all loss terms will (roughly) converge to a value of 1, and thus the loss output scales will be approximately matched. Further details are given in Section 5.4.

For AO, we cannot take the same approach since the "best-case" losses for $J(\cdot, \cdot)$ and $\mathcal{R}(\cdot)$ are 0. When Eq. (2) is derived from the maximum a posteriori (MAP) estimate, the scaling factors can be computed with respect to the parameters of the likelihood and/or prior distributions (Chambolle et al., 2010; Hoopes et al., 2021). However, in cases where the regularization term cannot be expressed as a valid probability or when multiple terms are used, such an approximation cannot be obtained. In this work, we tuned the scaling factors to align the magnitudes of the separate terms. Although this works well in practice,

this issue is still an open question in the general case and may be an important direction for further research.

5. Experiments

We evaluate our proposed method on two large, publicly-available MRI datasets consisting of coronal knees and axial brains for SL and AO classes of multi-term loss functions. For AO, we experiment with the CS-MRI reconstruction task. For SL, we experiment with CS-MRI, AWGN denoising, and image superresolution reconstruction tasks.

5.1 Data

We conducted our experiments using two large, publicly-available datasets. The first is the NYU fastMRI dataset (Zbontar et al., 2018) composed of proton density (PD) and proton density fat-suppressed (PDFS) weighted knee MRI scans. The second is the ABIDE dataset, composed of T1-weighted brain MRI scans (Di Martino et al., 2014).

For both datasets, we used 100, 25, and 50 subjects for training, validation, and testing, respectively. Volumes were separated into 2D slices and slices with only background were removed, resulting in a final train/val/test split of 3500/875/1750 for knee and 11400/2850/5700 for brain slices. All images were intensity-normalized to the range $[0, 1]$ and cropped and re-sampled to a pixel grid of size 256×256.

5.2 Model Details

This work presents a general strategy for computing hyperparameter-agnostic reconstructions with a single model, which is applicable to any main network architecture. In this work, the main network M_θ is a commonly-used Unet architecture (Ronneberger et al., 2015) adapted for reconstruction, where the network input is a 2-channel, complex-valued input and the network output is a single-channel, real-valued output. A hidden channel dimension of 32 is used for all layers, resulting in a total main network parameter count of $149, 409$.

Since M_θ receives noisy images as input, this network can be viewed as a post-processing network preceded by an optional model-based inversion step (e.g. applying the inverse Fourier transform to under-sampled measurements), which is a common reconstruction pipeline in the literature (Jin et al., 2017; Kang et al., 2017; Wang et al., 2021b). We expect the conclusions drawn in this work to hold for any deep learning-based main network, particularly state-of-the-art unrolled networks (Monga et al., 2019).

The hypernetwork H_ϕ consists of a 5-layer fully-connected (FC) architecture with intermediate LeakyReLUs. The input to the network is the number of hyperparameters, and the hidden dimension and output dimension are of size d. Each convolution layer in the main network takes as input the d-dimensional output embedding of the hypernetwork and linearly projects it to the size of the kernel and bias, where the parameters of the projection are learned. We treat the hypernetwork and the main network as a single, large network, and refer to the overall model as HyperRecon. We experiment with hypernetwork hidden dimensions $d \in \{4, 32, 128\}$, and refer to them as HyperRecon-{S, M, L}, respectively. We used a batch size of 32 and the Adam optimizer for all models (Kingma and Ba, 2017).

Table 1: Training time and number of parameters for all models on brain dataset. "∼" denotes approximation. "All Unets" denotes all 361 baselines across all hyperparameters. Inference time (defined as the runtime of one forward pass with a single λ and y input, averaged over the test set) for all models is ∼0.2 seconds.

Model	Train time (hrs)	# parameters
Unet	∼4	149K
All Unets	∼648	53M
HyperRecon-S	∼5	744K
HyperRecon-M	∼6	4.9M
HyperRecon-L	∼7	19M

All training and testing experiments were performed on a machine equipped with an Intel Xeon Gold 6126 processor and an NVIDIA Titan Xp GPU. All models were implemented in Pytorch. Table 1 outlines the training time, inference time, and number of parameters for all models used in experiments. We train each model until the loss converges on the validation set, typically for 1000 epochs.

5.3 Baselines

For comparison, we trained separate Unet reconstruction networks for each fixed hyperparameter value. We refer to these models as baselines and emphasize that they demand significant computational resources, since each of these models needs to be trained and saved separately (see Table 1). For SL experiments, we trained five baseline models for $\lambda \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. For AO experiments, we trained 361 baseline models with hyperparameters chosen non-uniformly on a 19×19 grid over the space $[0, 1] \times [0, 1]$, in order to more densely sample in high-performing regions[2].

5.4 Loss Functions

5.4.1 SUPERVISED LEARNING

For SL, we consider CS-MRI, denoising, and superresolution reconstruction tasks. For all tasks, we use MAE for L_1 and SSIM for L_2 (Wang et al., 2004) as the two loss terms in Eq. (9).

MAE is a global, intensity-based metric which is simple and widely-used. SSIM is a perception-based metric that considers image degradation as perceived change in structural information. Prior work has shown performance gain when using a combination of these two losses (Zhao et al., 2018). While we restrict our focus to these two losses in the SL setting, we emphasize that our method works for any choice of supervised loss functions.

Proper HyperRecon training requires approximately-matched loss output scales, i.e. α_1 and α_2 (see Section 4.4). To compute these values, we first train two Unet models for $\lambda = 0$ and $\lambda = 1$. The validation losses s_1 and s_2 for the two models are computed, and then we set $\alpha_1 = 1/s_1$ and $\alpha_2 = 1/s_2$ during HyperRecon training.

---

2. The 19 values were {0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.85,0.9,0.93,0.95,0.98, 0.99,0.995,0.999,1.0}.

5.4.2 Amortized Optimization

For AO, we consider the CS-MRI reconstruction task and experiment with two regularization terms $\mathcal{R}_1$ and $\mathcal{R}_2$ in Eq. (10): layer-wise total ℓ_1-penalty on the weights θ of the main reconstruction network and the anisotropic total variation of the reconstruction image:

$$\mathcal{R}_1 = \sum_{i=1}^{L} \|\theta_i\|_1\,, \qquad\qquad \mathcal{R}_2 = \sum_{i=1}^{N} \big|[D_1 x]_i\big| + \big|[D_2 x]_i\big|, \qquad (11)$$

where θ_i denotes the weights of the reconstruction network for the ith layer, L is the total number of layers of the reconstruction network, and D_1 and D_2 denote the finite difference operation along the first and second dimension of a two-dimensional image.

5.5 Evaluation

5.5.1 Supervised Learning

For SL experiments, in addition to evaluating on the trained losses MAE and SSIM, we also evaluate on PSNR and high-frequency error norm (HFEN) (Ravishankar and Bresler, 2011). Given a 15×15 Laplacian-of-Gaussian (LoG) filter with a standard deviation of 1.5 pixels, the HFEN is computed as the ℓ_2 difference between LoG-filtered ground truth and LoG-filtered reconstructions.

To demonstrate the downstream utility of HyperRecon, we evaluate segmentation performance of the reconstructions for varying λ on a deep learning-based segmentation model. The model has the same architecture as the baseline Unet M_θ, except that it has a 5-channel output and a final softmax layer. We train the model to minimize the soft-Dice loss (Sudre et al., 2017) on the same ABIDE dataset with the same data split as reconstruction training, and evaluate the hard-Dice score (Dice, 1945). Ground truth segmentation maps corresponding to 5 regions-of-interest[3] were anatomically segmented with FreeSurfer (Fischl, 2012) (see (Balakrishnan et al., 2019) for details). Batch size, optimizer, and other hyperparameters were identical to reconstruction training.

5.5.2 Amortized Optimization

For AO experiments, we report the *relative* PSNR (abbreviated rPSNR) for a reconstruction by subtracting the PSNR value for the zero-filled reconstruction from the PSNR value of the reconstruction. Positive values for rPSNR are preferable as they indicate that the regularization terms lead to improvement over the trivial zero-filled reconstructions.

For ease of visualization, we report the *negative* of certain metrics so that higher values are better for all metrics. These are abbreviated by an "n" in front of the metric name (e.g. nMAE, nHFEN).

5.6 Reconstruction Tasks

For all tasks, noisy inputs to our model were obtained retrospectively and performed *in-silico*. Although this is a slight simplification, we believe that these experiments are consis-

---

3. Background, gray matter, white matter, cerebrospinal fluid, and cortex.

tent with the main methodological message of this work, and furthermore that these results will translate to real-world scenarios.

5.6.1 CS-MRI

k-space data was generated by retrospective under-sampling using 4-fold and 8-fold acceleration under-sampling masks generated using a polynomial Poisson-disk variable-density sampling strategy (Geethanath et al., 2013; Lustig et al., 2007). The input into the models were the zero-filled reconstructions, i.e. the inverse Fourier transform of the under-sampled k-space data with zeros for missing values.

5.6.2 AWGN DENOISING

Ground truth images were used as clean images, which were corrupted with additive white Gaussian noise sampled from $\mathcal{N}(0, \sigma^2 I)$, where $\sigma = 0.1$.

5.6.3 SUPERRESOLUTION

Ground truth images were used as full-resolution images, which were down-sampled by a factor of 4 and subsequently up-sampled using nearest-neighbor interpolation back to the original grid size.

6. Results

6.1 Supervised Learning

We evaluate the performance of HyperRecon-L using metric curves over the space of permissible hyperparameter values $\lambda \in [0, 1]$. We generated the curves for visualization by densely sampling the support $[0, 1]$ to create 100 discrete samples. For each grid point, we computed the value by passing the corresponding hyperparameter values to the model along with each under-sampled measurement $\boldsymbol{y}$ in the test set and taking the average PSNR value.

Fig. 3 shows metric performance for brain data. Each panel shows Unet and HyperRecon-L performance for the specified metric across λ values. Similarly, Fig. 4 shows metric performance for knee data on three different reconstruction tasks. We observe very close matching of performance across all metrics between Unet and HyperRecon-L models, indicating that the hypernetwork is able to generate the optimal weights for the entire hyperparameter space. For downstream brain segmentation, the optimal λ for maximizing Dice is around 0.25. With HyperRecon-L, this optimal point can be easily obtained with a single model.

Fig. 5a shows representative brain and knee slices for the CS-MRI task, and Fig. 5b shows zoomed-in versions of the red arrow regions for the brain slices. We notice a significant visual similarity between the same hyperparameters for Unet and HyperRecon-L models. In general, a higher weight on MAE tends to lead to more blurry and smoother reconstructions, while a higher weight on SSIM tends to lead to more noisy reconstructions with more high frequency content. In addition, MAE tends toward less contrast between hypo-intense and hyper-intense regions, whereas SSIM accentuates this contrast more. With baseline models where $\boldsymbol{\lambda}$ must be chosen before training, these variations would be completely missed and
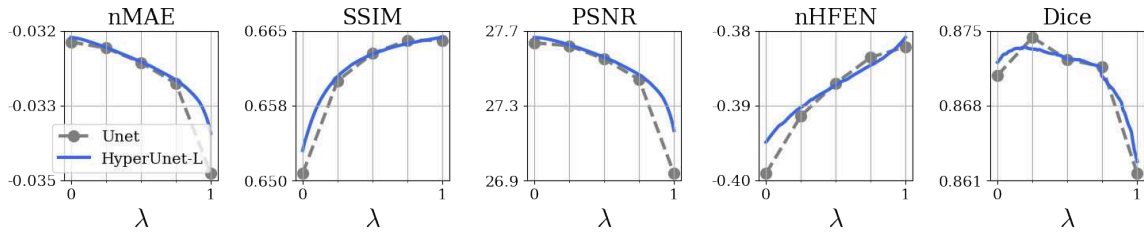
Figure 3: Model performance on brain dataset for SL loss functions. Higher is better. Each panel shows metric performance for Unet and HyperRecon-L models for varying λ. The loss function of the HyperRecon-L model is a sum of MAE and SSIM loss functions (first 2 panels), where $\lambda = 0$ corresponds to only MAE loss and $\lambda = 1$ corresponds to only SSIM loss (see Section 5.4.1).
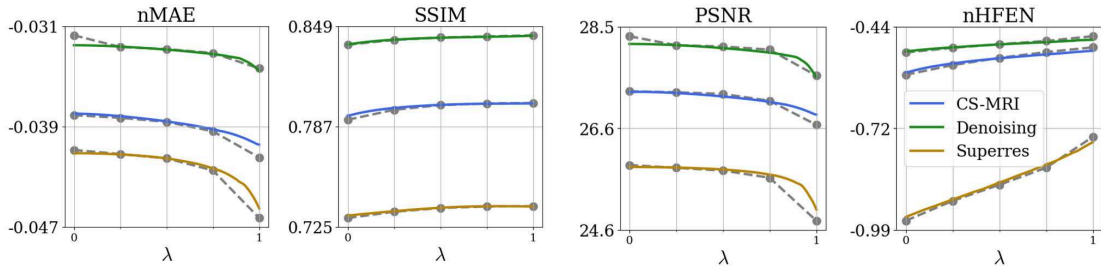

Figure 4: Model performance on knee dataset for SL loss functions. Higher is better. Each panel shows metric performance for Unet and HyperRecon-L models for varying λ, where each color corresponds to a different reconstruction task. The loss function of the HyperRecon-L model is a sum of MAE and SSIM loss functions (first 2 panels), where $\lambda = 0$ corresponds to only MAE loss and $\lambda = 1$ corresponds to only SSIM loss (see Section 5.4.1).

end users would be stuck with one reconstruction. Additional slices for CS-MRI, denoising, and superresolution tasks are presented in the Appendix.

6.2 Amortized Optimization

We evaluate the performance of HyperRecon-{S, M, L} using rPSNR landscapes over the space of permissible hyperparameter values $(\lambda_1, \lambda_2) \in [0,1] \times [0,1]$. We generated landscapes for visualization by densely sampling the support $[0,1] \times [0,1]$ to create a grid of size 100×100. For baselines, the 19×19 grid was linearly interpolated to 100×100 to match the hypernetwork landscapes.

Fig. 6 shows rPSNR landscapes for different reconstruction models. The top left map corresponds to baselines. The first image in the second row shows an example histogram of the regularization weight samples used for gradient computation over one DHS training epoch.

The remaining images in the top row correspond to UHS models with varying hypernetwork capacities. Similarly, the bottom row shows rPSNR landscapes for DHS models at the same capacities. We find that higher capacity hypernetworks approach the baseline
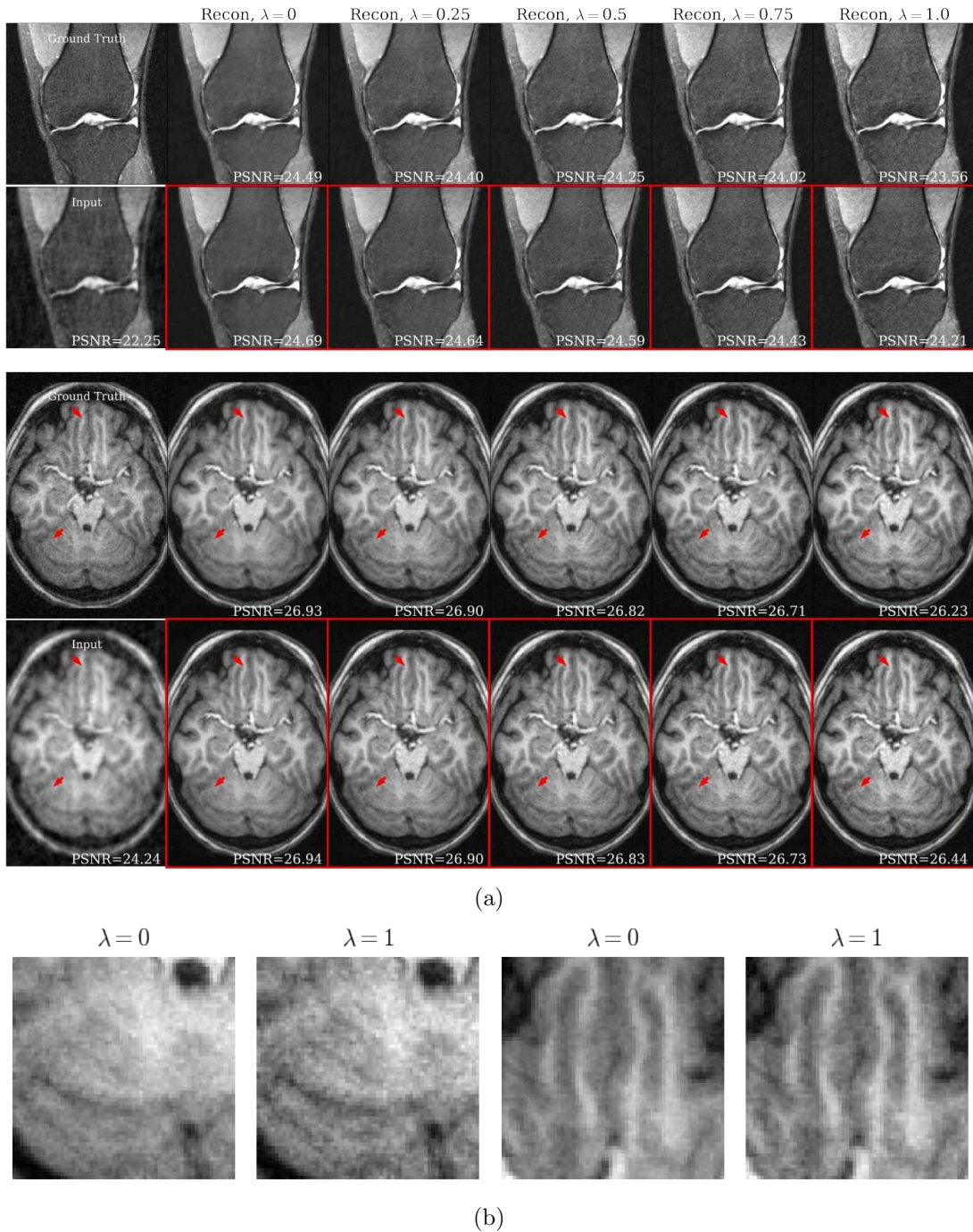
(a)

(b)

Figure 5: (a) Representative slices for SL on CS-MRI task with 8-fold under-sampling. Ground truth and input images are denoted in the first column. Top row are baseline Unet reconstructions with varying λ. Bottom row (red) are HyperRecon-L reconstructions with varying λ. PSNR values in reconstructions provide a basis of comparison between Unet and HyperRecon-L reconstructions of the same λ. Brain slices are cropped to show detail. Arrows indicate corresponding points where the difference between reconstructions can be appreciated. (b) Representative brain patches for HyperRecon-L, centered around red arrows.

14
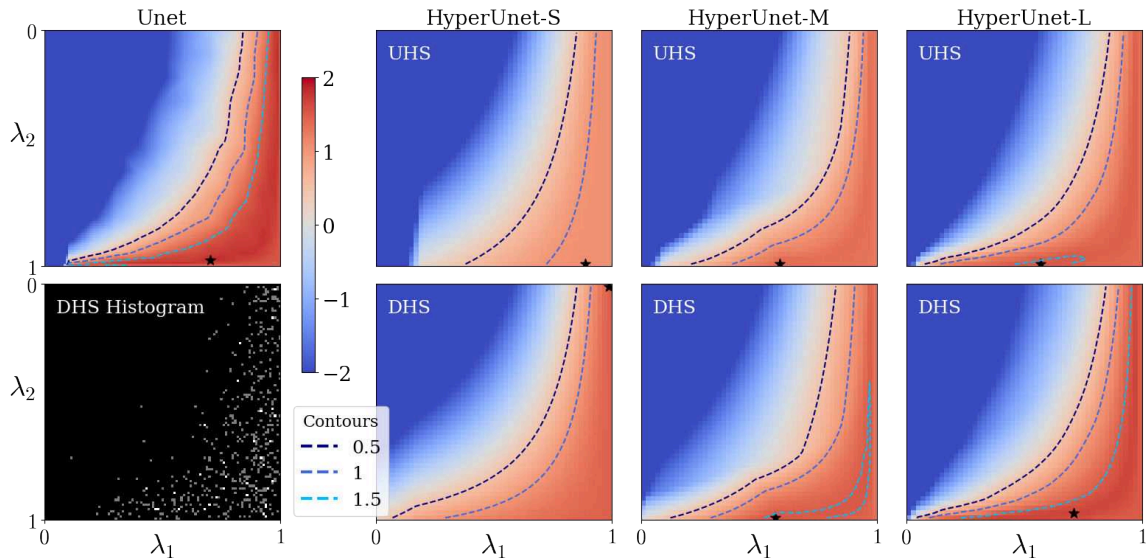
Figure 6: rPSNR values for knee dataset on AO loss over the hyperparameter support $[0,1] \times [0,1]$ for different hypernetwork capacity and sampling methods. The x-axis and y-axis denote the value of the hyperparameters λ_1 and λ_2, respectively. Contours denote level sets of fixed value (see legend). Stars in the landscapes denote maximum value. (Left) Top image depicts the baseline landscape. Bottom image shows an example histogram of hyperparameter values used for gradient computation during one epoch of training with the DHS strategy. (Right) Top and bottom row show the UHS and DHS model landscapes, respectively, for three different hypernetwork capacities.

models' performance, at the cost of computational resources and training time (see Table 1). We also observe significant improvement in performance using DHS as compared to UHS, given a fixed hypernetwork capacity. We find that the performance improvement achieved by DHS is less for the large hypernetwork, validating the expectation that the sampling distribution plays a more important role when the hypernetwork capacity is restricted.

Fig. 7 shows representative brain and knee slices from the DHS HyperRecon-L model. The two corresponding reconstructions are selected as follows. First, we densely sample uniformly from $[0,1] \times [0,1]$ to generate 100×100 reconstructions. Then, we filter out the reconstructions which are below some threshold PSNR value (we choose the 90th percentile). Finally, we choose the two reconstructions from this filtered set which are maximally separated by ℓ_2 distance.

We notice that the two reconstructions are significantly dissimilar despite the similarity in PSNR value. In the baseline setting, only one of these reconstructions would be available and finding another reconstruction would require training from scratch. With our model, users can search over the entire space of possible reconstructions and choose the one(s) they prefer. This highlights the value of this model as a tool for the interactive selection of many diverse reconstructions for further use based on visual inspection.
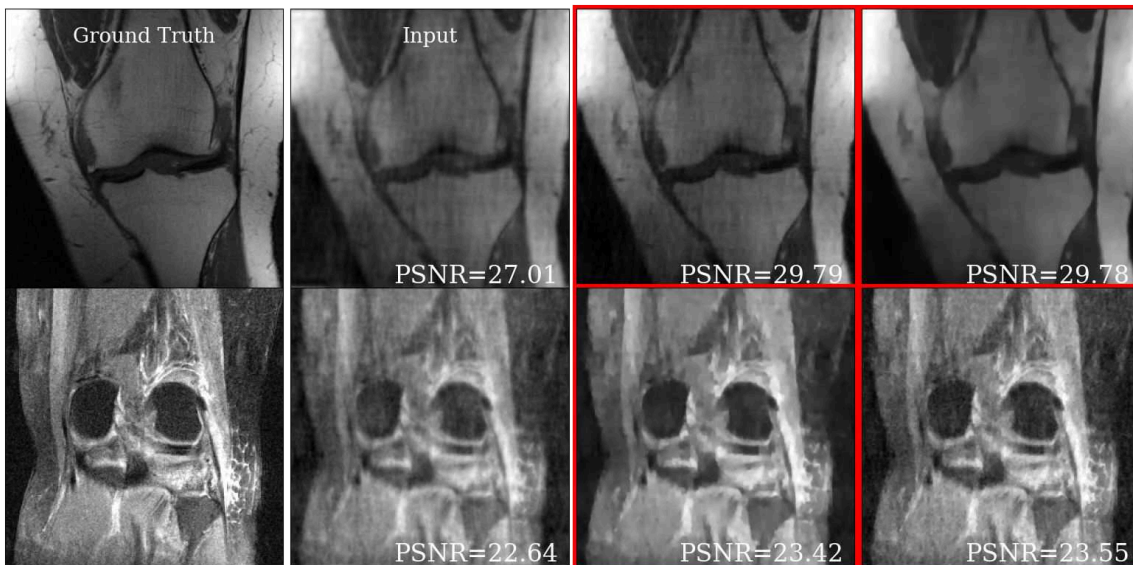
Figure 7: Representative slices for AO on CS-MRI task on 8-fold undersampled knees. Ground truth is denoted and two visually different but high-quality reconstructions (in terms of PSNR) are shown in red.

7. Conclusion and Future Work

We presented a method for controllable image reconstruction using hypernetworks, which enables the interactive selection from a dense set of reconstructions at test-time. We highlight and address several issues related to hypernetwork training associated with the reconstruction setting, and demonstrate empirically that our method works on two datasets and on a variety of multi-term loss functions and reconstruction tasks.

We believe this work opens many interesting directions for further research. A straightforward and promising future direction is to extend this to a higher number of loss coefficient hyperparameters. The challenges with this extension is two-fold. First, hypernetworks need to be designed to improve their expressivity for high-dimensional hyperparameter spaces. Second, since the space of possible reconstructions grows with additional hyperparameters, searching through all reconstructions becomes quickly intractable. Improved techniques for extracting the "best" reconstructions to show to the user would be necessary.

Improving the expressivity and efficiency of hypernetworks is still under-explored. From Table 1, it can be seen that even the largest hypernetwork we experimented with requires three times less the number of parameters as compared to the combined parameters of all baseline models, with only a slight increase in training time. However, there may be more efficient yet expressive parameterizations of hypernetworks (for example, hypernetworks which only generate weights for certain layers of the main network) that can enable further compression.

The matched scaling of loss functions discussed in section 4.4 is an important open question. Since the optimal hyperparameter landscape is in general dependent on the scale of the loss functions and furthermore can affect the training dynamics, an automated and

data-driven approach to loss scaling would be preferable. Insights may be gleaned from similar work in multi-task learning, where matching the contributions of losses associated with different tasks is important to the end performance (Kendall et al., 2018). Other ideas associated with matching the scale of the gradients for each loss (instead of the scale of the loss itself) may also be useful.

Acknowledgments

Ethical Standards

We used publicly available data that was anonymized/de-identified and originally collected with appropriate consents and approvals from the institutional IRBs and/or ethical review boards.

Conflicts of Interest

We declare we don't have conflicts of interest.

References

Guha Balakrishnan, Amy Zhao, Mert R. Sabuncu, John Guttag, and Adrian V. Dalca. Voxelmorph: A learning framework for deformable medical image registration. *IEEE TMI*, 38(8):1788–1800, Aug 2019. ISSN 1558-254X.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, March 2009.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011. ISSN 1935-8237.

Andrew Brock, Theo Lim, J.M. Ritchie, and Nick Weston. SMASH: One-shot model architecture search through hypernetworks. In *ICLR*, 2018.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40, 05 2011.

Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. *An Introduction to Total Variation for Image Analysis*, pages 263–340. De Gruyter, 2010.

Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. *arXiv preprint arXiv:0912.3522*, 2009.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *arXiv preprint arXiv:0307152*, 2003.

A. Di Martino, C. G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto, B. Deen, and et al. The autism brain imaging data exchange: Towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular Psychiatry*, 19(6):659–667, June 2014. ISSN 1359-4184.

Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26 (3):297–302, 1945.

Jeffrey A Fessler. Optimization methods for mr image reconstruction (long version). *arXiv preprint arXiv:1903.03510*, 2019.

M. A. T. Figueiredo and R. D. Nowak. An em algorithm for wavelet-based image restoration. *IEEE TIP*, 12(8):906–916, Aug 2003.

Bruce Fischl. Freesurfer. *NeuroImage*, 62(2):774–781, 2012.

Peter I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Sairam Geethanath, Rashmi Reddy Pabhati Reddy, Amaresha Shridhar Konar, Shaikh Imam, Rajagopalan Sundaresan, Ramesh Babu, and Ramesh Venkatesan. Compressed sensing mri: a review. *Critical reviews in biomedical engineering*, 41 3:183–204, 2013.

Vahid Ghodrati, Jiaxin Shao, Mark Bydder, Ziwu Zhou, Wotao Yin, Kim-Lien Nguyen, Yingli Yang, and Peng Hu. Mr image reconstruction using deep learning: evaluation of network structure and loss functions. *Quantitative Imaging in Medicine and Surgery*, 9 (9), 2019.

David Ha, Andrew Dai, and Quoc V. Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. *arXiv preprint arXiv:1810.03982*, 2019.

Andreas Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, Jun 2016.

Andrew Hoopes, Malte Hoffmann, Bruce Fischl, John Guttag, and Adrian V. Dalca. Hypermorph: Amortized hyperparameter learning for image registration. *IPMI*, 2021.

Y. Hu and M. Jacob. Higher degree total variation (hdtv) regularization for image recovery. *IEEE TIP*, 21(5):2559–2571, May 2012.

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. doi: 10.1109/TIP.2017.2713099.

Eunhee Kang, Junhong Min, and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *Medical Physics*, 44 (10):e360–e375, 2017. doi: https://doi.org/10.1002/mp.12344. URL https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.12344.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

Sylwester Klocek, Lukasz Maziarka, Maciej Wolczyk, Jacek Tabor, Jakub Nowak, and Marek Śmieja. Hypernetwork functional image representation. *Lecture Notes in Computer Science*, page 496–510, 2019. ISSN 1611-3349.

David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville. Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*, 2018.

Xi Lin, Zhiyuan Yang, Qingfu Zhang, and Sam Kwong. Controllable pareto multi-task learning. *arXiv preprint arXiv:2010.06313*, 2021.

Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S. Kamilov. Image restoration using total variation regularized deep image prior. *arXiv preprint arXiv:1810.12864*, 2018.

Jonathan Lorraine and David Duvenaud. Stochastic hyperparameter optimization through hypernetworks. *arXiv preprint arXiv:1802.09419*, 2018.

Michael Lustig, David Donoho, and John M. Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. *arXiv preprint arXiv:2106.04489*, 2021.

Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing, 2019. URL https://arxiv.org/abs/1912.10557.

Arghya Pal and Yogesh Rathi. A review of deep learning methods for mri reconstruction. *arXiv preprint arXiv:2109.08618*, 2021.

Zheyi Pan, Yuxuan Liang, Junbo Zhang, Xiuwen Yi, Yong Yu, and Yu Zheng. Hyperst-net: Hypernetworks for spatio-temporal forecasting. *arXiv preprint arXiv:1809.10889*, 2018.

Saiprasad Ravishankar and Yoram Bresler. Mr image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE Transactions on Medical Imaging*, 30 (5):1028–1041, 2011.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *arXiv preprint arXiv:1505.04597*, 2015.

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

Pranjal Sahu, Hailiang Huang, Wei Zhao, and Hong Qin. Interactive smoothing parameter optimization in dbt reconstruction using deep learning. In Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert, editors, *MICCAI*, pages 57–67. Springer, 2021.

Falong Shen, Shuicheng Yan, and Gang Zeng. Meta networks for neural style transfer. *arXiv preprint arXiv:1709.04111*, 2017.

Rui Shu, Hung H. Bui, Shengjia Zhao, Mykel J. Kochenderfer, and Stefano Ermon. Amortized inference regularization. *arXiv preprint arXiv:1805.08913*, 2018.

Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248. Springer, 2017.

Kenya Ukai, Takashi Matsubara, and Kuniaki Uehara. Hypernetwork-based implicit posterior estimation and model averaging of cnn. In Jun Zhu and Ichiro Takeuchi, editors, *Proceedings of The 10th Asian Conference on Machine Learning*, volume 95 of *Proceedings of Machine Learning Research*, pages 176–191. PMLR, 14–16 Nov 2018.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *International Journal of Computer Vision*, 128(7):1867–1888, Mar 2020. ISSN 1573-1405.

Alan Q. Wang, Adrian V. Dalca, and Mert R. Sabuncu. Neural network-based reconstruction in compressed sensing mri without fully-sampled training data. In *Machine Learning for Medical Image Reconstruction*, pages 27–37, Cham, 2020. Springer.

Alan Q. Wang, Adrian V. Dalca, and Mert R. Sabuncu. Hyperrecon: Regularization-agnostic cs-mri reconstruction with hypernetworks. In *Machine Learning for Medical Image Reconstruction*, pages 3–13, Cham, 2021a. Springer.

Alan Q. Wang, Aaron K. LaViolette, Leo Moon, Chris Xu, and Mert R. Sabuncu. Joint optimization of hadamard sensing and reconstruction in compressed sensing fluorescence microscopy. In Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 129–139, Cham, 2021b. Springer International Publishing. ISBN 978-3-030-87231-1.

Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.*, 13(4):600–612, April 2004. ISSN 1057-7149.

Doris Xin, Litian Ma, Jialin Liu, Stephen Macke, Shuchen Song, and Aditya Parameswaran. Accelerating human-in-the-loop machine learning: Challenges and opportunities. In *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning*. Association for Computing Machinery, 2018.

Nan Ye, Farbod Roosta-Khorasani, and Tiangang Cui. Optimization methods for inverse problems. *MATRIX Book Series*, page 121–140, 2019. ISSN 2523-305X.

Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J. Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, Nafissa Yakubova, James Pinkerton, Duo Wang, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastmri: An open dataset and benchmarks for accelerated mri. *arXiv preprint arXiv:1811.08839*, 2018.

Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph hypernetworks for neural architecture search. *arXiv preprint arXiv:1810.05749*, 2019.

Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for neural networks for image processing. *arXiv preprint arXiv:1511.08861*, 2018.
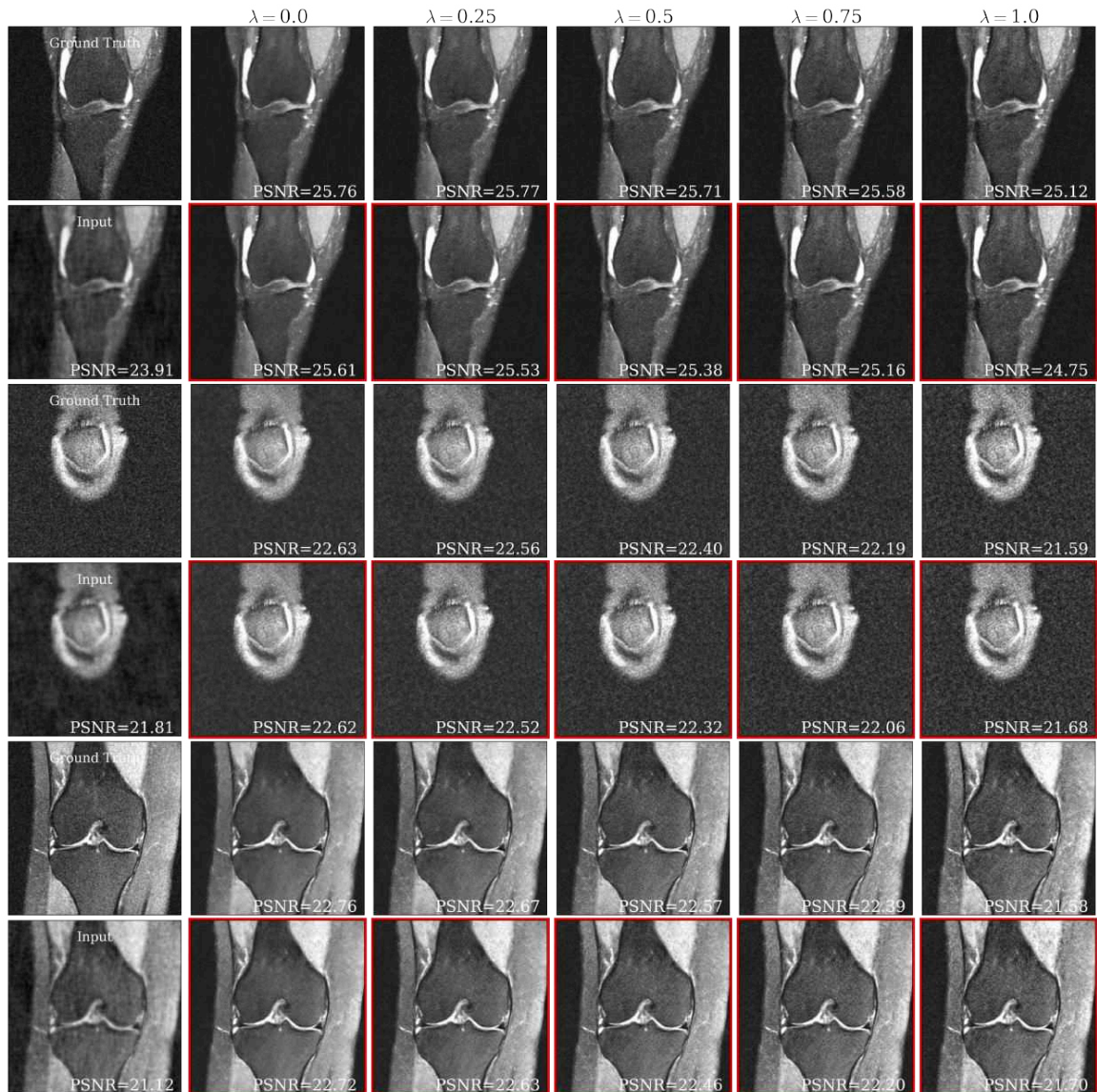
Appendix

Figure 8: Representative slices for knee on CS-MRI task with 8-fold under-sampling. Ground truth and input images are denoted in the first column. Top row are baseline Unet reconstructions with varying λ. Bottom row (red) are HyperUnet-L reconstructions with varying λ. PSNR values in reconstructions provide a basis of comparison between Unet and HyperUnet-L reconstructions of the same λ.
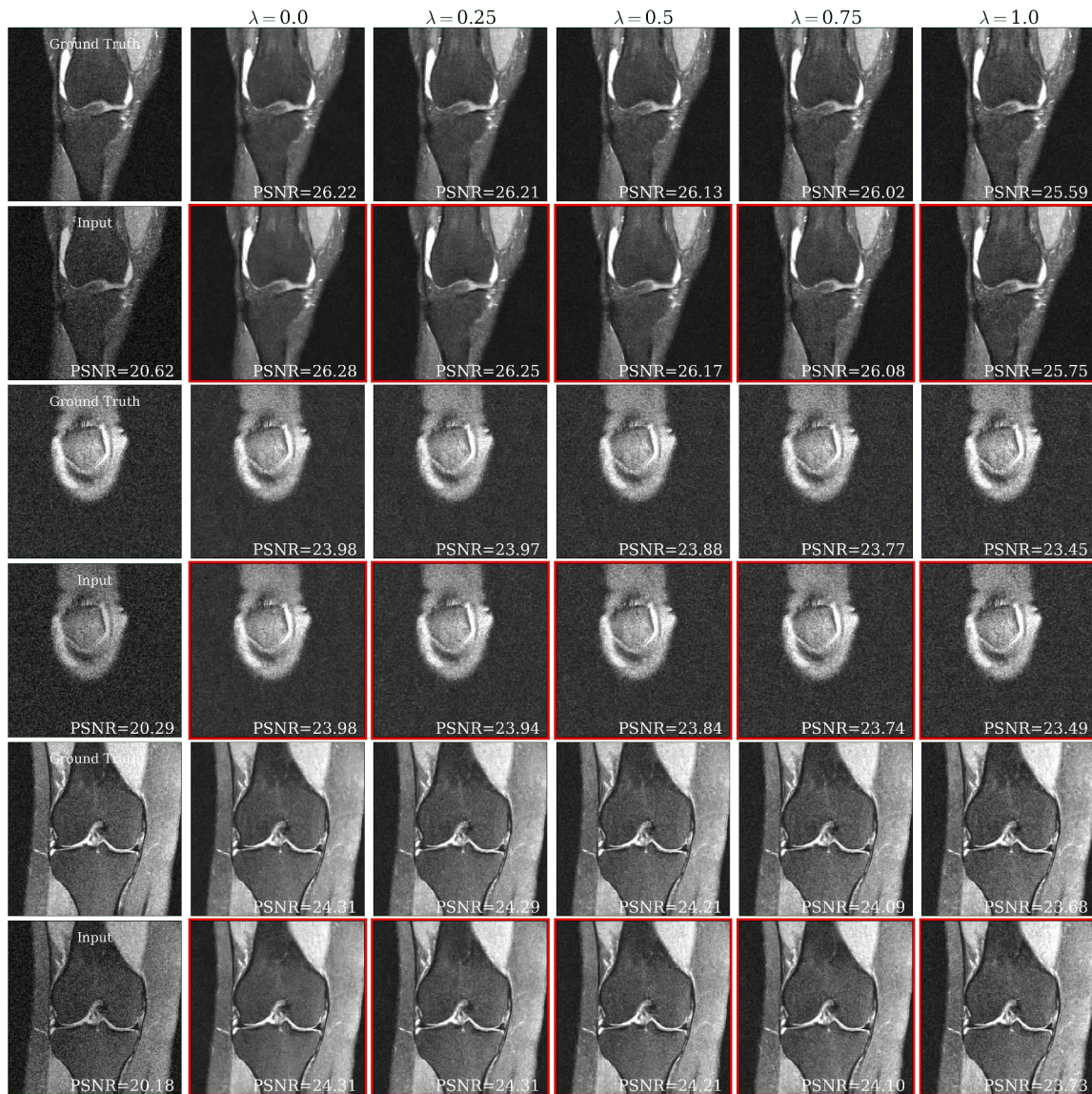
Figure 9: Representative slices for knee on AWGN denoising task with noise standard deviation $\sigma = 0.1$. Ground truth and input images are denoted in the first column. Top row are baseline Unet reconstructions with varying λ. Bottom row (red) are HyperUnet reconstructions with varying λ. PSNR values in reconstructions provide a basis of comparison between Unet and HyperUnet reconstructions of the same λ.
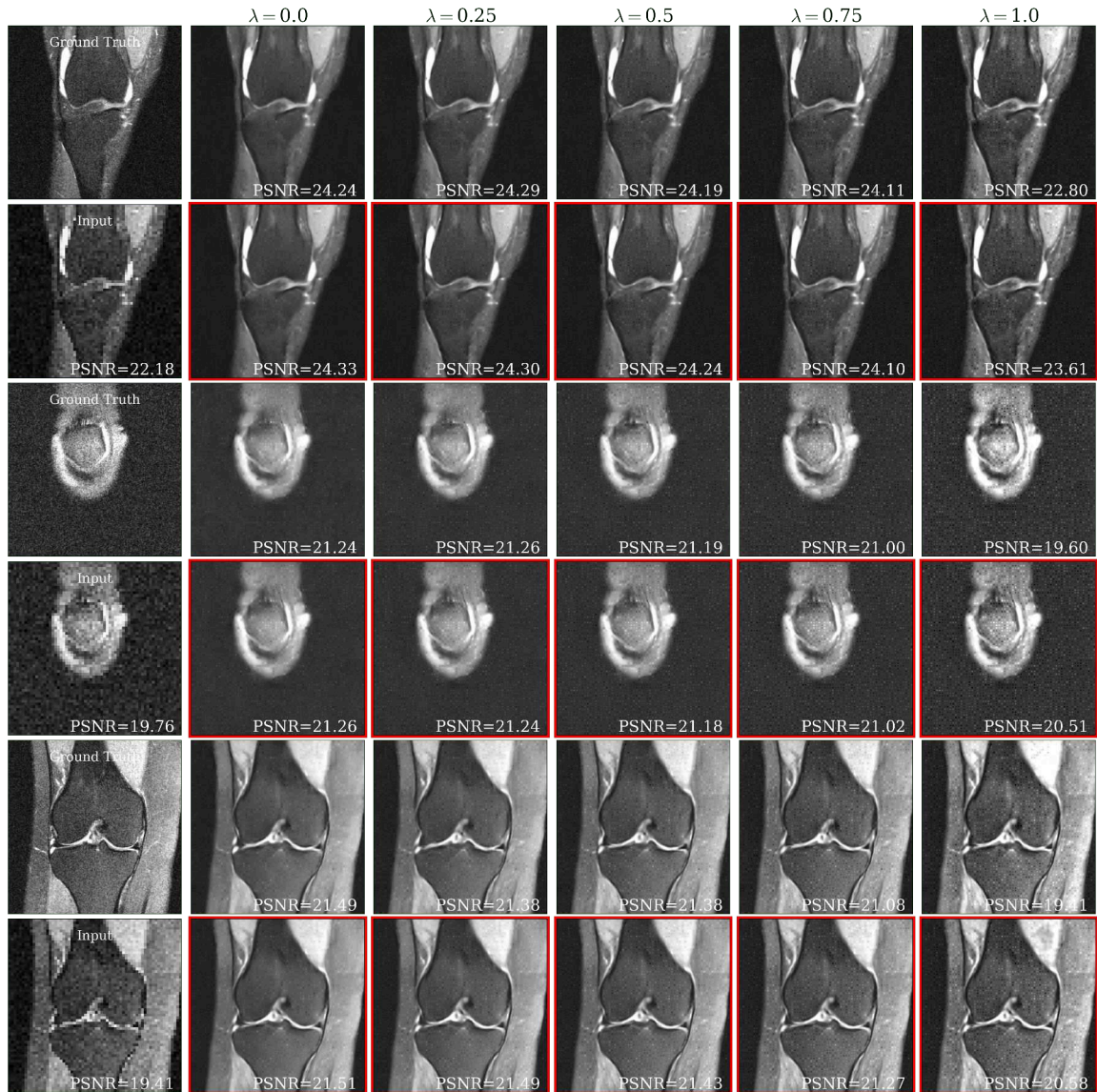
Figure 10: Representative slices for knee on superresolution task at 4x downsampling. Ground truth and input images are denoted in the first column. Top row are baseline Unet reconstructions with varying λ. Bottom row (red) are HyperUnet-L reconstructions with varying λ. PSNR values in reconstructions provide a basis of comparison between Unet and HyperUnet-L reconstructions of the same λ.
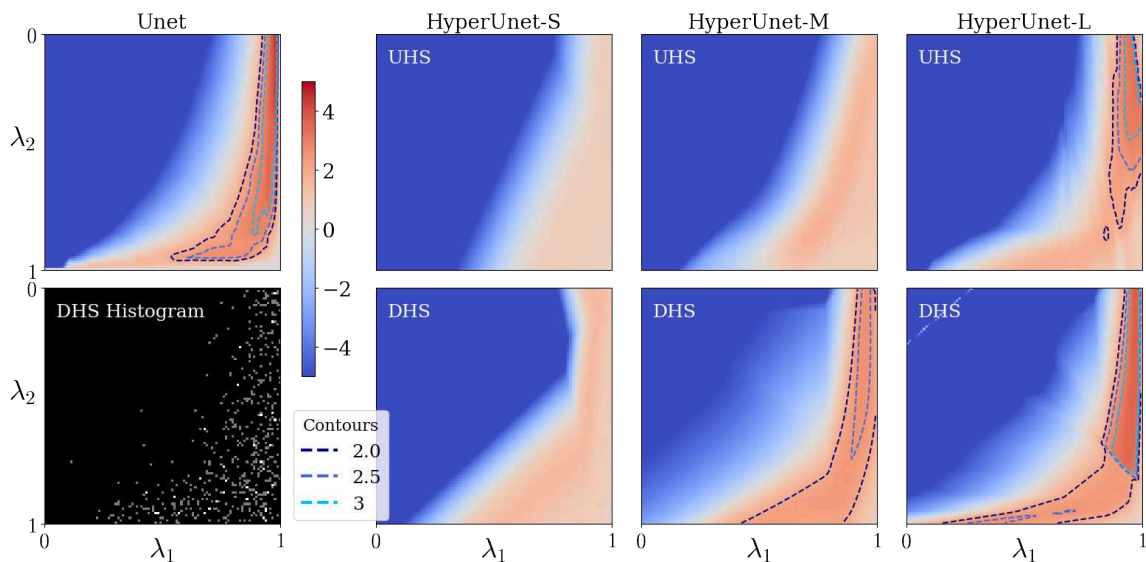
Figure 11: rPSNR values for brain dataset over the hyperparameter support $[0,1] \times [0,1]$ for different hypernetwork capacity and sampling methods. The x-axis and y-axis denote the value of the hyperparameters λ_1 and λ_2, respectively. Contours denote level sets of fixed value (see legend). Stars in the landscapes denote maximum value. (Left) The top image depicts the baseline landscape. The bottom image shows an example histogram of hyperparameter values used for gradient computation during one epoch of training with the DHS strategy. (Right) The top and bottom row show the UHS and DHS model landscapes, respectively, for three different hypernetwork capacities.