Martin Schwarzbach

Image Enhancement in ASL Perfusion Imaging
From Markov Random Fields to Variational Networks

Master Thesis

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Biomedical Engineering

submitted to

Graz University of Technology

Institute of Medical Engineering

Supervisors:

Dipl.-Ing. Stefan Spann
Univ.-Prof. Dipl.-Ing. Dr.techn. Rudolf Stollberger

Graz, June 2020

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Graz, am ....
(Unterschrift)

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the used sources. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

....
(date) (signature)

Danksagung

Zunächst möchte ich Herrn Professor Rudolf Stollberger für die Möglichkeit danken, dieses interessante Thema am Institut für Medizintechnik zu bearbeiten.

Weiters möchte ich mich bei Stefan Spann für die zahlreichen Gespräche und die hervorragende Unterstützung bedanken.

Bedanken möchte ich mich natürlich auch bei der restlichen Arbeitsgruppe für die schöne Büroatmosphäre und die netten Kaffeerunden.

Für die gesamte Studienzeit möchte ich meinen Wegbegleitern und Kommilitonen, insbesondere Felix Steiner danken.

Der größte Dank gilt natürlich meiner Familie: Danke an meine Eltern, die mir das Studium nicht nur in finanzieller Hinsicht überhaupt erst ermöglicht haben. Danke an meinen Bruder Wolfgang für all die Kraft tankenden Ausflüge gemeinsam. Danke an meine Lebensgefährtin Judith, die mich über die letzten Jahre hinweg immer unterstützt hat. Zu guter Letzt danke ich meiner Großmutter, die mich mein gesamtes Leben geprägt hat und mir immer zur Seite stand. Leider kann sie diesen Abschluss nicht mehr miterleben.

Abstract

Arterial spin labeling (ASL) perfusion imaging is a non-invasive technique capable of measuring the cerebral blood flow. Due to its poor signal-to-noise ratio, an efficient denoising method is required. Recently proposed methods like spatio-temporal total generalized variation (stTGV) improve the image quality but utilize full optimization procedures and thus are slow in inference. In contrast, deep learning (DL) based methods are fast in inference, but need much data and time for training. The aim of this thesis is to implement a co-sparse analysis model (CSM) and a variational network (VN) for ASL denoising to avoid both limitations.
A CSM uses learned filter kernels and applies a penalty function to the response of the filter. The result of this procedure forms a regularization term. In combination with a data fidelity term an "energy" is obtained which is topic to minimize w.r.t. an input image. In the framework of VNs, full optimization is replaced by an unrolled gradient descent scheme with a fixed number of steps. By using learnable penalty functions and a single parameter set for each descent step, a highly expressive and efficient model is obtained. Both models were trained with ASL data from 6 subjects and compared to stTGV on a quantitative and visual basis.
Although both models showed very good denoising performance, the VN outperformed the CSM. Despite visual differences, the VN and the stTGV performed on par in terms of structural similarity. However, the VN was about 50 times faster in denoising than stTGV. Further, the training of the VN lasts only 15 minutes.
This thesis highlighted the efficient ASL denoising capability of the VN. Its fast training and the ability to deal with few data makes the VN highly suited for more advanced applications in the field of arterial spin labeling.
Keywords: Magnetic Resonance Imaging, Arterial Spin Labeling, Image Denoising, Co-Sparse Analysis Model, Variational Network

Kurzfassung

Arterial Spin Labeling (ASL) Perfusionsbildgebung ist eine nicht-invasive Technik zur Quantifizierung des zerebralen Blutflusses, die durch einen sehr kleinen Signal-Rausch-Abstand gekennzeichnet ist. Kürzlich publizierte Methoden wie spatio-temporal Total Generalized Variation (stTGV) oder Deep Learning (DL) konnten die Bildqualität zwar deutlich verbessern, sind allerdings rechenintensiv bzw. benötigen viel Zeit und viele Daten für das Training. Das Ziel dieser Arbeit ist es beide Nachteile durch die Adaptierung eines Co-Sparse Analysis Models (CSM) sowie eines Variational Networks (VN) zu vermeiden.
Ein CSM verwendet lernbare Filter und penalisiert die Filterantwort mit Hilfe einer vorgegebenen Funktion. Dieser Regularisierungsterm in Kombination mit einem Datenterm bildet eine "Energie", welche in Bezug auf ein Eingangsbild minimiert wird. Diese vollständige Minimierung wird bei VNs umgangen, d.h. nach einer festen Anzahl an Gradientenschritten abgebrochen. Erlernbare Penalisierungsfunktionen sowie unterschiedliche Parametersätze für jeden Abstiegsschritt führen zu einem expressiven und effizienten Modell. Beide Methoden wurden mit ASL Daten von 6 Probanden trainiert und quantitativ sowie visuell mit stTGV verglichen.
Beide Modelle konnten die Bildqualität deutlich erhöhen, wobei das VN bessere Resultate erzielte. Trotz visueller Unterschiede erzielten VN und stTGV gleichwertige strukturelle Ähnlichkeits Indizes. Weiters ist das VN rund 50 mal schneller in der Bildverarbeitung als stTGV und benötigte nur 15 Minuten für das Training. Diese Arbeit unterstreicht die Fähigkeit des VNs zur effizienten Bildverbesserung. Aufgrund der wenig benötigten Daten und des kurzen Trainings ist das VN besonders für herausfordernde Aufgaben im Bereich des Arterial Spin Labelings geeignet.
Keywords: Magnetresonanztomographie, Arterial Spin Labeling, Bildverbesserung, Co-Sparse Analysis Model, Variational Network

Contents

List of Figures

List of Tables

List of Acronyms

ASL arterial spin labeling. ii, 1–3, 15, 16, 18, 28, 30, 32, 35, 43, 45, 49, 90, 92, 97

BOLD blood oxygenation level dependent. 1

CASL continuous ASL. 1, 2
CBF cerebral blood flow. iii–v, vii, viii, 1, 15–17, 44, 45, 52–61, 64, 67–71, 73–75, 78–80, 92, 95, 97
CD contrastive divergence. 11
CG conjugate gradient. 11, 42, 88, 89
CPU central processing unit. 89
CSF cerebrospinal fluid. 16
CSM co-sparse analysis model. iii–v, viii, 10, 35, 39, 40, 43, 47, 48, 51, 61, 65–77, 79, 81, 82, 85–89, 91–95, 97
csRoot center-smoothed root. 47, 64

DCE dynamic contrast enhanced. 1
DCT discrete cosine transform. ii, 10, 28, 30, 32, 35, 39
DSC dynamic susceptibility contrast. 1

EPI echo planar imaging. 15

fMRI functional magnetic resonance imaging. 1
FoE field of experts. 2, 9, 10, 12, 97
FOV field-of-view. 15, 16

GBCA gadolinium-based contrast agent. 1
GL generalized Laplace distribution. 7
GM gray matter. iii, iv, 16, 53, 61, 62, 65, 71, 91

GPU graphics processing unit. 12, 88, 95

HLP higher level problem. 11, 42, 43, 88

iPALM inertial proximal alternating linearized minimization. 3, 48

L/C label/control. i–v, vii, viii, 16, 22–25, 27–29, 32–34, 45, 46, 51, 52, 54–57, 59–78, 90, 94, 97, 98
L1msSSIM L1 with msSSIM. 64
L1SSIM L1 with SSIM. 64
LLP lower level problem. 11, 12, 36, 37, 39–42, 46, 60, 87, 88, 92

MAP maximum a posteriori. 5, 11, 35
MCMC markov chain Monte Carlo. 11
MH metropolis hastings. 11
MR magnetic resonance. 1, 15
MRF markov random field. 7–12, 94
MRI magnetic resonance imaging. 1, 18
MSE mean squared error. 43, 45, 91, 93, 97
msSSIM multiscale SSIM. 2, 44, 48, 63, 64, 67, 97

NSF nephrogenic systematic fibrosis. 1

PASL pulsed ASL. 2, 15, 18, 26, 51
pCASL pseudo-continuous ASL. 2, 90, 98
PDF probability density function. i, 18–20, 22–26, 28, 29, 31
PoE product of experts. 6, 7, 9
PSNR peak signal-to-noise ratio. iii–v, viii, 42–45, 52–59, 61–63, 66–78, 92, 93, 95, 97
PV partial volume. 16
PWI perfusion weighted image. i–iv, 1, 17, 18, 23–25, 27–32, 34, 52, 65, 66, 71, 86, 87, 95

RBF radial basis function. 47, 86

SciPy ScientificPython. 46
SGD Stochastic Gradient Descent. 41

SNR signal-to-noise ratio. 2, 27, 53, 90, 91

SSIM structural similarity. iii–v, viii, 2, 3, 42–44, 46–48, 52–78, 82, 88, 91–93, 95, 97

ST Student-t distribution. 7

stTGV spatio-temporal total generalized variation. 2, 3, 46, 48, 65, 76, 77, 79, 93, 94, 97, 98

TGV total generalized variation. v, viii, 76–78, 92, 93, 97

TV total variation. v, 7, 8, 10, 51, 79, 80

VN variational network. iii–v, viii, 2, 3, 12, 46, 47, 64–79, 81, 83, 85, 86, 88, 91–95, 97, 98

VU variational unit. 13

WB whole brain. iii, iv, 53, 65, 71

WM white matter. iii, iv, 16, 53, 56, 61, 62, 65, 71, 91

1. Introduction

Magnetic resonance (MR) perfusion weighted imaging summarizes several acquisition techniques which are capable of measuring signals proportional to the CBF. This is achieved by employing the impact of an intravascular, extracellular or diffusible tracer on the tissue magnetization. The gained information is used in clinics for the diagnosis and localization of diseases resulting in a pathologic cerebral blood flow (f.e. strokes and tumors) as well as for scientific research.
In principle, three different kinds of methods are distinguished: dynamic susceptibility contrast (DSC) perfusion magnetic resonance imaging (MRI), dynamic contrast enhanced (DCE) perfusion MRI and ASL. The first two techniques have the use of an intravenous injected gadolinium-based contrast agent (GBCA) in common. In contrast, ASL uses magnetically labeled bloodwater as an endogenous tracer. Some GBCAs are related to the development of nephrogenic systematic fibrosis (NSF) in patients with renal insufficiency [1]. Also gadolinium deposition in brain and body have been observed. In addition, qualified personal is needed for tracer injection. These properties make DSC and DCE less suited for research. The non-invasive and safe character of ASL overcomes these issues and thus is considered as an appropriate technique especially for research like f.e. functional magnetic resonance imaging (fMRI), where it is advantageous over blood oxygenation level dependent (BOLD) contrast imaging in the sense of directly measuring the perfusion.
In ASL, the PWI is obtained by the difference of an unaffected 'control' image and a 'label' image. The essential part of the ASL pulse sequences is the labeling of arterial blood water outside the region of interest. The first labeling technique, continuous ASL (CASL), was proposed by Williams et al. in 1992 [2]. Over the next decade the methods EPISTAR [3, 4], FAIR [5], PICORE [6], PULSAR [7]

and QUASAR [8] formed the new class of pulsed ASL (PASL) techniques. The main difference between these methods and CASL is the use of one adiabatic pulse for labeling the arterial blood in a broad area instead of continuous RF-irradiation which results in relevant tissue heatening. The latest technique, namely pseudo-continuous ASL (pCASL) was proposed by Dai et al. [9] in 2008 and is considered as hybrid form between CASL and PASL. Once labeled, the images can be acquired using 2D or 3D [10] readouts. In general, 3D methods yield a higher signal-to-noise ratio (SNR) but are less robust against motion. Nevertheless, depending on the labeling method the magnetisation is increased or decreased just slightly (1%-2%) which results in a very poor SNR. In order to increase the SNR and hence the image quality, several images are obtained and averaged. The resulting long scanning times are clinically not acceptable and additionally lead to movement artifacts.

To enhance the image quality and reduce the scanning time, ASL has been topic for many denoising techniques: anisotropic difusion filtering [11], adaptive wiener filtering [11], iterative soft thresholding [12], wavelet domain filtering [13], 3D block matching [14], spatio-temporal low rank total variation [15], spatio-temporal total generalized variation (stTGV) [16], deep learning methods [17, 18, 19, 20] and others [21, 22].

A very effective convential technique is given by stTGV denoising. However, this method needs a manual parameter tuning for different SNR cases and, due to full optimization, long inference (i.e. denoising) times. Neural network based approaches avoid long denoising times but have the need for large datasets (f.e. [19] used data from 20 subjects for learning, [20] 240 subjects) and long training times (f.e [19] trained 12h for low resolution images). In addition, all proposed learning based models exploit a squared L2 based loss, which is known to favour blurry solutions more then sharp ones [23]. On the other hand, the possible solution of using a perceptual based loss like the SSIM [24] or the multiscale SSIM (msSSIM) [25] lead to a very complex and non-convex energy function, which is likely to dramatically increase the computational costs.

The aim of this work is to tackle the reported problems by firstly use learnable filter operators in the framework of field of experts (FoE) [26, 27] and secondly by adapting a VN [28, 29] to the certain characteristics of high resolution 2D PASL data. The FoE method will help to overcome costly parameter sweeps like

needed for stTGV. The VN approach will reduce the image processing time to the millisecond range, which would increase the usability of the ASL denoisng procedure greatly. Further, the VN comes with a powerful optimization strategy (inertial proximal alternating linearized minimization (iPALM) [30]) which might be capable of performing efficient SSIM optimization.

1.1. Theory

Low level computer vision tasks like denoising, inpainting or non-blind deconvolution can be formulated as an energy based inverse problem. Eq. 1.1 describes the noise-free forward problem in terms of a system matrix $\mathbf{S}$, data $\hat{\mathbf{y}}$ and ground truth model parameters $\hat{\mathbf{x}}$.

$$\hat{\mathbf{y}} = \mathbf{S}\hat{\mathbf{x}} \tag{1.1}$$

If the problem is well-posed, $\hat{\mathbf{x}}$ is obtained by computing the inverse of $\mathbf{S}$. Unfortunately, the measured data is almost always subject to noise $\mathbf{n}$ (eq. 1.2)

$$\mathbf{y} = \mathbf{S}\hat{\mathbf{x}} + \mathbf{n} \tag{1.2}$$

and the system matrix might not be positive definite. In such cases, the ground truth $\hat{\mathbf{x}}$ is not obtainable any more and therefore an estimated solution $\mathbf{x}^*$ is computed as the minimizer of a suitable energy function (eq. 1.3).

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} E(\mathbf{S}, \mathbf{x}, \mathbf{y}) \tag{1.3}$$

The most simple form of an energy models only the distance between the measured (noisy) data $\mathbf{y}$ and the forward mapping $\mathbf{Sx}$. For instance, if a squared L2 norm is used to measure this difference, the well known least-square solution is obtained. Depending on the task and on the SNR, this distance measure, which is often called data fidelity term $D(\mathbf{S}, \mathbf{x}, \mathbf{y})$, does not yield acceptable results. Therefore, it is extended with a so called regularization term $R(\mathbf{x})$ (eq. 1.4).

$$E(\mathbf{S}, \mathbf{x}, \mathbf{y}) = R(\mathbf{x}) + D(\mathbf{S}, \mathbf{x}, \mathbf{y}) \tag{1.4}$$

The exponential of the negative energy is proportional to a Boltzmann distribution and, therefore, is proportional to a probability. If $R(\mathbf{x})$ and $D(\mathbf{S}, \mathbf{x}, \mathbf{y})$ are proportional the negative logarithm of the prior probability $P(\mathbf{x})$ and likelihood

$P(\mathbf{y}|\mathbf{x})$, than $e^{-E(\mathbf{S},\mathbf{x},\mathbf{y})}$ is proportional to the a posteriori probability (eq. 1.5).

$$e^{-E(\mathbf{S},\mathbf{x},\mathbf{y})} = e^{-R(\mathbf{x})} \cdot e^{-D(\mathbf{S},\mathbf{x},\mathbf{y})} \propto \frac{P(\mathbf{x}) \cdot P(\mathbf{y}|\mathbf{x})}{P(\mathbf{y})} = P(\mathbf{x}|\mathbf{y}) \tag{1.5}$$

Minimizing the energy $E(\mathbf{S}, \mathbf{x}, \mathbf{y})$ is equivalent to maximizing the joint probability $P(\mathbf{x}, \mathbf{y})$ and because the optimization is independent to the evidence $P(\mathbf{y})$, the whole procedure is equivalent to maximizing the posterior probability $P(\mathbf{x}|\mathbf{y})$ (eq. 1.6).

$$\max_{\mathbf{x}} \frac{P(\mathbf{x})P(\mathbf{y}|\mathbf{x})}{P(\mathbf{y})} = \max_{x} P(\mathbf{x})P(\mathbf{y}|\mathbf{x}) \tag{1.6}$$

Therefore, if the regularization term and the data term are modeling the corresponding probabilities, minimizing the energy of an inverse problem is equivalent to finding the maximum a posteriori (MAP) solution in a Bayesian framework.

The Modeling of the Data Term

The most important question to this point is how to model the parts of the energy function. The data fidelity term has to capture the imperfection of the data acquisition. If the error distribution is known, one can easily infer a suitable distance metric by maximizing the logarithmic likelihood function. As an example, eq. 1.7 shows the relation between spatial independent Gaussian noise and the L2 norm.

$$\begin{aligned} \max_{\mathbf{x}} \log P(\mathbf{y}|\mathbf{x}) &= \max_{\mathbf{x}} \log \left(\prod_{n=1}^{N_p} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_n - x_n)^2}{2\sigma^2}} \right) \\ &= \min_{\mathbf{x}} \sum_{n=1}^{N_p} \frac{1}{2}(y_n - x_n)^2 = \min_{\mathbf{x}} \frac{1}{2}||\mathbf{y} - \mathbf{x}||_2^2 \end{aligned} \tag{1.7}$$

Data $\mathbf{y}$ and estimate $\mathbf{x}$ can be considered as vectorized images with N_p pixels. Another well known data term function is the L1 norm. It maximizes the log-likelihood in case of Laplace distributed noise and leads to sharper solutions in

general. In case of a very complex error distribution, it might be beneficial to use a general formulated data term like a mixture of radial basis functions, where the parameters are learned from data. However, the central limit theorem is valid for many problems and thus the squared L2 norm is an appropriate choice. Nevertheless, for reasons of generality D is modeled in terms of an arbitrary distance measure

$$D(\mathbf{S}, \mathbf{x}, \mathbf{y}) = \psi(\mathbf{S}, \mathbf{x}, \mathbf{y}). \tag{1.8}$$

The Modeling of the Data Prior

In contrast to the data term, the choice of data prior is more critical and thus many different regularization terms have been proposed. The definition of the prior over the whole image as a N_p dimensional distribution, is not only computational infeasible in case of learning but also restricts the prior to a certain image size. To avoid these problems and to exploit the self-similarity of images, the majority of different variants of image priors have in common that they are defined over image patches.

Product of Experts

The product of experts (PoE) model formulated by Hinton et al. [31] provides a specific filter-based approach for modeling the prior distribution of (vectorized) image patches $\mathbf{u} \in \mathbb{R}^N$. In the framework of PoE, the prior probability $p(\mathbf{u})$ is written as formulated in eq. 1.9 with E_{PoE} being the energy of the model, $\mathbf{\Theta}$ model parameters, $Z(\mathbf{\Theta})$ the normalization, N_f the number of filter - expert pairs, $\mathbf{a}_i \in \mathbb{R}^N$ the i-th (vectorized) filter kernel and ρ_i the i-th expert function that aims to model the probability distribution of the i-th filter response.

$$p(\mathbf{u}) = \frac{1}{Z(\mathbf{\Theta})} e^{-E_{PoE}(\mathbf{u}, \mathbf{\Theta})} \quad \text{with} \quad E_{PoE} = -\sum_{i=1}^{N_f} \log \rho_i(\mathbf{a}_i^T \mathbf{u}) \tag{1.9}$$

The filter response of natural images are typically heavy-tailed distributed [32], therefore the Student-t distribution (ST) and generalized Laplace distribution (GL) are widely used expert functions. The scalar product $\mathbf{a}_i^T\mathbf{u}$ can be considered as filtering the image patch $\mathbf{u}$ with the filter $\mathbf{a}_i$. Further, it can be considered as projecting $\mathbf{u}$ on the basis vector $\mathbf{a}_i$. Assuming the matrix $\mathbf{A} \in \mathbb{R}^{N_f xN}$ is composed by stacking the N_f filters $\mathbf{a}_i^T$, then $\mathbf{Au}$ describes the linear transformation of $\mathbf{u}$ by the transformation matrix $\mathbf{A}$. As the heavy-tailed expert functions favor sparse filter responses, the model can be considered to transform the image patch onto a sparse feature space, where deviations from the expectation can be detected and penalized more reliable. For sake of completeness, instead of modeling the expert function it is common to directly model the penalty function $\phi_i = -\log \rho_i$. The link between expert function ρ_i and penalty ϕ_i can be seen in eq. 1.10 where α_i and β_i are some parametrization constants.

$$\begin{cases} \rho_i(\mathbf{u}) = (1+\mathbf{u}^2)^{-\alpha_i} & \Leftrightarrow \quad \phi_i(\mathbf{u}) = \alpha_i \log(1+\mathbf{u}^2) & \text{ST} \\ \rho_i(\mathbf{u}) = e^{-|\mathbf{u}|^{\beta_i}} & \Leftrightarrow \quad \phi_i(\mathbf{u}) = |\mathbf{u}|^{\beta_i} & \text{GL} \end{cases} \tag{1.10}$$

In case of setting $\beta_i = i$, $\forall i$ the penalty represents the L1 norm of the filter response. This is particularly interesting when building a connection from PoE to TV priors.

Markov Random Fields

A markov random field (MRF) is an undirected graphical model $G(V, E)$ with nodes V and edges E that fulfill the local Markov property. The nodes represent random variables and typically refer to image properties like intensity values, surface normals or optical flow estimations. The structure of the MRF enables a factorization of the probability distribution $p(V = \mathbf{X})$ by employing the maximal cliques of the graph defined by eq. 1.11. A clique is an undirected subgraph where every two distinct nodes are connected to each other. It is said to be maximal if no node could be added without violating the previous definition.

$$p(\mathbf{X}) = \frac{1}{Z(\mathbf{\Theta})} \prod_k^K f_k(\mathbf{X}_k) \tag{1.11}$$

Here, K defines the number of maximal cliques $\mathbf{X}_k$ of an image $\mathbf{X}$, f_k is the factorization function, Z is the normalization and $\boldsymbol{\theta}$ is the entire set of parameters. In analogy to the PoE, the above equation can also be written in terms of an energy

$$p(\mathbf{X}) = \frac{1}{Z(\boldsymbol{\Theta})} e^{-\sum_k^K q_k(\mathbf{X}_k)} \tag{1.12}$$

where q_k corresponds to the factorization function f_k and is termed potential function or clique potential. In order to establish translation invariance, the same potential function q_k is used for each clique. In this case the MRF is called homogeneous.
In general, the potential function is written in terms of a penalty function ϕ, whose argument is robust scalar mapping defined on the nodes of the cliques. Eq. 1.13 shows exemplarily how a clique potential $q(\mathbf{X}_k)$ could look in case of a pairwise MRF, where just the direct non-diagonal neighbors are connected.

$$q(\mathbf{X}_k) = q(x_i, x_j) = \phi(x_i - x_j) \tag{1.13}$$

The scalar mapping used above can be interpreted in the simplest form as a gradient estimation, i.e. forward differences. If the penalty ϕ is set to be the absolute function, the TV prior is obtained (eq. 1.14).

$$E_{pwMRF} = \sum_k^K q(\mathbf{X}_k) = \sum_{(i,j)=(1,1)}^{(N_x-1,N_y-1)} |x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}| = ||\nabla \mathbf{X}||_1 \tag{1.14}$$

Unfortunately, low-order MRF do not lead to satisfying results. For example, the TV prior introduces so called staircasing artifacts because it favors piecewise constant solutions. To obtain a more powerful model, higher order MRFs can be established by forming larger maximal cliques (larger neighborhoods). Among others, Geman and Reynolds [33] did so by using polynomial functions for scalar mapping. Roth and Black [26] reformulated this mapping by filtering the nodes of a (vectorized) clique $\mathbf{x}_k$ with suitable (vectorized) filters $\mathbf{a}_i$. Thus, each clique $\mathbf{x}_k$

is centered at pixel k and its size is determined indirectly via the filter size.

$$q(\mathbf{x}_k) = \sum_{i=1}^{N_f} \phi(\mathbf{a}_i^T \mathbf{x}_k) \tag{1.15}$$

Note that the higher order MRF clique potential defined in eq. 1.15 is equivalent to the PoE model if the image patch $\mathbf{u}$ is considered as a maximal clique. Therefore, Roth and Black termed this combination of MRF and PoE a field of experts (FoE).

Field of Experts

The field of experts (FoE) overcomes the problem of the PoE of just being defined over small image patches by incorporating the PoE in the framework of MRFs (eq. 1.16).

$$p(\mathbf{X}) = \frac{1}{Z(\mathbf{\Theta})} e^{-E_{FoE}(\mathbf{X},\mathbf{\Theta})} \quad \text{with} \quad E_{FoE} = -\sum_{k=1}^{N_p} \sum_{i=1}^{N_f} \log \rho_i((\mathbf{A}_i * \mathbf{X})_k) \tag{1.16}$$

In contrast to the PoE (eq. 1.9), in the FoE (eq 1.16) the filter $\mathbf{A}_i$ is applied to the whole image $\mathbf{X}$ instead to a image patch $\mathbf{U}$. As a consequence, a second sum over the number of pixels N_p is introduced which adds up the penalized filter responses for each center pixel k. This formulation instantly models the overlapping of different image patches and thus avoids any patch averaging procedures. In the following sections the FoE energy will also be written by using the filter operator $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N_p N_f \times N_p}$, where ϕ^{Σ} in combination with the latter also comprises the required integration steps. This leads to the following, more compact formulation.

$$R(\mathbf{x}) = E_{FoE}(\mathbf{x}) = \phi^{\Sigma}(\boldsymbol{\mathcal{A}}\mathbf{x}) \tag{1.17}$$

The Co-Sparse Analysis Model

Combining eq. 1.8 and eq. 1.17 yields the overall energy of the first model, the so-called CSM.

$$\min_{x} \quad \phi^{\Sigma}(\boldsymbol{\mathcal{A}}\mathbf{x}) + \psi(\mathbf{S}, \mathbf{x}, \mathbf{y}) \tag{1.18}$$

Assuming $\boldsymbol{\mathcal{A}}$ is invertible and defining $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{A}}^{-1}$ and $\mathbf{x} = \boldsymbol{\mathcal{D}}\boldsymbol{\chi}$ one can also write

$$\min_{\chi} \quad \phi^{\Sigma}(\boldsymbol{\chi}) + \psi(\boldsymbol{\mathcal{D}}\boldsymbol{\chi}, \mathbf{y}, \mathbf{S}), \tag{1.19}$$

which gives the so-called sparse synthesis model. The idea behind this model is to use a dictionary $\boldsymbol{\mathcal{D}}$ and synthesize the solution from dictionary atoms. As the penalty function promotes sparsity in the solution space, just a few atoms are needed for synthesis. The dictionary $\boldsymbol{\mathcal{D}}$ can be learned from data or set to be a rich transformation basis like DCT atoms or wavelets.

Inference and Learning in Markov Random Fields

Many parameters like the maximal clique, the neighboring structure, the filters and the penalty function exist and are often chosen by hand. The TV model, for instance, can be interpreted as hand-tuned MRF model. However, in the last 15 years, learning approaches have gained more and more attention. Therefore, it is natural to consider optimizing the parameters of the FoE model with an appropriate algorithm. Basically, there are two different strategies to learn the MRF parameters. The first approach tries to minimize the difference between the model distribution and data distribution. Hence, if sampling from the model distribution the data distribution should be obtained. This procedure can be considered as minimizing the Kullback-Leibler divergence between the model and data distribution and is also equivalent to maximizing the likelihood of the given training data $\mathbf{X}$. The gradient of the log-likelihood (eq. 1.20) can be used to employ any first order maximization method.

$$\frac{\partial \log L_X}{\partial \Theta_i} = \left\langle \frac{\partial E_{FoE}}{\partial \Theta_i} \right\rangle_p - \left\langle \frac{\partial E_{FoE}}{\partial \Theta_i} \right\rangle_X \tag{1.20}$$

Here, $\langle . \rangle_p$ and $\langle . \rangle_X$ denote the expectation value of parameter Θ_i for the model and data distribution. The data distribution is obtained by simply building the average over the data $\mathbf{X}$. The exact computation of the model distribution is not possible, because the partition function has to be evaluated over all possible energy configurations which is computationally not traceable. Therefore, suitable sampling strategy like markov chain Monte Carlo (MCMC) or metropolis hastings (MH) are employed to estimate the model distribution. However, to obtain a very accurate estimate of the model distribution, many sampling steps would be necessary, leading to a very slow learning procedure. Fortunately, Hinton et al. [34] found a very efficient way to overcome this issue. They suggest to initialize the MCMC with the training data and updating the chain just a few times. This concept is called contrastive divergence (CD) and works even for one single update. This probabilistic learning scheme represents a generative model and thus once learned, it can be used for any kind of problem. Nevertheless, for a specific task a discriminatively learned prior often leads to better results. Therefore, Samuel and Tappen [35] proposed a loss-specific learning approach that directly optimizes the MRF prior for MAP inference. This is achieved by minimizing the loss $L(\mathbf{x}^*(\boldsymbol{\Theta}), \hat{\mathbf{x}})$ which measures the distance between the ground truth $\hat{\mathbf{x}}$ and the minimizer of the inference scheme $\mathbf{x}^*$. This is formally written as bilevel optimization scheme (eq. 1.21) where the inference represents the lower level problem (LLP) and the loss minimization the higher level problem (HLP).

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \quad & L(\mathbf{x}^*(\boldsymbol{\Theta}), \hat{\mathbf{x}}) \\ & \mathbf{x}^* = \arg\min_{\mathbf{x}} \quad E_{FoE}(\mathbf{x}, \boldsymbol{\Theta}) + \psi(\mathbf{S}, \mathbf{x}, \mathbf{y}) \end{aligned} \tag{1.21}$$

The drawback of the bilevel approach is that the LLP has to be solved with a very high accuracy [36], which is particularly hard if the penalty function is also learnable. Therefore, the penalty function is typically chosen a-priori. However, the LLP is non-convex in general and thus hard to solve anyway, even if powerful optimization methods like conjugate gradient (CG) [35] or L-BFGS [36] are used. A different learning approach was performed by Barbu [37] and Domke [38] by

using truncated optimization. This active random field called technique stops the LLP optimization after a fixed, rather low number of iterations, leading to a fast but suboptimal inference. The lack for accurate inference is compensated by a prior, that is optimized not only for a certain inference scheme, but also for the exact number of inference steps. Nevertheless, solving the LLP with a very high accuracy still leads to better results than using truncated optimization [27].

Variational Networks

The proposed learning schemes for MRF-FoE are feasible, but still not very efficient. Other image restoration algorithms like BM3D [14] attend also good performance, but are difficult to incorporate into a parallelized GPU based training. Chen et al. [28] tackled this problem by formulating a VN. The VN's basic concept is similar to those of active random fields, but with the difference of learning a separate prior for each gradient descent step. In combination with learnable penalty functions this leads to a highly expressive model that gains very good performance. More formally, the reconstructed image $\mathbf{x}^*$ in the VN framework is written as:

$$\mathbf{x}^* = \mathbf{x}^T, \quad \mathbf{x}^{t+1} = \mathbf{x}^t - \frac{\partial}{\partial \mathbf{x}^t} \left[\sum_{k=1}^{N_p} \sum_{i=1}^{N_f} \phi_i^t \left(\left(\boldsymbol{\mathcal{A}}_i^t \mathbf{x}^t \right)_k \right) \; + \; \lambda^t \psi \left(\mathbf{S}\mathbf{x}^t - \mathbf{y} \right) \right], \quad 0 \leq t \leq T-1 \tag{1.22}$$

where $\boldsymbol{\mathcal{A}}_i^t$ and ϕ_i^t denote the i-th filter-penalty pair for the t-th gradient descent step. λ^t denotes the t-th non-negative data term weight. N_f refers to the number of filter-penalty pairs and T to the number of gradient descent steps. By computing the derivatives and rearranging some variables, eq. 1.23 is obtained.

$$\frac{\mathbf{x}^{t+1} - \mathbf{x}^t}{\Delta t} = - \sum_{i=1}^{N_f} \boldsymbol{\mathcal{A}}_i^{t\top} \phi_i^{t\prime} \left(\boldsymbol{\mathcal{A}}_i^t \mathbf{x}^t \right) \; - \; \lambda^t \frac{\partial}{\partial \mathbf{x}^t} \psi \left(\mathbf{S}\mathbf{x}^t - \mathbf{y} \right) \quad \text{with} \;\; \Delta t = 1 \tag{1.23}$$

This formulation is closely related to the well-known Perona-Malik model for anisotropic diffusion [39], with the first part being the so-called diffusion term and the second part being the reaction term. The models differ in the formulation

of the filter operator, which is composed of a horizontal and vertical gradient filter in case of the Perona-Malik model and N_f principally unconstrained learnable filters in case of the variational network. A second difference is given by the reaction term, which is not part of the original anisotropic diffusion model.
Further, Kobler et al. [40] showed the relation between VNs and deep learning. A single diffusion step, in this context also called variational unit (VU), can be interpreted as residual unit, which is the building block of residual neural networks. The central idea behind residual networks is to utilize short-cut links to skip certain layers, leading to efficiently learnable networks with up to 1000 layers. The short-cut link between input $\mathbf{x}^t$ and output $\mathbf{x}^{t+1}$ establishes a residue function (eq. 1.24) g_t which is typically formulated using filter kernels and thus is closely related to eq. 1.23.

$$\mathbf{x}^{t+1} - \mathbf{x}^t = g_t(\mathbf{x}^t) = \sum_{i=1}^{N_f} \boldsymbol{\mathcal{A}}_i^{t2} a\left(\boldsymbol{\mathcal{A}}_i^{t1} \mathbf{x}^t\right) \tag{1.24}$$

$\boldsymbol{\mathcal{A}}_i^{t1}$ and $\boldsymbol{\mathcal{A}}_i^{t2}$ refer to filter operators and $a(.)$ to the activation function, which is often set to be the well-known rectified linear function (rectified linear units - ReLU). The data term can be incorporated in the framework of residual neural networks by using a second residual mapping, yielding so-called multi residual units. It is obvious that the residual function estimates the gradient of the current image $\mathbf{x}^t$. Hence, variational networks combine the profound theoretical background of variational models and the efficiency of neural networks.

2. Methods

2.1. Data Acquisition and Preprocessing

The ASL data used throughout this work was acquired and preprocessed in context with the publication of Spann et al. [16]. The following chapters summarize the data recording and preprocessing steps performed in the context of the above publication.

2.1.1. Data Acquisition

ASL measurements were performed on ten healthy subjects (24-28 years, 4 women) after giving written informed consent (caffeine and tobacco consumption were avoided before the MR experiment). The latter is reasoned in alternations on the global and regional CBF caused by the mentioned substances. ([41], [42], [43]) Label and control images were acquired using a 3T MR system (Magnetom Skyra, Siemens Healthcare, Germany) performing pulsed ASL (PASL) measurements (PICORE [6] - Q2TIPS [44]) with a 32-channel head coil. For the reduction of motion artifacts small foam blocks were used to fixate the head of the subjects. In more detail: 12 slices with an in-plane resolution of 1.8x1.8 mm^2 (128x128 matrix, 230x230 mm^2 field-of-view (FOV)) and 3.6 mm thickness (distance factor 25%), 6/8 partial Fourier, GRAPPA-factor 2 and pre-scan normalize. Single-shot echo planar imaging (EPI) with TR/TE = 2800/19 ms was used for imaging with a flip angle of 90°, bolus duration TI_1 = 800 ms, labeling inversion time TI_2 = 1800 ms, labeling slab thickness of 100 mm (20 mm gap between slab and image slice), ascending slice order and a bandwidth of 1630 Hz/px. To estimate a noise-free ground truth

500 L/C-pairs and a proton density weighted image (M0) were acquired in about 45 min.

2.1.2. Tissue Masks

WM, GM and cerebrospinal fluid (CSF) masks were computed from the acquired T1w image. The anatomical T1 weighted images were measured using a 3D MPRAGE sequence with the following imaging parameters: 1 mm isotrop, FOV=256x256 mm^2, 144 slices, flip angle of 8° and TR/TE/TI = 1910/1.81/1000 ms). The segmentation was performed using the Statistical Parameter Mapping 12 toolbox (SPM12, Wellcome Trust Centre for Neuroimaging, London Uk, www.fil.ion.ucl.ac.uk/spm). The generated partial volume (PV) content maps were registered to the first ASL image and thresholded to obtain the corresponding tissue masks (threshold = 0.5). In addition, a whole brain mask was computed by summing up the PV-content maps and thresholding at 0.1.

2.1.3. Preprocessing

The ASL data was preprocessed as recommended in [45] using the SPM12 ASL toolbox [46, 47] and in-house MATLAB scripts. In a first step ASL data was motion corrected and de-trended using a Butterworth high-pass of 1st order with a cutoff frequency of 0.01 Hz [47, 48] followed by discarding the first and the last slice of each volume. Further, residual motion artifacts and global signal fluctuations were removed from the label-control time series and outlier L/C-pairs were discarded by performing z-score thresholding [49].

2.1.4. CBF Quantification

For the conversion of perfusion weighted images to CBF maps the general kinetic model of Buxton et al. [50] was exploited (eq. 2.1).

$$\mathbf{f}(x,y,z) = \frac{6 \cdot 10^6 \cdot \lambda \cdot \Delta\mathbf{M}(x,y,z)}{2\alpha \cdot \mathbf{M}_0(x,y,z) \cdot TI_1 \cdot e^{-\frac{TI_2}{T1_b}}} \tag{2.1}$$

In eq. 2.1 above $\Delta\mathbf{M}(x,y,z)$ denotes the PWI, TI_1 the labeling duration (800 ms), TI_2 the total delay time (1800 ms for the first slice, 80 ms added for each subsequent slice [51]), $T1_b$ the longitudinal relaxation time of blood at 3T (1650 ms [52]), α the labeling efficiency (0.98 [53]) and λ the blood-brain partition coefficient (0.9 ml/g [54]). The resulting CBF values $\mathbf{f}(x,y,z)$ are given in ml/100g/min. The proton density weighted image (M0) was smoothed with a Gaussian filter (FWHM=3 mm [51] to reduce the impact of noise.

2.2. Data Analysis

In the following sections different aspects regarding the preprocessed 2D ASL data are analyzed. The obtained results are important to gain a profound understanding of the characteristics of 2D PASL data. The differences to natural images (BSDS300, [55]), which are corrupted with spatial independent Gaussian noise, are from particular interest, because almost all denoising models are designed and tested for such cases. The knowledge gained from this analysis will be used for selecting theoretically reasonable hyperparameters for the two denoising approaches.

2.2.1. Temporal Distribution of Perfusion Weighted Images

Real and imaginary part of MRI voxels suffer from Gaussian noise, which causes the voxels of the corresponding magnitude images to be Rice distributed. Although the use of multiple coils in combination with GRAPPA transforms the noise distribution approximately to a non-central χ distribution [56], for reasons of interpretability this section focuses on the Rician distribution. Eq. 2.2 states the Rician probability density function for an arbitrary positive $x \in \mathbb{R}_+$ with ν being the truth value, σ the scaling factor and $\mathbf{I}_0$ the modified Bessel function of first kind and zero order.

$$\mathcal{R}_{pdf}(x|\nu,\sigma) = \frac{x}{\sigma^2} e^{-\frac{x^2+\nu^2}{2\sigma^2}} \mathbf{I}_0\left(\frac{x\nu}{\sigma^2}\right) \tag{2.2}$$

The truth value directly corresponds to the voxel intensity, i.e each voxel intensity suffers from its own error distribution. This makes the modeling of the error distribution particular difficult. Fortunately, the PWI is obtained from the difference of two very similar Rician distributed images. The theoretic PDF of a random variable being the difference of two Rice distributed random variables, is stated in eq. 2.3 where (μ_C,σ_C) and (μ_L,σ_L) correspond to the parameters of the control voxel and the label voxel, respectively.

$$\mathcal{R}^{2s}_{pdf}(x|\nu_C,\nu_L,\sigma) = \begin{cases} \mathcal{R}_{pdf}(x+m|\nu_C,\sigma) & \text{for} \quad x \geq m \\ \mathcal{R}_{pdf}(x+m|\nu_C,\sigma) + \mathcal{R}_{pdf}(-(x-m)|\nu_L,\sigma)) & \text{for} \quad |x| < m \\ \mathcal{R}_{pdf}(-(x-m)|\nu_L,\sigma) & \text{for} \quad x \leq -m \end{cases} \tag{2.3}$$

Figure 2.1 shows the estimated PDF of the ratio between ground truth control and label voxels. This plot indicates that the difference is beneath 2% for the large majority of voxels. In theory the ratio should be greater than one, but due to noise and artifacts (e.g. motion) this is not the case for all voxels.

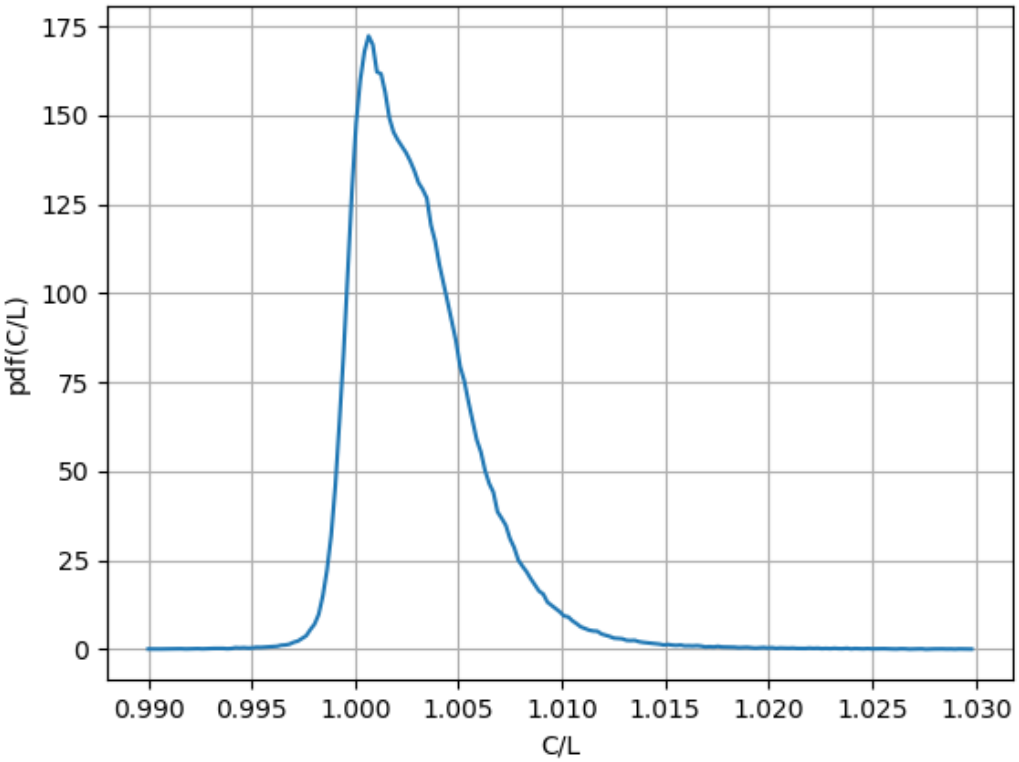

Figure 2.1.: Estimated PDF of the ratio between ground truth (400 averages) control (C) and label (L) voxels. This PDF indicates negative perfusion in some voxels (L>C), which is physically not possible and thus must be reasoned in the presence of errors in the ground truth.

Figure 2.2 shows the PDF eq. 2.3 with different parameter settings. For a small

ν (red curve) the one-sided character of the Rice distribution leads to two prominent non-smooth points left and right from the origin (red arrows). For larger ν (green curve) the non-smooth points migrate to outer regions where their impact is negligible. This behavior is to some extent independent to σ, because a larger σ will shift the single distributions even more apart from the origin. The parameter setting of the yellow dotted curve ($\nu_C = 1.1\nu_L$) is motivated by the PDF shown before (Figure 2.1) and shows that even an intensity difference of 10% has no observable effect on the corresponding 'two-sided' Rice distribution. The Gaussian PDFs depicted in this graphic show the remarkable similarity to their corresponding 'two-sided' Ricians, especially for large ν's. Note that for large ν's already the "one-sided" Rice distribution is quite Gaussian like.

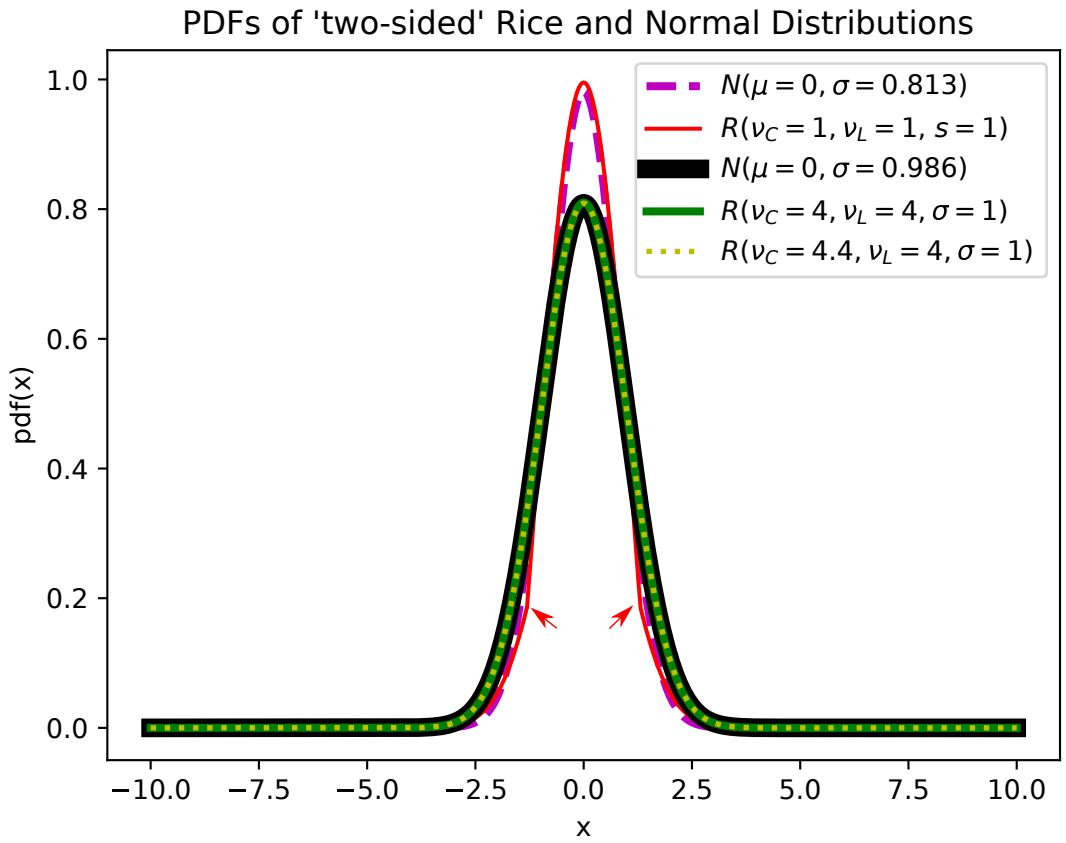

Figure 2.2.: Probability density functions of 'two-sided' Ricians and the corresponding Gausian distributions .

Statistical Testing for Normality

To verify the validity of a Gaussian approximation, a d'Agostino-Pearson test [57] was performed. The Null-Hypothesis H_0 of the voxel intensities being drawn from a normal distribution is rejected if the p-value is below a significance level of $\alpha = 0.001$. Figure 2.3 shows the ratio between rejected Null-Hypotheses and tested voxels for each slice and all subjects.

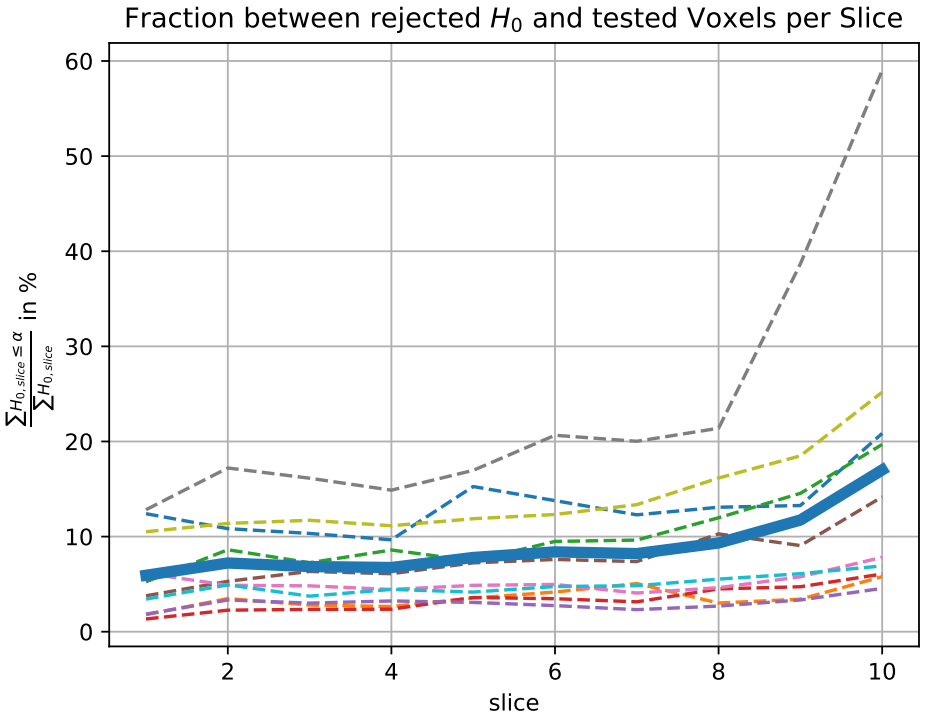

Figure 2.3.: Fraction between rejected H_0 ($\alpha = 0.001$) and tested voxels per slice for all subjects (dashed line). The bold line indecates the average over all subjects.

This Figure shows that H_0 is rejected for only a small fraction of voxels, with a slightly increase in rejections for upper slices. The prominent outlier in the last slice is founded in an acquisition artefact. The brain maps of rejected H_0 (Figure 2.4) indicate a spatial dependency of the rejections. The majority of rejected voxels are from cortical regions, where motion artefacts are more problematic. In these areas, motion will mix in-brain voxels with background voxels which result in a mean afflicted and asymmetric error distribution. Therefore, the approximation

of the temporal voxel distribution by a Gaussian can be considered valid for the majority of voxels, especially in the absence of motion.

Figure 2.4.: Brainmap (left subject VI slice 10, middle subject IX slice 8, right subject X slice 10) of rejected H_0 (black) and not rejected H_0 (gray) voxels

2.2.2. Data and Error Distribution

The estimated PDFs studied in this chapter were obtained for a certain number of L/C-pairs (N_{ave}) by repetitively (100 times) choosing N_{ave} random L/C-pairs from the acquired 400 L/C-pairs, followed by computing normalized histograms within masked regions. As the general data distribution is considered here, the data from all subjects is used without any seperation in subjects and slices. Additionally, the arithmetic mean μ, the standard deviation σ and the skewness s are calculated.

Data Distribution

Figure 2.5 depicts the estimated PDFs for a different number (N_{ave}) of L/C-pairs. For less averages the noise dominates and thus the shape of the PDF is Gaussian-like. For more averages the skewness (Table 2.1) increases and a shoulder evolves,

Table 2.1.: Estimated statistics of the data distribution for different numbers of averages N_{ave} (mean μ, standard deviation σ and skewness s).

$\mathbf{N}_{ave}$	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$	**s**
16	1.67	2.85	0.44
64	1.68	1.97	1.03
256	1.68	1.68	1.53
400	1.68	1.64	1.63

which is likely to correspond to a second modal value.

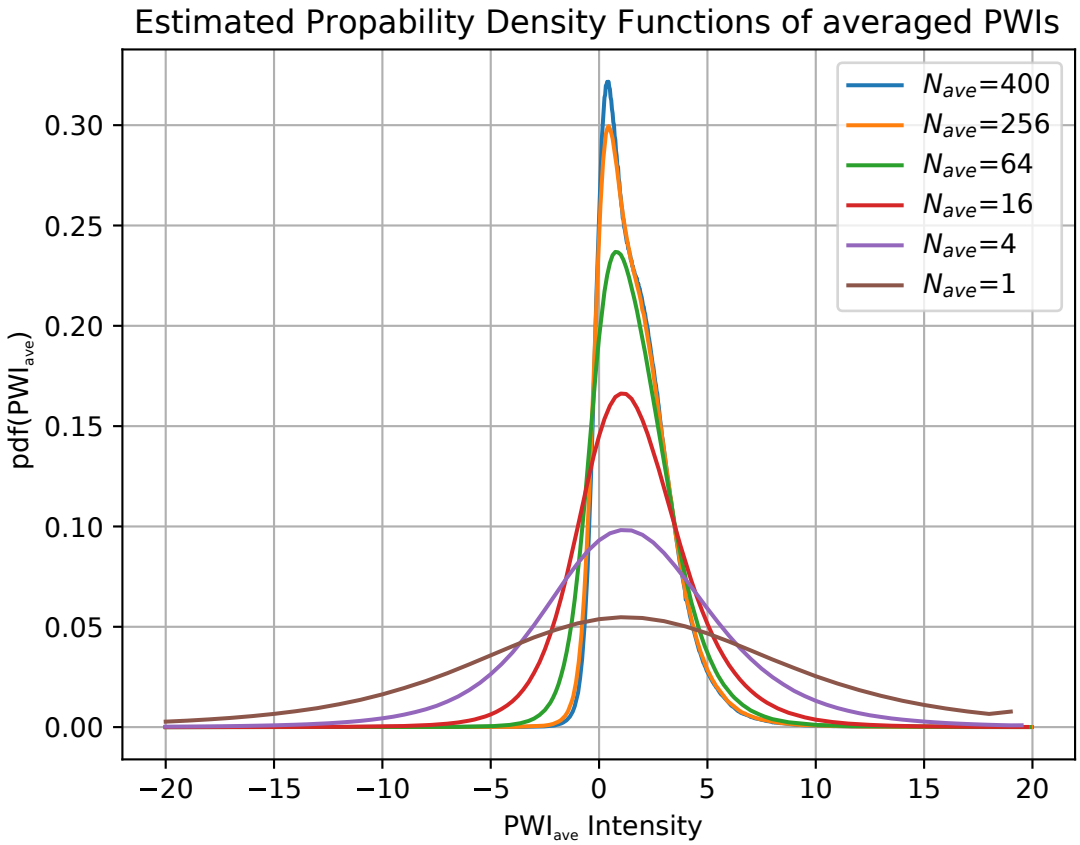

Figure 2.5.: Estimated probability density functions of the averaged PWI (PWI_{ave}) using a different number of averages N_{ave}.

In theory, two modes are expected, the first corresponds to white and the second to gray matter, respectively. Because of a high noise level, the bimodal characteristic vanishes and only a shoulder and consequently an increased skewness are observable. As expected, all curves share approximately the same mean value, which indicates an approximately mean-free error distribution. Regardless of the number of used L/C-pairs, the PDFs indicate negative perfusion, which is physically not possible and thus must be the effect of noise and artefacts (see also Figure 2.1).

Error Distribution

The PDF of the error shown in Figure 2.6 is derived by subtracting the estimated ground truth (400 L/C-pairs) from the mean PWI (N_{ave} pairs). For a higher number of L/C-pairs the standard deviation of the error decreases with about $\sqrt{N_{ave}}$. Table 2.2 confirms the validity of this relation for less N_{ave}. For more L/C-pairs, it deviats more and more, which is reasoned in the computation of the ground truth as the average of a finite number (400) of L/C-pairs, i.e. the ground truth still contains noise. This leads to an underestimation of the variance. Hence, for $N_{ave} = 400$ the error PDF would collapse to a Kronecker-Delta impulse.

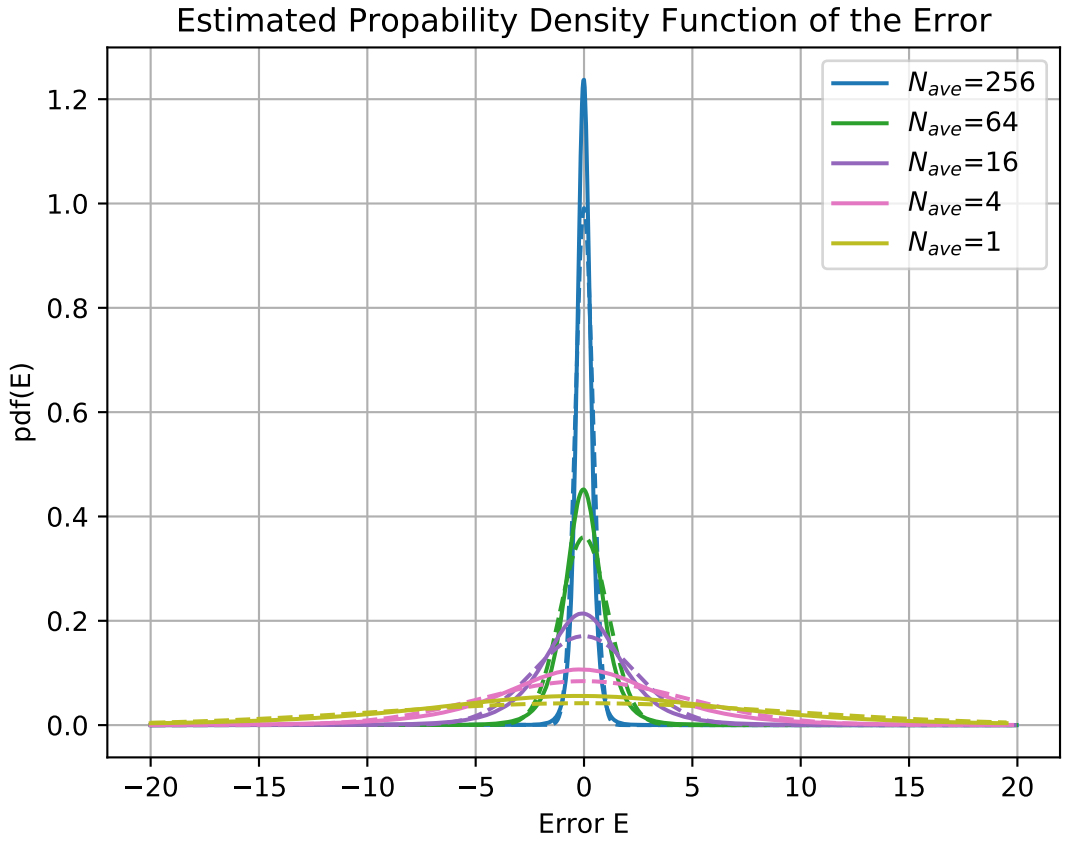

Figure 2.6.: Estimated Error distribution and the corresponding normal distributions for different number of L/C-pairs (N_{ave}). The ground truth is estimated from 400 L/C-pairs.

Although, the skewness and the mean differ significantly ($\alpha = 0.001$) from zero

(see Table 2.2, p_{mf} and p_{sym}) for some N_{ave}, both are very close to zero regardless the number of N_{ave}. This indicates an almost (0.42% bias for $N_{ave} = 16$) mean-free and symmetric distribution. Nevertheless, the dashed lines in the plot emphasize a notable deviation from normal distribution. This difference is also appearent in the p-values p_{AP} of Table 2.2 being smaller than $\alpha = 0.001$. Hence, the H_0 of the data being drawn from a normal distribution can be rejected with a significance level of $\alpha = 0.001$.

Table 2.2.: Estimated statistics and related p-values of the error distribution for different numbers of averages N_{ave} (mean μ, standard deviation σ, skewness s, test for normality p_{AP} [57], t-test for zero-mean p_{mf} and zero-skewness test p_{sym} [58]).

$\mathbf{N}_{ave}$	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$	**s**	$\mathbf{p}_{AP}$	$\mathbf{p}_{mf}$	$\mathbf{p}_{sym}$
1	2.51E-2	9.50E-0	2.82E-2	0	0	0
4	-1.77E-2	4.73E-0	4.03E-3	0	0	0
16	-7.00E-3	2.34E-0	-6.55E-3	0	0	0
64	3.21E-4	1.11E-0	9.37E-4	0	4.22E-2	7.47E-3
256	4.78E-5	4.02E-1	-2.19E-2	0	4.06E-1	0

The log PDFs depicted in Figure 2.7 visualizes the difference to logarithmized normal distributions (illustrated with dashed lines) more clearly. As Rician or χ distributions (degree of freedom ≥ 3) do not exhibit heavy-tailing (see section 2.2.1 "Temporal Distribution of Perfusion Weighted Images"), the occurence of heavy-tails in the error distribution must be reasoned in another error source. However, errors due to motion are in a mathematical sense very similar to the filter response of gradient kernels. As this response is generalized Laplace distributed, i.e. heavy-tailed, the tailing characteristic of the error's PDF could be explained by an imperfect motion correction due to distinct patient movement. But also other error sources must kept in mind.

It is remarkable that despite temporal voxel distributions is Gaussian like for about 90% of voxels (see Figure 2.3) the joint error distribution is heavy-tailed. In this context it is important to highlight that the temporal voxel distribution corresponds to the curve for $N_{ave} = 1$ L/C-pair, where the Gaussian noise fraction is very high and thus dominates the Laplacian error. For more PWI used, the fraction

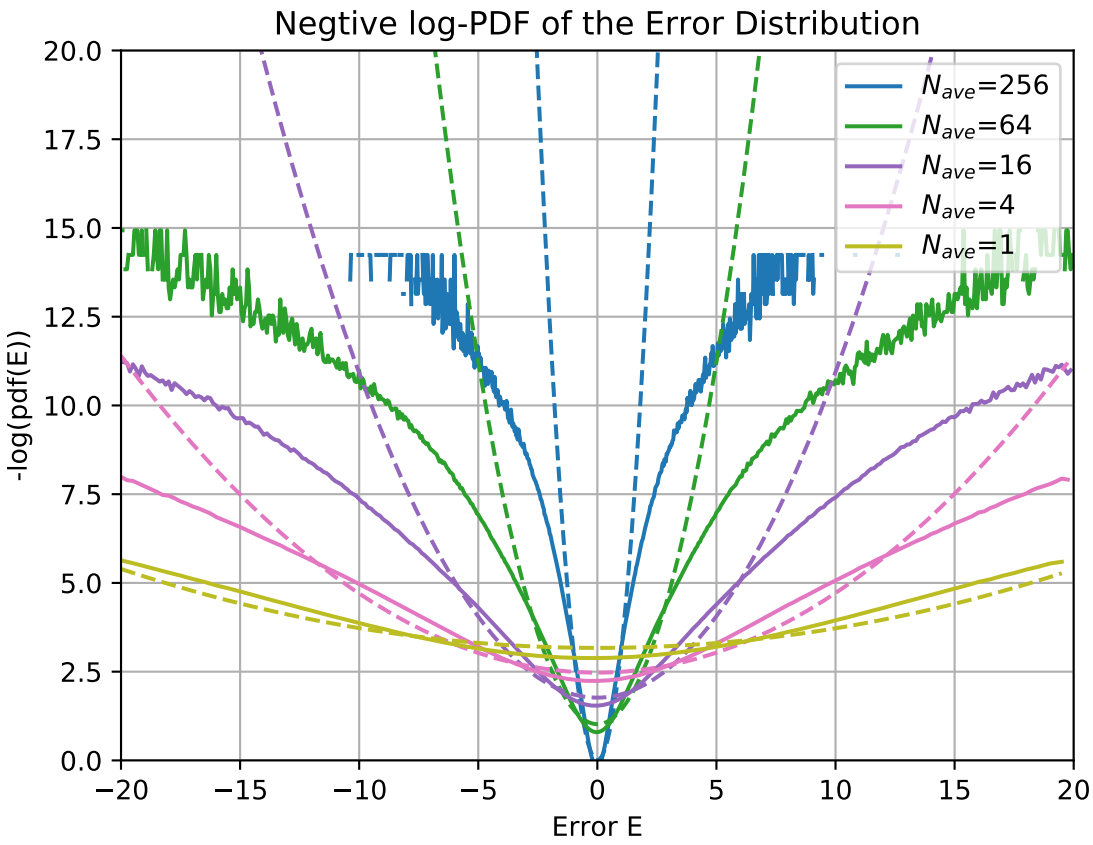

Figure 2.7.: Negative logarithm of the Error distribution and the corresponding normal distributions. The heavy-tailed character of the error PDF indicates the presence of Laplacian like noise, which is an indicator for residual motion artifacts.

of Gaussian noise decreases and the heavy-tailed generalized Laplace distribution becomes more prominent.

2.2.3. Slice Dependent Voxel Intensity and Intensity Deviation

The performed 2D PASL measurements lead to a very basic issue: As already described in the introduction section, arterial blood water is magnetically labeled. After waiting a period of time, allowing the blood to flow into the region of interest, the images are acquired. In case of 2D readouts the slices are acquired in ascending order, which leads to the issue that the magnetization of bloodwater in upper slices

is already more relaxed when read out. As a consequent, the label images' signal intensity is less decreased for upper slices which leads to less difference signal. Hence, PWIs from upper slices have less signal and less SNR, respectively.
In addtion, regions being more distant to the head coils (f.e. lower slices) are contributing less signal but undergo the same noise level. The effect of less signal is corrected by Siemens's Prescan Normalize algorithm, but this correction increases the standard deviation of the noise.

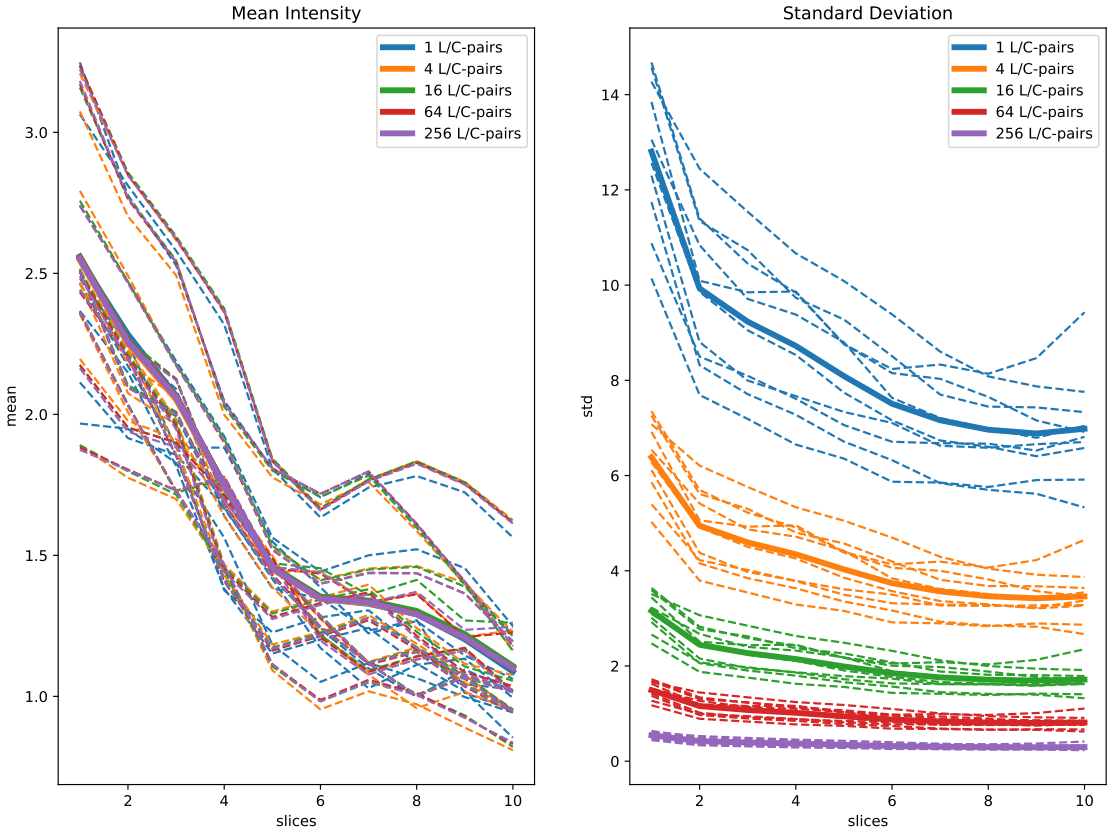

Figure 2.8.: Temporal mean and temporal standard deviation, averaged over all voxels within the masked regions of a slice. The colored dashed lines correspond to specific subjects, the colors to a specifc number of L/C-pairs and the bold lines to the average over all subjects.

Figure 2.8 depicts both phenomenons for different numbers of L/C-pairs by showing the temporal voxel mean and standard deviation, avaraged over the masked slices. The curvers corresponding to more than 1 L/C-pair are obtained by repet-

itively (400 times) building the mean over N randomly chosen PWIs, followed by computing the desired statistics. The curves in this graph show the intensity loss as well as the decrease in standard deviation for upper slices. In contrast to the intensity curves, the deviation curves depend on the number of used L/C-pairs. As expected, an increase in the number of L/C-pairs by a factor of four leads to a decrease in the standard deviation by a factor of two. This is consistent with the theory that the standard deviation is proportional to $\sqrt{N}$ with N being the number of measurements (L/C-pairs).

2.2.4. Filter Response

For the selection of an appropriate penalty function, the filter response of standard kernels applied to gold standard ASL data, is essential. As discribed in chapter 2.3.1 "Details on the Model Formulation", the filter kernels are composed as a weighted sum of the DCT basis. Consequently, in this chapter the response to DCT filters is analyzed. The distributions shown below are obtained by averaging over all filter responses corresponding to a certain DCT base. I.e. for DCT-7 PDFs all 48 non-constant filters are evaluated.
Figure 2.9 shows the negative log-probability of the DCT filter response for different image classes and different kernel sizes ks. All graphs are bias corrected and normalized by their maximum to emphasize their tailing behavior. This plot indicates several differences between ASL perfusion weight images (ASL-D, averaged over 400 L/C-pairs), ASL control images (ASL-C, averaged over 400 images) and natural images (BSDS300). First, unlike ASL-D, the shape of ASL-C is very similar to BSDS300. ASL-D is more quadratic around the origin, whereas ASL-C is more narrow and more heavy-tailed. For all image classes, larger kernels lead to a more distinct heavy-tailing. The graph of ASL-C for a kernel size ks=3 is very different to all other graphs. It is not clarified in total, but it is likely that the estimated distribution is unstable for larger responses. The normalization to a maximum value of one further distorts the shape of this graph.

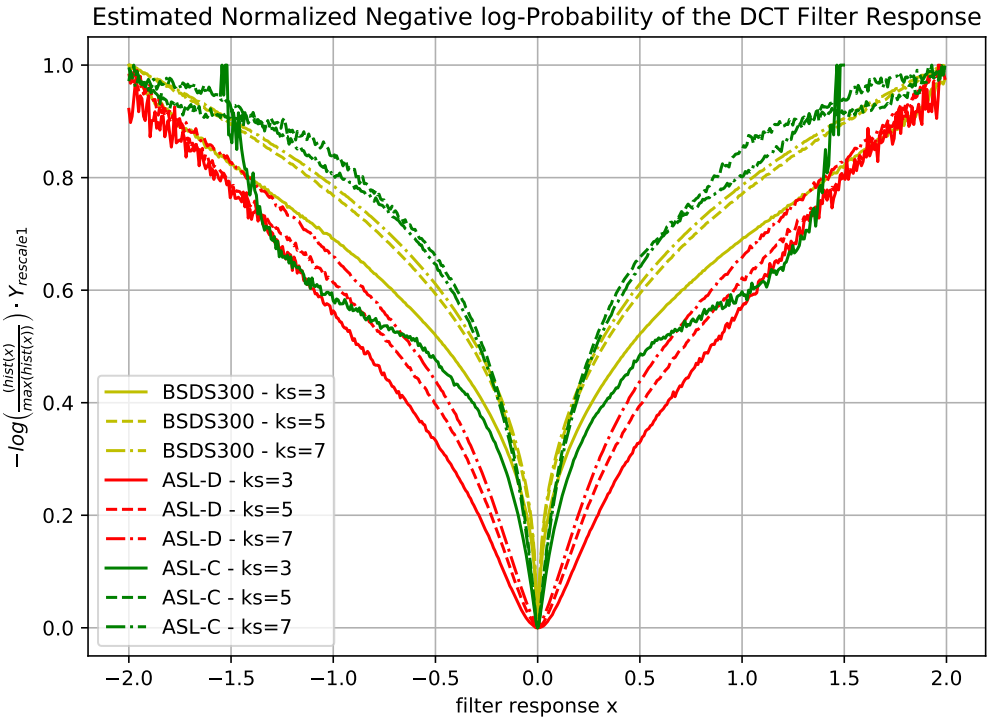

Figure 2.9.: Estimated negative log-probability of the DCT filter response for natural images (BSDS300), averaged PWIs (ASL-D, 400 L/C-pairs) and averaged control images (ASL-C, 400 images). The probability corresponds to all $ks^2 - 1$ non-constant filter kernels of the DCT-ks basis.

The distributions depicted in Figure 2.10 and Figure 2.11 uncover the noise-level dependency of the filter responses for PWI and natural images. More noise (less L/C-pairs) leads to a broadend center and to a less distinct heavy-tailing. The dashed lines indicate the distribution of the filter response to Gausian noise only. The comparison of the noise-only PDFs with the remaining PDFs leads to the assumption that the center broadening is caused by the ground truth's noise portion. As it is not clear which characteristics of the distributions are caused by noise and which are caused by information, a penalty selection based on the given ground truth is considered as inappropriate.

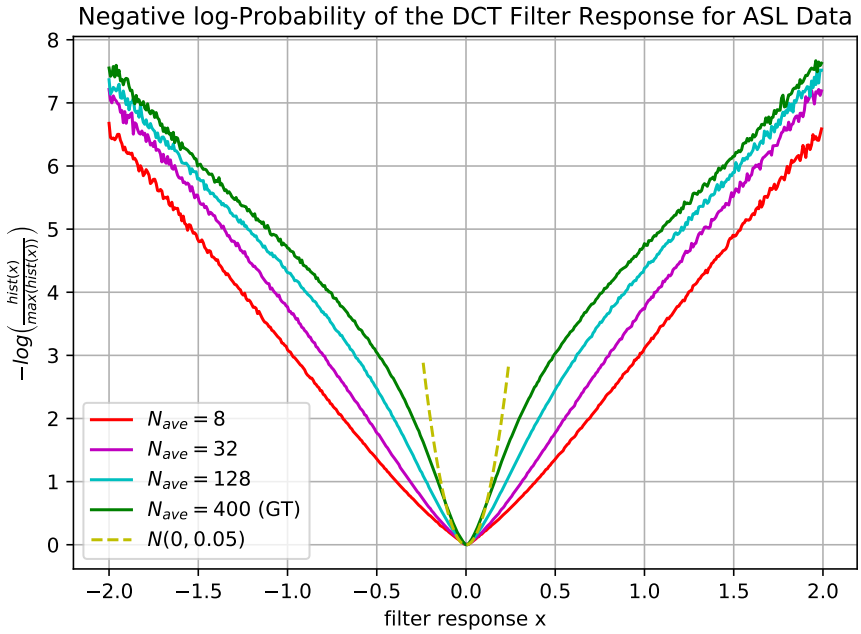

Figure 2.10.: Estimated negatavie log-probability of the ASL PWI's DCT filter response for different noise levels (number of averages N_{ave}).

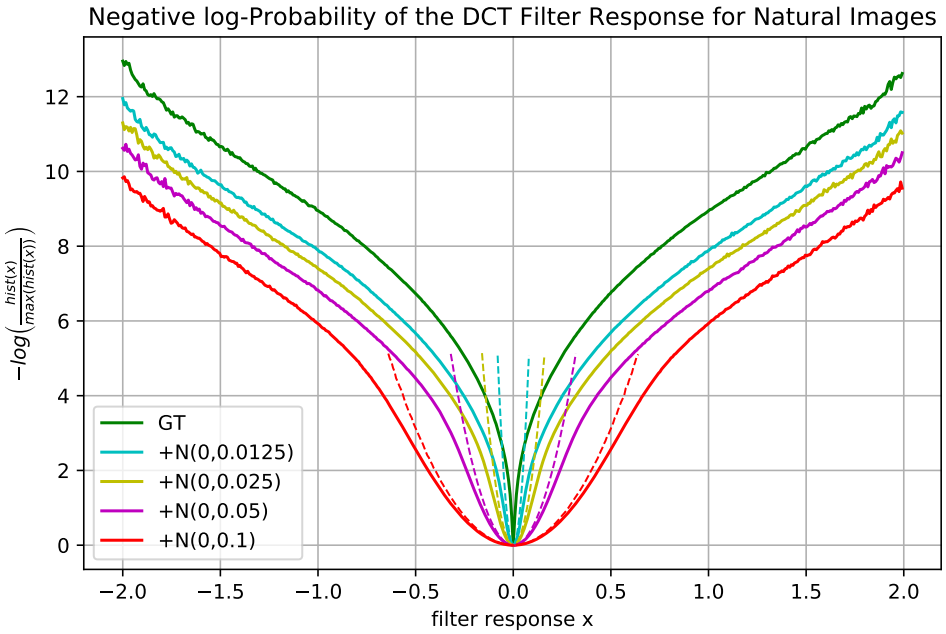

Figure 2.11.: Estimated negative log-probability of the DCT filter response of natural images for different noise levels. The dashed lines indicate the est. neg. log-P for Gaussian noise only.

An alternative interpretation of the filter response distributions is given by the data's degree of correlation. Fully correlated data, i.e. constant pixel values within an image batch, will result in a filter response of zero (mean-free filters). A Dirac impulse as PDF would be the consequence. Small deviations within the constant batches would cause a response that slightly differs from zero. The resulting PDF would be slightly broadened. In contrast, if the values of the batch are completly uncorrelated the response would be equally distributed. Hence, the broadened center of the filter response's negative log-PDF for noisy data is not directly because of the noise, but because of the decrease of correlation caused by the noise.

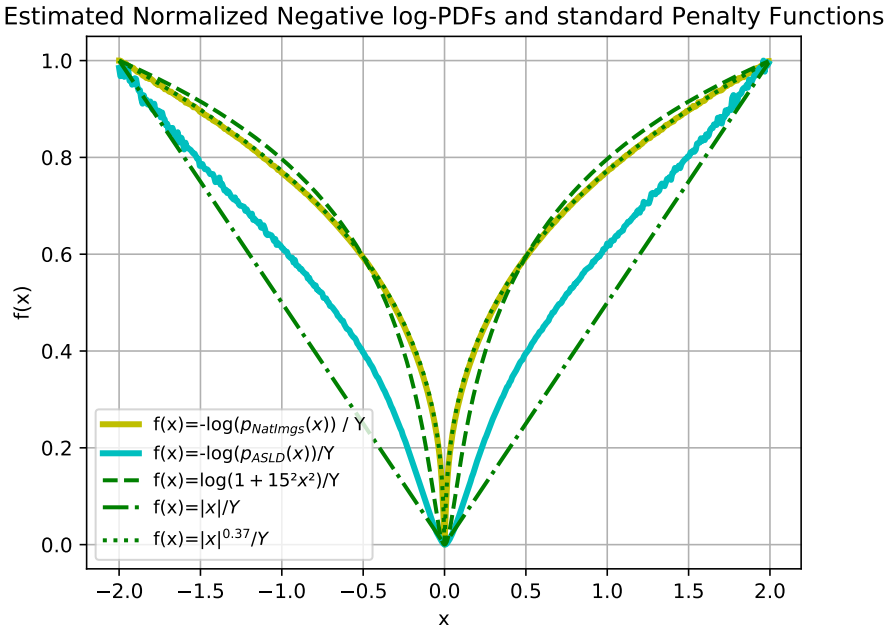

Figure 2.12.: Comparison of the estimated negative log-probability of the PWIs' and Natural Images' filter response and commonly used penalty functions.

Figure 2.12 shows the normalized negative log-probability for PWI and natural images as well as three often used penalties. The root function (green dotted, equivalent to logarithmized generalized Laplacian) adapts to the distribution for natural images almost perfectly but not to the distribution for PWIs. Neither the log-Cauchy penalty (green dashed) nor the absolute function (green dash-dotted) seem to be good estimates for the PWI distribution.

To conclude, the analysis of the filter responses did not emphasize the use of any specific penalty function. For reasonable decision making a less noisy ground truth would be needed. In general, it might be better to use a penalty that fits to the expected filter response of noise-free data. However, in a learning based model this would probably lead to bad solutions because of the discrepancy between noisy ground truth and hyperparameters that are adapted to an ideal ground truth. In this context also the convexity of the chosen penalty has to be considered. A non-convex penalty might be theoretical more reasonable, but might also lead to bad local optimal solutions.

2.2.5. Summary

At the beginning of this chapter it was found that the error distribution is a mixture of at least a Gaussian and a Laplace distribution. The ratio depends mainly on the number of L/C-pairs, i.e. more used pairs are reducing the portion of Gaussian noise. For training and testing, at least 30 L/C-pairs will be used. Less pairs are not considered to yield acceptable image quality, regardless of the specific denoising approach. For more than 30 L/C-pairs the error is more Laplacian like distributed and thus a L1 norm as data term function is preferable against a squared L2 norm. As a Gausian noise portion is still assumed to be present, a center smoothed approximation of the L1 norm is prefered over an ideal L1 norm. This has the additional advantage that a non-continuously differentiable function can be avoided.
It was also found that the voxel intensity and standard deviation is dependent on the position of the slice within the volume. Different intensities are handled by normalizing the data by an appropriate measure (see section 2.2.6 "Data Normalization"). Different standard deviations and thus different noise levels could be handled in principle by using a regularization map. However, the computation of the latter would include the need for a robust estimation of the temporal standard deviation, especially when considering learning based approaches.
The last part of the ASL data analysis dealt with the DCT filter response to PWIs and highlighted the impact of noise to the filter responses. The remaining noise within the ground truth makes a penalty selection based on the responses inappro-

priate. Hence, the final penalty will be chosen not only on theoretic assumptions but also on the obtained image quality and convergence criteria.

2.2.6. Data Normalization

The most simple way to normalize the data would be by dividing each slice by its maximum intensity. Unfortunatelly, the data is corrupted by noise and artifacts, so the maximum is very likely to be an outlier. To overcome this problem, a robust maximum is found by analysing the standard deviation of different percentiles for 100 random combinations for each slice, subject and noise level (number of L/C-pairs). The effects of different brain dimensions per slice are handled by using just the inner 64x64 patch for computation. It was found that the 94 - percentile yields a good tradeoff between maximum correlation and stability for all subjects and noise levels.
Figure 2.13 shows the normalized average intensity and standard deviation per slice for different numbers of L/C-pairs. As expected, the intensities are less dependent to the slice order compared to the unnormalized case (see Figure 2.8). Due to different normalization factors for different noise levels, the intensities are now dependent on the noise levels. This is not an issue, because data with different numbers of L/C-pairs are not mixed up during learning and testing. Additionally, because negative perfusion is not possible, negative values in the PWIs are clipped to zero. This increases the error's bias but decreases the error's standard deviation.

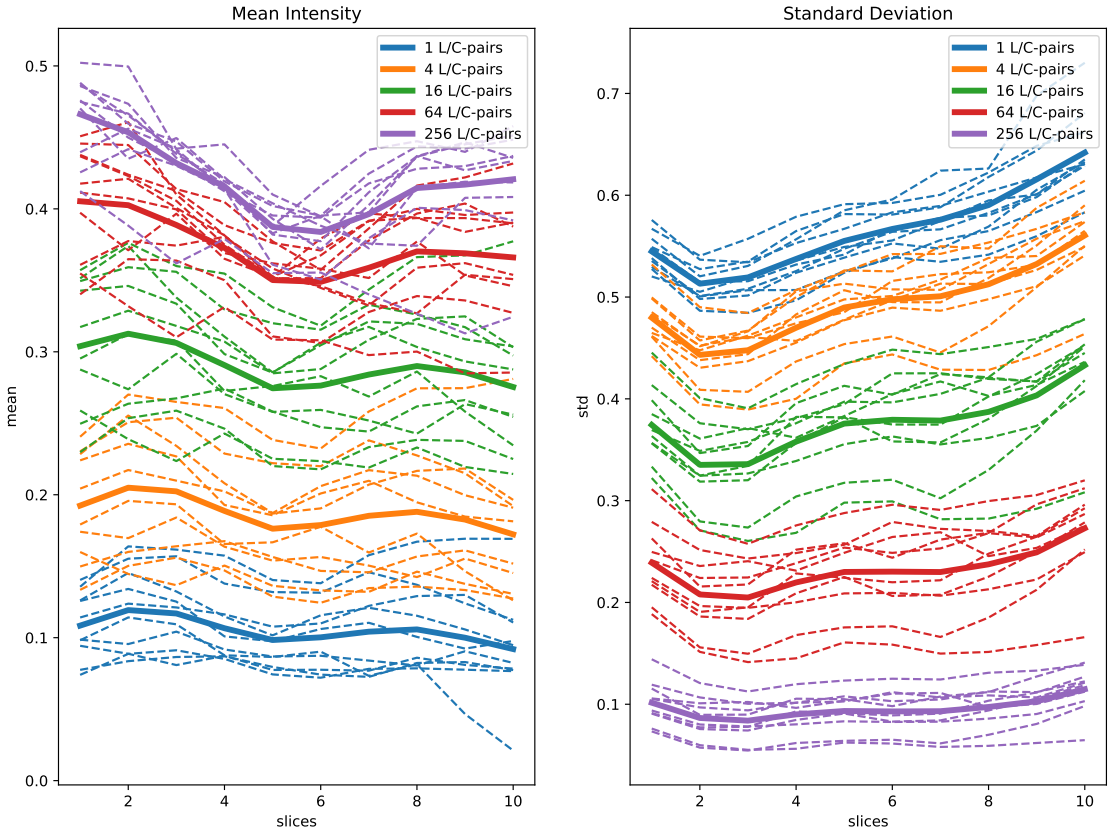

Figure 2.13.: Temporal statistics of the normalized PWIs for different number of L/C-pairs. The colored dashed lines correspond to specific subjects, the colors to a specifc number of L/C-pairs and the bold lines to the average over all subjects.

2.3. A Markov Random Field for ASL denoising

The first model investigated for ASL denoising was the co-sparse analysis model (CSM). The basic form of this model is already explained, therefore the following will focus on the formulation of the kernels, the exact inference scheme, the loss based learning of the free parameters as well as the choice of the hyperparameters.

2.3.1. Details on the Model Formulation

Filter Operator

Chen et al. [27] found that using mean-free filters lead to a better performance than non mean-free filters. Therefore, one could either apply constrained optimization methods which would make the MAP inference more difficult to solve or use a suitable mean-free filter basis. As this part of the thesis was based on the work of Chen, the filter kernels were defined as linear combination of the discrete cosine transform (DCT) basis. This does not only yield a meaningful basis, but also mean-free filters if the constant-entry atom of the DCT basis is omitted. Eq. 2.4 shows how the i-th filter kernel $\mathbf{A}_i \in \mathbb{R}^{k_s \times k_s}$ with k_s being the kernel size, is composed using the DCT basis $\mathbf{B} = \{\mathbf{B}_i, \dots, \mathbf{B}_{N_b}\}$ where $\mathbf{B}_i \in \mathbb{R}^{k_s \times k_s}$ denotes the i-th DCT atom and $N_b = k_s^2 - 1$ the number of basis atoms. The learnable basis weights are written as $\boldsymbol{\beta} \in \mathbb{R}^{N_f \times N_b}$

$$\mathbf{A}_i = \sum_{j=1}^{N_b} \beta_{ij} \mathbf{B}_j \tag{2.4}$$

If the filter process is stated as matrix-vector multiplication, the filter matrix $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N_p N_f \times N_p}$ is used. It is obtained from eq. 2.4, by replacing the DCT filters kernels $\mathbf{B}_i$ with its sparse matrix formulation $\boldsymbol{\mathcal{B}}_i \in \mathbb{R}^{N_p \times N_p}$, followed by subsequently stacking of the resulting row vectors.
To control the ideal amount of regularization, each filter is weighted separately by introducing an additional non-negative parameter $\boldsymbol{\alpha} \in \mathbb{R}_+^{N_f \times 1}$. The objective function of the CSM for denoising E_{CSM} is formulated in eq. 2.5. In contrast to eq.

1.18, the sampling matrix $\mathbf{S}$ was set to be the identity matrix and an additional regularization factor γ was introduced.

$$E_{CSM}(\mathbf{x}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \gamma \sum_{k=1}^{N_p} \sum_{i=1}^{N_f} \alpha_i \phi \left(\left(\left[\sum_{j=1}^{N_b} \beta_{ij} \boldsymbol{\mathcal{B}}_j \right] \mathbf{x} \right)_k \right) + \psi(\mathbf{x}, \mathbf{y}) \quad where \quad \boldsymbol{\alpha} \geq \mathbf{0}. \tag{2.5}$$

Penalty Function

Although learnable penalty functions are desired, the minimization of the LLP with a high level of accuracy would become too difficult and time consuming. Therefore, several meaningful penalties were investigated. A probably powerful penalty is given by the logarithm of the smooth and heavy-tailed Lorentzian distribution, also known as Cauchy distribution, which is equivalent to a student-t distribution with one degree of freedom. Eq. 2.6 shows the primitive as well as the corresponding first and second derivatives of a log-Cauchy penalty.

$$\text{log-Cauchy} \begin{cases} \phi(x) = \kappa_1 \log\left(1 + \kappa_2^2 x^2\right) \\ \phi'(x) = 2\kappa_1 \kappa_2^2 \dfrac{x}{1 + \kappa_2^2 x^2} \\ \phi''(x) = 2\kappa_1 \kappa_2^2 \dfrac{(1 - \kappa_2^2 x^2)}{(1 + \kappa_2^2 x^2)^2} \end{cases} \tag{2.6}$$

The parametrization of the penalty could theoretically be omitted by incorporating κ_1 and κ_2 into the learnable weights $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$, but in practice they simplify the choice for a robust initialization.
Another distribution that is likely to model the filter response well is the Generalized Laplacian (see eq. 1.10) with the parameter $\beta_i = \frac{1}{3} \forall i$. To obtain a continuous differentiateable penalty, the absolute function of the GL was replaced by a smooth approximation $|x| = \sqrt{x^2 + \varepsilon^2}$. The compact formulation of the approximated log-GL penalty and its derivatives are stated in eq. 2.7.

$$\text{approx. GL} \begin{cases} \phi(x) = \kappa_1 \left(x^2 + \varepsilon^2\right)^{\frac{1}{6}} \\ \phi'(x) = \dfrac{\kappa_1}{3} \dfrac{x}{\left(x^2 + \varepsilon^2\right)^{\frac{5}{6}}} \\ \phi''(x) = \dfrac{\kappa_1}{3} \dfrac{\varepsilon^2 - \frac{2}{3}x^2}{\left(x^2 + \varepsilon^2\right)^{\frac{11}{6}}} \end{cases} \tag{2.7}$$

For further comparison also the performance of a smooth approximated L1 penalty (eq. 2.8) and a squared L2 penalty (eq. 2.9) were evaluated. However, the squared L2 for a scalar is actually just a square function, which is indeed not heavy-tailed and thus not very suited. This theoretic limitation is probably counteracted by a very accurate and fast LLP solution as the quadratic penalty causes a quadratic LLP, whose global minimum could be found by using on Newton update step.

$$\text{approx L1} \begin{cases} \phi(x) = \kappa_1 \sqrt{x^2 + \varepsilon^2} \\ \phi'(x) = \kappa_1 \dfrac{x}{\sqrt{x^2 + \varepsilon^2}} \\ \phi''(x) = \kappa_1 \dfrac{\varepsilon^2}{\left(x^2 + \varepsilon^2\right)^{\frac{3}{2}}} \end{cases} \tag{2.8}$$

$$\text{squared L2} \begin{cases} \phi(x) = \dfrac{\kappa_1}{2} x^2 \\ \phi'(x) = \kappa_1 x \\ \phi''(x) = \kappa_1 \end{cases} \tag{2.9}$$

Table 2.3 shows the parameters used for the different penalty functions as well as their identifiers. Two different parameterizations were used for the approximated absolute function. The first, EstAbs aims to estimate the absolute function as exact as possible. For ε smaller than 0.01, the LLP optimizer was not able to find an acceptable solution within reasonable time. The second, SmoAbs, models a function that shares the tailing behavior with the absolute function and the shape around the origin with a quadratic function.

Table 2.3.: Parametrization of the used penalty functions. EstAbs is considered as an estimation to an absolute function and SmoAbs as an absolute function with a distinct quadratic shape around the origin.

Name	**Penalty**	$\boldsymbol{\kappa_1}$	$\boldsymbol{\kappa_1}$	$\boldsymbol{\varepsilon}$
Cauchy	log-Cauchy	0.05	4.0	-
xRoot	log-GL	0.1	-	0.05
EstAbs	approx. L1	0.1	-	0.01
SmoAbs	approx. L1	0.1	-	0.6
Square	squared L2	0.1	-	-

Data Term Function

Although chapter 2.2 "Data Analysis" showed that the error distribution is heavy-tailed, for reasons of convergence the data term function $\psi(\mathbf{x}, \mathbf{y})$ was chosen to be a Huber norm (eq. 2.10). The Huber norm is a simple extension to the Huber loss $h(u)$ that also works with multivariate input data. It is received by computing the Huber loss for each element of the input, followed by an integration of the results over all elements.

$$
\begin{cases} \psi(\mathbf{x},\mathbf{y}) = \mathbf{1}^\top \mathbf{h}(\mathbf{x}-\mathbf{y}) \\ \psi'(\mathbf{x},\mathbf{y}) = \mathbf{h}'(\mathbf{x}-\mathbf{y}) \\ \psi''(\mathbf{x},\mathbf{y}) = \operatorname{diag}(\mathbf{h}''(\mathbf{x}-\mathbf{y})) \end{cases} \quad \text{with} \quad \begin{cases} \mathbf{h}(\mathbf{u}) = (h(u_1),\dots,h(u_N))^\top \\ \mathbf{h}'(\mathbf{u}) = (h'(u_1),\dots,h'(u_N))^\top \\ \mathbf{h}''(\mathbf{u}) = (h''(u_1),\dots,h''(u_N))^\top \end{cases}
$$

$$
\text{and} \quad \begin{cases} h(u) = \begin{cases} \frac{u^2}{2\varepsilon} + \frac{\varepsilon}{2} & |u| \le \varepsilon \\ |u| & \text{else} \end{cases} \\ h'(u) = \begin{cases} \frac{u}{\varepsilon} & |u| \le \varepsilon \\ \operatorname{sign}(u) & \text{else} \end{cases} \\ h''(u) = \begin{cases} \frac{1}{\varepsilon} & |u| \le \varepsilon \\ 0 & \text{else} \end{cases} \end{cases} \tag{2.10}
$$

The Huber loss shares the shape of a quadratic function around the origin and the linear tailing of an absolute function. The parameter ε controls the switchover of

the multipart function. I.e. a small ε will lead to a smooth approximation of the absolute function and a large one to a squared function that is more robust against outliers. For comparison some trained CSM models use the squared L2 norm (eq. 2.11) as data term function.

$$\begin{cases} \psi(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 \\ \psi'(\mathbf{x}, \mathbf{y}) = \mathbf{x} - \mathbf{y} \\ \psi''(\mathbf{x}, \mathbf{y}) = \mathbf{I} \end{cases} \tag{2.11}$$

Kernel Size and the Number of Filters

Theoretically, more and larger filter kernels lead to a more expressive prior and thus to a better denoising performance. In practice, the size and the amount of filters are limited by computational feasibility and the aggregation of numerical errors, which introduce further problems when solving the LLP. Kernel sizes of 3x3, 5x5 and 7x7 were selected for testing the models. Previous work [59, 60, 61] often focused on cases where the number of filter kernels N_f exceeded the kernel dimension. Chen et al. [27] showed that it is sufficient to chose N_f to be exactly the kernel dimension, which is in the current case $k_s^2 - 1$ due to the excluded DCT atom.

Initial Values

The used initial values were inspired by Chen et al. and adapted slightly to ensure fast and stable converging properties for the given data. For all performed experiments, the initial value for α_i was set to 1 over the number of filters. The initials for $\boldsymbol{\beta}$ were drawn from a normal distribution and normalized such that $\left\|\boldsymbol{\beta}_{\cdot j}\right\|_2 = 1$. For inference as well as for learning, the $\mathbf{x}_0$ was set to be the input image $\mathbf{y}$. The Huber loss's ε was set to 10^{-1}, which yield a good tradeoff between accuracy and convergence stability. For smaller values the training procedure becomes more likely to fail. Unless stated otherwise, the additional regularization factor γ was set to 1.

2.3.2. Inference and Learning

The bilevel learning scheme (eq. 1.21) can be interpreted as constrained optimization problem. Assuming the existence of $\mathbf{x}^*$ and a convex LLP, the first order optimality condition is sufficient and thus can be used to further simplify the expression. Using the fully parametrized CSM (eq. 2.5) as LLP, the corresponding Lagrangian $\mathcal{L}$ with $\boldsymbol{\lambda} \in \mathbb{R}^{N_p \times 1}$ being the Lagrange multiplier writes as follows:

$$\begin{aligned}
\mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) &= L(\mathbf{x}, \hat{\mathbf{x}}) + \nabla_x E_{CSM}(\mathbf{x}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta})\boldsymbol{\lambda} \\
&= L(\mathbf{x}, \hat{\mathbf{x}}) + \left(\sum_{i=1}^{N_f} \alpha_i \boldsymbol{\mathcal{A}}_i^\top \phi' \left(\boldsymbol{\mathcal{A}}_i \mathbf{x} \right) + \psi' \left(\mathbf{x}, \mathbf{y} \right) \right) \boldsymbol{\lambda} \\
&= L(\mathbf{x}, \hat{\mathbf{x}}) + \left(\sum_{i=1}^{N_f} \alpha_i \left[\sum_{j=1}^{N_b} \beta_{ij} \boldsymbol{\mathcal{B}}_j \right]^\top \phi' \left(\left[\sum_{j=1}^{N_b} \beta_{ij} \boldsymbol{\mathcal{B}}_j \right] \mathbf{x} \right) + \psi' \left(\mathbf{x}, \mathbf{y} \right) \right) \boldsymbol{\lambda}
\end{aligned} \tag{2.12}$$

The inequality constraint of $\boldsymbol{\alpha}$ is omitted in the formula above and will be handled as a simple box constraint by the optimizer itself. To solve the bilevel problem at least the gradient of the Lagrangian is needed.

$$\begin{aligned}
\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) &= L'(\mathbf{x}, \hat{\mathbf{x}}) + \mathbf{H}_E \boldsymbol{\lambda} \\
&= L'(\mathbf{x}, \hat{\mathbf{x}}) + \nabla_x^2 E_{CSM}(\mathbf{x}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta})\boldsymbol{\lambda} \\
&= L'(\mathbf{x}, \hat{\mathbf{x}}) + \left(\sum_{i=1}^{N_f} \alpha_i \boldsymbol{\mathcal{A}}_i^\top \operatorname{diag}(\phi''(\boldsymbol{\mathcal{A}}_i \mathbf{x})) \boldsymbol{\mathcal{A}}_i + \psi''(\mathbf{x}, \mathbf{y}) \right) \boldsymbol{\lambda} \\
\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) &= \nabla_x E_{CSM}(\mathbf{x}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= \sum_{i=1}^{N_f} \alpha_i \boldsymbol{\mathcal{A}}_i^\top \phi' \left(\boldsymbol{\mathcal{A}}_i \mathbf{x} \right) + \psi' \left(\mathbf{x}, \mathbf{y} \right) \\
\nabla_{\alpha_i} \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) &= \left(\boldsymbol{\mathcal{A}}_i^\top \phi'(\boldsymbol{\mathcal{A}}_i \mathbf{x}) \right)^\top \boldsymbol{\lambda} \\
\nabla_{\beta_{ij}} \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) &= \alpha_i \left(\boldsymbol{\mathcal{B}}_j^\top \phi'(\boldsymbol{\mathcal{A}}_i \mathbf{x}) + \boldsymbol{\mathcal{A}}_i^\top \operatorname{diag}(\phi''(\boldsymbol{\mathcal{A}}_i \mathbf{x})) \boldsymbol{\mathcal{B}}_j \mathbf{x} \right)^\top \boldsymbol{\lambda}
\end{aligned} \tag{2.13}$$

Eq. 2.13 shows the first order partial derivatives of the Lagrangian $\mathcal{L}$ with respect to $\mathbf{x}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\lambda}$. In principle, the gradient above would be sufficient to start an iterative optimization procedure, but it is possible to simplify the learning scheme. A closer look at the dependencies of the partial derivatives uncovers that $\nabla_{\boldsymbol{\lambda}}\mathcal{L}$ just depends on $\mathbf{x}$ and $\nabla_{\mathbf{x}}\mathcal{L}$ just on $\mathbf{x}$ and $\boldsymbol{\lambda}$. In addition, the latter can be obtained in closed form by using the first order optimality condition $\nabla_{\mathbf{x}}\mathcal{L} = 0$. Hence, if $\mathbf{x}$ is solved first with a high level of accuracy using a suitable optimization algorithm, followed by computing $\boldsymbol{\lambda}$, $\nabla_{\mathbf{x}}\mathcal{L} = 0$ and $\nabla_{\boldsymbol{\lambda}}\mathcal{L} = 0$ are fulfilled and thus more effective optimization steps are obtained for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. It is further noteworthy that $\nabla_{\boldsymbol{\lambda}}\mathcal{L}$ is equivalent to $\nabla_{\mathbf{x}}E_{CSM}$, the gradient of the LLP w.r.t to $\mathbf{x}$ and that H_E notes the corresponding Hessian matrix. This raises an important clue to the existence of the inverse Hessian needed for the computation of the gradient w.r.t to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$: In case of a convex LLP, hence a convex penalty and a convex data norm, the Hessian is always positive definite and thus invertible. In case of a non-convex LLP, the Hessian is not positive definite for all possible $\mathbf{x} \in \mathbb{R}^{N_p}$, but it follows from the second order optimality condition that the Hessian is positive definite if $\mathbf{x}$ is a local minimum of the LLP.

In general, the gradient of the whole test set could be obtained by summing up all gradients w.r.t to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of single image patches. A more sophisticated approach is given by so called mini-batches. Here, the gradients are computed over small subsets rather than over the whole dataset. For each iteration a different mini-batch is used, leading to faster but more inaccurate updates. This inaccuracy is reflected by a loss evolution that is superimposed with more or less strong oscillations. This sometimes called Stochastic Gradient Descent (SGD) known technique has shown to work out well for many different kind of problems. Apart from performance considerations, the stochastic nature of this procedure might also help to escape from local minima, which is important when considering a non-convex loss function. The full iterative learning scheme looks like follows:

1. Initialization
2. Select N_B batches from the training dataset
3. Solve the LLP with a suitable optimizer (f.e. Newton's Method, CG)
4. Compute $\boldsymbol{\lambda}$ in closed form (exploiting sparsity)
5. Compute metrics (loss, PSNR, SSIM)
6. Compute the gradient w.r.t to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
7. Perform a parameter update on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ (f.e. Adam)
8. Apply box constraints to $\boldsymbol{\alpha}$
9. If convergence criteria is not reached, go back to 2.

Optimizers

Basically, there is no limitation in which optimizer to use, but in preliminary tests it was found that Newton's Method performed well on the LLP for patch sizes up to 64x64 pixels. For larger patch sizes the computation of the Hessian matrix needed for Newtons method becomes too costly and thus CG was used in such cases. For both methods the step size was found according to a linesearch.
The HLP, i.e. the learning of the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, was optimized by using Adam [62], which is a first order method using an adaptive momentum term. It is often used in state of the art machine learning tasks and determines its step size automatically. This is particularly important, because a line search or similar techniques would imply the evaluation of the loss for each different step. This evaluation would just be valid after solving the LLP and the Hessian. Hence, a stepsize selection scheme for the HLP would slow down the learning a lot.

Stopping Criterion

As already mentioned, to reach acceptable convergence properties it is important to solve the LLP with a high level of accuracy. Chen et al. stopped the LLP optimization as soon as the L2 norm of the LLP's gradient, normalized by the

number of pixels, falls below 10^{-5} (input range [0-255]) [36]. Kunisch et al. stopped it at 10^{-9} (input range [0-1]) [63]. As those values are related to images with a known maximum intensity they can not by transfered directly to the used ASL input data. Nevertheless, for training a stopping epsilon of $2 \cdot 10^{-7}$ turned out to work well. In case of inference this high level of accuracy is not needed and thus an epsilon of $5 \cdot 10^{-4}$ was used.
The HLP optimization was early stopped after 500 iterations. A stopping based on vanishing gradients or vanishing loss was not used as the stochastic learning scheme as well as the outlier prone data lead to oscillations of the loss and of the gradients during training.

Loss Function

The CSM was trained using a squared L2, a Huber L1 ($\varepsilon = 10^{-3}$) and a SSIM Loss. Apart from its smooth and thus simple to optimize nature, the squared L2 norm is closely related to the mean squared error (MSE) and therefore to the PSNR. As it penalizes large deviations stronger than the L1 loss it favors smooth solutions rather then sharp ones.
The optimization of the non continuously differentiateable L1 norm is often done by exploiting proximal gradient methods like the primal dual algorithm [64]. Another, simpler way is to use a smooth approximation like the Huber norm (eq. 2.10). This approximation does not affect the tailing behavior of the loss and thus is especially suited when sharp edges are desired.
The SSIM [24] measures the structural similarity between two image patches and is designed to model the human perception. Compared to the PSNR it favors sharp solutions rather than blurry ones and is therefore preferable as quantitative metric although its relation to human perception might be doubtful [23].

$$SSIM(\mathbf{x}, \mathbf{y}) = \left(\frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \right) \left(\frac{2\sigma_{xy} + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \right) = l(\mathbf{x}, \mathbf{y}) \cdot cs(\mathbf{x}, \mathbf{y}) \tag{2.14}$$

Eq. 2.14 states the SSIM computations between the image patches $\mathbf{x}$ and $\mathbf{y}$ of

size N_{SSIM} x N_{SSIM} ($N_{SSIM} = 11$ px). μ_x and μ_y refer to the mean values of the patches. σ_x^2 and σ_y^2 to the variances of the patches and σ_{xy} to the covariance between those two patches. The originally proposed SSIM version weights the elements within the patches according to a 2D-Gaussian function ($\sigma = 1.5$ px). The constants c_1 and c_2 are set to 0.01^2 and 0.03^2 respectively. There are several hyperparameters for calculating the SSIM which effect the result. A very important one is given by the patch size: large patch sizes favor sharp but noisy solutions, small patch sizes favor more blurry ones. The authors proposed to use the SSIM with a patch size of 11 pixels, which is found to work well on natural images. However, to overcome the problem of selecting an appropriate patch size the authors also proposed a multiscale SSIM (msSSIM) [25].
The msSSIM is obtained by computing the contrast and structure $cs(x, y)$ part of the SSIM on different scales and multiplying all results with the luminescence part $l(x, y)$ at maximum scale. The formula for the msSSIM is given as follows:

$$msSSIM(\mathbf{x}, \mathbf{y}) = l_{\sigma_M}(\mathbf{x}, \mathbf{y}) \cdot \prod_{i=1}^{M} cs_{\sigma_i}(\mathbf{x}, \mathbf{y}) \tag{2.15}$$

The subindex σ_i indicates the changed Gaussian weighting which is interpreted as a change in scale. The msSSIM used throughout this work is performed with $\sigma = [0.5, 2, 4]$. The gradients of the SSIM and msSSIM w.r.t. $\mathbf{x}$ can be found in appendix A.

2.3.3. Metrics for Evaluation

The obtained results were compared on the basis of the peak signal-to-noise ratio (PSNR) and the SSIM 2.14. The PSNR is computed as follows:

$$PSNR(\mathbf{x}, \mathbf{y})_{dB} = 10 \log_{10} \left(\frac{y_{max}^2}{MSE(\mathbf{x}, \mathbf{y})} \right) \tag{2.16}$$

where $\mathbf{x}$ notes the denoised and $\mathbf{y}$ the GT CBF map, MSE denotes the mean squared error and y_{max}^2 is the maximum intensity within the reference slice $\mathbf{y}$. It is noteworthy that in case of natural images y_{max} is typically set to the maximum

possible intensity (255 in case 8bit). The PSNR computation above states its relation to the MSE and as a consequence the relation to the squared L2 norm. In fact, minimizing the squared L2 also minimizes the MSE and thus the PSNR gets maximized. Both metrics are calculated slicewise within given brain masks (whole brain, gray matter and white matter). To evaluate the performance on whole datasets, the metrics are averaged over all slices.

2.3.4. Datasets

The averaged ASL difference data 2.17 from 10 subjects was splitted into 3 subsets.

$$\mathbf{y}_{in} = \frac{1}{N_{ave}} \sum_{i=1}^{N_{ave}} \mathbf{C}_i - \mathbf{L}_i \tag{2.17}$$

The first subset was the learning set which was formed from all slices (1-10) except 3,6,9 from subject I-VI. The skipped slices 3,6,9 formed the intra-subject test set. The last subset was the inter-subject test set which contained all data from subject VII - X. Although the basic error level differs quite a lot for different subjects, especially for men and women (higher CBF for women), the sets can be considered equal because the ratio between 'good' and 'bad' quality slices is about the same. For each of the 128x128 test set slices, 10 randomly selected combinations of L/C-pairs were used to build the input. Therefore, the inter-subject test set contained 400 images and the intra-subject test set 120 images. A second version of the inter-subject test set was obtained by using just the first N_{ave} L/C-pairs. Thus this set is named InARow test set.
During training, for each iteration new N_{ave} randomly chosen L/C-pairs were used. An alternative is given by using subsequently acquired L/C-pairs, which would enhance the quality of the input data due to less motion. However, the 'random load' approach overcomes the effects of bad input combinations to some extent. I.e. it is more robust against outliers. Additionally, this method yields a more comprehensive trainingset as for each iteration different datat is used. As the mini-batches were composed of all trainig slices within a single subject, 7 slices were used for each iteration. Hence, after 500 training iterations the model has seen

3500 different input slices. However, they must be considered as highly correlated due to the 42 basic image slices (7 slices from 6 subjects).

2.3.5. Experimental Setup

To evaluate the basic principle, initial tests on a synthic dataset [16] were performed at the very beginning. Afterwards three groups of experiments were carried out to chose appropriate hyperparameters. The first group of expermiments focused on the preselection of different penalty functions. The second group explored the impact and the interaction between loss, data term and penalty. The last group investigated the impact of the number and size of the used filters. All of these experiments were carried out using the inter-subject test set and 50 L/C-pairs. After appropriate hyperparameters were found, the performance of the SSIM loss was tested. Further, the impact of the additional regularization factor γ during inference was explored. Once the final model was chosen, the performance on the remaining test sets with different numbers of input L/C-pairs was evaluated and compared to the VN and stTGV [16].

2.3.6. Implementation Details

The model was implemented using TensorFlow in combination with PYTHON 3.6. TensorFlow does not support sparse matrix formats which are needed to efficiently compute the Hessian matrix. Therefore, the $\boldsymbol{\lambda}$ computation was performed outside the TensorFlow graph using the ScientificPython (SciPy) library. The inversion of the Hessian was performed using the sparse Cholesky decomposition provided by the Skimage library. Convolutions were performed using zero padding at boundaries. To increase the LLP accuracy, all computations were carried out using double precision floating point operations. All experiments were carried out on an Intel i5-2500K @ 3.30GHz x 4.

2.4. A Variational Network for ASL denoising

2.4.1. Details on the Model Formulation

Filter Operator

In contrast to the CSM, the VN uses basis free filters. Hence, the filter parameters were optimized directly. Therefore, mean-free filters were ensured by adding a constraint to the optimization scheme. A second constraint fixed the filters to have unity norm, i.e. $||\mathbf{k}_i^t||_2 = 1$.

Penalty Function

A fundamental improvement of the VN compared to the CSM is the use of learnable penalty functions for each filter. Therefore, the gradient of the penalty (i.e. the activation) $\phi_i^{t\prime}(x)$ (eq. 2.18) of the i-th filter in stage t is modeled as a weighted combination of several radial basis functions (RBFs).

$$\phi_i^{t\prime}(x) = \sum_{j=1}^{N_w} w_{ij}^t e^{-\frac{(x-\mu_j)^2}{2\sigma^2}} \tag{2.18}$$

Here, N_w is the number of radial basis functions which were set up linearly between $v_{min} = -0.5$ and $v_{min} = +0.5$ and was set to 31. The center of the j-th RBF is termed as μ_j and the standard deviation σ is calculated as follows: $\sigma = \frac{v_{max}-v_{min}}{N_w-1}$

Data Term Function

For the data term ψ three different functions were tested: a squared L2, a L1 and a center-smoothed root (csRoot) ($\psi(\mathbf{x}) = ||\mathbf{x}^{\odot 2} + \mathbf{1}||_\gamma^\gamma,\ \gamma = 0.15$). Additional experiments were carried out using temporal data terms as well as SSIM based data terms. However, these data term approaches were not further investigated as they do not improve the result for the given model-optimizer setup.

Kernel Size, the Number of Filters and the Number of Stages

In analogy to the CSM the number of filters were set to $k_s^2 - 1$, with k_s being the kernel size. Tested kernel sizes include 3x3, 5x5, 7x7 as well as 9x9. The tested number of stages were set to 3,5,7 and 9.

Initial Values

The filters were initialized with normal distributed random values followed by applying the zero mean and the unit ball constraint. The penalty weights were initialized linearly between -0.02 and 0.02. The initial weight for the data term was set to 0.1 for all stages. Like for the CSM, the initial image $\mathbf{x}_0$ was set to be the input image $\mathbf{y}$.

2.4.2. Inference and Learning

The network was trained using either a L2, L1, SSIM or msSSIM loss for 500 iPALM [30] iterations. For the iPALM, a fixed momentum of 0.4 and a maximum of 40 backtracking iterations were used. The Lipschitz constant was initialized with 1000. The data was preprocessed and loaded exactly as described in 2.3.4 "Datasets". For training, a batch size of 42 images was used (7 slices of 6 training subjects).

2.4.3. Experimental Setup

At first, an appropriate model size was found by varying the number of stages between 3 and 9, and the number and size of filters between 8x3x3 and 48x7x7. Further, the impact of different losses and data term functions on the denoising performance was investigated using the InARow test set. At the end, the performance on the inter-subject and intra-subject test set was evaluated and compared to the CSM and stTGV.

2.4.4. Implementation Details

The model was implemented in TensorFlow using the Framework provided by Kobler [40] and adapted to the peculiarities of ASL denoising. Convolutions were performed using reflected boundary conditions. All experiments were carried out on a Nvidia Titan Xp.

3. Results

3.1. Phantom Data, Convergence and Input

At the very beginning a proof of concept is established by training the CSM with synthetic data [16]. The obtained kernels for this test set are shown in Figure 3.1 (EstAbs penalty, squared L2 loss, squared L2 data term and 10 filters of size 5x5). The four non-zero weighted kernels could be paired to form symmetric difference kernels. This highlights the close relation of the learned operator to total variation (TV) regularization. This behavior is caused by the piecewise constant nature of the used synthetic data, which is known to be attracted by TV regularized solutions.

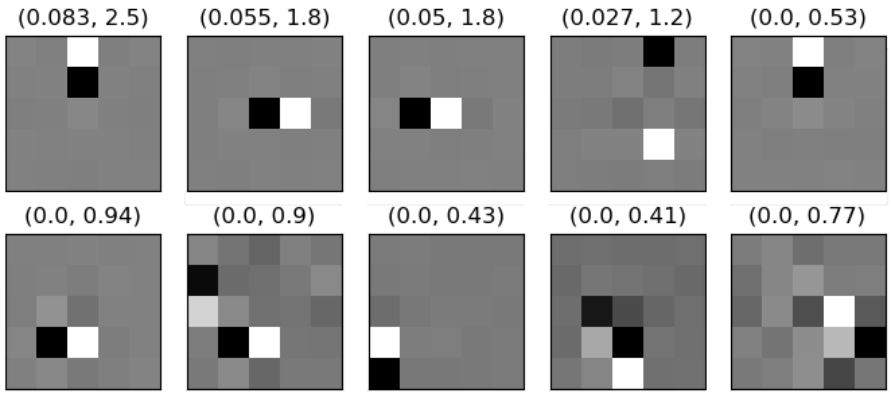

Figure 3.1.: Learned kernels for EstAbs penalty, squared L2 loss, squared L2 data term, 10 filters of size 5x5, 50 L/C-pairs and phantom data.The corresponding weight and norm of the filter is stated in brackets.

Figure 3.2 shows exemplarily the training progress for real PASL data (inter-subject test set). After an initial loss decrease the latter begins to oscillate. This is

probably due to the rather small batch-size and the randomly chosen PWI for each iteration. On a first glance this might be an issue, but actually these stochastic updates help to escape from local minima.

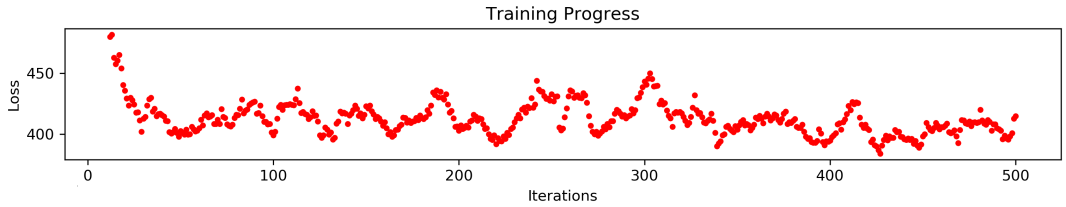

Figure 3.2.: Training progress for SmoAbs penalty, squared L2 loss, squared L2 data term, 24 filters of size 5x5 and 50 L/C-pairs.

Figure 3.3 shows several input and reference CBF maps for different subjects and slices as well as the initial SSIM and PSNR for 50 L/C-pairs.

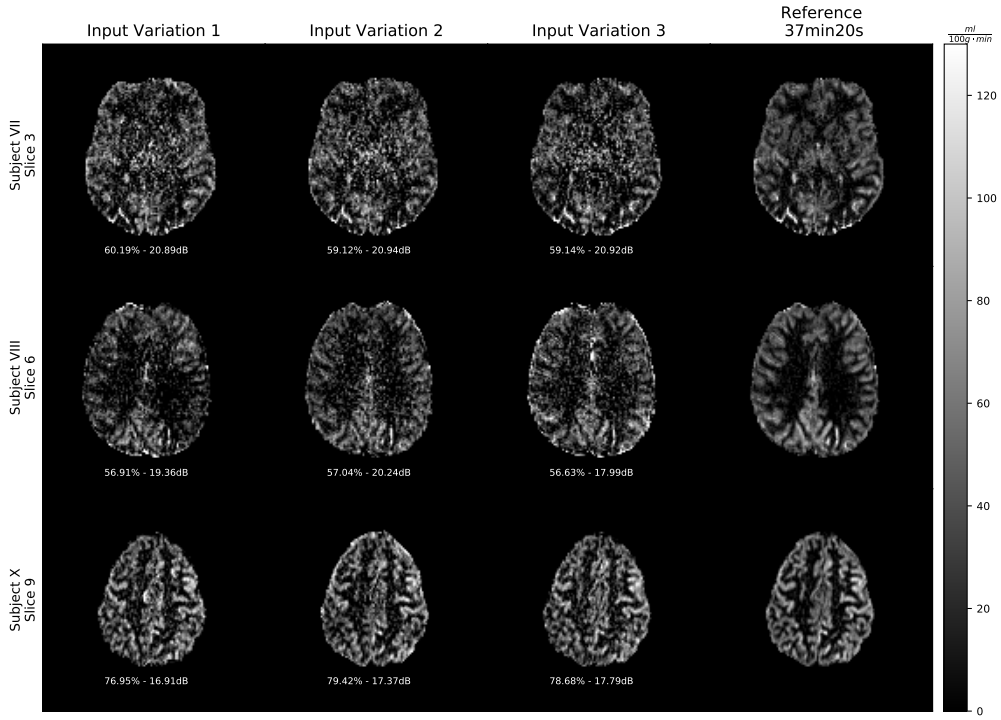

Figure 3.3.: Input CBF maps for several subjects and slices (50 L/C-pairs) as well as the corresponding reference CBF map. The three given maps per slice differ in the used L/C-pair combination. The stated SSIM and PSNR are computed between the corresponding slice and its reference.

3.2. Hyperparameters of the CSM

3.2.1. Impact of the Penalty Functions

The results of the first group of model selection experiments can be seen in Table 3.1 as well as in Figure 3.4. Log-Cauchy, SmoAbs and also Square penalties lead to very similar WB PSNR values (22.55dB - 22.61dB), whereas the xRoot penalty is clearly worse considering PSNR (22.45dB) and SSIM (68.62%). This is particular interesting as the xRoot penalty is in theory the most appropriate penalty function. A reason for this phenomenon might be explained by the non-convexity as well as the non-smoothness of this function. Due to the weak performance and the long training time, the xRoot penalty approach is not pursued. The CBF maps show that the xRoot penalty yields very blurry results and therefore a lower SSIM (68.62%). The Square penalty maps contain more noise than the remaining ones and thus a lower PSNR (22.57dB) is obtained. However, they are also sharper which leads to a higher SSIM (69.70%). EstAbs and SmoAbs behave to some extend like xRoot and Square penalties, respectively. This is surprising, because in general one would expect that more quadratic penalties lead to more blurry solutions because they penalize larger filter responses stronger. A possible explanation is derived from another viewpoint: SmoAbs and Square have a quadratic center, which means that small filter responses are less penalized compared to the linear shaped center of xRoot and EstAbs penalty. Small filter responses are likely in rather constant valued regions which are corrupted with noise. This behavior might be increased by the bias which is introduced by the clipping of negative values (especially in WM). During learning, accepting small filter responses reduces the amount of regularization of the learned prior. This leads to less regularization and thus to sharper images. Despite its acceptable performance, the square penalty will not be part of further experiments, because the SmoAbs penalty combines the advantages of the Square penalty and the theoretic assumptions made upon the log-probability of the filter response.
This first set of experiments (as well as all proceeding ones) also highlights the different SNR levels in GM and WM. Although this was expected, the ideal model should perform equally well in both regions.

Table 3.1.: Parameter setup, training loss and testing results (CBF, inter-subject test set) for the penalty selection tests. All experiments were carried out using 50 L/C-pairs, squared L2 loss, squared L2 data term and 24 filters of size 5x5. The stated loss is averaged over the last 60 iterations. Training was early stopped after 500 iterations.

Setup		Training			Testing (WB / GM / WM)	
Penalty	Function	Time	$\alpha_i \leq 0$	Loss	PSNR in dB	SSIM in %
SmoAbs	$\frac{\sqrt{x^2+0.6^2}}{10}$	2h40	0-5	408.0	22.61 / 20.74 / 17.59	69.74 / 85.41 / 57.61
EstAbs	$\frac{\sqrt{x^2+0.01^2}}{10}$	5h20	0-1	410.2	22.55 / 20.64 / 17.64	69.07 / 85.08 / 56.64
Cauchy	$\frac{\log\left(1+(4x)^2\right)}{20}$	4h20	0-4	408.3	22.60 / 20.69 / 17.70	69.87 / 85.32 / 57.90
xRoot	$\frac{(x^2+0.05^2)^{\frac{1}{3}}}{10}$	14h10	0-2	419.8	22.45 / 20.52 / 17.60	68.62 / 84.75 / 56.50
Square	$\frac{x^2}{20}$	2h40	0	417.8	22.57 / 20.72 / 17.48	69.70 / 85.44 / 57.48

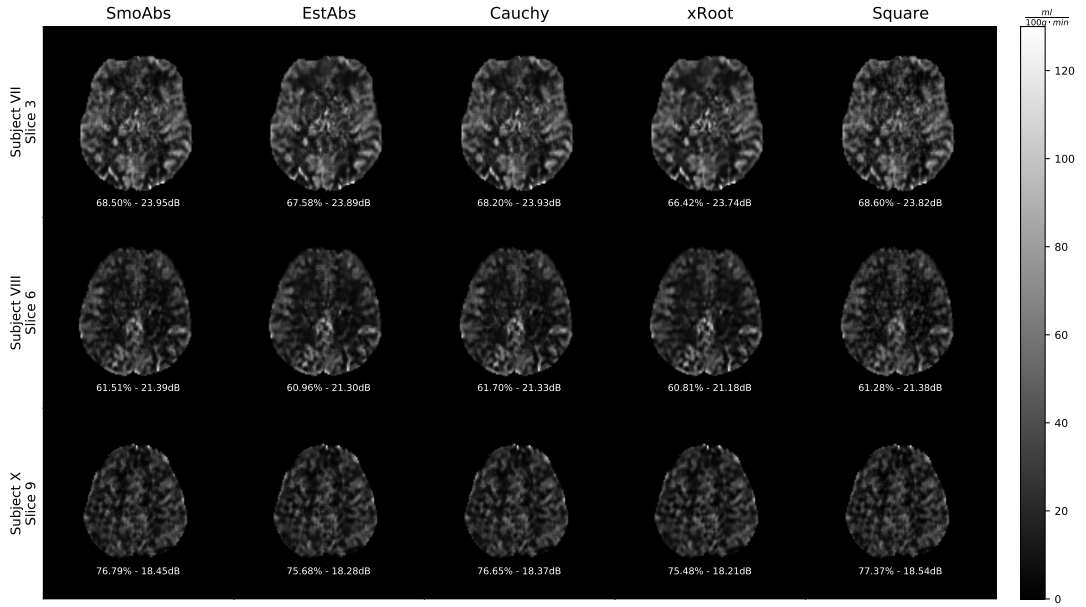

Figure 3.4.: Penalty Evaluation. CBF maps, SSIM and PSNR for different slices using 50 L/C-pairs. All models use a L2 loss, a L2 data term and 24 filters of size 5x5.

3.2.2. Interaction between Loss, Data Term and Penalty

In the following section the results for different combinations of data term (squared L2, smooth Huber-L1 with $\varepsilon = 10^{-1}$ ($\mathrm{HL1_{E\text{-}1}}$)) and loss (squared L2 and Huber-L1 with $\varepsilon = 10^{-3}$ ($\mathrm{HL1_{E\text{-}3}}$)) with the previously selected penalty functions (SmoAbs, EstAbs, Cauchy) are stated. Firstly, the influence of loss and data term for SmoAbs

penalty are investigated and the results are listed in Table 3.2. From a quantitative point of view, a Huber loss with squared L2 data term ($HL1_{E\text{-}3}L2$) performs best on the given dataset (22.61dB / 70.92%). The other setups can not be ranked clearly: L2L2 gives a better PSNR (22.61dB), whereas $L2HL1_{E\text{-}1}$ and $HL1_{E\text{-}3}HL1_{E\text{-}1}$ a better SSIM (70.39% and 70.38%). Despite trained with different losses, $L2HL1_{E\text{-}1}$ and $HL1_{E\text{-}3}HL1_{E\text{-}1}$ share quite the same metrics.

Table 3.2.: Parameter setup, training loss and testing results (CBF, inter-subject test set) for different loss-data term combinations using the SmoAbs penalty. All experiments were carried out using 50 L/C-pairs and 24 filters of size 5x5. The stated loss is averaged over the last 60 iterations. Training was early stopped after 500 iterations.

Setup		Training			Testing (WB / GM / WM)	
Loss	Data Term	Time	$\alpha_i \leq 0$	Loss	PSNR in dB	SSIM in %
L2	L2	2h40	0-5	408.8	22.61 / 20.74 / 17.59	69.74 / 85.41 / 57.61
$HL1_{E\text{-}3}$	L2	2h40	4-8	3590.7	22.61 / 20.72 / 17.74	70.92 / 85.60 / 58.81
L2	$HL1_{E\text{-}1}$	8h30	0-15	416.6	22.54 / 20.48 / 18.11	70.39 / 84.52 / 59.56
$HL1_{E\text{-}3}$	$HL1_{E\text{-}1}$	8h30	1-14	3512.9	22.54 / 20.46 / 18.17	70.38 / 84.49 / 59.75

Comparing the CBF maps (Figure 3.5) it is observable that a L1 data term leads

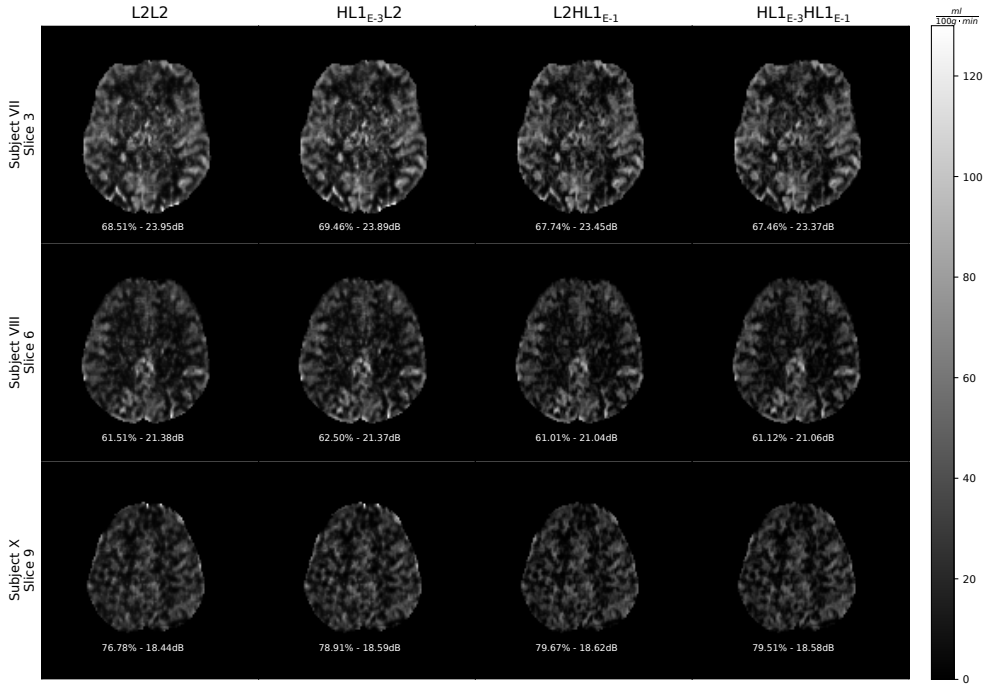

Figure 3.5.: Loss-data term evaluation for SmoAbs penalty. CBF maps, SSIM and PSNR for different slices and 50 L/C-pairs. All models using 24 filters of size 5x5.

to lower CBF values in regions were less CBF is assumed (WM regions). This is expected because larger deviations from the input are assumed as more likely and thus less penalized. Despite a lower SSIM, L1 data term images appear sharper than corresponding L2 data term images. A drawback due to allowing larger deviations from the input is the clipping of large CBF values (potentially hyperperfusion).

Table 3.3.: Parameter setup, training loss and testing results (CBF, inter-subject test set) for different loss-data term combinations using the EstAbs penalty. All experiments were carried out using 50 L/C-pairs and 24 filters of size 5x5. The stated loss is averaged over the last 60 iterations. Training was early stopped after 500 iterations.

Setup		Training			Testing (WB / GM / WM)	
Loss	Data Term	Time	$\alpha_i \leq 0$	Loss	PSNR in dB	SSIM in %
L2	L2	5h20	0-1	410.2	22.55 / 20.63 / 17.64	69.09 / 85.06 / 56.68
$HL1_{E-3}$	L2	5h20	3-4	3601.9	22.56 / 20.64 / 17.77	70.64 / 85.38 / 58.29
L2	$HL1_{E-1}$	22h50	0-3	415.4	22.33 / 20.14 / 18.24	69.19 / 83.69 / 59.00
$HL1_{E-3}$	$HL1_{E-1}$	22h50	0-2	3499.7	22.46 / 20.30 / 18.35	69.68 / 84.00 / 59.28

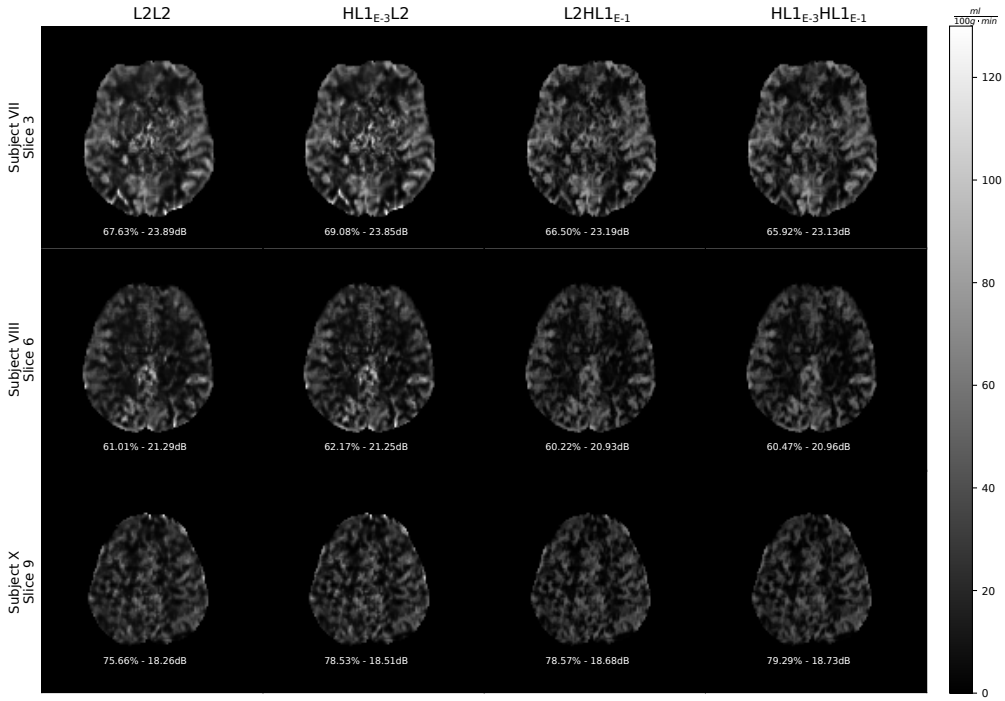

Figure 3.6.: Loss-data term evaluation for EstAbs penalty. CBF maps, SSIM and PSNR for different slices and 50 L/C-pairs. All models using 24 filters of size 5x5.

For the EstAbs penalty, the combination of $HL1_{E\text{-}3}L2$ attains also the best quantitative results (Table 3.3). However, compared to SmoAbs all setups are worse. A squared L2 loss leads to particular blurry CBF maps (Figure 3.6) and a L1 data term suppresses large CBF values even stronger compared to a SmoAbs penalty. All in all, the SmoAbs penalty outperforms the EstAbs penalty. Hence, an EstAbs penalty will not be considered in future experiments.

Table 3.4.: Parameter setup, training loss and testing results (CBF, inter-subject test set) for different loss-data term combinations using the Cauchy penalty. All experiments were carried out using 50 L/C-pairs and 24 filters of size 5x5. The stated loss is averaged over the last 60 iterations. Training was early stopped after 500 iterations.

Setup		Training			Testing (WB / GM / WM)	
Loss	Data Term	Time	$\alpha_i \leq 0$	Loss	PSNR in dB	SSIM in %
L2	L2	4h20	0-4	408.3	22.60 / 20.69 / 17.70	69.85 / 85.32 / 57.88
$HL1_{E-3}$	L2	4h20	7-11	3586.4	22.59 / 20.67 / 17.81	70.84 / 85.48 / 58.85
L2	$HL1_{E-1}$	14h10	0-1	410.9	22.51 / 20.36 / 18.38	70.09 / 84.10 / 60.02
$HL1_{E-3}$	$HL1_{E-1}$	14h10	0	3482.3	22.46 / 20.35 / 18.26	70.91 / 84.40 / 60.31

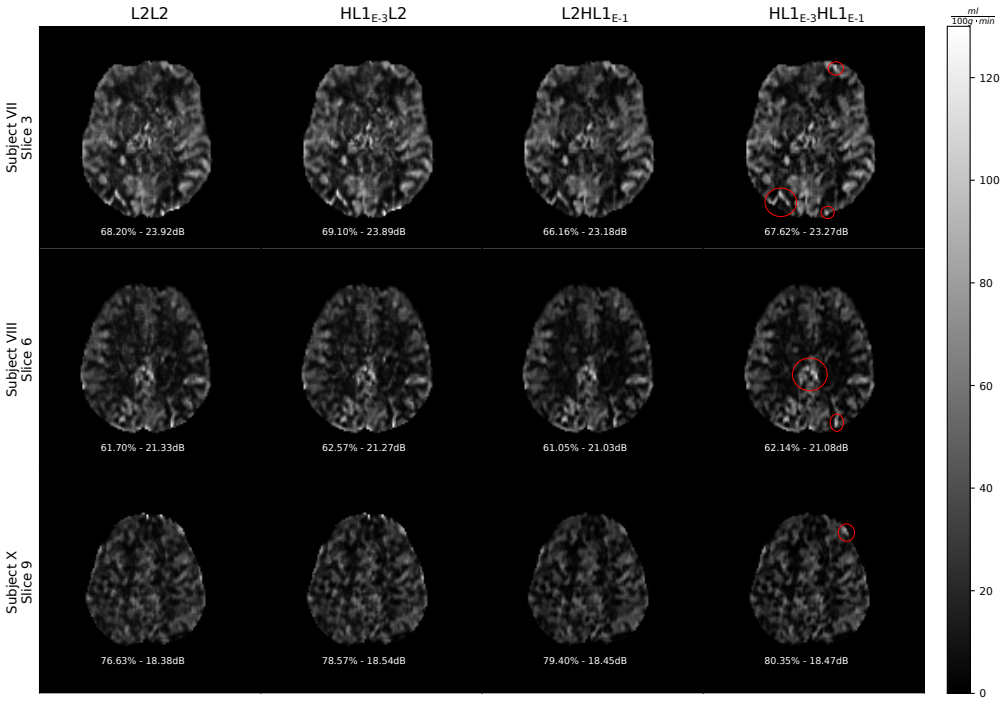

Figure 3.7.: Loss-data term evaluation for Cauchy penalty. CBF maps, SSIM and PSNR for different slices and 50 L/C-pairs. All models using 24 filters of size 5x5.

The results of the third investigated penalty, the Cauchy penalty, are listened in Table 3.4. The best SSIM (70.91%) is obtained using a Huber loss and a Huber-L1 data term ($HL1_{E\text{-}3}HL1_{E\text{-}1}$). Regardless of the loss, the best PSNR (22.59dB) is obtained when using a squared L2 data term. However, a higher SSIM is obtained by using a Huber loss (70.84%) instead of a L2 loss (69.85%). In general, the Cauchy penalty leads to better metrics than an EstAbs penalty and to comparable results using the SmoAbs penalty. The CBF maps in Figure 3.7 uncover the most important property of the Cauchy penalty: The preservation of large CBF values (see marked regions).
To sum up the results, this study showed that the highest SSIM and PSNR are obtained by using a Huber loss with a squared L2 data term and a SmoAbs penalty. However, a Cauchy penalty with Huber loss and Huber data term provides the same SSIM but lower PSNR. This combination is considered superior compared to the first one because of the following reasons:

- The clipping of high CBF values is considered as a knockout criteria.
- A Cauchy penalty and $HL1_{E\text{-}3}HL1_{E\text{-}1}$best meets the theoretic considerations. This leads to a more general model.
- The more than 5 times longer training time (14h10min) compared to SmoAbs $HL1_{E\text{-}3}L2$ (2h40Min) is still acceptable.

3.2.3. Impact of the Filter Size

In this section the kernel size and the number of filters for a Cauchy penalty, L1 data term (Huber loss approximation) and L1 loss is explored. Table 3.5 states a non-expected result, the small model with 8 filters of size 3x3 (8x3x3) attains better metrics than the large model with 48 filters of size 7x7 (48x7x7). From a PSNR point-of-view it is also on par with the 24x5x5 model.

Table 3.5.: Parameter setup, training loss and testing results (CBF, inter-subject test set) for the size selection tests for Cauchy penalty, Huber loss (L1) and Huber loss (smooth L1) data term. All experiments were carried out using 50 L/C-pairs and 24 filters of size 5x5. The stated loss is averaged over the last 60 iterations. Training was early stopped after 500 iterations.

Setup		Training			Testing (WB / GM / WM)	
Kernel Size	#Filters	Time	$\alpha_i \leq 0$	Loss	PSNR in dB	SSIM in %
3x3	8	8h20	0-1	3499.9	22.46 / 20.35 / 18.28	70.46 / 84.19 / 60.27
5x5	24	14h10	0	3482.3	22.46 / 20.35 / 18.26	70.91 / 84.40 / 60.31
7x7	48	81h40	0-15	3510.2	22.33 / 20.08 / 18.42	67.83 / 82.89 / 58.77

The CBF maps (Figure 3.8) emphasize the quantitative results, i.e 48x7x7 gives blurry images whereas 8x3x3 and 24x5x5 shows comparable results. Since a larger model should perform at least as well as a small model, it is likely that the found solution is not a very deep local minimum or even no minimum at all. This is verified by the training loss which is higher than for the other two models.

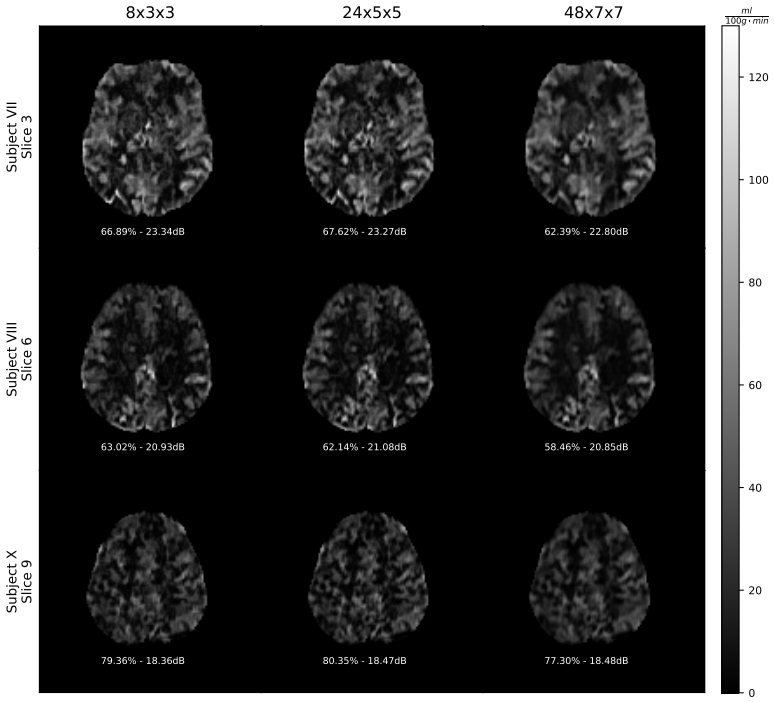

Figure 3.8.: Model size evaluation. CBF maps, SSIM and PSNR for different slices using 50 L/C-pairs. All models use a $HL1_{E\text{-}3}$ loss, a Cauchy penalty and a $HL1_{E\text{-}1}$ data term.

Although the 8x3x3 model is 40% faster to train, all further experiments are carried out with a 24x5x5 model due to the slightly better SSIM metrics.
An aspect which was not focused on yet is the number of negative or zero elements in $\boldsymbol{\alpha}$. Non-positive α_i during the last few iterations, where the model is assumed to be close to a minimum, means that the number of filters could be reduced without loss of performance. Nevertheless, the reduction of the number of filters from 24 to 22 (Cauchy $\mathrm{HL1_{E\text{-}3}HL1_{E\text{-}1}}$) would be only around 10% and thus is considered as negligible.
The training time mainly depends on the used data term function as well as on the used penalty (see training time results). In contrast, the used loss does not affect the training time at all. This is explained by the computationally costly solution of the LLP, which is further hardened by the use of non-convex penalties and non-quadratic data term functions on the one hand and the similarity between a smooth L1 loss and a squared L2 loss on the other hand.

3.2.4. SSIM loss and additional Regularization

In the previous section it was shown that a L1 loss produces sharper images than a squared L2 loss and thus obtains a higher SSIM. Although highly non-linear and non-convex, a SSIM loss model is trained as well and compared to the L1 loss model. The hyperparameters are chosen according to the findings in the previous section. Hence, a Huber loss approximation of a center smooth L1 norm as data term function, a Cauchy penalty function and 24 filter of size 5x5 are used. Further, the effect of an additional regularization factor is investigated. To find a regularization factor that is stable across different noise levels, all experiments are carried out for $N_{ave} = \{30, 40, 50, 60, 80, 100\}$.

Table 3.6.: Training and test results (CBF, inter-subject test set) for SSIM and L1 loss both with L1 data term, Cauchy penalty, 50 L/C-pairs and 24 filters of size 5x5. The stated loss is averaged over the last 60 iterations. Training was early stopped after 500 iterations.

Setup	Training			Testing (WB / GM / WM)	
Loss	Time	$\alpha_i \leq 0$	Loss	PSNR in dB	SSIM in %
SSIM	16h30	3-7	0.71	22.21 / 20.22 / 17.72	71.27 / 84.41 / 59.73
$\mathrm{HL1}_{E-1}$	14h10	0	3482.3	22.46 / 20.35 / 18.26	70.91 / 84.40 / 60.31

Table 3.6 and Figure 3.9 show the training and testing results for the SSIM model in comparison to the L1 loss model. The SSIM model takes slightly longer to finish the 500 training iterations, but attains a higher SSIM (71.27%). This result has to be taken with care, because both models perform on par in gray matter and in white matter the L1 model performs even better. Therefore, the increased whole brain SSIM for the SSIM loss model must be reasoned in regions which are neither in GM, nor in the WM mask. The relevance of this phenomenon is not clear, because it might be founded in imperfect brain masks, i.e. not all white (gray) matter is necessarily part of the WM (GM) mask.
The visual quality of depicted CBF maps agree with the quantitative results. The maps of the SSIM loss model appear sharper but also contain more noise than the L1 loss model's CBF maps.

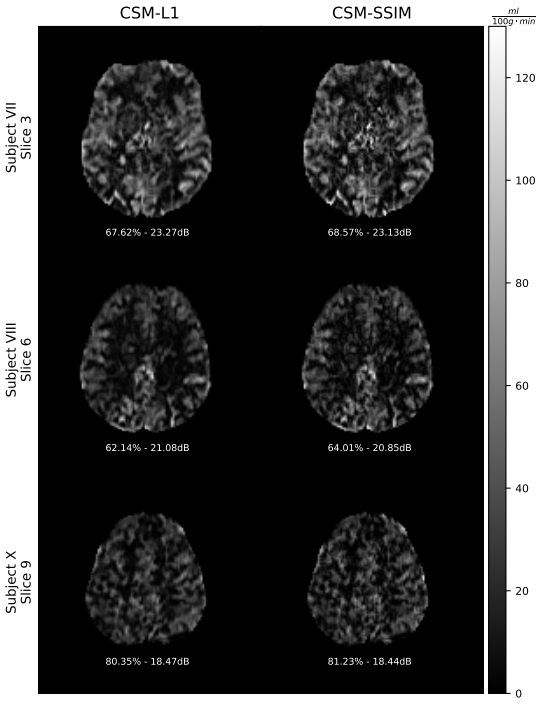

Figure 3.9.: SSIM vs L1 loss CSMs. CBF maps, SSIM and PSNR for different slices and 50 L/C-pairs. Both models use a Cauchy penalty and a L1 data term.

Figure 3.10 contains the SSIM and PSNR graphs for different regularization parameters using the L1 and SSIM loss model with different numbers of L/C-pairs (N_{ave}). It is observable that for the L1 loss, the best SSIM is obtained for η be-

tween 0.7 and 0.8 and the best PSNR for η between 0.8 and 1.0. The SSIM model is in general less dependent to the choice of η. Here, the best SSIM is obtained for $\eta = 0.9$ and the best PSNR for $\eta = [0.9, 0.95]$. In theory, the SSIM loss model should have best SSIM for a regularization factor of 1. This contradiction is not clear at all. However, it is most likely that the overregularized learning is based in a lower GM/WM ratio in the 64x64 patches for learning than in the 128x128 patches for testing. A more detailed explanation is given in section 4.3.2 "Additional Regularization of the CSM". For the final tests a regularization factor of $\eta = 0.8$ for the L1 model and $\eta = 0.9$ for the SSIM model is chosen.

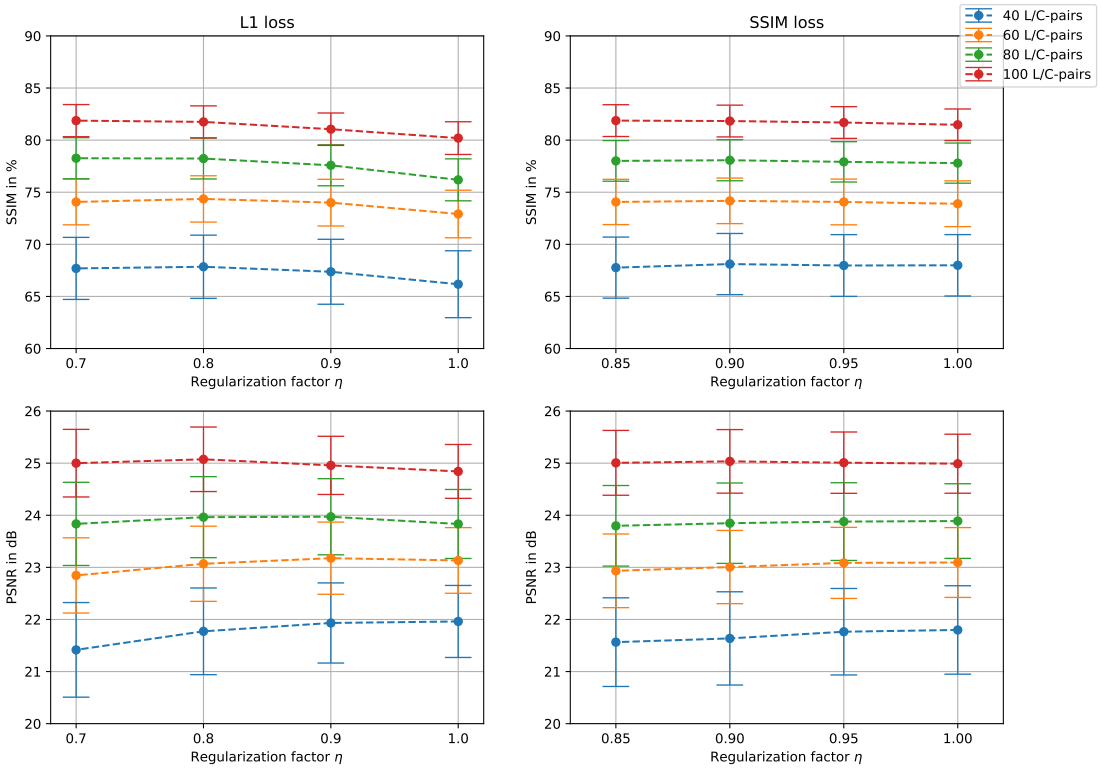

Figure 3.10.: SSIM and PSNR results of the SSIM and L1 loss model for different regularization factors. The error bars indicate the averaged (over subjects and slices) estimated standard deviation for the different variations of L/C-pairs.

3.3. Hyperparameters of the VN

3.3.1. Model Size

To determine the most appropriate model size in terms of diffusion stages, number of filters and kernel size, models with $\{3, 5, 7, 9\}$ stages and 8 filters of size 3x3, 24 filters of size 5x5, 48 filters of size 7x7 and 80 filters of size 9x9 were trained (squared L2 and L1 loss) and evaluated (PSNR and SSIM). No explicit results are depicted here, but the outcome of the tests are summarized. All experiments were carried out with 50 L/C-pairs as input, a squared L2 data term and the InARow test set.
In general larger models yield a lower loss during training. When using a squared L2 loss, 5x5 and 7x7 kernel models perform better on test data than 3x3 kernel models. However, for more than 5 stages, 7x7 models started to overfit to the training data. When using a L1 loss, the 5x5 model with 5 stages attains a higher SSIM than all 7x7 models. Therefore, the 24x5x5 model with 5 stages is considered as an appropriate choice which yields sharp results and is robust against overfitting. Additionally, some of these experiments were also performed using just the first 50 L/C-pairs for training. As assumed theoretically, this increased the effects of overfitting (see section 2.3.4 "Datasets").

3.3.2. Loss and Data Term

In analogy to the section above, the results are only summarized here. All preliminary results are given in Appendix B.
For a squared L2 data term, the highest PSNR values are obtained using a squared L2 loss. However, this combination leads to blurry images and consequently to a low SSIM. A tradeoff between noise removal and sharp images is obtained by using a L1 loss. In contrast to a L2 loss, a SSIM based loss delivers the best SSIM metrics but due to less noise reduction also a lower PSNR.
The drawback of the SSIM is its single scale definition. In this definition a 11x11 SSIM patch is used which averages out noise deviations. Hence, less noise is suppressed when optimizing the single scale SSIM. The msSSIM overcomes this issue

by computing the SSIM on several scales. Also hybrid loss forms like L1 with SSIM (L1SSIM) or L1msSSIM were tested. However, a msSSIM loss generally worked best.
For a msSSIM based loss additionally to the L1 and L2 data terms a log-Cauchy and a center-smoothed root (csRoot) were used. The best quantitative results were obtained for a squared L2 and a center-smoothed root, which performed very similar. However, the latter has the theoretical advantage of being heavy-tailed and thus fits better to the data characteristics. Therefore, the center-smoothed root function is chosen for all remaining experiments.
Additionally to the previous investigation of spatial denoising, the denoising performance of the VN with additional temporal information as input was evaluated. Therefore, also a squared L2 and L1 with estimated temporal voxelwise variances (corresponds to a Mahalanobis distance) and temporal data terms (all L/C-pairs as input) were tested. They yield a working denoising, but the obtained metrics and CBF maps were not able to compete with non-temporal data terms at all. This might be reasoned in a more difficult training and hence a bad local solution. Nevertheless, the choice of the model based on the test data has to be considered critical: Firstly, TensorFlow is not deterministic at all and secondly, the test data varies a lot for different combinations of L/C-pairs. Hence, another model might attain better metrics for different test conditions. In general, statistical testing would be used to cope with this uncertainty, but due to variances caused by TensorFlow, by the slice quality and by 'good' and 'bad' L/C-pair combinatios, this is not profitably for this task. Therefore, the model with the best (not-significant) results and the best theoretic agreement was chosen.

3.4. Final Test and Comparison to TGV

In this section the final results for the chosen CSM and VN models are shown for intra-subject and inter-subject data. In addition, they are compared with the results of stTGV variants on the basis of the InARow dataset.

3.4.1. Inter-Subject Test Set

Figure 3.11 shows the SSIM in gray matter, white matter and in the whole brain for the CSM-L1 loss, CSM-SSIM loss and the VN model for {30,40,50,60,70,80,90,100} L/C-pairs. The black curves indicate the input of the models i.e the temporal mean over the used PWI.

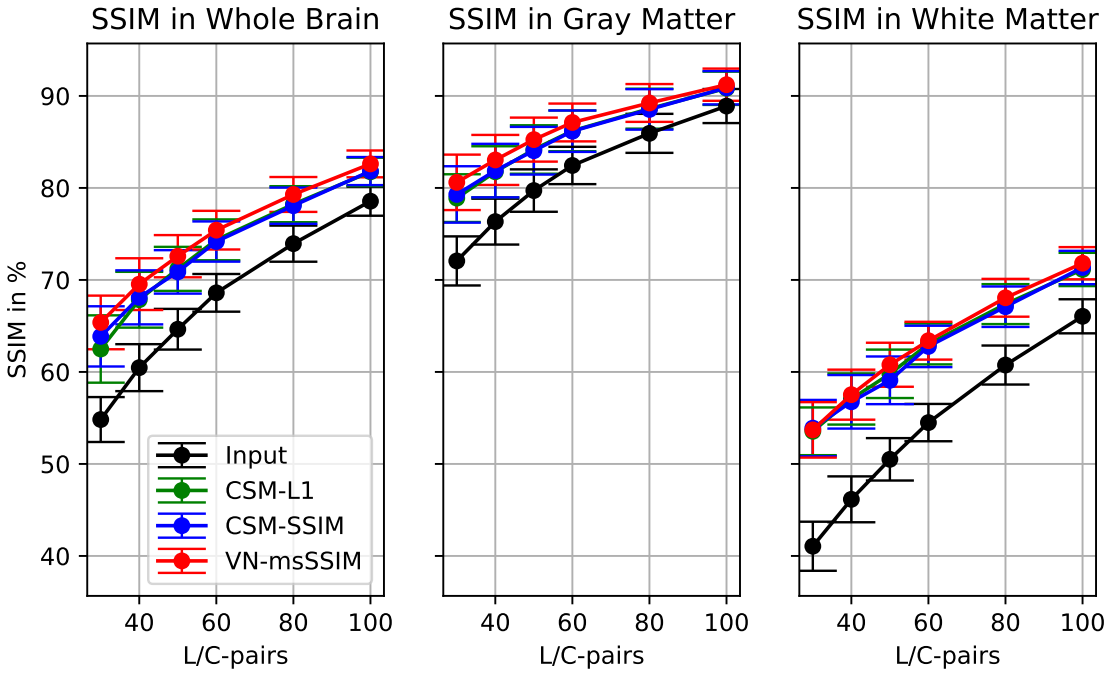

Figure 3.11.: Inter-Subject Testing. SSIM in GM, WM, and WB, for the CSM-L1, CSM-SSIM and VN model using a different number of PWIs. The errorbars indicate the standard deviation over the used different L/C-pair combinations, which is averaged over all slices and subjects.

It is clearly visible that all models outperform the input by a margin larger than the estimated standard deviation. In addition, all models attain better quantitative

results in gray matter than in white matter. This is expected because in gray matter the blood supply is approximately 3 times as large as in white matter. Regardless of matter and the number of used PWI, the variational network attains a higher SSIM than the CSM models. Further, it is observable that the SSIM increases monotonically with the number of L/C-pairs. However, the increase in SSIM gets lower for higher numbers of averages due to the $\sqrt{N}$ improvement. I.e. the gap between 30 and 50 pairs is larger than for 80 and 100 pairs. Additionally, also a decrease of variation for more L/C-pairs is visible.

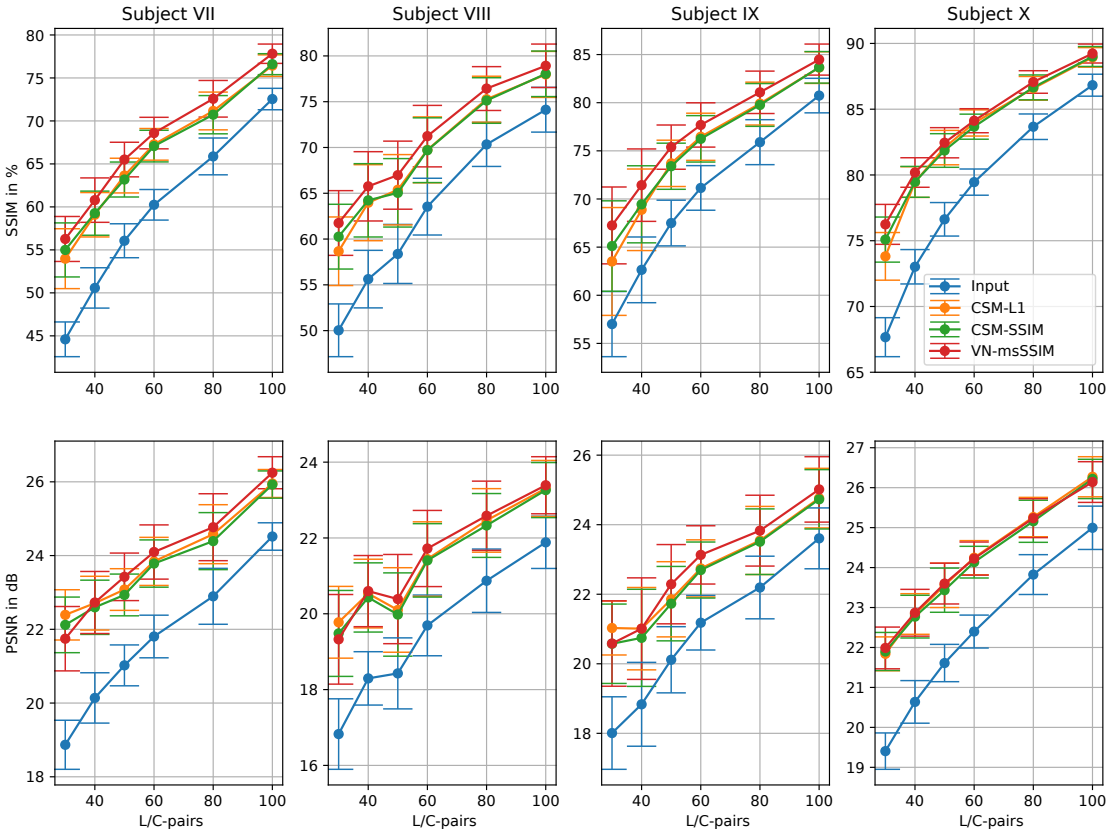

Figure 3.12.: Inter-Subject Testing. SSIM in whole brain for all testing subjects, for the CSM-L1, CSM-SSIM and VN model using a different number of PWIs. The errorbars indicate the standard deviation over the used different L/C-pair combinations, which is averaged over all slices.

Figure 3.12 shows the SSIM and the PSNR for different noise levels (L/C-pairs) and all subjects of the inter-subject test set. Three new observations can be made here: First, the single subjects attain quite different metrics. For example, subject X has

approximately the same SSIM for 30 L/C-pairs as Subject VII for 100 L/C-pairs. In contrast, the PSNR of both subjects is nearly identical. Second, for subject VIII all models obtain a higher PSNR for 40 pairs than for 50 pairs. As this behavior is not observable for the SSIM, it might be the case that the input data of subject VIII is more noisy than expected. This might leads to an underregularization in terms of PSNR. Third, the VN yields a lower PSNR for 30 L/C-pairs for subject VII - IX than the CSM models. This might be reasoned in the used msSSIM loss, which does not guarantee high PSNR values.
Table 3.7 shows the SSIM and PSNR, computed over all subjects and for different numbers of L/C-pairs.

Table 3.7.: Inter-Subject Testing. SSIM and PSNR for the final models using a different number of L/C-pairs. The error stated is the standard deviation over the input variations, averaged over all slices and subjects.

Metric	L/C-pairs	Input	CSM-L1	CSM-SSIM	VN
SSIM in %	30	54.83 ± 2.44	62.49 ± 3.66	63.86 ± 3.27	65.38 ± 2.91
	40	60.47 ± 2.55	67.85 ± 3.03	68.11 ± 2.94	69.54 ± 2.81
	50	64.64 ± 2.21	71.20 ± 2.39	70.87 ± 2.36	72.58 ± 2.29
	60	68.60 ± 2.05	74.36 ± 2.22	74.17 ± 2.18	75.41 ± 2.10
	80	73.94 ± 1.96	78.23 ± 1.96	78.07 ± 1.97	79.29 ± 1.90
	100	78.56 ± 1.58	81.75 ± 1.54	81.83 ± 1.53	82.61 ± 1.46
PSNR in dB	30	18.28 ± 0.77	21.26 ± 0.71	21.02 ± 0.88	20.91 ± 0.95
	40	19.48 ± 0.78	21.77 ± 0.83	21.64 ± 0.89	21.80 ± 0.96
	50	20.29 ± 0.73	22.15 ± 0.83	22.02 ± 0.82	22.42 ± 0.87
	60	21.27 ± 0.64	23.07 ± 0.72	23.01 ± 0.70	23.29 ± 0.75
	80	22.45 ± 0.75	23.96 ± 0.78	23.85 ± 0.77	24.11 ± 0.83
	100	23.75 ± 0.62	25.07 ± 0.62	25.03 ± 0.61	25.20 ± 0.66

Figure 3.13, 3.14 and 3.15 show the CBF maps for subject VII slice 3, subject VII slice 6 and subject X slice 9. Although the visual difference between the CSM (SSIM loss) and the VN is in general small, the VN's CBF maps appear sharper and less noisy.

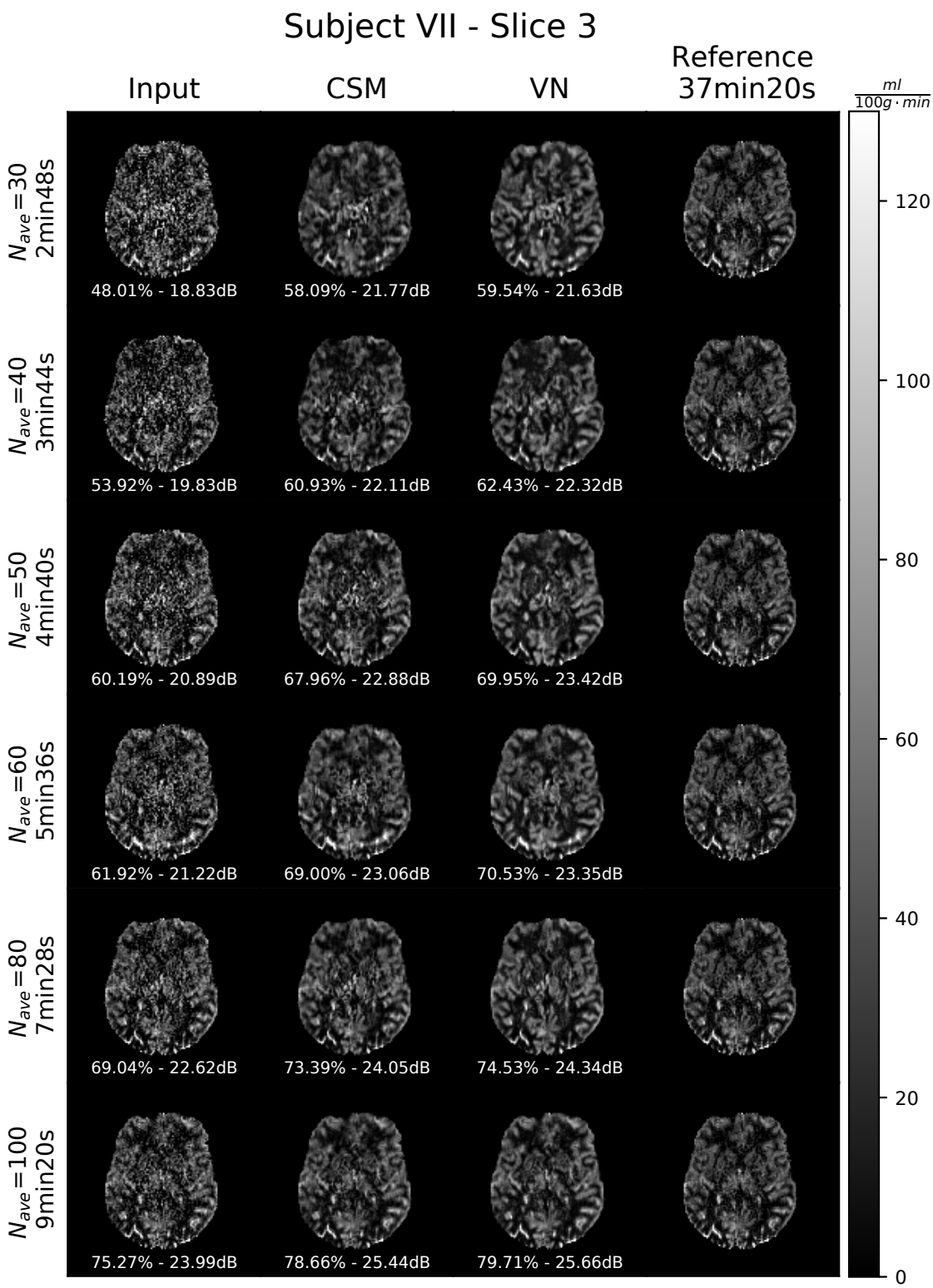

Figure 3.13.: CBF maps, SSIM and PSNR for different numbers of L/C-pairs (N_{ave}) of subject VII slice 3 (inter-subject test set) for the CSM-SSIM and VN model. For N_{ave}=60 a chemical shift artifact is visible.

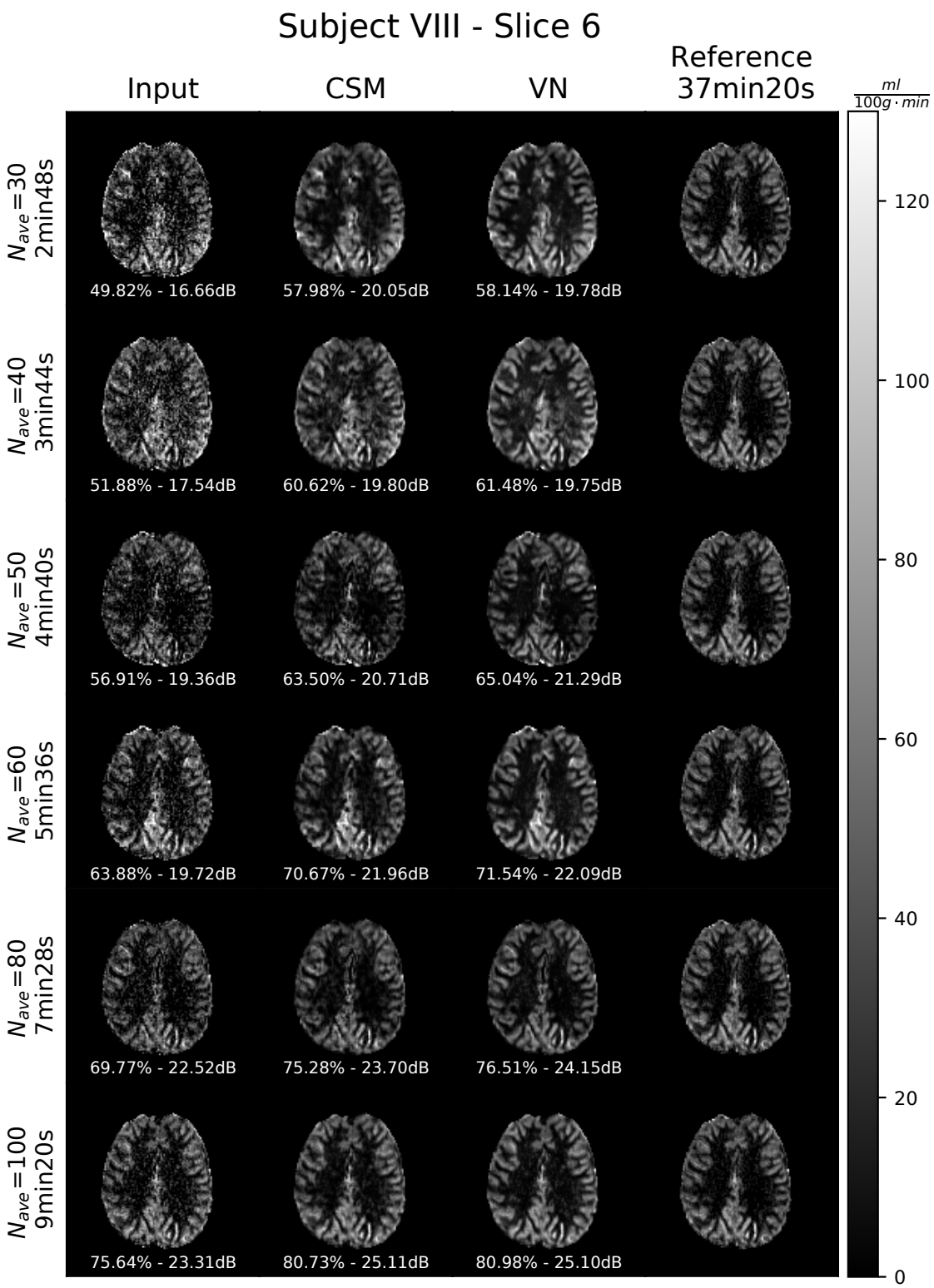

Figure 3.14.: CBF maps, SSIM and PSNR for different numbers of L/C-pairs (N_{ave}) of subject VIII slice 6 (inter-subject test set) for the CSM-SSIM and VN model.

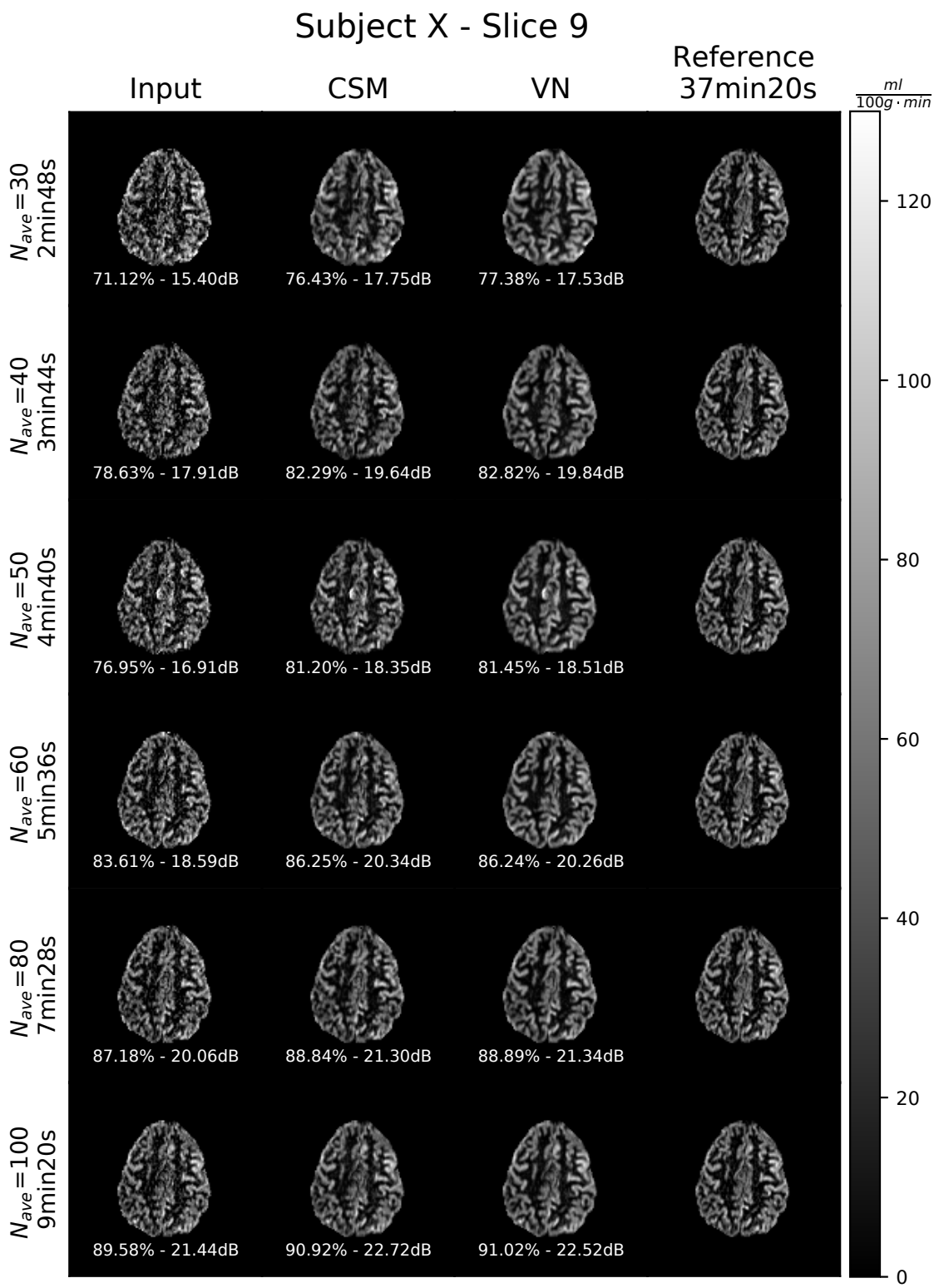

Figure 3.15.: CBF maps, SSIM and PSNR for different numbers of L/C-pairs (N_{ave}) of subject X slice 9 (inter-subject test set) for the CSM-SSIM and VN model.

3.4.2. Intra-Subject Test Set

The aim of this test set is to evaluate if there are any differences in behavior of the models compared to the inter-subject test set. Figure 3.16 shows the SSIM in GM, WM and WB for different models and a different number of L/C-pairs.

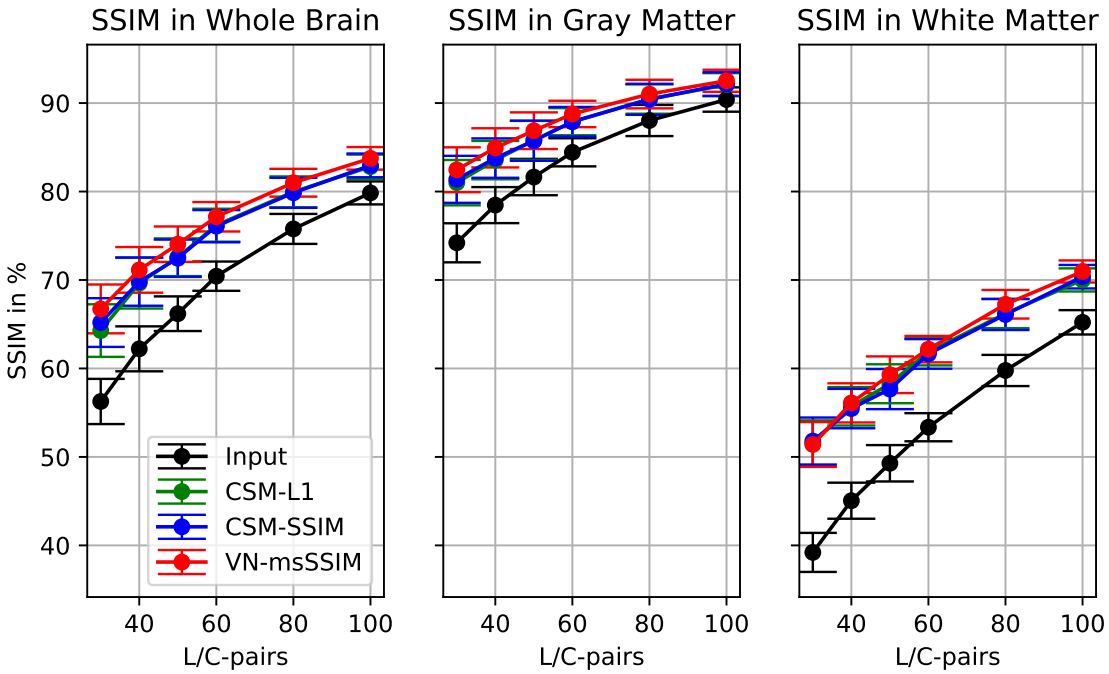

Figure 3.16.: Intra-Subject Testing. SSIM in GM, WM and WB, for the CSM-L1, CSM-SSIM and VN model using a different number of PWIs. The errorbars indicate the standard deviation over the used different L/C-pair combinations, which is averaged over all slices and subjects.

Compared to the inter-subject case, the trends and the behavior of the models are the same. The single subject results are shown in Figure 3.17. Principally these results emphasize the same behavior as the curves for the inter-subject test case. Table 3.8 indicates a slightly higher SSIM (1.1%-1.9%) and PSNR (0.6dB-1.1dB) for the intra-subject test set than for the inter-subject test set. As this is also the case for the input data, the better metrics can be assumed to be caused by a lower noise level of the intra-subject test set.
Figure 3.18, 3.19 and 3.20 show the CBF maps for subject I slice 9, subject II slice 6 and subject V slice 3. In analogy to the inter-subject test case, the VN attains sharper and less noisy CBF maps than the CSM.

Table 3.8.: Intra-Subject Testing. SSIM and PSNR for the final models using a different number of L/C-pairs. The error stated is the standard deviation over the input variations, averaged over all slices and subjects.

Metric	L/C-pairs	Input	CSM-L1	CSM-SSIM	VN
SSIM in %	30	56.27 ± 2.55	64.28 ± 2.97	65.19 ± 2.75	66.73 ± 2.76
	40	62.22 ± 2.54	69.64 ± 2.87	69.82 ± 2.73	71.15 ± 2.58
	50	66.19 ± 1.96	72.57 ± 2.13	72.46 ± 2.11	74.05 ± 2.00
	60	70.43 ± 1.65	76.20 ± 1.86	76.08 ± 1.82	77.15 ± 1.66
	80	75.77 ± 1.69	79.96 ± 1.75	79.86 ± 1.70	81.00 ± 1.57
	100	79.84 ± 1.30	82.81 ± 1.37	82.95 ± 1.34	83.75 ± 1.28
PSNR in dB	30	18.90 ± 0.68	21.84 ± 0.66	21.72 ± 0.69	21.56 ± 0.81
	40	20.24 ± 0.62	22.53 ± 0.69	22.46 ± 0.70	22.65 ± 0.74
	50	21.15 ± 0.65	23.04 ± 0.65	22.90 ± 0.70	23.34 ± 0.74
	60	22.11 ± 0.52	23.97 ± 0.60	23.90 ± 0.60	24.20 ± 0.60
	80	23.44 ± 0.62	25.03 ± 0.64	24.90 ± 0.63	25.17 ± 0.68
	100	24.61 ± 0.52	25.91 ± 0.52	25.89 ± 0.50	26.07 ± 0.58

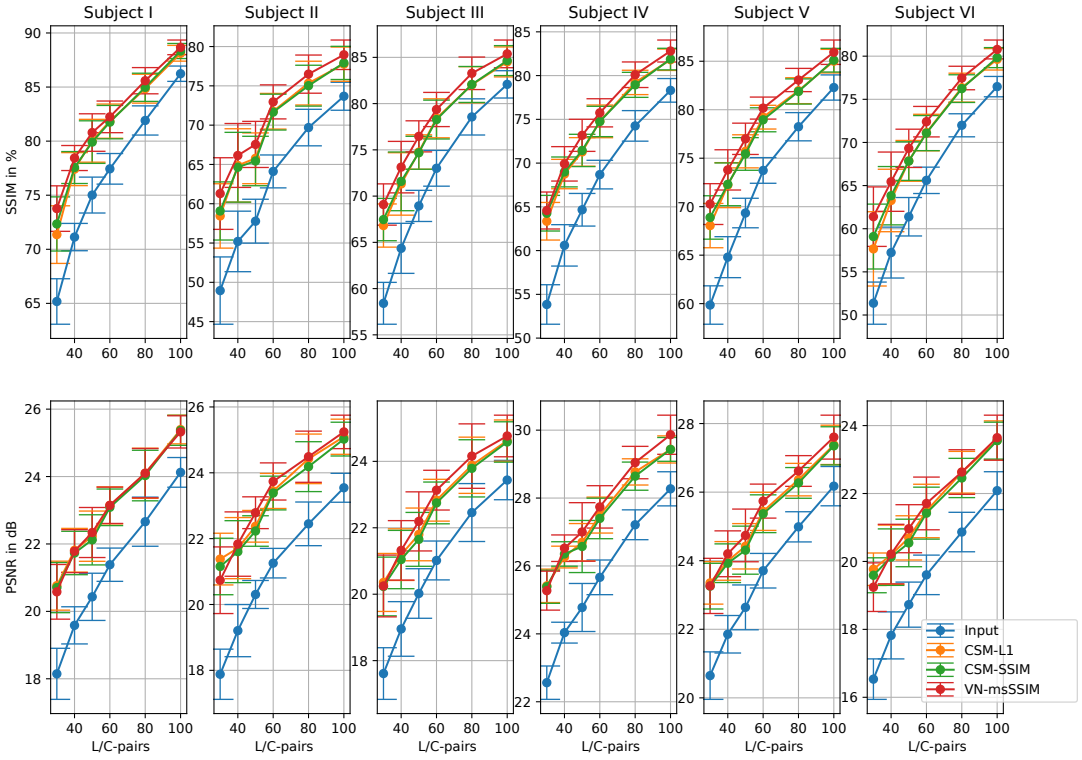

Figure 3.17.: Intra-Subject Testing. SSIM in whole brain for all testing subjects, for the CSM-L1, CSM-SSIM and VN model using a different number of PWIs. The errorbars indicate the standard deviation over the used different L/C-pair combinations, which is averaged over all slices.

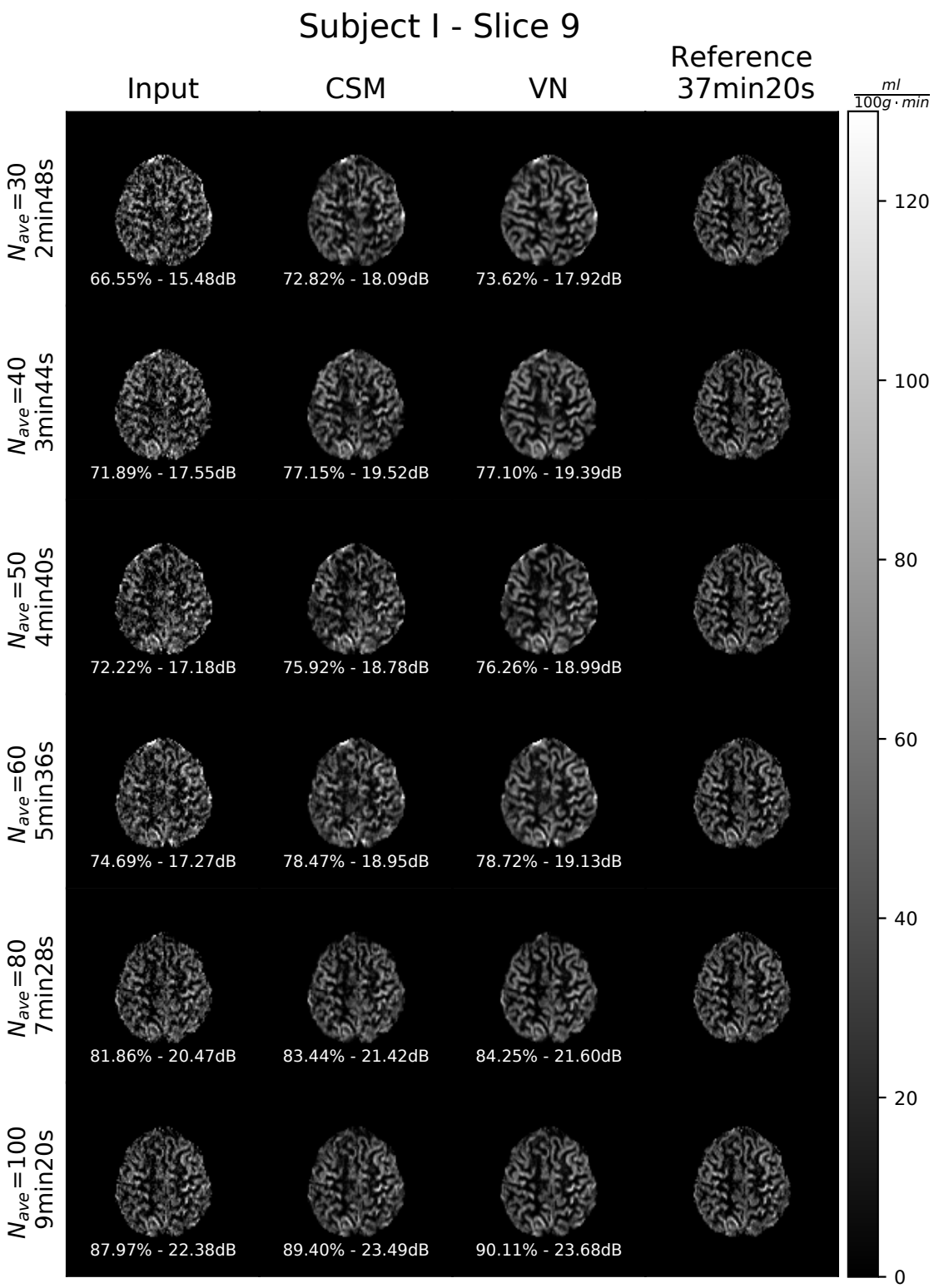

Figure 3.18.: CBF maps, SSIM and PSNR for different numbers of L/C-pairs (N_{ave}) of subject I slice 9 (intra-subject test set) for the CSM-SSIM and VN model.

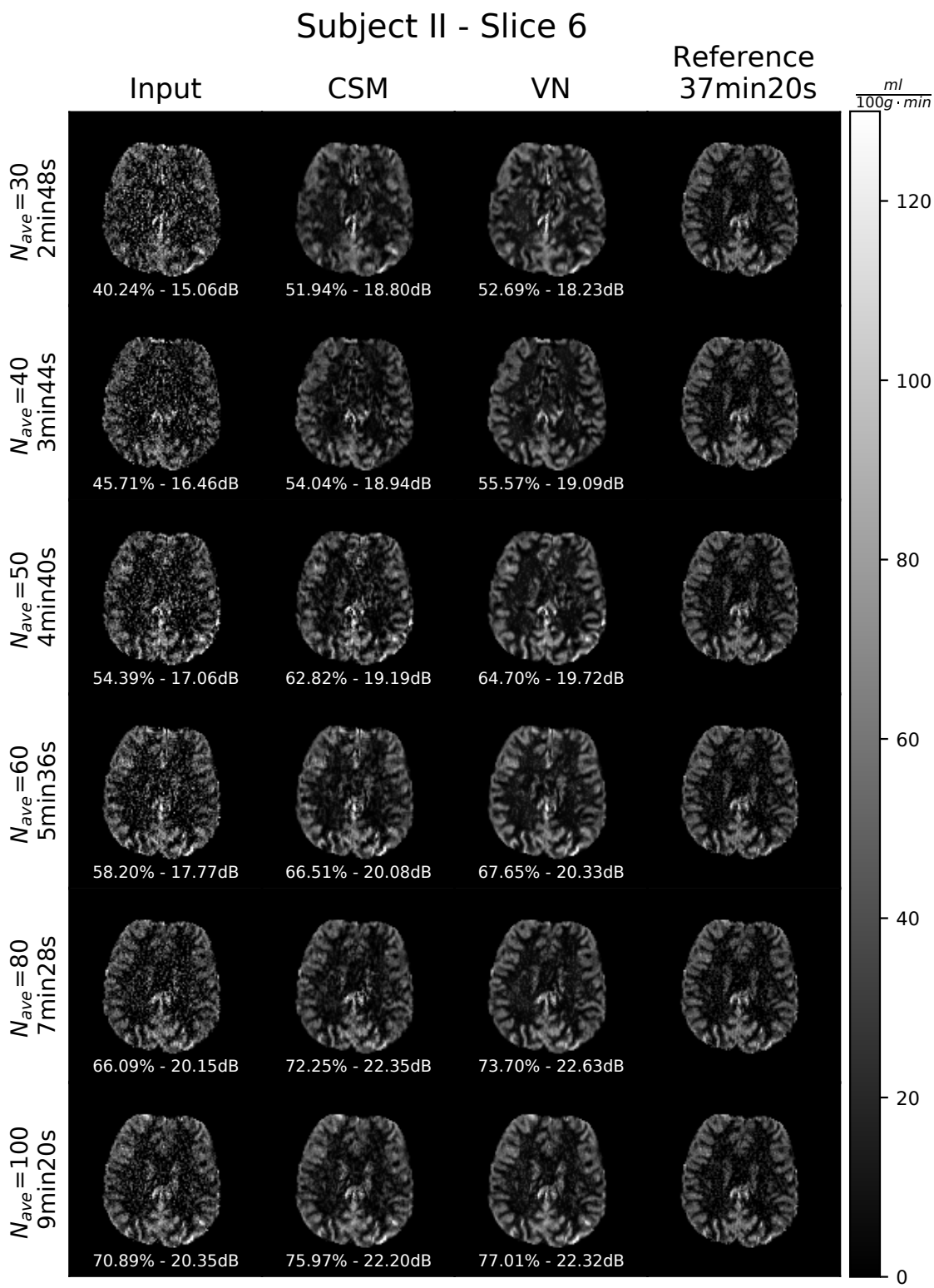

Figure 3.19.: CBF maps, SSIM and PSNR for different numbers of L/C-pairs (N_{ave}) of subject II slice 6 (intra-subject test set) for the CSM-SSIM and VN model.

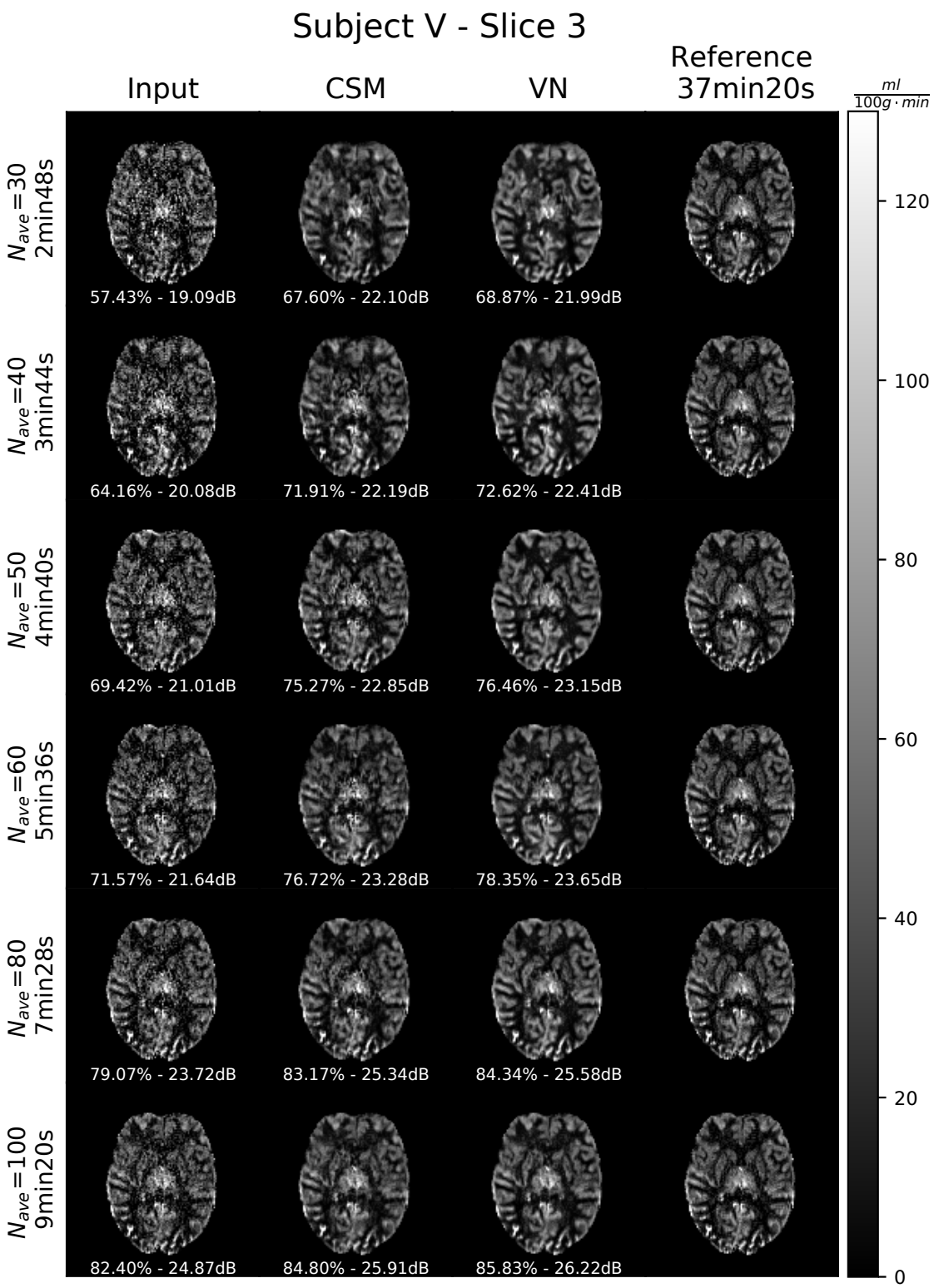

Figure 3.20.: CBF maps, SSIM and PSNR for different numbers of L/C-pairs (N_{ave}) of subject V slice 3 (intra-subject test set) for the CSM-SSIM and VN model.

3.4.3. Comparison to TGV

In this section, the final models are compared to different variants of stTGV denoising used by Spann et al. [16]. Beside of the full spatio-temporal approach (TGV-L1-LC-temporal), also a variant without temporal information (TGV-L1-LC) and without L/C-separation (TGV-L1-dM-temporal) is tested. In addition, also a spatial dependent squared L2 data terms (Mahalanobis L2) is used in a 2D and 3D setting.

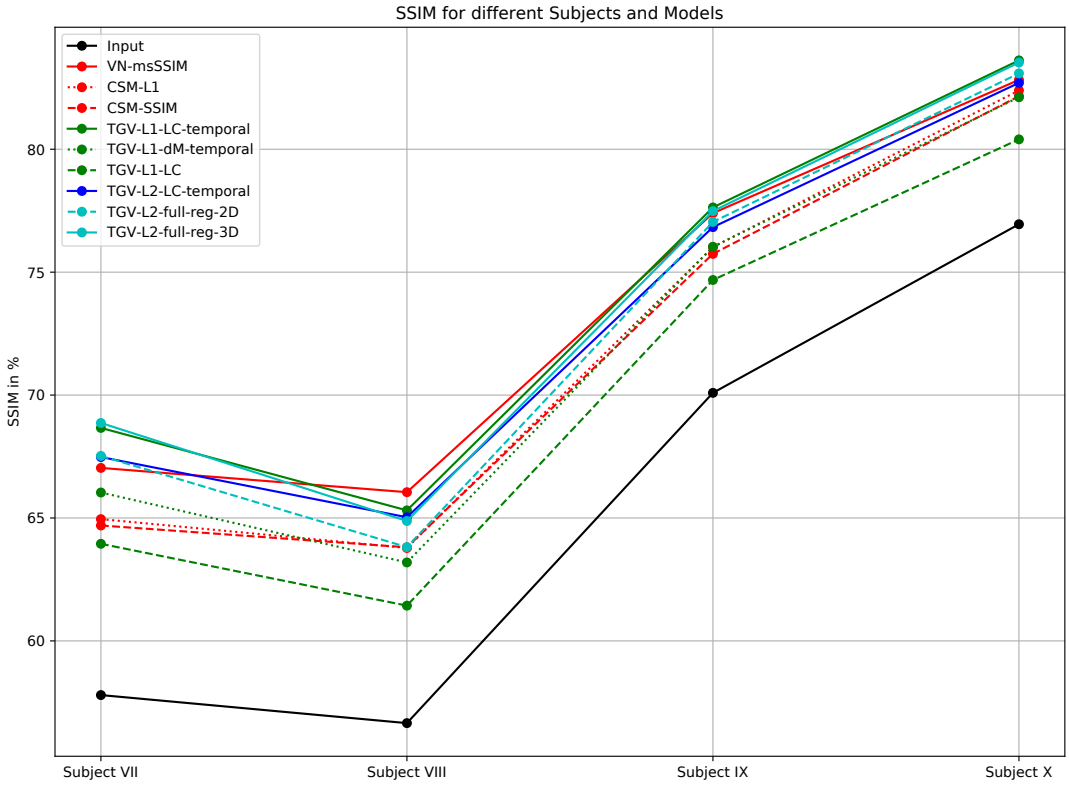

Figure 3.21.: SSIM based comparison between the learned models (VN and CSM) and different TGV models for 50 L/C-pairs.

Figure 3.21 and 3.22 show the SSIM and the PSNR for all subjects (InARow test set). Each model performs clearly better than the input. On the basis of SSIM, both CSM perform worse than the best TGV models. The VN attains a higher SSIM for Subject VIII, a comparable for subject IX and a lower SSIM for the

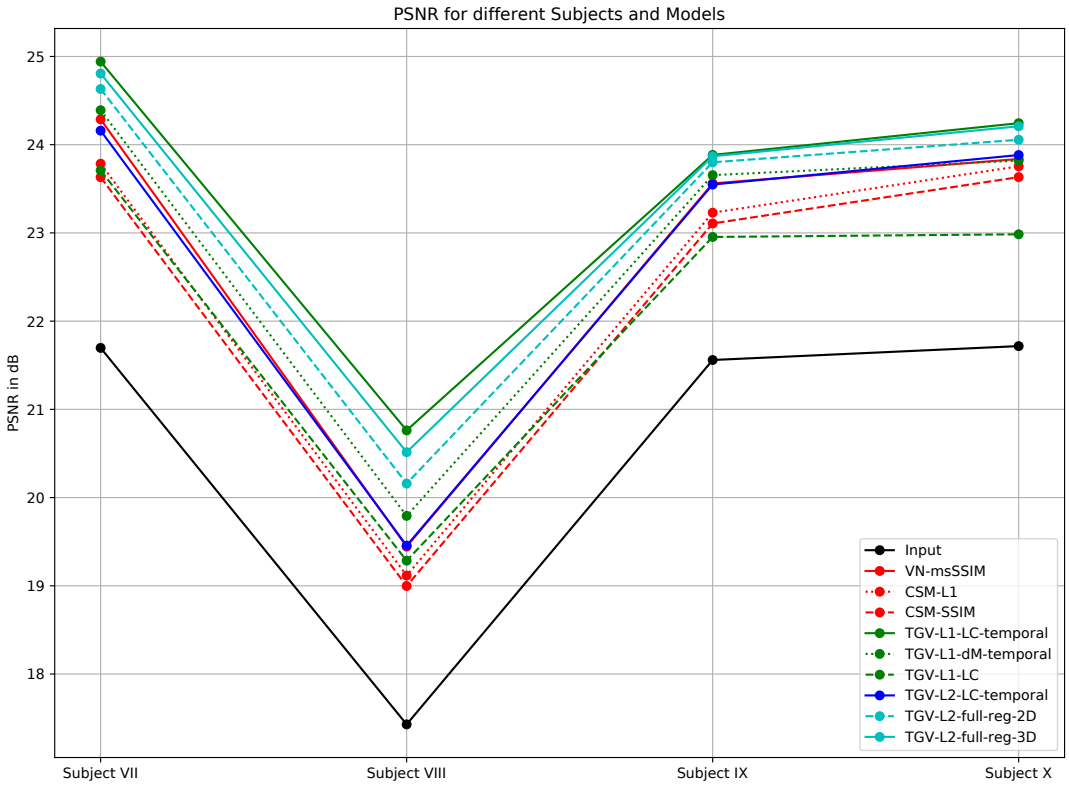

Figure 3.22.: PSNR based comparison between the learned and TGV models for 50 L/C-pairs.

remaining two subjects. On the basis of PSNR, the TGV models outperform the learned models. The results for all subjects are summarized in Table 3.9.

Table 3.9.: SSIM and PSNR results for 50 L/C-pairs for the learned models (VN and CSM) and the TGV models.'C' indicate a CSM model, 'T' a TGV model. 'dM' states temporal TGV without L/C-splitting and 'LC' termes non-temporal TGV. '2D' and '3D' stand for the 2D and 3D spatial regularized TGV models.

Metric	Input	VN	C-L1	C-SSIM	T-L1	T-L1-dM	T-L1-LC	T-L2	T-2D	T-3D
SSIM in %	65.37	73.33	71.79	71.60	73.81	71.84	70.12	73.01	72.87	73.69
PSNR in dB	20.60	22.78	22.47	22.34	23.46	22.91	22.23	22.76	23.16	23.35

Figure 3.23 compares the learned VN and the manually tuned stTGV. Due to their very different formulation, a clear difference between the models is visible. For the stated slices, the VN seems to suppress more noise but preserves less edge information than the TGV model.

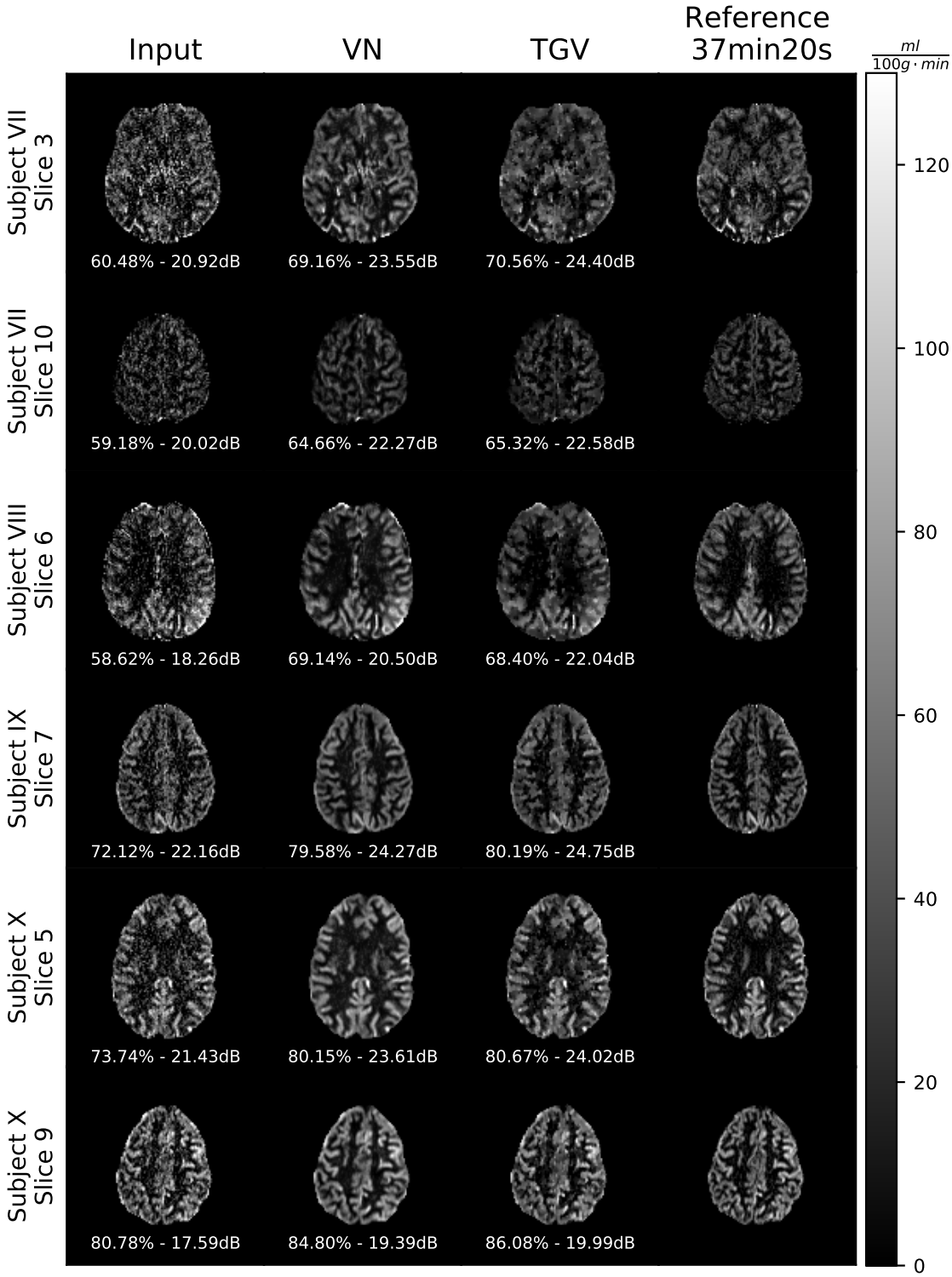

Figure 3.23.: CBF maps, SSIM and PSNR for different slices of the inter-subject test set for the VN and TGV (L1-LC-temporal) model and 50 L/C-pairs.

3.4.4. Edge preservation

To examine the models's capability of preserving edges, Figure 3.24 shows the TV maps for the input, the reference and the denoised CBF maps using the CSM, VN and stTGV. Large values indicate an large intensity change between the center pixel and its neighborhood, like f.e. at edges or in noisy regions. These maps show the highest supression of noise like structures in case of the VN. The TV maps show narrower edges for the stTGV model. I.e. the results of the VN seems to be less noisy but more blurry than the results of the stTGV approach. The CSM suppresses less noise than the other two models and blures edges about the same as the VN.

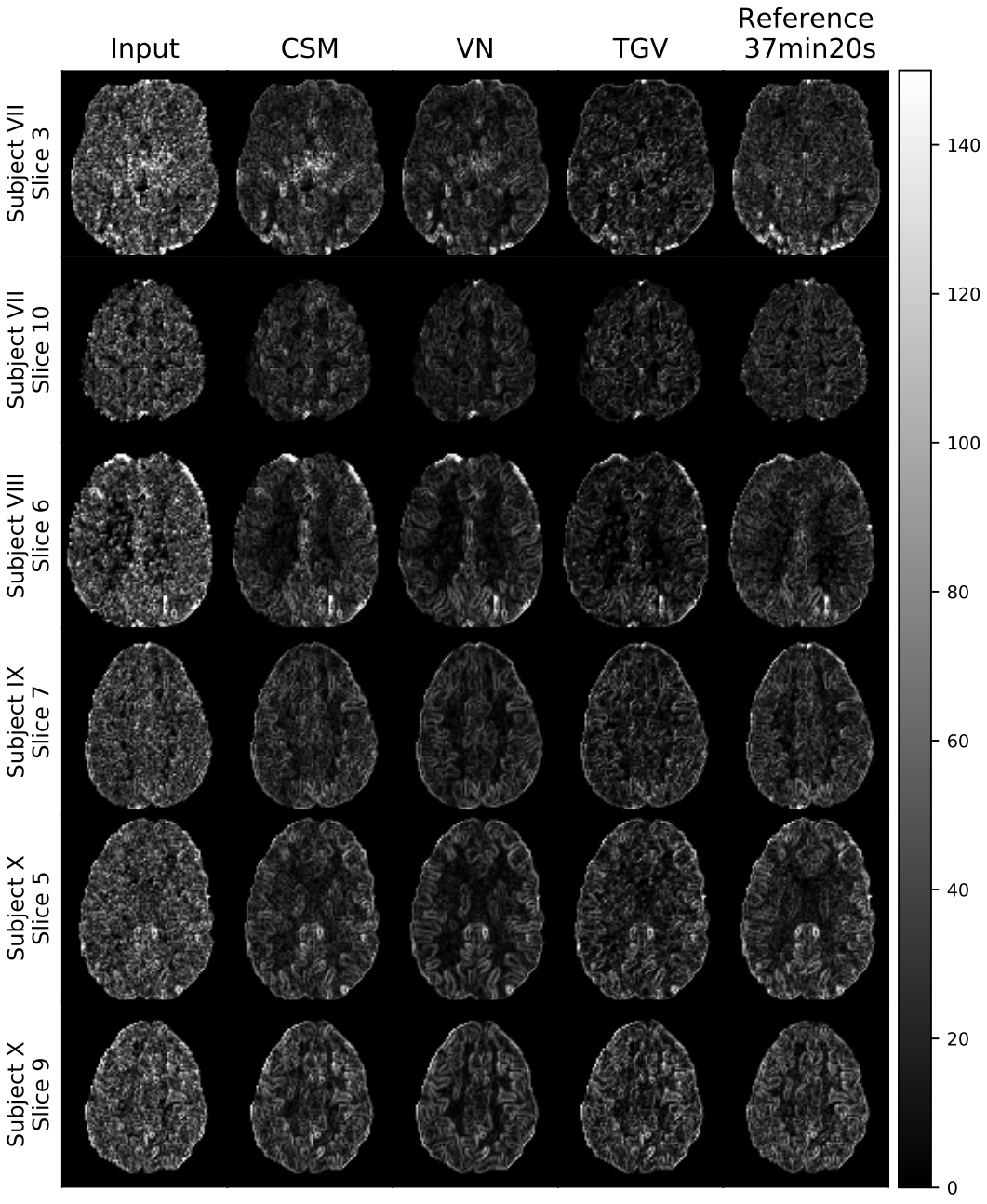

Figure 3.24.: TV maps of input, reference and denoised CBF maps. The TV maps are obtained by summing the absolute value of the gradient maps using forward differences in x and y direction.

3.4.5. Learned Parameters

Figure 3.25 and 3.26 depict the 24 learned kernels for the two CSM models. Figure 3.27 and 3.28 show the learned kernels and the corresponding activation and penalty functions of the VN in the first and fifth stage. Some of the stated kernels can be interpreted as gradient filters, some as second order (Laplace) filters, and some of them as edge detectors. However, the majority does not seem to have a particular structur. Many of the learned penalty functions are of quadratic nature and some undergo a heavy-tailed characteristic. These penalties fit with the theoretic expectations, but there are also unsymmetric bimodal penalties.

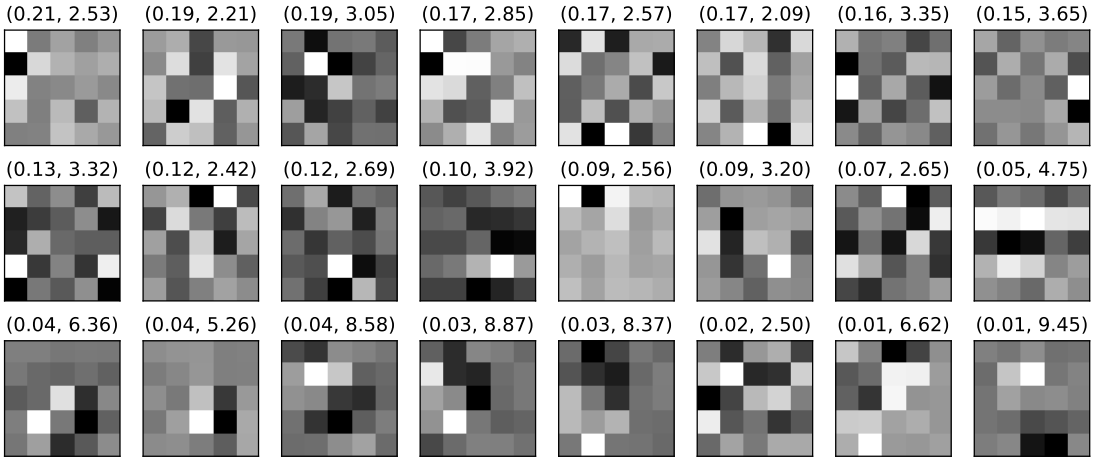

Figure 3.25.: Learned filter kernels for the final CSM with L1 loss. The corresponding weight and norm of the filter is stated in brackets.

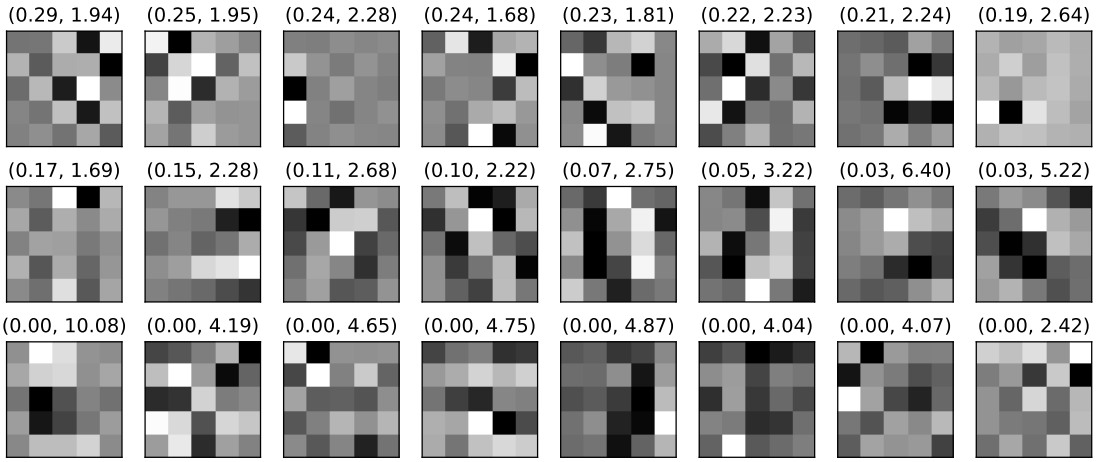

Figure 3.26.: Learned filter kernels for the final CSM with SSIM loss. The corresponding weight and norm of the filter is stated in brackets.

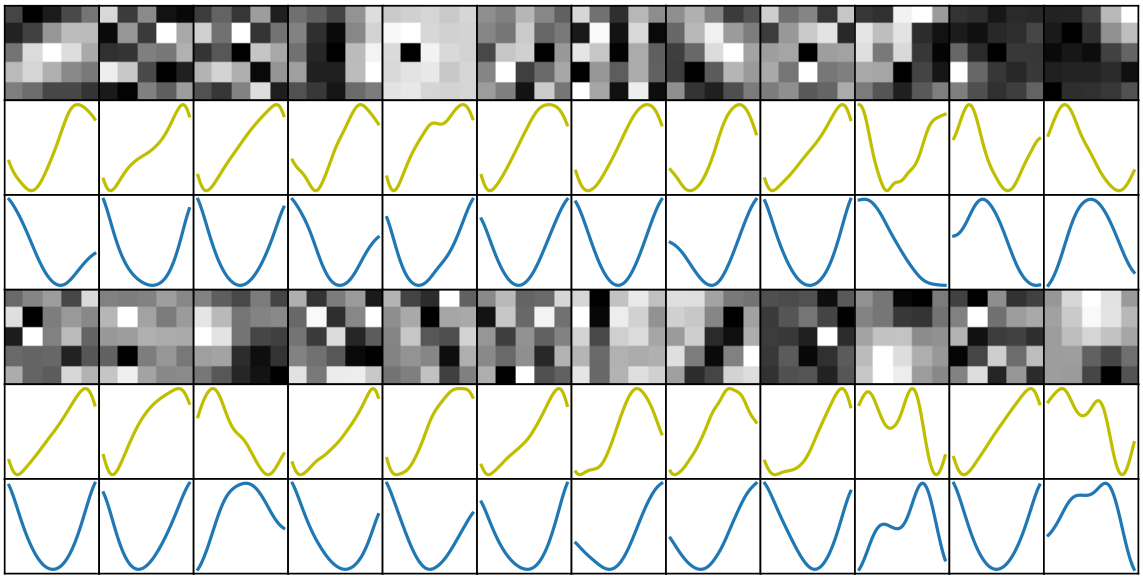

Figure 3.27.: Learned filter kernels, activation function (yellow) and corresponding penalty function (blue) for the VN at the first stage.

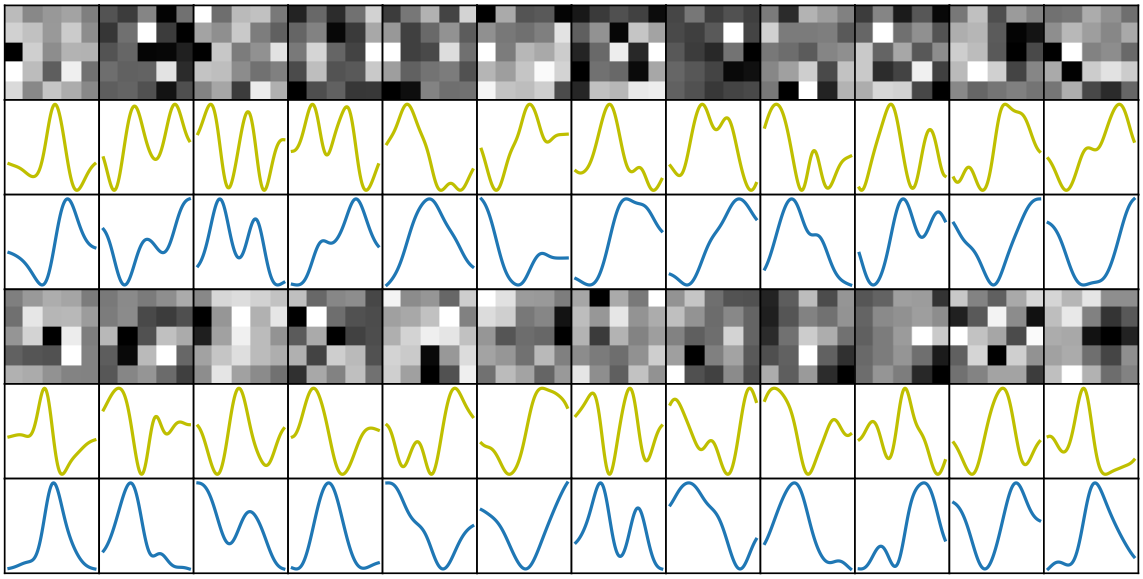

Figure 3.28.: Learned filter kernels, activation function (yellow) and corresponding penalty function (blue) for the VN at the fifth stage.

4. Discussion

4.1. Quality of the learned Solution

In optimization the quality of the found parameters always has to be considered. Is the solution a deep local minimum or even a global minimum? Is it a bad minimum or just a saddle point? Is it robust to noise, i.e. does it generalize well? In case of a convex problem, the solver should converge to a unique global solution immediately. Unfortunately, the used models exploit non-convex penalty functions and non-convex loss functions, leading to a non-convex optimization problem. Consequently, the learned parameters are potentially related to low local minima. Apart from this, very flat regions of the energy landscape can also lead to bad solutions, even in convex settings. The impact of flat regions might become worse when numerical limitations arise. Therefore, badly scaled input values have to be avoided.

4.1.1. Local Optima

The used CSM as well as the VN are non-convex models and thus prone to get stuck in bad local optima or in very flat regions of the energy landscape. A simple way to overcome these issues is to use mini-batches for training. The sampling noise introduced by this method helps to escape from local solutions or flat regions. As mini-batch training is performed, it can be assumed that the found solution is at least a moderately low minimum or valley. An exact evaluation of the quality of the solution is computationally not feasible because this would include the evaluation of the whole energy landscape.

In case of the VN, the batch based learning approach leads to large oscillations of the loss during training. This might be reasoned in large variations of the input noise level for different PWIs of different slices and subjects.
Anyway, there is no guarantee that the training will converge to stationary point. Beside the right training approach the initial values are also very important for a successful training. Bad initial values or hyperparameters can lead to poor convergence and hence to bad results. Fortunately, the proposed hyperparameters have turned out to be very reliable in yielding convergence.

4.1.2. Overfitting

A simple and effective strategy against overfitting is given by reducing the number of learnable parameters, i.e. small models are preferred. The proposed VN utilizes 6725 free parameters (5 stages with 24 filters of size 5x5, each of which is related to a 31 RBF based penalty function as well as a data term weight for each stage), which is pretty little compared to more than 86 million different data points presented during training ($64^2 \cdot 42$ data points per iteration, for each iteration a new training set is used, see 2.3.4 "Datasets"). It is obvious that the CSM with its 600 free parameters is even less prone to overfitting. Additionally, the large variations of the inputs's noise level avoid overfitting too. These more theoretical considerations are proofed by the similar performance of the models on the intra-subject and inter-subject test set. Overfitting would have lead to a better performance of the models on the intra-subject test set, which is highly correlated to the training data.

4.1.3. Input Scaling and Numerical Problems

Float numbers are stored using two values, the mantissa and the exponent. Both of them are represented using a fixed number of bits (8 exponent + 23 mantissa + 1 sign in case of single precision, 11+54+1 for double precision). When performing computations on values with very different scaling, information is lost leading to inaccurate results. A simple way to overcome this issue is by scaling the values to

be in about the same range. In the present case, the intensities of PWIs are in the range of about 0-10 and the final parameters about 0-1. Hence, it is useful to scale down the input before denoising. This scaling yields in general more accurate gradients and thus a faster convergence. Nevertheless, in case of the CSM and its bilevel learning approach, numerics are still a limiting factor, because it is crucial to solve the LLP with a very high accuracy. In this context two phenomenons have to be reported:

In case of using an EstAbs penalty, which approximates a linear function, the additional scaling of the single filters (parameter $\boldsymbol{\alpha}$) is redundant, because the scaling of the filters could be incorporated in the learned kernels. ($\alpha_i|\mathbf{A}_i\mathbf{x}| = |\alpha_i\mathbf{A}_i\mathbf{x}| \; \forall\alpha_i \geq 0$). The existence of this additional parameter and the convexity of the problem leads to a non-strict global minimum, i.e. the single minimum of the function is not a point, but a 24 (length of $\boldsymbol{\alpha}$) dimensional space. The fact that the EstAbs penalty just approximates a linear function turns to above stated equality to be slightly inexact. A unique strict global minimum is the consequence. The combination of both aspects, leads to the assumption that the unique global minimum must be located in an extremely flat 24 dimensional region of the energy landscape. Several trainings with the same EstAbs setting and different random initials have been carried out. Although the same results would be expected, all tests lead to different kernels. Obviously the energy landscape became too flat to perform further gradient descent steps.

As a second example, even if the LLP optimization is subject to an unlimited number of descent steps, the theoretical possible residual of zero is not reached. The gradient simple vanishes before.

These two examples show the impact of numerical errors due to the finite resolution of the data type. Especially when model parameters and gradients get close to the machine epsilon, the approximation of the discrete optimization problem as continuous valued problem becomes inaccurate.

4.2. Optimization of the CSM Parameters

4.2.1. Choice of the HLP Solver

Beside of numerical considerations, the bilevel learning undergoes another issue: Solving the LLP to a high level of accuracy is computational very costly, f.e. a whole update lasts 165s, including 157s for the solution of the LLP (averaged over 500 iterations for the CSM with log-Cauchy penalty, 24 filters of size 5x5, SSIM loss and L1 data term; training batch of 7 slices). Therefore, to reduce the number of HLP iterations, it is very important to perform a powerful update on the model parameters. On a first glance it would make sense to compute the Hessian of the HLP and apply Newton's method or approximate second order information by using f.e L-BFGS. The problem with those methods is that they would require a step size selection to perform well. In the case of bilevel learning, a linesearch procedure is not efficient because it would include the solving of the LLP for each loss evaluation. For this reason, Adam as a state of the art first order solver was chosen for the HLP update.

4.2.2. GPU Acceleration Potential

As the training of the CSM is particular expensive, an arising question is how it could be accelerated. One issue considering GPU parallelization is the computation of the inverse of the sparse Hessian. The fastest way to do so is probably based on a sparse Cholesky decomposition. Unfortunately, there is less support for sparse computations on GPUs and the efficient implementation of the latter would exceed the effort of this thesis by far.
Therefore, only investigations to accelerate the LLP were done. However, preliminary test on the GPU where the LLP was solved using CG were not very promising. This might be reasoned in unoptimized TensorFlow code and the little suitability ofCG with its linesearch (evaluation of the LLP loss has to be started from outside of the TensorFlow graph, leading to much overhead). Nevertheless, there is no reason why an optimization of the LLP on the GPU is not possible, but as with the VN a faster and actually more powerful model is available, there was no focus

on accelerating the CSM inference (inference time on a 128^2 slice using CG about 25s, using the CSM log-Cauchy L1 model and an i5-2500K CPU @ 3.30GHz x 4).

4.3. The used ASL Data

4.3.1. Different Noise Levels and Regularization Maps

Natural images used for evaluating new machine learning approaches are mostly corrupted with rather simple noise distributions. In general, Gaussian noise with the same variance for each pixel is added to the noise-free ground truth, leading to high-quality datasets for learning. In contrast, the used ASL data undergoes different noise characteristics in each voxel. Even if the basic shape of the distributions is the same, the variance differs a lot between voxels within a slice, between slices within a volume and between volumes of different subjects.
To overcome these SNR variances, the learned priors form a tradeoff between the low and high SNR input for a certain number of L/C-pairs. The unavoidable drawback is that some areas will be underregularized and others overregularized. In some cases, fewer averages of a specific subject yield better results than more averages of another subject. Due to outliers, it can not be excluded that fewer averages of the same voxel would yield better results than more averages. The separation of the noise levels based on the number of L/C-pairs is thus not ideal. The stated problems could be solved using regularizations maps and skipping the separation based on the number of L/C-pairs. Unfortunately, the regularization maps estimated from the temporal standard deviations within each voxel lead to a more difficult training. In addition, the estimation is prone to outliers. The implemented preliminary tests with regularization maps were not very promising at all. As the data related problems could also be solved to some extend by more sophisticated acquisition techniques, the regularization map approach was not further investigated. An improvement in data-quality could be achieved with newer labeling schemes like pCASL, which reduces outliers due to labeling in different states of the cardiac cycle or by using efficient background-suppression techniques to reduce the influence of physiological noise. An further improvement in SNR could be achieved with 3D readout strategies.

4.3.2. Additional Regularization of the CSM

In general, if the training set is drawn from the same distribution as the test set, an additional regularization factor would not be not needed. Nevertheless, it was shown that an additional regularization factor is beneficial in terms of SSIM, even if the model was trained using a SSIM based loss. The simplest explanation would be, that training and test set differ in terms of noise level. Another reason for this observation could be caused by the 64x64 subset used for training, which is characterized by a decreased GM/WM ratio and thus in average has less signal per voxel ($\mathrm{GM/WM}_{128^2} = 1.28$ and $\mathrm{GM/WM}_{64^2} = 1.09$). This results in lower SSIM and higher MSE ($\mathrm{SSIM}_{128^2} = 78.56\%$, $\mathrm{SSIM}_{64^2} = 75.43\%$, $\mathrm{MSE}_{128^2} = 47.23$, $\mathrm{MSE}_{64^2} = 141.75$). This means, that using the full 128^2 patch for training would lead to better results. In case of the CSM, 128^2 patches are not used because of infeasible long training times. In case of the VN, no benefit could be observed using the full 128^2 patch. The latter contradicts the GM/WM ratio explanation, although theoretically solid. However, one might assume that the VN is more robust to slight variations of the input SNR between test and training set.
Apart the clear impact of an additional regularization factor and the probably not ideal 64^2 patch, the relative shape of the learned filters are in principle independent from the noise level. The ideal amount of regularization will always be a matter of the used metric, which might be adapted to different needs and subjective preferences of the radiologist or any other expert. Hence, once learned an expressive prior it might be more useful to provide the practitioner the possibility to chose her/his preferred amount of regularization. In case of the VN, due to the it's fast inference, this would be possible to perform on-line.

4.4. Interpretation of the Results

The interpretation and comparison of the results are not straightforward. One image attains a high SSIM but a low PSNR, another image a low SSIM and a high PSNR, so it is not clear which image is closer to the reference. Additionally, the used reference is not noise-free, hence obtaining a better metric does not necessarily mean to be closer to the unknown noise-free reference. An alternative way of comparison, with the drawback of being subjective, is given by the direct visual evaluation of the CBF maps.

4.4.1. CSM and VN Results in General

Both models successfully improve the image quality of the input, regardless of the input noise level. As the VN can be interpreted as an advanced CSM, it is absolutely consistent with the theory that the VN exceeds the performance of the CSM.
Both learned models outperform the non-temporal TGV on the basis of the given dataset. Including temporal information, the TGV's performance increase and attain better quantitative results for some subjects than the VN. This emphasizes the importance of temporal or variance information for denoising of artifact prone ASL data.
A difference in performance between the TGV denoising and the learned models is founded in the use of local and global information. The latter is potentially useful, f.e. for tracking of long edges or homogeneous areas. The TGV based models utilize a few thousand gradient based optimization steps on the 128^2 image patch, resulting in whole image information for each pixel. In contrast, the VN uses 5 stages and hence each pixel receives information from a r=11 neighborhood. The CSM needs 28 LLP iterations on average, which results in information from a r=57 neighborhood.
The CBF maps (Figure 3.23) highlight the difficulty of a direct interpretation. For instance in the first row, the VN result clearly looks more natural and more denoised than the TGV result, but the latter attains higher SSIM and PSNR. It is unknown if this is reasoned by the metric calculation itself or by the noise within

the ground truth. This raises the question if the use of metrics for evaluation is meaningful at all.

4.4.2. Metrics for Evaluation

There are many ways for measuring the distance between two images. MSE, absolute error, PSNR, SSIM to name a few. But which metric should be used to rank images? On the one hand, PSNR and MSE are linked to a physical quantity, the power of the error. On the other hand, the SSIM tries to imitate human perception. It can not be answered if a physical quantity is a better choice than a psychological or vice versa, but it might be better to weight metrics stronger which are more attracted by desired image properties. As the SSIM is more attracted by sharp images, which are preferred against blurry images, the SSIM is weighted stronger than the PSNR.
Beside of the principal choice for a certain metric, there are different ways for computing the certain metric. As equally weighted slices are desired, the here used PSNR and SSIM are computed slicewise and averaged over the whole volume. Due to different foreground fractions (brain volume) per slice this leads to the drawback that single voxels are not weighted equally. F.e. in a slice with less foreground voxels, voxel deviations are weighted stronger than in a slice with more foreground voxels. This fact must be considered with care, as the standard deviation of the brain volume per slice is not negligible. (16.8% from the mean brain volume per slice, computed over all 10 subjects)

4.4.3. Edge Preservation

In section 3.4.4 "Edge preservation" it was shown that the full stTGV model preserves edges better than the VN and CSM. In general, the nature of the TGV functional is attracted by sharp images and thus preserves edges well. However, the additional inclusions of temporal information and the splitting of label and control images have a positive impact on the edge preservation. The SSIM and PSNR graphs (Figure 3.21 and 3.22) indicate a constant increase for the non-temporal,

temporal dM and full stTGV model. This raises the assumption that both model improvements (temporal input, L/C separated input) increase the capability in preserving edges.

4.4.4. Interpretation of the Learned Parameters

Theoretically, the learned parameters could be interpreted as MRF prior. They assign an energy proportional to the negative logarithm of the prior probability to each image patch, which could be used to identify likely and unlikely, maybe pathologic regions. Preliminary tests showed that the interpretation of the learned MRF as prior probability is not very promising. This is probably reasoned in the difference between discriminative and generative models. As the priors of the VN and the CSM are optimized for denoising, the filters mainly extract features to improve the performance in reducing noise and preserving image structures like edges. The learned priors should thus not be interpreted as generative priors.

4.5. Miscellaneous

4.5.1. CBF denoising

Actually, it would have been more reasonable to perform the image enhancement directly on the CBF maps and not on the PWIs. However, it has to be noted that an essential part of the CBF calculation is the division of the PWI by the M0 image. This could introduce problems like a division by zero or a rescaling of the noise. Nevertheless, CBF denoising was tested too and it performed very similar as PWI denoising.

4.5.2. Statistical Testing

In this thesis no statistical tests were performed for comparing different methods, because of the following reasons:
For an expressive testing rather 100 than just 10 repetitions would have been needed, which would last approximately 28h per test case (CSM) and thus is not feasible considering all different tested regularization factors, model hyperparameters etc.
As GPU computations are designed to be fast, they have the side effect of lacking for accuracy. In fact, the errors have a stochastic nature and thus the VN inference is not deterministic. In addition, TensorFlow is not completely deterministic either.
The result of a statistical test would only be useful in stating which method yields a significant better metric. However, a better metric does not include better performance. For instance, if one method attains significant higher PSNR and the other significant higher SSIM, the question which model should be chosen remains. In addition, the choice for the significance level is subjective too.
Summarized, statistical testing was not performed because it would be computational very costly (CSM), the inference is not deterministic (VN) and the meaningfulness is still low.

5. Conclusion

In the first part of the work it was shown that a co-sparse analysis model (CSM) as variant of a field of experts (FoE) is able to learn a discriminative prior for ASL image enhancement. In terms of PSNR and SSIM the CSM performs about as well as the non-temporal TGV approach but worse than stTGV. Nevertheless, the objective of designing an usable image enhancement method without the need for manually tuned parameters was fulfilled.
In the second part the drawbacks of the CSM, i.e the long inference time and the simplicity of the model (fixed penalty functions, etc.) were tackled by the use of a more powerful and flexible variational network (VN). Although the VN yields better results than the previously tested CSM, on a quantitative base it is only able to compete with the stTGV in terms of SSIM. However, the qualitative comparison shows a visual improvement of the VN CBF-maps compared to the CBF-maps from the stTGV. For reasons of a time consuming parameter tuning process for the stTGV, just results for 50 L/C-pairs were compared. It is reasonable to suspect that the impact of the temporal approach will decrease for more L/C-pairs used or for a less outlier prone ASL sequences.
The huge benefit of the VN is founded in its fast inference (50ms for a 128x128 patch) and fast learning (15min with a highly non-convex msSSIM loss) in combination with the ability to deal with a very low amount of data (data from 6 subjects for learning). For comparison, the stTGV denoising process takes about 2.5s (stTGV and VN experiments carried out on a Nvidia Titan Xp). Other learning based methods have shown to require costly trainings ([19] take 12h training in combination with a low resolution dataset and a MSE loss) as well as data from numerous subjects (f.e. Xie et al. [20] used 240 subjects for training). This highlights the VN's suitability for further applications in context with ASL image and

volume enhancement. As shown, even without temporal information the VN is able to perform almost equally well as the stTGV. Unfortunately, in the presence of large outliers or artifacts, this property reduces the ability to detect doubtful data like outliers and artifacts. In extreme cases this can lead to non-physiological structures.

Outlook

For future work, superior results are expected for the VN if additional information like temporal data, separate label and control images and regularization maps are included. The inclusion of the regularization maps might be difficult, because it seems to make the training more complex. Additionally, it would be important to obtain trust full variance statistics. This could be accomplished using the bootstrapping technique. Further, regularization maps could be used to skip the separation in different models for different numbers of used L/C-pairs. Hence, only one model would have to be trained. The amount of regularization would be determined by the regularization map. However, this would only increase the usability and not the performance. A probably fast way to increase the model capacity would be to include a learnable data term. This was actually part of preliminary tests, but the learning did not work well for this case. Beside of the mentioned model improvements, a very promising step goes towards the use of sophisticated 3D pCASL sequences. The increased basic image quality of these techniques makes highly resolved voxels possible (1.5mm isotropic). For such resolutions 3D denoising with 3D filter kernels becomes useful. Parameter learning in this case is particularly costly, thus a fast and effective model is needed. The VN might be the only model which meets the requirements of efficient inference and efficient learning.

References

[1] Rogosnitzky, M. and Branch, S.: "Gadolinium-based contrast agent toxicity: a review of known and proposed mechanisms". In: *BioMetals* 29.3 (June 2016), pp. 365–376.

[2] Williams, D. S. et al.: "Magnetic resonance imaging of perfusion using spin inversion of arterial water". In: *Proceedings of the National Academy of Sciences* 89.1 (1992), pp. 212–216.

[3] Edelman, R. R. et al.: "Qualitative mapping of cerebral blood flow and functional localization with echo-planar MR imaging and signal targeting with alternating radio frequency." In: *Radiology* 192.2 (1994), pp. 513–520.

[4] Edelman, R. R. and Chen, Q.: "EPISTAR MRI: multislice mapping of cerebral blood flow". In: *Magnetic resonance in medicine* 40.6 (Dec. 1998), pp. 800–805.

[5] Kim, S.-G.: "Quantification of relative cerebral blood flow change by flow-sensitive alternating inversion recovery (FAIR) technique: Application to functional mapping". In: *Magnetic Resonance in Medicine* 34.3 (1995), pp. 293–301.

[6] Wong, E. C., Buxton, R. B., and Frank, L. R.: "Implementation of quantitative perfusion imaging techniques for functional brain mapping using pulsed arterial spin labeling". In: *NMR in Biomedicine* 10.4-5 (1997), pp. 237–249.

[7] Golay, X., Petersen, E. T., and Hui, F.: "Pulsed star labeling of arterial regions (PULSAR): A robust regional perfusion technique for high field imaging". In: *Magnetic Resonance in Medicine* 53.1 (2005), pp. 15–21.

[8] Petersen, E. T., Lim, T., and Golay, X.: "Model-free arterial spin labeling quantification approach for perfusion MRI". In: *Magnetic Resonance in Medicine* 55.2 (2006), pp. 219–232.

[9] Dai, W. et al.: "Continuous flow-driven inversion for arterial spin labeling using pulsed radio frequency and gradient fields". In: *Magnetic Resonance in Medicine* 60.6 (2008), pp. 1488–1497.

[10] Günther, M., Oshio, K., and Feinberg, D. A.: "Single-shot 3D imaging techniques improve arterial spin labeling perfusion measurements". In: *Magnetic Resonance in Medicine* 54.2 (2005), pp. 491–498.

[11] Wells, J. A. et al.: "Reduction of errors in ASL cerebral perfusion and arterial transit time maps using image de-noising". In: *Magnetic Resonance in Medicine* 64.3 (2010), pp. 715–724.

[12] Daubechies, I., Defrise, M., and De Mol, C.: "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". In: *Communications on Pure and Applied Mathematics* 57.11 (2004), pp. 1413–1457.

[13] Bibic, A. et al.: "Denoising of arterial spin labeling data: wavelet-domain filtering compared with Gaussian smoothing". In: *Magnetic Resonance Materials in Physics, Biology and Medicine* 23.3 (June 2010), pp. 125–137.

[14] Dabov, K. et al.: "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering". In: *IEEE Transactions on Image Processing* 16.8 (Aug. 2007), pp. 2080–2095.

[15] Fang, R., Huang, J., and Luh, W.: "A spatio-temporal low-rank total variation approach for denoising arterial spin labeling MRI data". In: *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. Apr. 2015, pp. 498–502.

[16] Spann, S. M. et al.: "Spatio-temporal TGV denoising for ASL perfusion imaging". In: *NeuroImage* 157 (2017), pp. 81–96.

[17] Kim, K. H., Choi, S. H., and Park, S.-H.: "Improving Arterial Spin Labeling by Using Deep Learning". In: *Radiology* 287.2 (2018), pp. 658–666.

[18] Ulas, C. et al.: "DeepASL: Kinetic Model Incorporated Loss for Denoising Arterial Spin Labeled MRI via Deep Residual Learning". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by A. F. Frangi et al. Cham: Springer International Publishing, 2018, pp. 30–38.

[19] Owen, D. et al.: "Deep Convolutional Filtering for Spatio-Temporal Denoising and Artifact Removal in Arterial Spin Labelling MRI". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by A. F. Frangi et al. Cham: Springer International Publishing, 2018, pp. 21–29.

[20] Xie, D., Bai, L., and Wang, Z.: "Denoising Arterial Spin Labeling Cerebral Blood Flow Images Using Deep Learning". In: *CoRR* abs/1801.09672 (2018).

[21] Petr, J. et al.: "Denoising arterial spin labeling MRI using tissue partial volume". In: *SPIE Medical Imaging 2010: Image Processing*. Vol. 7623. San Diego, United States, Feb. 2010, n.a.

[22] Zhu, H., Zhang, J., and Wang, Z.: "Arterial spin labeling perfusion MRI signal denoising using robust principal component analysis". In: *Journal of Neuroscience Methods* 295 (2018), pp. 10 –19.

[23] Dosselmann, R. and Yang, X. D.: "A comprehensive assessment of the structural similarity index". In: *Signal, Image and Video Processing* 5.1 (Mar. 2011), pp. 81–91.

[24] Wang, Z. et al.: "Image quality assessment: From error visibility to structural similarity". English (US). In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612.

[25] Wang, Z., Simoncelli, E. P., and Bovik, A. C.: "Multiscale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*. Vol. 2. Nov. 2003, pp. 1398–1402.

[26] Roth, S. and Black, M. J.: "Fields of Experts". In: *International Journal of Computer Vision* 82.2 (Jan. 2009), p. 205.

[27] Chen, Y., Ranftl, R., and Thomas, P.: "Insights Into Analysis Operator Learning: From Patch-Based Sparse Models to Higher Order MRFs". In: *IEEE Transactions on Image Processing* 23.3 (Mar. 2014), pp. 1060–1072.

[28] Chen, Y., Yu, W., and Pock, T.: "On Learning Optimized Reaction Diffusion Processes for Effective Image Restoration". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[29] Hammernik, K. et al.: "Learning a Variational Model for Compressed Sensing MRI Reconstruction". English. In: *Proceedings of the International Society of Magnetic Resonance in Medicine (ISMRM)*. 2016.

[30] Pock, T. and Sabach, S.: "Inertial Proximal Alternating Linearized Minimization (iPALM) for Nonconvex and Nonsmooth Problems". In: *SIAM Journal on Imaging Sciences* 9.4 (2016), pp. 1756–1787.

[31] Hinton, G. E.: "Products of experts". In: *IET Conference Proceedings* (Jan. 1999), 1–6(5).

[32] Huang, J. and Mumford, D.: "Statistics of natural images and models". In: *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*. Vol. 1. June 1999, pp. 541–547.

[33] Geman, D and Reynolds, G: "Constrained Restoration and the Recovery of Discontinuities". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.03 (Mar. 1992), pp. 367–383.

[34] Hinton, G. E.: "Training Products of Experts by Minimizing Contrastive Divergence". In: *Neural computation* 14 (Sept. 2002), pp. 1771–800.

[35] Samuel, K. G. G. and Tappen, M. F.: "Learning optimized MAP estimates in continuously-valued MRF models". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. June 2009, pp. 477–484.

[36] Chen, Y. et al.: "Revisiting Loss-Specific Training of Filter-Based MRFs for Image Restoration". In: *Pattern Recognition*. Ed. by J. Weickert, M. Hein, and B. Schiele. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 271–281.

[37] Barbu, A: "Training an Active Random Field for Real-Time Image Denoising". In: *IEEE Transactions on Image Processing* 18.11 (Nov. 2009), pp. 2451–2462.

[38] Domke, J.: "Generic Methods for Optimization-Based Modeling". In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*. Ed. by N. D. Lawrence and M. Girolami. Vol. 22. Proceedings of Machine Learning Research. La Palma, Canary Islands: PMLR, Apr. 2012, pp. 318–326.

[39] Perona, P. and Malik, J.: "Scale-space and edge detection using anisotropic diffusion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7 (July 1990), pp. 629–639.

[40] Kobler, E. et al.: "Variational Networks: Connecting Variational Methods and Deep Learning". In: *Pattern Recognition*. Ed. by V. Roth and T. Vetter. Cham: Springer International Publishing, 2017, pp. 281–293.

[41] Addicott, M. A. et al.: "The effect of daily caffeine use on cerebral blood flow: How much caffeine can we tolerate?" In: *Human brain mapping* 30.10 (Oct. 2009), pp. 3102–3114.

[42] Domino, E. F. et al.: "Regional cerebral blood flow and plasma nicotine after smoking tobacco cigarettes". In: *Progress in Neuro-Psychopharmacology and Biological Psychiatry* 28.2 (2004), pp. 319–327.

[43] Vidyasagar, R. et al.: "The Effect of Black Tea and Caffeine on Regional Cerebral Blood Flow Measured with Arterial Spin Labeling". In: *Journal of Cerebral Blood Flow & Metabolism* 33.6 (2013), pp. 963–968.

[44] Luh, W. M. et al.: "QUIPSS II with thin-slice TI1 periodic saturation: a method for improving accuracy of quantitative perfusion imaging using pulsed arterial spin labeling". In: *Magnetic resonance in medicine* 41.6 (June 1999), pp. 1246–1254.

[45] Fazlollahi, A. et al.: "Reproducibility of multiphase pseudo-continuous arterial spin labeling and the effect of post-processing analysis methods". In: *NeuroImage* 117 (2015), pp. 191–201.

[46] Wang, Z. et al.: "Empirical optimization of ASL data analysis using an ASL data processing toolbox: ASLtbx". In: *Magnetic Resonance Imaging* 26.2 (2008), pp. 261 –269.

[47] Wang, Z.: "Improving cerebral blood flow quantification for arterial spin labeled perfusion MRI by removing residual motion artifacts and global signal fluctuations". In: *Magnetic Resonance Imaging* 30.10 (2012), pp. 1409–1415.

[48] Wang, Z. et al.: "Arterial spin labeled MRI in prodromal Alzheimer's disease: A multi-site study". In: *NeuroImage: Clinical* 2 (2013), pp. 630–636.

[49] Tan, H. et al.: "A fast, effective filtering method for improving clinical pulsed arterial spin labeling MRI". In: *Journal of Magnetic Resonance Imaging* 29.5 (2009), pp. 1134–1139.

[50] Buxton, R. B. et al.: "A general kinetic model for quantitative perfusion imaging with arterial spin labeling". In: *Magnetic Resonance in Medicine* 40.3 (1998), pp. 383–396.

[51] Alsop, D. C. et al.: "Recommended implementation of arterial spin-labeled perfusion MRI for clinical applications: A consensus of the ISMRM perfusion study group and the European consortium for ASL in dementia". In: *Magnetic Resonance in Medicine* 73.1 (2015), pp. 102–116.

[52] Lu, H. et al.: "Determining the longitudinal relaxation time (T1) of blood at 3.0 Tesla". In: *Magnetic Resonance in Medicine* 52.3 (2004), pp. 679–682.

[53] Wong, E. C., Buxton, R. B., and Frank, L. R.: "A theoretical and experimental comparison of continuous and pulsed arterial spin labeling techniques for quantitative perfusion imaging". In: *Magnetic Resonance in Medicine* 40.3 (1998), pp. 348–355.

[54] Herscovitch, P. and Raichle, M. E.: "What is the Correct Value for the Brain-Blood Partition Coefficient for Water?" In: *Journal of Cerebral Blood Flow & Metabolism* 5.1 (1985), pp. 65–69.

[55] Arbelaez, P. et al.: "Contour Detection and Hierarchical Image Segmentation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 33.5 (May 2011), pp. 898–916.

[56] Aja-Fern, S. and Trist, A.: "A review on statistical noise models for Magnetic Resonance Imaging". In: 2013.

[57] d'Agostino, R. B. and Pearson, E. S.: "Tests for departure from normality. Empirical results for the distributions of b2 and âˆšb1". In: *Biometrika* 60.3 (Dec. 1973), pp. 613–622.

[58] d'Agostino, R. B., Belanger, A., and Jr., R. B. d'Agostino: "A Suggestion for Using Powerful and Informative Tests of Normality". In: *The American Statistician* 44.4 (1990), pp. 316–321.

[59] Yaghoobi, M. et al.: "Analysis operator learning for overcomplete cosparse representations". In: *2011 19th European Signal Processing Conference*. Aug. 2011, pp. 1470–1474.

[60] Rubinstein, R., Faktor, T., and Elad, M.: "K-SVD dictionary-learning for the analysis sparse model". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2012, pp. 5405–5408.

[61] Hawe, S., Kleinsteuber, M., and Diepold, K.: "Analysis Operator Learning and its Application to Image Reconstruction". In: *IEEE Transactions on Image Processing* 22.6 (June 2013), pp. 2138–2150.

[62] Kingma, D. P. and Ba, J.: "Adam: A Method for Stochastic Optimization". In: *arXiv e-prints* (Dec. 2014).

[63] Kunisch, K. and Pock, T.: "A Bilevel Optimization Approach for Parameter Learning in Variational Models". In: *SIAM Journal on Imaging Sciences* 6.2 (2013), pp. 938–983.

[64] Chambolle, A. and Pock, T.: "A First-Order Primal-Dual Algorithm for Convex Problems withÂ Applications to Imaging". In: *Journal of Mathematical Imaging and Vision* 40.1 (May 2011), pp. 120–145.

A. Appendix - Derivations

A.1. SSIM

$$SSIM(\mathbf{x},\mathbf{y}) = \left(\frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}\right)\left(\frac{2\sigma_{xy} + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}\right) = l(\mathbf{x},\mathbf{y}) \cdot cs(\mathbf{x},\mathbf{y}) \tag{A.1}$$

$$\mu_x = \sum_{p=1}^{P} w_p x_p \qquad \frac{\partial \mu_x}{\partial x_i} = w_i \tag{A.2}$$

$$\sigma_x^2 = \sum_{p=1}^{P} w_p x_p^2 - \mu_x^2 \qquad \frac{\partial \sigma_x^2}{\partial x_i} = 2w_i(x_i - \mu_x) \tag{A.3}$$

$$\sigma_{xy} = \sum_{p=1}^{P} w_p x_p y_p - \mu_x\mu_y \qquad \frac{\partial \sigma_{xy}}{\partial x_i} = w_i(y_i - \mu_y) \tag{A.4}$$

With x_p and y_p being the p-th pixel of patch $\mathbf{x}$ and $\mathbf{y}$, respectively. w_p notes the Gaussian weighting of the p-th pixel of a patch.

$$\frac{\partial}{\partial x_i} SSIM(\mathbf{x},\mathbf{y}) = \frac{\partial l(\mathbf{x},\mathbf{y})}{\partial x_i} cs(\mathbf{x},\mathbf{y}) + l(\mathbf{x},\mathbf{y}) \frac{\partial cs(\mathbf{x},\mathbf{y})}{\partial x_i} \tag{A.5}$$

$$\begin{aligned} \frac{\partial l(\mathbf{x},\mathbf{y})}{\partial x_i} &= \frac{(2w_i\mu_y)\left(\mu_x^2 + \mu_y^2 + c_1\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)^2} - \frac{(2\mu_x\mu_y + c_1)(2w_i\mu_x)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)^2} \\ &= 2w_i \frac{\mu_y - \mu_x l(\mathbf{x},\mathbf{y})}{\mu_x^2 + \mu_y^2 + c_1} \end{aligned} \tag{A.6}$$

$$\begin{aligned}\frac{\partial cs(\mathbf{x},\mathbf{y})}{\partial x_i} &= \frac{2w_i(y_i-\mu_y)(\sigma_x^2+\sigma_y^2+c_2)}{\left(\sigma_x^2+\sigma_y^2+c_2\right)^2} - \frac{(2\sigma_{xy}+c_2)2w_i(x_i-\mu_x)}{\left(\sigma_x^2+\sigma_y^2+c_2\right)^2} \\ &= 2w_i\frac{(y_i-\mu_y)-(x_i-\mu_x)cs(\mathbf{x},\mathbf{y})}{\sigma_x^2+\sigma_y^2+c_2}\end{aligned} \tag{A.7}$$

A.2. msSSIM

$$msSSIM(\mathbf{x},\mathbf{y}) = l_{\sigma_M}(\mathbf{x},\mathbf{y})\cdot\prod_{s=1}^{M} cs_{\sigma_s}(\mathbf{x},\mathbf{y}) \tag{A.8}$$

$$\begin{aligned}\frac{\partial}{\partial x_i}msSSIM(\mathbf{x},\mathbf{y}) &= \frac{\partial l_{\sigma_M}(\mathbf{x},\mathbf{y})}{\partial x_i}\cdot\prod_{s=1}^{M} cs_{\sigma_s}(\mathbf{x},\mathbf{y}) \\ &\quad + \sum_{k=1}^{M}\frac{\partial cs_{\sigma_k}(\mathbf{x},\mathbf{y})}{\partial x_i} l_{\sigma_M}(\mathbf{x},\mathbf{y})\prod_{\substack{s=1\\s\neq k}}^{M} cs_{\sigma_s}(\mathbf{x},\mathbf{y}) \\ &= \left(\frac{\partial l_{\sigma_M}(\mathbf{x},\mathbf{y})}{\partial x_i} + l_{\sigma_M}(\mathbf{x},\mathbf{y})\sum_{s=1}^{M}\frac{\partial cs_{\sigma_s}(\mathbf{x},\mathbf{y})}{\partial x_i}\frac{1}{cs_{\sigma_s}(\mathbf{x},\mathbf{y})}\right)\prod_{s=1}^{M} cs_{\sigma_s}(\mathbf{x},\mathbf{y})\end{aligned} \tag{A.9}$$

B. Appendix - Results

Table B.1.: Quantitative results for different VN setups. All results for 24 filters of size 5x5 in 5 stages with 31 RBFs - 30 and 100 L/C-pairs.

Patchsize: 64x64 – 30 L/C-pairs

Loss / DT			L1					L2					SSIM					msSSIM					L1SSIM					L1msSSIM				
			S-VII	S-VIII	S-IX	S-X	mean	S-VII	S-VIII	S-IX	S-X	mean	S-VII	S-VIII	S-IX	S-X	mean	S-VII	S-VIII	S-IX	S-X	mean	S-VII	S-VIII	S-IX	S-X	mean	S-VII	S-VIII	S-IX	S-X	mean
L1	SSIM	WB	55.76	52.16	67.58	73.66	62.29	54.80	50.71	66.56	72.20	61.07	57.60	54.46	69.78	76.35	64.55	49.78	51.39	65.11	72.66	59.73	57.55	54.02	69.27	75.49	64.08	57.20	54.66	69.53	75.81	64.30
		GM	76.86	68.98	83.39	87.25	79.12	78.32	71.24	84.00	87.58	80.29	76.15	69.43	83.75	88.57	79.48	70.84	68.83	81.46	88.06	77.30	76.67	69.32	83.75	88.09	79.46	77.91	71.31	84.53	88.93	80.67
		WM	43.27	47.39	55.73	61.94	52.08	40.95	46.47	53.72	59.00	50.03	43.00	47.41	56.77	62.61	52.45	34.09	41.57	49.22	54.66	44.88	43.91	47.89	56.92	62.88	52.90	42.51	47.85	56.25	62.30	52.23
	PSNR	WB	23.13	18.52	22.24	21.94	21.46	23.10	18.82	22.26	21.86	21.51	22.52	18.21	22.04	22.13	21.22	20.18	16.68	20.61	21.05	19.63	22.94	18.48	22.26	22.12	21.45	22.77	18.44	22.25	22.27	21.43
		GM	18.95	16.75	20.83	20.68	19.30	19.23	17.20	21.03	20.78	19.56	18.34	16.49	20.65	21.03	19.13	16.38	15.54	19.48	20.64	18.01	18.73	16.69	20.83	20.91	19.29	18.81	16.85	20.97	21.24	19.47
		WM	14.46	13.82	19.30	19.65	16.81	13.84	13.49	18.80	18.96	16.27	13.84	13.46	19.23	19.58	16.53	10.59	11.56	17.02	17.20	14.09	14.30	13.76	19.42	19.78	16.82	13.74	13.49	19.16	19.56	16.49
L2	SSIM	WB	56.14	55.11	67.92	73.95	63.28	54.40	51.56	65.76	70.71	60.61	56.81	56.05	69.79	76.28	64.73	56.62	56.07	70.09	76.69	64.87	57.10	56.20	69.36	75.67	64.58	56.85	55.89	69.09	75.24	64.27
		GM	79.34	74.22	84.89	89.11	81.89	79.46	74.08	84.62	88.07	81.56	77.15	72.84	84.44	89.27	80.92	76.70	72.42	84.32	89.40	80.71	78.30	73.61	84.79	89.41	81.53	78.74	73.85	84.94	89.44	81.74
		WM	40.19	47.03	53.67	59.14	50.01	38.65	45.34	51.31	55.29	47.65	41.12	47.82	55.59	61.74	51.57	41.11	47.67	56.08	61.99	51.71	41.23	47.83	55.07	61.01	51.28	40.92	47.58	54.73	60.27	50.88
	PSNR	WB	22.70	18.50	22.11	22.11	21.36	22.85	18.84	22.14	21.69	21.38	22.04	18.13	21.85	22.08	21.03	21.92	18.00	21.81	22.13	20.96	22.36	18.31	22.02	22.17	21.22	22.50	18.40	22.09	22.19	21.29
		GM	19.08	17.17	21.08	21.37	19.67	19.32	17.51	21.19	20.98	19.75	18.16	16.70	20.63	21.22	19.18	18.02	16.58	20.56	21.26	19.10	18.59	16.92	20.89	21.37	19.44	18.78	17.03	20.99	21.42	19.55
		WM	13.08	13.08	18.43	18.68	15.82	12.97	13.00	18.06	18.00	15.51	12.84	12.95	18.62	19.02	15.86	12.77	12.92	18.73	19.13	15.89	12.99	13.04	18.60	18.95	15.90	13.04	13.08	18.60	18.88	15.90
L1 ZMF		WB											56.61	55.06	69.23	75.91	64.20	57.39	54.14	70.12	76.38	64.51	58.26	54.60	69.52	74.93	64.33	57.71	53.09	69.16	74.63	63.65
	SSIM	GM											76.92	72.15	84.10	89.03	80.55	75.24	68.26	83.89	88.64	79.01	75.94	67.92	83.37	86.98	78.55	75.88	67.49	83.29	86.83	78.37
		WM											40.86	47.21	55.13	61.30	51.13	42.49	45.63	55.77	60.98	51.22	44.95	48.25	57.68	62.83	53.43	44.26	46.84	57.05	62.22	52.59
	PSNR	WB											22.07	18.13	21.77	21.99	20.99	22.26	18.04	22.05	22.11	21.12	22.96	18.49	22.36	21.92	21.43	23.01	18.57	22.34	21.86	21.44
		GM											18.14	16.64	20.53	21.11	19.11	18.09	16.39	20.67	21.05	19.05	18.67	16.65	20.81	20.52	19.16	18.72	16.70	20.78	20.43	19.16
		WM											12.90	12.95	18.57	18.92	15.84	13.56	13.11	19.10	19.46	16.31	14.52	13.96	19.70	20.02	17.05	14.56	14.02	19.71	20.03	17.08
L2 ZMF	SSIM	WB	57.63	53.01	67.00	71.70	62.34	56.80	52.27	66.99	71.81	61.97	56.76	55.78	69.77	76.34	64.66	57.73	54.71	70.22	76.36	64.75	58.42	55.08	69.62	75.02	64.54	58.21	54.43	69.20	74.69	64.13
		GM	75.84	66.76	82.00	84.86	77.37	77.70	69.79	82.92	85.55	78.99	76.90	72.63	84.30	89.24	80.77	75.19	68.23	83.80	88.59	78.96	76.28	68.36	83.48	87.08	78.80	76.38	68.22	83.60	87.32	78.88
		WM	44.93	46.75	56.27	61.05	52.25	43.95	48.31	55.99	61.27	52.38	41.26	47.68	55.71	61.78	51.61	43.20	46.13	56.11	60.99	51.61	44.65	48.23	57.48	62.78	53.29	44.29	46.78	56.52	61.86	52.36
	PSNR	WB	23.20	18.52	22.19	21.45	21.34	23.39	19.07	22.37	21.58	21.60	22.00	18.08	21.81	22.08	20.99	22.26	18.00	22.05	22.12	21.11	22.92	18.42	22.35	21.94	21.41	22.89	18.39	22.33	21.97	21.39
		GM	18.84	16.59	20.52	19.87	18.96	19.26	17.24	20.78	20.04	19.33	18.07	16.65	20.56	21.19	19.12	18.06	16.35	20.64	21.05	19.02	18.70	16.63	20.82	20.56	19.18	18.69	16.63	20.85	20.65	19.21
		WM	14.90	14.21	19.67	19.91	17.17	14.68	14.26	19.53	19.86	17.08	12.85	12.93	18.66	19.08	15.88	13.67	13.18	19.18	19.50	16.38	14.33	13.88	19.68	20.01	16.97	14.25	13.77	19.49	19.84	16.84
temporal L1	SSIM	WB	57.28	53.17	68.38	74.68	63.38	56.44	50.82	66.92	73.17	61.84	56.04	52.59	69.05	75.92	63.40	56.49	53.62	69.50	76.05	63.92	57.22	53.92	69.32	75.91	64.09	57.32	53.96	69.43	75.94	64.16
		GM	76.65	69.40	84.10	88.63	79.69	76.71	69.08	83.70	87.93	79.35	74.35	68.60	83.40	88.85	78.80	75.21	68.96	84.06	89.13	79.34	75.96	69.39	84.21	89.13	79.68	76.02	69.38	84.28	89.16	79.71
		WM	41.44	44.10	55.36	60.95	50.46	41.64	43.95	54.52	60.17	50.07	38.50	42.66	54.33	59.25	48.69	38.95	43.21	55.07	60.32	49.39	40.05	43.67	55.25	60.79	49.94	40.30	43.83	55.60	61.11	50.21
	PSNR	WB	22.82	18.28	22.18	22.15	21.36	23.11	18.66	22.25	22.04	21.52	21.81	17.77	21.58	21.85	20.75	22.02	17.79	21.86	22.07	20.94	22.36	18.02	22.00	22.17	21.14	22.42	18.03	22.06	22.21	21.18
		GM	18.70	16.64	20.91	21.19	19.36	18.96	16.95	20.97	20.98	19.47	17.69	16.15	20.26	20.99	18.77	17.99	16.23	20.63	21.25	19.03	18.29	16.43	20.76	21.29	19.19	18.34	16.44	20.81	21.32	19.23
		WM	13.99	13.35	18.96	19.14	16.36	14.25	13.49	18.92	19.07	16.43	12.91	12.74	18.55	18.76	15.74	12.94	12.83	18.68	18.91	15.84	13.40	13.03	18.81	19.05	16.07	13.46	13.08	18.90	19.14	16.15
temporal L2	SSIM	WB	58.33	54.15	68.64	74.83	63.99	55.92	50.67	67.53	73.04	61.79	56.50	53.18	69.37	75.68	63.68	56.28	53.41	69.37	75.94	63.75	57.21	53.58	69.39	75.84	64.00	57.38	53.54	69.46	75.95	64.08
		GM	76.98	69.21	84.32	88.57	79.77	76.02	68.18	83.68	87.85	78.93	75.27	68.89	84.16	88.81	79.28	74.76	68.58	83.87	88.96	79.04	75.77	68.92	84.14	89.00	79.46	76.13	69.01	84.33	89.09	79.64
		WM	41.37	44.26	55.35	61.13	50.53	39.93	43.24	53.93	59.01	49.03	39.12	43.05	54.86	60.53	49.39	38.49	42.87	54.66	59.80	48.95	39.88	43.35	55.27	60.60	49.77	40.46	43.90	55.51	61.32	50.30
	PSNR	WB	22.78	18.09	22.11	22.05	21.26	22.58	18.24	22.01	21.76	21.15	22.09	17.87	21.86	21.95	20.94	21.86	17.66	21.72	21.94	20.80	22.25	17.86	21.92	22.06	21.02	22.38	17.94	22.02	22.16	21.13
		GM	18.72	16.49	20.91	21.09	19.31	18.55	16.57	20.81	20.83	19.19	18.05	16.29	20.69	21.10	19.03	17.83	16.11	20.51	21.13	18.89	18.18	16.29	20.69	21.19	19.09	18.34	16.36	20.80	21.28	19.20
		WM	13.92	13.15	18.87	19.05	16.25	13.37	12.98	18.58	18.60	15.88	13.02	12.80	18.60	18.85	15.82	12.82	12.71	18.57	18.83	15.73	13.31	12.93	18.79	19.02	16.01	13.39	13.01	18.86	19.14	16.10

Patchsize: 64x64 – 100 L/C-pairs

Loss / DT			L1					L2					SSIM					msSSIM					L1SSIM					L1msSSIM				
			S-VII	S-VIII	S-IX	S-X	mean	S-VII	S-VIII	S-IX	S-X	mean	S-VII	S-VIII	S-IX	S-X	mean	S-VII	S-VIII	S-IX	S-X	mean	S-VII	S-VIII	S-IX	S-X	mean	S-VII	S-VIII	S-IX	S-X	mean
L1	SSIM	WB	79.21	79.24	84.36	88.60	82.85	78.70	78.77	83.70	87.60	82.19	79.48	80.06	84.21	88.59	83.08	79.71	80.28	85.00	89.23	83.55	79.60	79.76	84.79	89.04	83.30	79.69	80.17	84.70	88.96	83.38
		GM	88.98	88.18	92.17	95.33	91.17	89.14	88.33	92.12	95.19	91.19	89.48	89.03	92.26	95.53	91.58	89.11	88.82	92.46	95.70	91.52	88.96	88.32	92.31	95.53	91.28	89.53	89.03	92.53	95.75	91.71
		WM	64.25	69.43	74.74	78.81	71.81	63.02	68.58	72.94	76.01	70.14	64.15	70.25	73.78	78.11	71.57	64.13	69.31	75.13	78.76	71.83	64.51	69.47	75.25	79.08	72.08	64.46	70.20	74.56	78.66	71.97
	PSNR	WB	27.15	24.29	25.57	26.19	25.80	27.19	24.32	25.54	26.00	25.76	27.05	24.42	25.53	26.20	25.80	26.93	24.30	25.54	26.30	25.77	27.08	24.28	25.57	26.27	25.80	27.13	24.47	25.65	26.41	25.91
		GM	22.64	22.14	24.25	25.16	23.55	22.73	22.19	24.23	25.02	23.54	22.70	22.38	24.28	25.36	23.68	22.48	22.25	24.30	25.43	23.61	22.57	22.14	24.28	25.31	23.58	22.76	22.42	24.42	25.58	23.79
		WM	18.37	20.17	23.28	23.36	21.29	18.36	20.11	23.10	22.96	21.13	17.89	19.94	22.86	22.87	20.89	18.04	19.96	23.19	23.23	21.10	18.30	20.11	23.31	23.37	21.27	18.12	20.08	23.16	23.18	21.14
L2	SSIM	WB	79.42	80.00	84.36	88.44	83.05	78.59	78.82	83.31	87.01	81.93	79.40	80.03	83.98	88.38	82.95	79.83	80.71	85.00	89.12	83.66	79.75	80.47	84.69	88.86	83.44	79.73	80.42	84.71	88.91	83.44
		GM	89.61	89.17	92.45	95.58	91.70	89.50	88.77	92.14	95.16	91.40	89.35	89.04	92.10	95.48	91.49	89.45	89.29	92.56	95.77	91.77	89.64	89.35	92.52	95.67	91.79	89.65	89.28	92.57	95.77	91.82
		WM	63.93	70.10	73.99	77.81	71.46	62.28	68.47	71.68	74.44	69.22	63.97	70.25	73.35	77.64	71.30	64.32	70.18	74.85	78.48	71.96	64.38	70.41	74.49	78.49	71.94	64.20	70.27	74.39	78.34	71.80
	PSNR	WB	27.18	24.53	25.64	26.28	25.91	27.21	24.41	25.52	25.90	25.76	26.95	24.40	25.47	26.12	25.74	26.99	24.46	25.60	26.35	25.85	27.11	24.53	25.62	26.33	25.90	27.13	24.52	25.66	26.41	25.93
		GM	22.82	22.48	24.42	25.44	23.79	22.87	22.35	24.27	25.04	23.63	22.61	22.37	24.19	25.31	23.62	22.60	22.43	24.36	25.53	23.73	22.77	22.51	24.41	25.49	23.79	22.78	22.49	24.44	25.60	23.83
		WM	18.13	20.11	23.08	23.03	21.09	18.13	20.00	22.84	22.50	20.87	17.75	19.87	22.78	22.72	20.78	17.95	20.02	23.16	23.13	21.06	18.04	20.07	23.10	23.06	21.07	18.02	20.08	23.12	23.12	21.08
L1 ZMF		WB											78.81	78.91	83.82	88.67	82.55	79.93	80.42	85.10	89.36	83.70	79.79	79.72	84.59	88.95	83.26	79.32	78.93	84.31	88.45	82.75
	SSIM	GM											88.08	87.51	91.57	95.28	90.61	89.12	88.95	92.54	95.72	91.58	88.92	88.19	92.13	95.40	91.16	88.41	87.51	91.92	95.11	90.74
		WM											63.74	69.08	73.39	78.10	71.08	64.57	69.32	74.78	78.66	71.83	65.02	69.67	74.97	79.16	72.21	64.05	67.98	74.51	77.90	71.11
		WB											26.60	23.97	25.28	26.13	25.50	26.94	24.36	25.57	26.32	25.80	27.09	24.26	25.53	26.21	25.77	27.02	24.09	25.43	25.98	25.63
	PSNR	GM											22.07	21.81	23.90	25.16	23.23	22.50	22.33	24.35	25.47	23.66	22.58	22.13	24.20	25.19	23.52	22.44	21.91	24.07	24.90	23.33
		WM											17.54	19.64	22.79	22.96	20.74	18.02	19.92	23.13	23.14	21.06	18.31	20.08	23.25	23.34	21.25	18.38	20.02	23.28	23.30	21.24
L2 ZMF	SSIM	WB	79.46	78.79	84.06	87.97	82.57	78.95	78.83	83.81	87.61	82.30	79.39	79.42	84.05	88.56	82.85	79.93	80.18	85.01	89.11	83.56	79.92	79.84	84.63	88.85	83.31	79.75	79.43	84.57	88.65	83.10
		GM	88.71	87.69	91.85	94.99	90.81	88.97	87.97	91.92	94.82	90.92	88.57	87.98	91.83	95.23	90.90	89.05	88.81	92.54	95.65	91.51	88.94	88.19	92.13	95.30	91.14	88.81	87.96	92.13	95.30	91.05
		WM	64.43	68.17	74.31	77.52	71.11	63.65	69.00	73.42	76.63	70.67	64.36	69.37	73.32	77.96	71.25	64.17	68.56	74.41	77.88	71.26	65.02	69.63	74.87	78.95	72.12	64.65	68.62	74.63	78.08	71.50
	PSNR	WB	27.10	24.17	25.46	25.96	25.67	27.22	24.29	25.50	25.85	25.71	26.80	24.12	25.38	26.03	25.58	26.94	24.32	25.57	26.27	25.77	27.08	24.28	25.54	26.17	25.77	27.07	24.22	25.53	26.14	25.74
		GM	22.55	21.99	24.08	24.84	23.37	22.69	22.07	24.11	24.67	23.38	22.30	21.99	24.03	25.06	23.35	22.48	22.30	24.36	25.42	23.64	22.56	22.13	24.20	25.11	23.50	22.54	22.06	24.20	25.10	23.47
		WM	18.38	20.07	23.28	23.27	21.25	18.55	20.31	23.26	23.19	21.33	17.80	19.78	22.82	22.90	20.83	18.04	19.87	23.12	23.08	21.03	18.27	20.11	23.26	23.35	21.25	18.31	20.05	23.29	23.32	21.24
temporal L1	SSIM	WB	79.30	79.61	84.60	88.93	83.11	78.64	78.51	83.85	87.85	82.21	78.27	78.16	82.55	86.91	81.47															
		GM	89.01	88.80	92.34	95.65	91.45	88.69	88.21	91.89	95.08	90.97	88.24	87.78	91.10	94.68	90.45															
		WM	62.51	67.65	74.95	78.94	71.01	62.63	67.26	74.64	77.68	70.55	62.53	67.52	72.37	76.02	69.61															
	PSNR	WB	26.91	24.33	25.53	26.36	25.78	27.01	24.24	25.44	25.98	25.67	26.54	23.97	25.02	25.54	25.27															
		GM	22.53	22.34	24.31	25.49	23.67	22.54	22.17	24.11	24.97	23.45	22.15	21.93	23.67	24.64	23.10															
		WM	17.79	19.75	23.07	23.14	20.94	18.11	19.86	23.12	22.98	21.02	17.23	19.30	22.30	22.07	20.23															
temporal L2	SSIM	WB	79.33	79.35	84.51	88.74	82.98	79.03	78.86	84.29	88.20	82.60	78.95	79.19	83.72	88.11	82.49															
		GM	89.01	88.59	92.29	95.54	91.36	88.88	88.20	92.11	95.23	91.10	88.64	88.38	91.78	95.15	90.99															
		WM	62.81	67.71	75.13	78.90	71.14	62.83	67.75	75.07	78.10	70.94	62.64	67.63	73.79	77.71	70.44															
	PSNR	WB	26.94	24.28	25.53	26.29	25.76	27.00	24.23	25.51	26.07	25.70	26.69	24.13	25.25	25.90	25.49															
		GM	22.56	22.27	24.30	25.40	23.63	22.56	22.15	24.21	25.08	23.50	22.31	22.12	23.98	24.96	23.34															
		WM	17.87	19.75	23.13	23.12	20.97	18.01	19.84	23.18	23.04	21.02	17.53	19.55	22.70	22.69	20.62															

Table B.2.: Quantitative results for different VN setups. All results for 24 filters of size 5x5 in 5 stages with 31 RBFs - {30,40,50,60,80,100} L/C-pairs.

Abbreviations			
L:	Loss		
D/T:	Datatern		
L2:	squared L2		
L1s:	approximated L1		
sXRT:	center smothed root		
IC:	logCauchy		
ZMT:	zeromean filters true		
ZMF:	zeromean filters false		