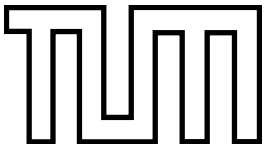
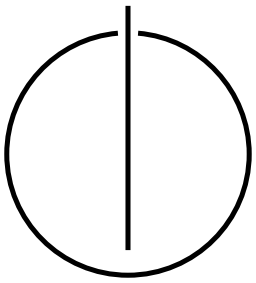


FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Deep Learning and Variational Analysis for High-Dimensional and Geometric Biomedical Data

Vladimir Golkov



Technische Universität München

Fakultät für Informatik
Lehrstuhl für Bildverarbeitung und Künstliche Intelligenz

Deep Learning and Variational Analysis for High-Dimensional and Geometric Biomedical Data

Vladimir Golkov

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Matthias Althoff
Prüfer der Dissertation:	1. Prof. Dr. Daniel Cremers
	2. Prof. Dr. Bastian Goldlücke

Die Dissertation wurde am 12.07.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 06.08.2021 angenommen.

Abstract

Deep learning and variational analysis have revolutionized various fields of science and engineering, for example the processing of 2D images. In this thesis, we take on the challenge of transferring this enormous success to areas of application in which geometric and high-dimensional data structures play a central role. We propose methods for predicting properties of proteins using data structures such as four-dimensional arrays containing evolutionary statistics, and for diffusion-weighted magnetic resonance imaging (diffusion MRI), which uses six-dimensional images.

One of the central questions in biology is to determine the structure and function of molecules such as specific proteins. The first method presented here predicts information about the three-dimensional structure of proteins, namely physical contacts between their amino acid residues. We derive a neural-network architecture from hypothetically meaningful biological features that the network might learn to extract. This approach, combined with end-to-end trained processing of rich evolutionary statistics, directly outperforms state-of-the-art methods. The second group of methods presented in this thesis predicts the function of molecules from their three-dimensional structure. Our method focuses on features of the structure that are responsible for how the molecule in question can interact with other molecules, namely the electron density and electrostatic potential fields. These fields are used directly as input to a neural network that is trained to predict the function of molecules. We also propose multi-channel representations that make it easier for the neural network to distinguish different amino acid types and hence structural patterns. Our methods achieve state-of-the-art results on small molecules and promising results on proteins. This indicates that neural networks can learn how electron density and electrostatic potential dictate the interactions between molecules.

Furthermore, we solve various challenges associated with diffusion MRI. Diffusion MRI provides unique information about the microstructure of living tissue, and thus is valuable for diagnosis, but so far its clinical application has been limited by long scan time requirements. Our deep-learning-based methods for diffusion MRI extract information from the images in an optimized way, and thus allow to reduce the scan duration by a factor of twelve. We also propose variational methods that use all raw measurements of a scan to reconstruct the image, instead of treating each image slice and each diffusion direction separately. By using synergies between neighboring slices and diffusion directions, our method considerably improves the quality of reconstructed images.

Zusammenfassung

Deep Learning und Variationsrechnung haben zahlreiche Teilgebiete der Wissenschaft und Technik revolutioniert, beispielsweise die Verarbeitung von 2D-Bildern. In dieser Dissertation übertragen wir diese Erfolge in Anwendungsbereiche, in denen geometrische und hochdimensionale Datenstrukturen eine zentrale Rolle spielen. Wir stellen Methoden vor, die Proteineigenschaften aus Datenstrukturen wie vierdimensionalen Arrays von Evolutionsstatistiken vorhersagen, sowie Methoden für diffusionsgewichtete Magnetresonanztomographie (Diffusions-MRT), welche sechsdimensionale Bilder verwendet.

Eine der zentralen Fragestellungen in der Biologie ist es, die Struktur und Funktion von Molekülen wie beispielsweise bestimmten Proteinen zu ermitteln. Die erste hier vorgestellte Methode sagt Informationen über die dreidimensionale Struktur von Proteinen vorher, und zwar physikalische Kontakte zwischen ihren Aminosäureresten. Wir leiten eine Architektur für neuronale Netze aus hypothetisch sinnvollen biologischen Merkmalen her, die das Netz lernen könnte zu extrahieren. Dieser Ansatz, kombiniert mit Ende-zu-Ende trainierter Verarbeitung von aussagekräftigen Evolutionsstatistiken, liefert direkt bessere Ergebnisse als bisherige Methoden. Die zweite Gruppe von Methoden, die wir in dieser Dissertation vorstellen, sagt die Funktion von Molekülen aus ihrer dreidimensionalen Struktur vorher. Das Hauptaugenmerk liegt dabei auf bestimmten Merkmalen der Struktur, die dafür verantwortlich sind, wie das jeweilige Molekül mit anderen Molekülen interagieren kann. Diese Merkmale sind die Elektronendichte und das durch Partialladung bedingte statische elektrische Feld. Diese Felder werden direkt als Eingabe für ein neuronales Netz verwendet, welches trainiert wird, die Funktion von Molekülen vorherzusagen. Wir stellen auch Mehrkanalrepräsentationen vor, die es dem Netz erleichtern, Aminosäuretypen und deshalb auch strukturelle Muster zu unterscheiden. Unsere Methoden erreichen Spitzenergebnisse bei kleinen Molekülen und vielversprechende Ergebnisse bei Proteinen. Dies deutet darauf hin, dass neuronale Netze lernen können, wie Elektronendichte und das elektrische Feld die Interaktionen zwischen Molekülen diktieren.

Darüber hinaus lösen wir mehrere Probleme im Bereich Diffusions-MRT. Diffusions-MRT liefert einzigartige und wertvolle Information über die Mikrostruktur von lebendem Gewebe und ist deshalb wertvoll für Diagnostik, doch die klinische Anwendung war bisher durch lange Scanzeiten beschränkt. Unsere auf tiefen neuronalen Netzen basierenden Methoden für Diffusions-MRT extrahieren Information aus den Bildern auf eine optimierte Art und Weise und erlauben deshalb, die Scanzeit um den Faktor zwölf zu verkürzen. Wir stellen auch eine Variationsmethode vor, die alle rohen Messungen eines Scans verwendet, um das Bild zu rekonstruieren, statt jede Bildschicht und jede Diffusionsrichtung getrennt zu behandeln. Durch die Verwendung von Synergien zwischen benachbarten Bildschichten und Diffusionsrichtungen ist unsere Methode in der Lage, die Qualität der rekonstruierten Bilder deutlich zu verbessern.

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Daniel Cremers for his wise, solid support throughout the years and for providing an excellent work environment at the TUM Computer Vision Group that encourages great research, creativity, fun, success, and collaboration with top researchers in the group and worldwide. I would also like to thank my project partners (of course including the many talented students) for the enjoyable and productive collaborations; my wonderful colleagues and many other members of the scientific community for their inspiring work and the illuminating conversations; for example the people who helped me explore new scientific territories: Alexey Dosovitskiy, Olivier Pauly, Claudia Nieuwenhuis, and others for teaching me about machine learning; Antonij Golkov, Remco Duits, Luc Florack, Jorg Portegies, Jon Sporring, Chantal Tax, Emanuele Rodolà, and others for bringing me into the wonderful world of differential geometry; Thomas Möllenhoff, Evgeny Strekalovskiy, Mohamed Souiai, and others for being great coaches in variational methods; Daniel Cremers for teaching me about all of the above topics and more; Marcin Skwark, Jeffrey Mendenhall, Jens Meiler, and others for offering valuable and fascinating insights into biology; my school teachers and university lecturers for sharing their knowledge and wisdom; Sebastian Weingärtner, Chantal Tax, and many others for inspiring me to set high standards for my research; Christiane Frense-Heck, Mohamed Souiai, Axel Wehmeier, and others for encouraging me to join the TUM Computer Vision Group full-time; Quirin Lohr for setting up a world-class computing infrastructure; Sabine Wagner for her support in organizational matters; Olivier Pauly and Till von Feilitzsch for encouraging me to apply for a scholarship at the Deutsche Telekom Foundation; the Deutsche Telekom Foundation and the Foundation of German Business for the scholarships and amazing event programs. Last but not least, I would like to thank my family for their support.

Contents

1 Introduction

Deep learning and variational analysis have revolutionized the field of computer vision, particularly the processing of 2D images. For example, in classifying photographs from the ImageNet dataset, methods based on deep learning have been ahead of other methods by a large margin ever since 2012, currently reaching an accuracy of over 90% [Pham et al., 2020]. Also methods based on variational analysis rather than machine learning provide good solutions to many difficult ill-conditioned and ill-posed problems, see [Bredies and Lorenz, 2011, Vese and Le Guyader, 2015]. They do not require any training data and are often the best choice when training data are costly to obtain or contain biases that are difficult [Mehrabi et al., 2019, Du et al., 2020] to deal with. Additionally, variational methods also allow solving inverse problems by using our knowledge of the forward mapping (the mapping we aim to invert) in a straightforward manner.

The challenge we take on here is to transfer this enormous success of deep learning and variational methods to new areas with complex geometric and high-dimensional data structures. We have developed appropriate methods for magnetic resonance imaging (MRI), particularly diffusion MRI, which uses six-dimensional images (with mutually informative values living on curved submanifolds of $\mathbb{R}^6$ such as $\mathbb{R}^3 \times S^2$), and for proteins, which can have various data structures associated with them, such as four-dimensional arrays (containing information about coevolution of amino acids within a protein, defined for all pairs of protein-chain positions and all pairs of amino acid types).

1.1 Outline of this Thesis

This thesis is structured as follows. We list our contributions in Section 1.2 and describe good practices for inventing new methods in Section 1.3. Section 2 describes the theoretical background: magnetic resonance imaging (MRI) and diffusion MRI in Section 2.1, proteins in Section 2.2, deep learning in Section 2.3, and variational analysis on Riemannian manifolds in Section 2.4. This is followed by the four publications included in this cumulative dissertation in Section 3 and by conclusions in Section 4.

1.2 Our Contributions

This cumulative thesis consists of the following four publications (included as Section 3):

[Golkov et al., 2016b] Golkov, V., Skwark, M. J., Golkov, A., Dosovitskiy, A., Brox, T., Meiler, J., and Cremers, D. (2016b). Protein contact prediction from amino acid co-evolution using convolutional networks for graph-valued images. In *Annual Conference on Neural Information Processing Systems (NIPS)*, Barcelona, Spain.

[Golkov et al., 2020b] Golkov, V., Skwark, M. J., Mirchev, A., Dikov, G., Geanes, A. R., Mendenhall, J., Meiler, J., and Cremers, D. (2020b). 3D deep learning for biological function prediction from physical fields. In *International Conference on 3D Vision (3DV)*. © 2020 IEEE. Reprinted with permission.

[Golkov et al., 2016a] Golkov, V., Dosovitskiy, A., Sperl, J. I., Menzel, M. I., Czisch, M., Sämann, P., Brox, T., and Cremers, D. (2016a). q-Space deep learning: Twelve-fold shorter and model-free diffusion MRI scans. *IEEE Transactions on Medical Imaging*, 35. © 2016 IEEE. Reprinted with permission.

[Golkov et al., 2015b] Golkov, V., Portegies, J. M., Golkov, A., Duits, R., and Cremers, D. (2015b). Holistic image reconstruction for diffusion MRI. In *Computational Diffusion MRI*. Springer. Reprinted/adapted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature: Computational Diffusion MRI (Proceedings of the 2015 MICCAI Workshop), Editors: Fuster, A., Ghosh, A., Kaden, E., Rathi, Y., Reisert, M. COPYRIGHT 2016

During the doctoral studies, also the following publications have been created:

[Aljalbout et al., 2018] Aljalbout, E., Golkov, V., Siddiqui, Y., Strobel, M., and Cremers, D. (2018). Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648*.

[Della Libera et al., 2019] Della Libera, L., Golkov, V., Zhu, Y., Mielke, A., and Cremers, D. (2019). Deep learning for 2D and 3D rotatable data: An overview of methods. *arXiv preprint arXiv:1910.14594*.

[Do et al., 2018] Do, B. T., Golkov, V., Gürel, G. E., and Cremers, D. (2018). Precursor microRNA identification using deep convolutional neural networks. In *bioRxiv preprint*.

[Dosovitskiy et al., 2015] Dosovitskiy, A., Fischer, P., Ilg, E., Haeusser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet: Learning Optical Flow with Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*.

[Fabbro et al., 2020] Fabbro, G., Golkov, V., Kemp, T., and Cremers, D. (2020). Speech synthesis and control using differentiable DSP. *arXiv preprint arXiv:2010.15084*.

[Golkov et al., 2020a] Golkov, V., Becker, A., Plop, D. T., Čuturilo, D., Davoudi, N., Mendenhall, J., Moretti, R., Meiler, J., and Cremers, D. (2020a). Deep learning for virtual screening: Five reasons to use ROC cost functions. *arXiv preprint arXiv:2007.07029*.

[Golkov et al., 2015a] Golkov, V., Dosovitskiy, A., Sämann, P., Sperl, J. I., Sprenger, T., Czisch, M., Menzel, M. I., Gómez, P. A., Haase, A., Brox, T., and Cremers, D. (2015a). q-Space deep learning for twelve-fold shorter and model-free diffusion MRI scans. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Munich, Germany.

[Golkov et al., 2014a] Golkov, V., Menzel, M., Sprenger, T., Haase, A., Cremers, D., and Sperl, J. (2014a). Semi-joint reconstruction for diffusion MRI denoising imposing similarity of edges in similar diffusion-weighted images. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Golkov et al., 2013a] Golkov, V., Menzel, M., Sprenger, T., Menini, A., Cremers, D., and Sperl, J. (2013a). Corrected joint SENSE reconstruction, low-rank constraints, and compressed-sensing-accelerated diffusion spectrum imaging in denoising and kurtosis tensor estimation. In *ISMRM Workshop on Diffusion as a Probe of Neural Tissue Microstructure*.

[Golkov et al., 2013b] Golkov, V., Menzel, M., Sprenger, T., Menini, A., Cremers, D., and Sperl, J. (2013b). Reconstruction, regularization, and quality in diffusion MRI using the example of accelerated diffusion spectrum imaging. In *16th Annual Meeting of the German Chapter of the ISMRM*.

[Golkov et al., 2014b] Golkov, V., Menzel, M., Sprenger, T., Souiai, M., Haase, A., Cremers, D., and Sperl, J. (2014b). Direct reconstruction of the average diffusion propagator with simultaneous compressed-sensing-accelerated diffusion spectrum imaging and image denoising by means of total generalized variation regularization. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Golkov et al., 2014c] Golkov, V., Menzel, M., Sprenger, T., Souiai, M., Haase, A., Cremers, D., and Sperl, J. (2014c). Improved diffusion kurtosis imaging and direct propagator estimation using 6-D compressed sensing. In *Organization for Human Brain Mapping (OHBM) Annual Meeting*.

[Golkov et al., 2014d] Golkov, V., Sperl, J., Menzel, M., Sprenger, T., Tan, E., Marinelli, L., Hardy, C., Haase, A., and Cremers, D. (2014d). Joint super-resolution using only one anisotropic low-resolution image per q-space coordinate. In *Computational Diffusion MRI*. Springer.

[Golkov et al., 2013c] Golkov, V., Sprenger, T., Menini, A., Menzel, M., Cremers, D., and Sperl, J. (2013c). Effects of low-rank constraints, line-process denoising, and q-space compressed sensing on diffusion MR image reconstruction and kurtosis tensor estimation. In *European Society for Magnetic Resonance in Medicine and Biology (ESMRMB) Annual Meeting.*

[Golkov et al., 2013d] Golkov, V., Sprenger, T., Menzel, M., Cremers, D., and Sperl, J. (2013d). Line-process-based joint SENSE reconstruction of diffusion images with intensity inhomogeneity correction and noise non-stationarity correction. In *European Society for Magnetic Resonance in Medicine and Biology (ESMRMB) Annual Meeting.*

[Golkov et al., 2013e] Golkov, V., Sprenger, T., Menzel, M., Tan, E., King, K., Hardy, C., Marinelli, L., Cremers, D., and Sperl, J. (2013e). Noise reduction in accelerated diffusion spectrum imaging through integration of SENSE reconstruction into joint reconstruction in combination with q-space compressed sensing. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting.*

[Golkov et al., 2016c] Golkov, V., Sprenger, T., Sperl, J. I., Menzel, M. I., Czisch, M., Sämann, P., and Cremers, D. (2016c). Model-free novelty-based diffusion MRI. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, Prague, Czech Republic.

[Golkov et al., 2018a] Golkov, V., Swazinna, P., Schmitt, M. M., Khan, Q. A., Tax, C. M. W., Serahlazau, M., Pasa, F., Pfeiffer, F., Biessels, G. J., Leemans, A., and Cremers, D. (2018a). q-Space deep learning for Alzheimer's disease diagnosis: Global prediction and weakly-supervised localization. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting.*

[Golkov et al., 2018b] Golkov, V., Vasilev, A., Pasa, F., Lipp, I., Boubaker, W., Sgarlata, E., Pfeiffer, F., Tomassini, V., Jones, D. K., and Cremers, D. (2018b). q-Space novelty detection in short diffusion MRI scans of multiple sclerosis. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting.*

[Gómez et al., 2015] Gómez, P., Sprenger, T., López, A., Sperl, J., Fernandez, B., Molina-Romero, M., Liu, X., Golkov, V., Czisch, M., Saemann, P., Menzel, M., and Menze, B. (2015). Using diffusion and structural MRI for the automated segmentation of multiple sclerosis lesions. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting.*

[Haeusser et al., 2018] Haeusser, P., Plapp, J., Golkov, V., Aljalbout, E., and Cremers, D. (2018). Associative deep clustering - training a classification network with no labels. In *Proc. of the German Conference on Pattern Recognition (GCPR).*

[Kukačka et al., 2017] Kukačka, J., Golkov, V., and Cremers, D. (2017). Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*.

[Menini et al., 2015] Menini, A., Golkov, V., and Wiesinger, F. (2015). Free-breathing, self-navigated RUFIS lung imaging with motion compensated image reconstruction. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Menzel et al., 2015] Menzel, M., Sprenger, T., Tan, E., Golkov, V., Hardy, C., Marinelli, L., and Sperl, J. (2015). Robustness of phase sensitive reconstruction in diffusion spectrum imaging. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Müller et al., 2021a] Müller, P., Golkov, V., Tomassini, V., and Cremers, D. (2021a). Rotation-equivariant deep learning for diffusion MRI. *arXiv preprint*.

[Müller et al., 2021b] Müller, P., Golkov, V., Tomassini, V., and Cremers, D. (2021b). Rotation-equivariant deep learning for diffusion MRI (short version). In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Naeyaert et al., 2020] Naeyaert, M., Aelterman, J., Audekerke, J. V., Golkov, V., Cremers, D., Pižurica, A., Sijbers, J., and Verhoye, M. (2020). Accelerating in vivo fast spin echo high angular resolution diffusion imaging with an isotropic resolution in mice through compressed sensing. *Magnetic Resonance in Medicine*, 85(3):1397–1413.

[Naeyaert et al., 2021] Naeyaert, M., Golkov, V., Cremers, D., Sijbers, J., and Verhoye, M. (2021). Faster and better HARDI using FSE and holistic reconstruction. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Pasa et al., 2019] Pasa, F., Golkov, V., Pfeiffer, F., Cremers, D., and Pfeiffer, D. (2019). Efficient deep network architectures for fast chest X-ray tuberculosis screening and visualization. *Scientific Reports*, 9(1):6268.

[Peeken et al., 2017] Peeken, J., Knie, C., Golkov, V., Kessel, K., Pasa, F., Khan, Q., Seroglazov, M., Kukačka, J., Goldberg, T., Richter, L., Reeb, J., Rost, B., Pfeiffer, F., Cremers, D., Nüsslin, F., and Combs, S. (2017). Establishment of an interdisciplinary workflow of machine learning-based radiomics in sarcoma patients. In *23. Jahrestagung der Deutschen Gesellschaft für Radioonkologie (DEGRO)*.

[Schuchardt et al., 2019] Schuchardt, J., Golkov, V., and Cremers, D. (2019). Learning to evolve. *arXiv preprint arXiv:1905.03389*.

[Sperl et al., 2014] Sperl, J., Sprenger, T., Tan, E., Golkov, V., Menzel, M., Hardy, C., and Marinelli, L. (2014). Total variation-regularized compressed

sensing reconstruction for multi-shell diffusion kurtosis imaging. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Sperl et al., 2013] Sperl, J., Tan, E., Sprenger, T., Golkov, V., King, K., Hardy, C., Marinelli, L., and Menzel, M. (2013). Phase sensitive reconstruction in diffusion spectrum imaging enabling velocity encoding and unbiased noise distribution. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Sprenger et al., 2013] Sprenger, T., Fernandez, B., Sperl, J., Golkov, V., Bach, M., Tan, E., King, K., Hardy, C., Marinelli, L., Czisch, M., Sämann, P., Haase, A., and Menzel, M. (2013). SNR-dependent quality assessment of compressed-sensing-accelerated diffusion spectrum imaging using a fiber crossing phantom. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Sprenger et al., 2014] Sprenger, T., Sperl, J., Fernandez, B., Golkov, V., Tan, E., Hardy, C., Marinelli, L., Czisch, M., Sämann, P., Haase, A., and Menzel, M. (2014). Novel acquisition scheme for diffusion kurtosis imaging based on compressed-sensing accelerated DSI yielding superior image quality. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Sprenger et al., 2016] Sprenger, T., Sperl, J. I., Fernandez, B., Golkov, V., Eidner, I., Sämann, P. G., Czisch, M., Tan, E. T., Hardy, C. J., Marinelli, L., Haase, A., and Menzel, M. I. (2016). Bias and precision analysis of diffusional kurtosis imaging for different acquisition schemes. *Magnetic Resonance in Medicine*.

[Swazinna et al., 2019] Swazinna, P., Golkov, V., Lipp, I., Sgarlata, E., Tomassini, V., Jones, D. K., and Cremers, D. (2019). Negative-unlabeled learning for diffusion MRI. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Vasilev et al., 2019] Vasilev, A., Golkov, V., Meissner, M., Lipp, I., Sgarlata, E., Tomassini, V., Jones, D. K., and Cremers, D. (2019). q-Space novelty detection with variational autoencoders. In *MICCAI 2019 International Workshop on Computational Diffusion MRI*.

1.3 Good Practices for Inventing New Methods

In this section, we briefly outline a few techniques we used to invent new successful methods.

1.3.1 Analysis of Requirements

A useful step in the creation of methods is to formulate the requirements for an ideal method, for example intuitively in words, or even to formalize them mathematically. Subsequently, some (out of possibly many) algorithms that satisfy these requirements can be formulated, in some cases they can be even formally derived from the formalized requirements.

For example, in [Schuchardt et al., 2019], we formulated the requirements for fair treatment of different parts of the input, formalized the requirements as equivariance (see Section 2.3.2) under permutations of data along certain dimensions of the input, and subsequently used the most general affine neural networks that fulfill these requirements in combination with common nonlinearities that also fulfill these requirements.

As another example, in [Golkov et al., 2016b] (included here as Section 3.1), we hypothesized what biological features that are useful for the final goal can be extracted at each processing stage. The number of hypothetically meaningful processing stages of features building upon each other provided an estimate for the ideal number of neural-network layers. The location of lower-level information about these hypothetical features provided an estimate of the ideal receptive field size for each neural-network layer. The number of these hypothetical features provided an estimate of the ideal number of channels/filters in each neural-network layer. Moreover, we also identified biological reasons for a requirement for equivariance under translations. Together with the common practice of using the most general trainable affine mappings followed by popular nonlinearities (as seen in the majority of network layers nowadays), all requirements fully dictated our neural-network architecture. This architecture worked immediately well, outperforming state-of-the-art methods, and further hyperparameter tuning did not considerably improve the results, i.e. our hypothesis based on domain knowledge directly yielded successful network hyperparameters. For details about neural networks in general, see Section 2.3.

1.3.2 Analysis of Weaknesses

A similar technique is to identify the weaknesses and failure cases of existing algorithms. Weaknesses can be considered as non-fulfillment of requirements, i.e. the procedure can be similar to the one described in Section 1.3.1. Moreover, the weaknesses and the techniques to overcome them might have analogs in existing literature from other fields.

For example, when analyzing existing algorithms that process diffusion MRI data (see Section 2.1 for details about diffusion MRI), we noticed the following "red flags":

- Large datasets with valuable information existed, but that information was not being used while processing small datasets.
- Most processing steps were handcrafted, i.e. probably suboptimal rather than jointly optimized to work together well.

A good practice in such cases is to use machine learning in order to make use of the valuable information present in existing data. More specifically, so-called end-to-end training of deep neural networks allows to jointly optimize all feature-extraction stages to work together optimally. This is what we did in [Golkov et al., 2016a, Golkov et al., 2016c, Golkov et al., 2018a, Golkov et al., 2018b, Vasilev et al., 2019, Swazinna et al., 2019, Müller et al., 2021a] with many benefits such as the shortening of the scan time by a factor of twelve, which strongly improves patient comfort and reduces costs.

As another example, we noticed that most image reconstruction methods for diffusion MRI treat each diffusion-encoded image from the same scan independently, without making use of the strong correlations and potential synergies between these different parts of the scan. That additional information has the potential to improve the imperfect signal-to-noise ratio and even the image resolution. Therefore, we used several measurements with different diffusion weightings in a joint optimization procedure to improve each other's resolution and signal-to-noise ratio [Golkov et al., 2014d, Golkov et al., 2015b]. Particularly in [Golkov et al., 2015b] (included herein as Section 3.4), we mathematically formulated advanced prior knowledge about the geometrical properties of the data and used it to improve the signal-to-noise ratio and resolution.

1.3.3 Data Flow between Mappings

Visualizing the data flow within existing algorithms can help to invent new algorithms. There are different ways to visualize data flow. The following main categories of "visual languages" for data flow can be distinguished, depending on how mappings (i.e. data-processing steps) and variables (i.e. data) are visualized:

- A node (e.g. rectangle) represents a mapping. An arrow represents a variable and connects outputs of some mappings to inputs of subsequent mappings.
- A node (e.g. rectangle) represents a variable. An arrow represents a mapping and connects the input variables of that step to its output variables. See for example Fig. 1 in [Golkov et al., 2016a] (included here as Section 3.3)
- Some nodes (e.g. rectangles) represent variables, other nodes (e.g. rectangles of a different color) represent mappings. An arrow represents data flow and connects a mapping to its output variables, or input variables of a mapping to that mapping.
- We are working on a publication about an advanced visual language for data flow (and for other things) with "syntactic sugar".

Arrows can merge or branch to represent usage of several variables as inputs to one mapping, or usage of one variable as input to several mappings.

If several algorithms share some mappings or variables, these algorithms can be combined into one diagram without visualizing these mappings or variables redundantly. For example, see Fig. 1 in [Golkov et al., 2016a], where several algorithms are shown in one diagram. If some mappings or variables are not identical but similar, the differences can be abstracted from, so that the algorithms can be combined into one diagram.

To invent new methods, these data-flow diagrams can be modified, for example by adding/replacing arrows. This corresponds for example to finding new combinations of inputs and outputs (and thus streamlining the data processing and/or using synergies between variables).

Due to the progress of deep learning methods and availability of training data, an important trend is to replace a handcrafted (hence suboptimal) mapping (or especially a sequence of several) by one direct mapping performed by a neural network, which is more optimal due to end-to-end training and goal-oriented (see [Golkov et al., 2020a]) cost functions. For an example of introducing new learning methods into data-flow diagrams, see the green arrows in Fig. 1 in [Golkov et al., 2016a]. More specifically, a trend is to make classical processing steps deeper (have more processing steps / network layers), wider (have more features / channels), trainable (have additional parameters with respect to which a data-driven objective function is being optimized), have nonzero gradients (for example by smoothing the mapping, or more generally replacing the cost function by one that hat has nonzero gradients and possibly highly nonlinear but still strong correlation with the original one [Yan et al., 2003]) for compatibility with gradient-based optimization of the previous/current/subsequent processing steps, and/or to train several trainable processing steps jointly (especially train all of them end-to-end). For example, classical optimization algorithms (not to be confused with the "outer loop" of optimization that trains the neural network) can be combined with recurrent or feed-forward networks by using supervised learning or reinforcement learning for improving the update steps, or using an optimization procedure as a layer in a larger neural network [Andrychowicz et al., 2016, Li and Malik, 2017, Adler and Öktem, 2017, Moeller et al., 2019, Mensch and Blondel, 2018, Amos and Kolter, 2017].

A special case of this trend is to use the architecture of classical methods as inspiration for the neural-network architecture of new methods, for example by choosing a neural-network architecture for which it is easy to learn to do what the classical method does. Alternatively, the neural-network weights can be initialized by hand such that the network initially does *(almost) exactly* what the classical method does. If the gradient is zero in that case because such weights are exactly at a saddle point of the loss landscape, i.e. training cannot progress, then slight noise can be added to the network weights without affecting the output much, in order to break the symmetry. Then the network training can start from there. An alternative to such handcrafted initialization is to *pretrain* the neural network to do *approximately* what the classical method does,

and then fine-tune on another training objective.

1.3.4 Data Flow between Data Domains

A variable carries certain information. Mapping/associating the value of the variable to some "high-level representation" (for example an intuitive formulation of the information content in words) of that information can be referred to as the *"meaning" of that variable*; the result of this mapping can be referred to as the *"meaning" of that given value* of the variable.

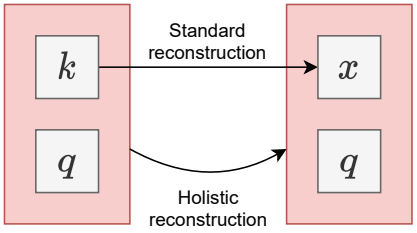
Irrelevant low-level details of the information are discarded by this mapping. Thus, several values can have the same meaning. A meaning shared by several values is similar to the concept of a *macro-state* in physics, i.e. the grouping of several of the possible states of a system into a set based on some high-level similarity.

A variable can take on values from a set of possible values. This set can be referred to as the *domain of the mappings* that take this variable as input. One can also consider the *mathematical space* in which this set lives. However, unlike the mathematical definitions of *domain* and *space*, in fields such as image processing and machine learning, these terms can be additionally associated with the "meaning" of the variable. For example, *domain adaptation* refers to not only adapting to a different set of possible variable values, but also "shifting the meaning" to the new set. Another example are the terms *k-space* from MRI and *q-space* from diffusion MRI. They refer not only to the mathematical space ($\mathbb{R}^2$, $\mathbb{R}^3$) in which the variable lives, but also to the meaning of the variable, i.e. k-space data represents the frequencies of the MRI image, q-space data represents the relationship between diffusion measurement vectors and signal decay.

Thus, a variable can be associated with its domain/space (accompanied by the "meaning" of the variable). Each step (mapping) of the data-processing pipeline thus maps a variable from some domain to a variable from a possibly different domain. This data flow between domains can be visualized. A rectangle can represent a domain, an arrow can represent a mapping.

Some spaces can be meaningfully decomposed into subspaces whose product spans the entire space. For example, raw measurements in diffusion MRI can be considered as separately acquired 3D k-spacemeasurements for each coordinate in 3D q-space, or as data in six-dimensional k×q-space where joint information along all dimensions is used. The joint k×q-space can be visualized as a rectangle containing two smaller rectangles – one for k-space and one for q-space. Separate processing is visualized as arrows that leave/enter the smaller rectangles, whereas processing that uses joint information in the 6D product space is visualized as arrows that enter/leave large rectangles. An example of this is Fig. 1 in [Golkov et al., 2013b]. The same can be summarized as a table. See Fig. 1 here for a comparison of the two visualizations.

When acquiring all data has disadvantages, acquiring partial but complementary data and using joint information can be a good approach. For example, in [Golkov et al., 2014d] we use different parts of *k*-space for different *q*-space



Method	Input space	Output space
Standard reconstruction	k	x
Holistic reconstruction [Golkov et al., 2015b]	$k \times q$	$x \times q$

Figure 1: Data flow between data domains (Fourier space k, physical space x, diffusion-encoding space q) in diffusion MRI visualized as a diagram (left) and as a table (right). Standard reconstruction maps k-space to x-space for each q-space coordinate independently, i.e. it does not use valuable information contained in other q-space coordinates. On the other hand, holistic reconstrution [Golkov et al., 2015b] uses all the information from joint $k \times q$-space to produce the output in $x \times q$-space, i.e. it uses synergies between the measurements along all dimensions. Our analysis of existing methods in terms of data flow between data domains [Golkov et al., 2013b] inspired the creation of methods with better data flow and better results [Golkov et al., 2014d, Golkov et al., 2015b, Golkov et al., 2016a, Golkov et al., 2018a].

coordinates.

1.3.5 General Formulations of Methods

It is worthwhile to find a general formulation/formula whose special cases are existing methods. This allows systematically creating more special cases and systematically searching the space of methods for the best methods. See for example Table 1 in [Duran et al., 2016].

Getting an overview and finding common patterns can be facilitated by bringing existing formulas into similar forms with consistent variable names. See for example Figure 4 in [Goldluecke et al., 2012], Table 6.1 in [Couprie, 2011], and Table 2 in [Aja-Fernández et al., 2009].

Tables of Method Properties A specific way to systematize methods is to identify the atomic properties of existing methods (possibly by decomposing some properties as far as possible into several independent sub-properties) and list the properties for each method in a table (one method per row, one property per column). For example, see Tables 1–2 in [Kukačka et al., 2017], Tables 1–4 in [Della Libera et al., 2019], Table II in [Justesen et al., 2019], Table 1 in [Tewari et al., 2020], Figs. 2–3 in [Poorman et al., 2020].

Subsequently, new combinations of existing properties can be created in a straightforward manner. By using this technique, we created a clustering method that outperformed state-of-the-art methods, see Table 1 in [Aljalbout

et al., 2018]. Another example where a new combination of existing properties yielded a new successful method is Table 1 in [Tzeng et al., 2017].

Information about impossible combinations of method properties can also be added to such a table, namely by filling in the conflicting/incompatible method properties in the respective columns, putting wildcards (e.g. "Any" or "*") into the other columns, and writing "Not possible because ..." in the header column.

An alternative to a table of binary properties of methods is a Venn diagram or an Euler diagram of methods, for example Fig. 2 in [Tay et al., 2020]. For additional examples of alternative visualizations, see [Vegas et al., 2009].

Some properties, or combinations of properties, can have advantages in certain situations. This allows identifying which methods have an optimal combination of properties, or whether such methods are yet to be created. For an example where advantages/disadvantages are marked, see Tables 1–4 in [Della Libera et al., 2019].

A table of method properties can be considered an *ontology* (in the sense of information science), but is often called *taxonomy* to avoid confusion with the term ontology from philosophy.

1.3.6 Family Trees of Methods

In the strict sense, a *taxonomy* is a family tree (of methods). Such a visualization can help gain an overview of which methods inherited what properties from which other methods, and what properties contributed to good results. Arrows (indicating kinship relations) and methods can be annotated with the properties of methods, for example via additional text or colors. An example is Fig. 4 in [Justesen et al., 2019].

2 Theoretical Background

2.1 MRI and Diffusion MRI

Magnetic resonance imaging (MRI) is an imaging technique that noninvasively measures millimeter-resolved 3D maps of various physical properties (such as water content or diffusivity) inside of objects and living beings. These measurements are based on various physical phenomena. Section 2.1.1 describes the physical phenomenon of nuclear magnetic resonance. Based on that, Section 2.1.2 explains basic techniques used in MRI, which yield 3D images based on properties of (biological) tissue such as water content and nuclear relaxation times. Section 2.1.3 explains basic techniques used in diffusion MRI, which yield 6D images of diffusion, i.e. 3D statistics of water self-diffusion for each 3D voxel of the object. Section 2.1.4 describes how an effect called the spin echo is used to improve the signal-to-noise ratio in diffusion MRI.

In these sections, the term *gradient* is used not in the sense of *a derivative*, but in its more mundane sense, namely a *spatial gradient*, i.e. a scalar field whose derivative is nonzero, and in our case equal in all points of the imaged volume.

2.1.1 Magnetic Resonance

In a strong constant magnetic field (like the field maintained by an MRI scanner), the *nuclear spins* (intrinsic angular momenta of atomic nuclei) exhibit *precession* (i.e. their orientation rotates, like the axis of an evenly wobbling spinning top or gyroscope) with the so-called *Larmor frequency* around the magnetic field direction. The Larmor frequency is proportional to the magnetic field strength.

The applied magnetic field is slightly modified locally due to *magnetic susceptibility* and *magnetic moments* of the electrons surrounding the atomic nuclei. The distribution of electron density around a nucleus depends on the chemical structure of the molecule. Thus, nuclei of the same isotope but with different chemical surroundings perceive a slightly different magnetic field strength. Therefore, they also have slightly different Larmor frequencies. This shift of the Larmor frequency due to chemical surroundings is referred to as the *chemical shift*. The most abundant *species* in living tissue are the hydrogen nuclei of water molecules. Thus (and due to having nonzero spin), they provide a strong signal and are most often used in MRI (i.e. it is their Larmor frequency that is used in the pulses described in the following).

At equilibrium, the spins have a very slight tendency to be aligned with the magnetic field direction, resulting in a very slight net magnetization in that direction. An electromagnetic pulse with the Larmor frequency flips the net magnetization away from the direction aligned with the field. This effect is called *magnetic resonance*. The pulse is a radiofrequency pulse because the Larmor frequency in MRI is within the radiofrequency spectrum. A pulse whose duration achieves a flip of the net magnetization by 90° is called a 90° pulse (or an *excitation pulse* because it brings the net magnetization out of its equilibrium

state). A signal can then be measured with a receiving radiofrequency coil (which may be the same coil that was used to emit the radiofrequency pulse).

For a more detailed overview of magnetic resonance and a refutation of common misconceptions about it, see [Hanson, 2008].

2.1.2 Magnetic Resonance Imaging

The transversal magnetization (the component of the net magnetization of spins that is perpendicular to the background magnetic field lines) rotates with the Larmor frequency and thus provides a measurable radiofrequency signal. This signal can be represented as a scalar: the magnitude of the radiofrequency wave and, if the signal is represented as a complex scalar rather than a real scalar, the phase of the wave. In order to obtain a 3D image, i.e. to measure the contributions from different parts of the object (e.g. due to different amount of water in different parts) rather than just one scalar for the entire object, several techniques are typically used in sequence, namely rotating the spins away from their equilibrium state only in a part of the object, then dephasing these spins in many different ways in order to measure many Fourier components of the spatial distribution of the spins, and then repeating these steps for a different part of the object. Details are described in the following.

An MRI scanner has additional electromagnetic coils that can be switched on to modify the magnetic field. By adding a weak linear magnetic field gradient (i.e. a field strength whose equidistant level sets are equidistant planes within the scanner, with several millitesla per meter) to the strong magnetic field (which is constant in space, at several tesla), the Larmor frequency in different slices of the volume (level sets of the magnetic field strength) becomes different. Thus, a radiofrequency pulse with a narrow frequency band rotates only the spins in a narrow slice of the volume, whose Larmor frequency corresponds to the pulse.

Subsequently, linear magnetic field gradients are applied in different directions perpendicularly to the selected slice, so that the spins dephase in a structured manner and their contributions to the signal become weighted by 2D cosine waves of different 2D directions and frequencies. Applying certain linear magnetic field gradients (thus changing the dephasing of spins and the directions and frequencies of the 2D cosine wave) and measuring the signal is repeated many times. The obtained measurements correspond to a Fourier transform of the 2D image of the selected slice. This process is repeated for each slice, with short delays in which the spins of the previously measured slice relax, which is outlined in the following paragraph.

The longitudinal magnetization (net magnetization parallel to the constant magnetic field) and the transversal magnetization (net magnetization perpendicular to the constant field) both return to their equilibrium states at different rates that depend on the local physical tissue properties. This is referred to as *relaxation*.

To reconstruct the 2D image of each slice, the inverse Fourier transform can be used. Alternatively, reconstruction algorithms can be used that suppress the measurement noise by taking advantage of prior knowledge about typical im-

ages. Such prior knowledge can be formulated mathematically as regularization terms [Knoll et al., 2011], also leveraging synergies between different slices and image contrasts [Golkov et al., 2015b] (presented here as Section 3.4), or learned from datasets [Zhu et al., 2018].

For an in-depth discussion of MRI, see [Brown et al., 2014].

2.1.3 Diffusion MRI

Water in biological tissue exhibits *(self-)diffusion*, i.e. the water molecules move randomly, colliding with each other and with other types of molecules. The statistics of this diffusion are affected by restrictions and hindrances that the water molecules encounter, such as membranes, organelles, large molecules, as well as subtle molecular interactions that contribute to effects such as viscosity. Thus, diffusion statistics contain unique valuable information about the tissue microstructure.

A group of MRI techniques called *diffusion MRI* provides a unique way to measure certain statistics of diffusion, and thus to non-invasively obtain unique information about the tissue microstructure. Many diseases, such as neurodegenerative diseases, affect the tissue microstructure and the measurable diffusion statistics. Diffusion MRI thus enables a more precise diagnosis and more targeted treatment.

Making the measured signal strongly dependent on the diffusion works as follows. After the excitation pulse, a relatively strong linear magnetic field gradient is applied in the direction in which the diffusion should be measured. This creates a spatial gradient in the Larmor frequencies of the spins, i.e. the spins dephase. After this first diffusion-encoding gradient is switched off, a certain amount of time is allowed to pass, during which most of the measured diffusion happens. Then, another, opposite gradient is used. This second diffusion-encoding gradient reverses the dephasing effect of the first diffusion-encoding gradient for spins that did not change their position. On the other hand, for molecules that moved much, the effects of the two gradients on their phase do not cancel each other out in general. These spins are not in phase with the spins that did not move much, and thus the overall signal is weaker.

How much diffusion affects the diffusion-weighted signal depends on the value b of the *diffusion weighting*, which in turn depends on the strength G of the diffusion-encoding gradients (in millitesla per meter), their durations δ, and the time Δ between their beginnings. For simplified transitions between "on" and "off" states of the gradients, the diffusion weighting is

$$b = \gamma^2 G^2 \delta^2 \left(\Delta - \frac{\delta}{3} \right), \tag{1}$$

where $\gamma \approx 2.675\,\mathrm{rad/s/T}$ is the gyromagnetic ratio of the hydrogen nucleus. Usually G is modified to achieve different b-values during the same scan, whereas Δ and δ are kept constant.

Note that the gradients mentioned in Section 2.1.2 also create a diffusion

weighting, but a relatively weak one, due to their smaller magnitude and duration [Pipe, 2014].

Diffusion can be quantified by comparing the diffusion-weighted signal (image intensity) S with the non-diffusion-weighted signal S_0. If the b-value is too low, the diffusion-weighted signal is similar to S_0, and similar in all diffusion-encoding directions. If the b-value is too high, the signal is similar to zero, and similar in all directions. In both cases, the measurement noise prevents an accurate quantification of the diffusivity and of any kind of diffusion anisotropy. Diffusion is best quantified at intermediate b-values, such as $1000\,\mathrm{s/mm^2}$ in biological tissue, where the signal is different from S_0 and from 0, and – if the tissue microstructure is anisotropic – considerably different in different directions.

The direction and weighting of the diffusion encoding can be expressed as a vector in diffusion-encoding space (*q-space*), where the direction of the vector corresponds to the diffusion direction and its length is proportional to $\sqrt{b}$. Usually a few dozen q-space coordinates are sampled during a scan.

Diffusion MRI is richly illustrated and described in Chapters 1–3 of [Mori, 2007]. It is also described in detail in [Johansen-Berg and Behrens, 2013, Jones, 2010]

2.1.4 The Spin Echo in Diffusion MRI

The aforementioned diffusion time Δ is relatively long (many milliseconds). The transversal magnetization decays during that time due to relaxation. This causes the signal to drop, whereas the considerable thermal noise in the measurements remains the same. In other words, the signal-to-noise ratio (SNR) drops. To prevent a low SNR, a radiofrequency pulse between the excitation pulse and the signal readout is used that flips the magnetization by 180°. This causes spins that dephased due to field inhomogeneity before the 180° pulse to rephase after it. The reason for the rephasing is the following. Spins that saw a slightly stronger field due to field inhomogeneity precessed slightly faster than other spins and were "ahead" of other spins in terms of phase; the 180° pulse flips them such that they are "behind" other spins in terms of phase and gradually catch up with the other spins because the other spins still precess more slowly due to the largely unchanged field inhomogeneity. When the rephasing is complete (i.e. after the same amount of time as the dephasing time between the 90° pulse and the 180° pulse), the signal reaches another peak, called the *spin echo*. This is when signal readout is performed in order to ensure the highest possible SNR before the signal decays again.

The 180° pulse not only inverts the dephasing caused by the field inhomogeneity, but also inverts the dephasing caused by the first diffusion-encoding gradient (which was applied before the 180° pulse). Therefore, the second diffusion-encoding gradient must be applied in the same direction as the first one (and not in the opposite direction as in the case described in Section 2.1.3 without a 180° pulse) in order to rephase the non-diffusing spins.

Apart from the described sequence of radiofrequency pulses and (pulsed) gradients, myriads of other pulse sequences are possible, yielding different im-

age contrasts. For an in-depth introduction, see [Bernstein et al., 2004]. Notable recent examples are random pulse parameters [Ma et al., 2013] and pulse sequences optimized to make the loud vibration of coils sound like music for improved patient comfort [Ma et al., 2016].

2.2 Proteins

Proteins are molecules and molecular complexes that are among the main building blocks of life. Different proteins carry out a multitude of functions related to metabolism, reproduction, signal transduction, repair, immune response, development and maintenance of body structure, motion, and more. Proteins play a central role in processes of life as well as many diseases. Understanding their function is essential.

The structure of a molecule dictates how the molecule can spatially interact with other molecules physically and chemically, i.e. the structure dictates the function. Therefore, inferring the structure of proteins is a central problem in biology and medicine.

2.2.1 Protein Structure

A protein chain is a molecule that consists of any long sequence of amino acid residues. There are 20 main types of amino acid residues. Their linear arrangement in the chemical structure of the protein chain is called the *sequence* of the protein or its *primary structure*.

Physically, a protein chain has many freely rotatable chemical bonds, i.e. degrees of freedom for its 3D structure. Due to Brownian motion (random collisions with other molecules), parts of the molecule rotate against each other. Depending on relative positions between atoms, van der Waals forces (e.g. attraction due to opposite partial charges) act between them, making certain conformations (3D arrangements of the molecule) more energetically stable than others. In this dynamic process, the molecule tends to stay in energetically stable conformations, as it is much more difficult to undo them through Brownian motion at moderate temperatures. Immediately after and even as early as during its synthesis, a protein chain begins this "folding" process.

The local 3D arrangement of residues that are neighbors in the primary structure is referred to as the *secondary structure*. In other words, a local segment of the sequence may fold into a so-called secondary-structure element, for example a helix. There are several types of helices with different tightness (about 3.0 residues per helical turn in 3_{10}-helices, about 3.6 residues per turn in α-helices, and about 4.1 residues per turn in π-helices) and handedness (most of the naturally occurring helices are right-handed). This classification of secondary-structure elements is based on the angles (ϕ, ψ) of the chemical bonds in the protein backbone (i.e. between the amino acid residues). For example, the backbone angles ϕ and ψ within an α-helix are very roughly $-60°$ and $-45°$, respectively. A typical helix is about a dozen residues long. However, for example the yeast protein Bud6 (PDB ID 3okq) contains an α-helix that is 79 residues long. The other common secondary-structure element apart from the α-helix is the β-strand. A β-strand is structurally stable if it forms a so-called β-sheet together with other β-strands.

Overall, the secondary-structure elements of one or several protein chains fold together into a global, energetically stable 3D conformation. For each

protein chain, the global 3D structure is called the *tertiary structure*. If several protein chains form a stable 3D structure together, it is called the *quaternary structure*.

The term *protein* may refer either to one protein chain, or to a structural complex consisting of several protein chains.

Most (but not all) natural proteins have a unique stable energy well that is easily accessible by the random folding process. This is referred to as Anfinsen's dogma. Some exceptions exist, such as fold-switching [Porter and Looger, 2018]. Even after folding, and even if the structure is stable, the bonds of the molecule remain in constant motion, varying slightly around the energy minimum due to Brownian motion. In other words, the stable energy well is not infinitely narrow, i.e. due to incessant random collisions with other molecules such as water molecules, the atoms within each molecule are incessantly moving, most of them only slightly and in an elastic manner, maintaining their rough location relative to each other. See for example the videos in the Supplementary Information of [Bock et al., 2013] for a comparison of such random motion of the atoms on a nanosecond scale and the slower stochastic transitions between slightly different folds throughout the functional cycle of the molecular complex on a microsecond scale.

2.2.2 From DNA to Protein

The main steps in the creation of a protein from the gene that codes for it are the following:

$$\text{DNA} \xrightarrow[\text{scription}]{\text{Tran-}} \text{pre-mRNA} \xrightarrow[\text{etc.}]{\text{Splicing}} \underset{\text{mRNA}}{\text{Mature}} \xrightarrow[\text{lation}]{\text{Trans-}} \underset{\text{protein}}{\text{Nascent}} \xrightarrow[\text{modifications}]{\text{Posttranslational}} \text{Protein}$$

Each processing step is catalyzed by specific biomolecules. In the following, we describe only the main products of each step, rather than the catalysts and byproducts.

First, the gene is *transcribed* from the DNA into a so-called *precursor messenger RNA* (also called *precursor mRNA* or *pre-mRNA*). This *primary transcript* molecule consists of a sequence of nucleotides that is complementary in terms of base pairing to the template DNA strand of the respective gene. This complementarity means that each cytosine base in the DNA is transcribed into a guanine base in the RNA, and analogously guanine to cytosine, thymine to adenine, and adenine to uracil (note that the uracil is rare in DNA, whereas thymine is rare in RNA). Also note that not only pre-mRNA is transcribed from DNA, but also *noncoding RNA* (ncRNA) that take on a multitude of functions [Cech and Steitz, 2014].

The pre-mRNA then undergoes *posttranscriptional modifications* that turn it into a *mature mRNA*. A modification that can influence the protein sequence in several different ways is *splicing*, i.e. the removal of non-coding regions (*introns*) from the pre-mRNA. A large percentage of genes yield RNA that has several *splicing variants*, i.e. splicing of the respective pre-mRNA sequence can happen in more than one way, in general yielding different protein sequences.

This is referred to as *alternative splicing*. While the number of splicing variants can be moderate for most genes, a gene of the "fruit fly" *Drosophila* potentially has up to 38000 splicing variants [Schmucker et al., 2000]. The resulting cell surface proteins (produced in neurons) can be used to distinguish individual neurons from each other. This is thought to contribute to guiding the neuronal connectivity [Schmucker et al., 2000, Neves et al., 2004]. Other posttranscriptional modifications are *capping* and *polyadenylation* – chemical reactions that add extensions to the two ends of the RNA molecule. The result of all these modifications is a *mature mRNA*.

The mature mRNA is *translated* into a protein chain according to the *genetic code*, i.e. each triplet of consecutive bases in the mRNA is either translated into a certain amino acid type out of 20 or terminates the chain. For example, the nucleotide base triplet uracil–cytosine–uracil codes for the amino acid serine, whereas uracil–adenine–adenine is one of the three *stop codons*, i.e. it terminates the translation. Note that insertions or deletions of bases in the RNA sequence (for example due to DNA mutations or alternative splicing) whose length is not a multiple of 3 cause a shift of the *reading frame* (subdivision of the mRNA sequence into coding triplets), such that an entirely different protein sequence is produced.

There are many mechanisms by which the aforementioned steps are *regulated* depending on the location (tissue type), time, and environmental conditions. Very different species share many of their genes, i.e. reuse similar "building blocks" (biomolecules). Many differences between species are due to differences in the regulation of gene expression (i.e. when which "building blocks" are created in what amounts) rather than merely due to differences between genes that code for biomolecules.

2.3 Deep Learning

Machine learning (ML) is a large and successful family of heavily dataset-driven algorithms. Instead of processing each *sample* of data (stand-alone piece of data) independently in a handcrafted hard-wired way like many other algorithms do, ML algorithms use entire datasets of samples to adjust the rules of their behavior towards each sample. This allows them to process each sample more optimally, if additional information which is valuable for the processing of this sample is contained in some of the other samples.

2.3.1 Supervised Learning and Generalization

One of the most popular settings of ML is *supervised learning*. In supervised learning, a *training set* consisting of samples is given. Each such training sample number i consists of an *input* $x_i \in X$ (from some set X called the *input feature space*) and an *output target* $t_i \in Y$ (from some set Y called the *output feature space*). The goal is to find a mapping f that satisfies the property $f(x_i) \approx t_i$ for many of the training samples (x_i, t_i) as well as additional properties called *inductive bias*, detailed below. In other words, the training set provides examples of what approximately f should output for certain inputs. After optimizing the mapping f to match the training set, f is applied to a *test set*, i.e. samples for which only inputs but no ideal outputs are known. For test samples that are not in the training set, the mapping f should output values that are in certain ways consistent with the training set, i.e. *generalize* well to previously unseen samples. Which properties of f are considered as good generalization depends on the application at hand. Some properties can be hardwired by limiting the search for f to candidates that have these properties. Other properties can be approximated by choosing a search procedure that is likely to find an f that fulfills these properties to a large extent. These "hard" and "soft" constraints on f are also called *inductive bias*. For details, see [Kukačka et al., 2017].

For a more formal description of good generalization, one can assume that data samples come from an unknown probability distribution P over the product space $X \times Y$. The goal of supervised learning is then to find a mapping $f : X \to Y$ such that for most inputs $x \in X$ which have relatively high probability density according to (the marginal of) P, the output $f(x)$ is close to a high-density region of the conditional (given x) probability distribution of the data over Y.

If the mapping f is probabilistic, it is called a *generative model* and usually tries to model the entire distribution (conditional distribution given the conditioning variable x). More often though, the mapping f is deterministic. In that case, if several output values are similarly valid for a given input, some compromise value is used.

Failures to perform well on test samples can be roughly divided into two categories: *underfitting* and *overfitting*. In the case of underfitting, the model performs badly on training samples *and* on test samples. A common reason for underfitting is that the family of models over which the search for f happens

consists of models that are too simple for the dataset, or that the search procedure fails to discover models that are good at least on the training set, let alone the test set. An example is when too little time is allotted to the search. In the case of overfitting, the model performs well on training samples, but badly on test samples. A common reason for overfitting is that the family of models is too large and the models are too complicated, i.e. models that perform well on the training set offer too many inappropriate possibilities of what to output for test samples, rather than only a few appropriate possibilities. For an overview of techniques that improve the results, see [Kukačka et al., 2017].

The term *feature* can have two meanings: a dimension of a feature space (each sample has a value; for example each tree has a height), or a subset of a feature space (each sample either belongs to that subset, i.e. has that feature, or not; for example each tree either has a height and thickness such that it withstands a certain wind speed without breaking, or not). These two meanings of the term "feature" can be to a certain extent considered two sides of the same coin, because *feature extractors* (for example layers of a neural network) can map (low-level) features such as tree height and thickness into a different feature space in which one of the dimensions describes (higher-level) features such as wind-endurance of a tree.

2.3.2 Popular Neural-Network Architectures

The term *artificial neural network*, often simply *neural network* (NN), nowadays can refer to almost any mapping f_w with parameters w. These parameters are the optimization variables in a "training" procedure that seeks to optimize some loss function. Neural networks are an important subfield of machine learning.

Network of Layers A neural network f_w can be decomposed into a directed acyclic graph of "intermediate" mappings called layers, for example

$$f_w(x) \equiv f_w^{(5)}\left(f_w^{(4)}\left(f_w^{(2)}(x)\right), f_w^{(3)}\left(f_w^{(1)}(x), f_w^{(2)}(x)\right)\right) \tag{2}$$

with layers $f_w^{(1)}$ to $f_w^{(5)}$, where not necessarily each of the parameters w influences each layer. This decomposition of f_w into layers is not unique, it can be performed in many ways, but usually only one or a few ways are chosen. An example of several competing decompositions is that some publications and programming frameworks consider dropout (setting random features to zero [Srivastava et al., 2014]) and/or nonlinearities as standalone layers, whereas others consider them as part of the layer that performs the preceding affine mapping.

Features that are the outputs of some layer and the inputs of subsequent layers (rather than outputs of the neural network) are referred to as *latent features*.

The graph representation of so-called recurrent neural networks can be drawn with cycles, or alternatively "unrolled" into a directed acyclic graph.

The success of neural networks stems from the fact that, even though each layer performs a simple mapping (for example an affine mapping followed by

a simple nonlinear function), there are many layers, i.e. all layers together can learn to perform quite complicated mappings, and all of them are usually trained jointly (end-to-end) to optimize some final goal, which is defined using all available (training) data. Over the training iterations, all layers are iteratively improved to work optimally with each other.

Multilayer Perceptrons The most famous classical NN architecture is the *multilayer perceptron* (MLP): a function composition (i.e. the graph of layers is a directed path, i.e.

$$f_w = f_w^{(L)} \circ f_w^{(L-1)} \circ \cdots \circ f_w^{(2)} \circ f_w^{(1)}, \tag{3}$$

where L is the number of layers) of so-called *fully connected layers*. A fully connected layer number ℓ performs an arbitrary affine mapping followed by a simple nonlinear function, i.e. it has the form

$$f^{(\ell)}_{W^{(\ell)},b^{(\ell)}}(x) = s^{(\ell)}\left(W^{(\ell)}x + b^{(\ell)}\right) \tag{4}$$

with trainable *weight matrix* $W^{(\ell)}$, trainable *bias vector* $b^{(\ell)}$, and a fixed nonlinearity $s^{(\ell)}(\cdot)$.

The Terms "Neural", "Network", "Weights" Multilayer perceptrons and similar NN architectures have coined the terms "neural network" and "weights". Interestingly, these terms are less appropriate for some newer NN architectures, but are still in use for historical reasons. A fully connected layer has a graphical interpretation as a network of neurons, see Fig. 2.

However, in some newer layer types, identification of individual neurons is more difficult or less commonly done; or biological neural networks are not the primary inspiration for their low-level structure. The term "neural network" is nonetheless used for many types of parametric mappings.

On the other hand, the term "network" has seen an increase in importance. MLPs have a dense network of connections within each layer (hence the term "neural network") but a very simple network of connections between layers. Newer architectures have a more complicated network of connections between layers, for example due to skip-connections [Ronneberger et al., 2015, He et al., 2016, Huang et al., 2017] or parallel processing paths [Szegedy et al., 2016, Szegedy et al., 2017].

In MLPs, most of the trainable parameters (W but not b) are *multiplied* by a layer's inputs, i.e. they *weight* the layer's inputs. For this historical reason, the term "weights" is often extended to trainable parameters that are used differently than for weighting the inputs.

Equivariance An important family of properties of neural network layers is *equivariance* (and its special cases *invariance* and *same-equivariance*). Intuitively, invariance of a mapping f (e.g. of a neural network) means that applying certain transformations to the input of f does not affect the output. This

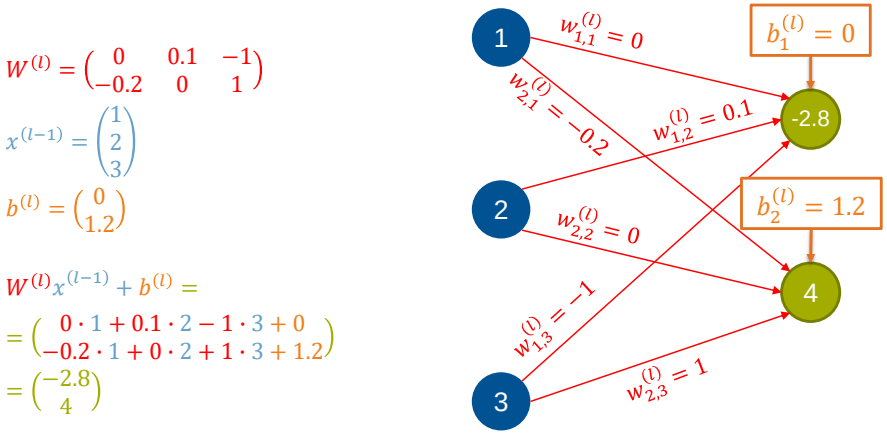


Figure 2: Graphical interpretation of a fully connected layer (formula on the left) as a network of neurons (right). The input is the feature vector $(1, 2, 3)^T$, the connections (red) of the three respective input neurons (blue) to the two output neurons (green) have associated weights (red). The activation of each output neuron is the respective bias value (orange) plus a weighted sum of the inputs. A nonlinearity (not shown) would further modify the activations before they serve as input to the next layer. Similar biological neural networks have inspired the usage of such artificial neural network architectures and coined the term "artificial neural network".

is useful for example if objects in images should be detected well, regardless of their location and orientation, i.e. with invariance under spatial shifts and rotations. Same-equivariance means that applying certain transformations to the input of f changes the output via the same transformation. For example, rotating the input rotates the output in the same way, which is useful in applications such as image segmentation. More generally, equivariance of f means that applying some transformation ϕ_g to some input x causes the output to change via a certain corresponding transformation ψ_g that does not depend on x:

Definition 1. A function $f : X \to Y$ is **equivariant** under a group G (with some *group actions* ϕ and ψ of G that transform X and Y, respectively) if

$$f(\phi_g[x]) = \psi_g[f(x)] \quad \forall g \in G \quad \forall x \in X, \tag{5}$$

where ϕ_g is the action of g on X, i.e. a transformation (for example rotation) of the input of f, and ψ_g is the action of g on Y, i.e. an "associated" (via the same g) but possibly different transformation (for example linear transformation) of the output of f. The usage of $g \in G$ to associate ψ_g with ϕ_g is important for correct composition of several transformations. For example, if ϕ_g is a 180°

rotation, i.e. $\phi_g \phi_g$ is the identity mapping, then $\psi_g \psi_g$ should also be the identity mapping.

Definition 2. A special case of equivariance is **same-equivariance** [Dieleman et al., 2016], when $Y = X$ and $\psi = \phi$. In some sources, same-equivariance is called equivariance, and what we call equivariance is called covariance.

Definition 3. A special case of equivariance is **invariance**, when ψ is the identity, e.g. when rotating the input of f has no influence on the output.

Invariance and same-equivariance are not the only types of equivariance that are common in deep learning. Some network layers possess an equivariance where in general $\psi_g \neq \phi_g$ (unlike same-equivariance), but where there are nonetheless as many different group actions ψ_g as there are different group actions ϕ_g (like in same-equivariance; unlike invariance, where all ψ_g are equal). Using the example where ϕ_g are rotations of images, such an equivariance ensures that spatial features are detected equally well regardless of their orientation, but nonetheless information about their orientation is not discarded (unlike in the case of invariance) and can be used by the following neural network layers to assess the relative orientation of several features to each other, or their absolute orientation. For example, convolutions with so-called steerable filter banks possess an equivariance where ϕ_g are 2D spatial rotations of the input and ψ_g represent not only the corresponding rotations of the output but also a corresponding rotation of the feature-space coordinate system, i.e. a certain linear reweighting of the image channels.

Note that invariant mappings discard information about which transformation ϕ_g was applied to the input. Even if the entire network is overall invariant, it is often better to discard information about ϕ_g in late layers than in early ones, in order to be able to assess the relative orientations between intermediate-level features in intermediate layers. Thus, oftentimes the first many layers are equivariant, and the last ones are invariant, all with compatible transformations (group actions), making the network overall invariant.

Note that terms like "equivariant under rotations" usually imply that ϕ_g are *spatial* rotations for the input layer, but not necessarily for every layer; rather, the abstract group G is the rotation group $\mathrm{SO}(n)$, whereas the group *actions* are not always simple spatial rotations.

If the content of a rectangular image is rotated (and/or translated), the "field of view" changes, i.e. features that used to be in the corners disappear and new features appear in the corners. This has caused some confusion as to how rectangular images can be processed in a rotation-equivariant way. The explanation is this: An output value of the neural network is only affected if the change of features is within its receptive field and the network has not learned from data that such a feature change should be irrelevant for the output. For example, if the network has learned from data that the detection of cats should be unaffected by the presence of clouds in other parts of the image, then a cloud appearing in the corner due to rotating the camera will not affect the detection of a cat in the image.

Similarly, if two input features are rotated relatively to each other, an output value changes only if both input features are within its receptive field and the network has not learned that such a relative rotation should be processed equivariantly.

For details about equivariant neural networks, see [Della Libera et al., 2019, Cohen et al., 2019, Weiler and Cesa, 2019, Kondor and Trivedi, 2018] and resources available from `https://e3nn.org/#`.

Layer Types An overview of popular types of neural-network layers is given in Table 1. Layers should be chosen such that the overall network architecture (not only some of the layers) has properties that are appropriate for the dataset/application at hand, such as equivariance under certain transformations.

The overall architecture is equivariant if each layer $f^{(\ell)}$ is equivariant with the same group actions $\psi_g^{(\ell)}$ at the output (see Definition 1) as the group actions $\phi_g^{(\ell+1)}$ at the input of the subsequent layer $f^{(\ell+1)}$, i.e. $\phi_g^{(\ell+1)} = \psi_g^{(\ell)}$ for all $g \in G$. This can be shown by applying Eq. (5) twice as follows:

$$\begin{aligned} & f^{(\ell+1)} \circ f^{(\ell)} \circ \phi^{(\ell)} \\ = \; & f^{(\ell+1)} \circ \psi^{(\ell)} \circ f^{(\ell)} \\ = \; & f^{(\ell+1)} \circ \phi^{(\ell+1)} \circ f^{(\ell)} \\ = \; & \psi^{(\ell+1)} \circ f^{(\ell+1)} \circ f^{(\ell)}. \end{aligned} \tag{6}$$

Receptive Field The *receptive field* is the part of the input that influences a given output value. For example, in a pixels-to-pixels convolutional network, not necessarily all input pixels influence each output pixel. The receptive field size depends on the filter sizes of the layers and on the number of layers. Choosing an appropriate receptive field size is an important step in network architecture design. It should be chosen such that it contains the features that are relevant for the respective output pixel, but not too much more, in order to avoid having to learn from data that far-away features are irrelevant.

Convolutional Layers Convolutional layers (see Table 1) provide equivariance under translations (i.e. under spatial shifts) and locality of feature extraction (i.e. their receptive field is box-shaped) for numerical data defined on a regular grid. Translation-equivariance is useful when features should be detected equally well regardless of their location (spatial position). Locality of feature extraction is useful when features should be detected equally well regardless of what other features are present at other locations (beyond the receptive field). For example:

- An object in a photograph should be detected equally well regardless of its location in the photograph (translation-equivariance) and regardless of the presence of other objects elsewhere in the photograph (locality of feature extraction).

Layer type	Operation	Important properties
Fully connected layer	Affine mapping with trainable parameters W, b followed by hardwired nonlinearity $s(\cdot)$: $s(Wx + b)$	Can approximate arbitrary function if layer has enough neurons (but many neurons may cause overfitting if training set is limited)
Convolutional layer	Discrete multi-channel convolution with trainable filters, addition of trainable bias values, hardwired nonlinearity: Eq. (7)	- Same-equivariance under translations (spatial shifts) - Along dimensions with filter size 1: specifically permutation-equivariance - Locality of feature extraction (box-shaped receptive field)
Global pooling along certain dimensions	Aggregation[a] of values along one or several array dimensions	Invariance under permutations along pooled dimensions
Local pooling	Aggregation[a] of values over a local neighborhood	Invariance under certain slight spatial distortions (certain slight changes of spatial locations of features) and certain slight changes of feature values, see [Kukačka et al., 2017]
Random noise such as batch normalization[b] or dropout	Modify feature values, for example set random entries to zero, or scale each channel to have a certain mean and variance across the batch	Throughout the training iterations, prevents overfitting certain feature configurations

[a] Typical aggregation functions are maximum, average, or sum.
[b] Batch normalization is random because the composition of the batches is random.

Table 1: Common types of neural-network layers and the properties that they have by design or that emerge through training. Layers with certain properties can be chosen depending on the project goals.

- A musical motif in a spectrogram should be detected equally well regardless of when it is played (equivariance under translations along the time dimension), of how its pitch is transposed (equivariance under translations along the frequency dimension), of what is played before and after it (locality along the time dimension), and of what is played in other registers (locality along the frequency dimension).
- A sequence pattern in a protein sequence should be detected equally well regardless of where in the protein sequence it appears (translation-equivariance) and of what other patterns appear elsewhere (locality).

For these and other useful properties of data processing methods, see the *scale-space axioms* for example in Chapter 5.1.1 of [Bredies and Lorenz, 2011].

The inputs and outputs of convolutional layers are *feature maps* represented

as multi-channel arrays, i.e. arrays with a "channels dimension" and $d \in \mathbb{N}$ "spatial" (or temporal) dimensions. Examples include $d = 1$ for sequence data such as audio waveforms, protein sequences, or text, $d = 2$ for images or audio spectrograms, $d = 3$ for videos or 3D volumes, and $d = 4$ for sequences of volumetric data. The trainable parameters of a convolutional layer are the *bias values* and the entries of the *kernels*. There is one kernel for each output channel and one bias value for each output channel. Each kernel has the same number of channels as the input of the layer. The inputs, kernels, and outputs have the same number d of "spatial" dimensions. The size of the kernel along the "spatial" dimensions corresponds to the size of the receptive field of the layer.

The output y of a convolutional layer is computed as follows from its input x, kernels w and bias values b:

$$y_{k,i_1,\ldots,i_d} = s\left(\sum_{c,j_1,\ldots,j_d} w_{k,c,j_1,\ldots,j_d} x_{c,i_1-j_1,\ldots,i_d-j_d} + b_c\right), \tag{7}$$

where k is the output channel and thus also the index of the kernel that produces that output channel, $(i_1, \ldots, i_d)$ are the coordinates in the output feature map, c enumerates all input channels and thus also the channels of the kernels as well as the bias values, and $(j_1, \ldots, j_d)$ enumerate all coordinates in the kernel.

In other words, the convolutional layer performs a multi-channel discrete convolution of the input with the filters.

Padding *Padding* is the practice of extending spatial data beyond its borders by fake data, for example by zeros. When the mapping is "pixels-to-pixels" and the receptive field is larger than 1, then the output would have a reduced field of view compared to the input. If the field of view should not be so reduced (for example because the user requires predictions for every single pixel), padding can be used. In other cases and some programming frameworks, using padding can be more convenient than not using it.

However, the main problems with padding are:

1. Every layer whose input feature map gets padded (and all subsequent layers) have to jointly learn what to output for all likely arrangements of real and fake input data.

2. They might also have to learn to distinguish fake from real data, if the two can look similar and this distinction is important.

Learning these things in addition to the main learning goals is difficult, it may require a slightly better network architecture, and/or worsen the results of the main learning goals. The solutions and remedies to these problems are:

- By avoiding padding when possible, both aforementioned problems can be avoided.

- A slight remedy to the second problem is to introduce an additional channel that informs the neural network which data are fake [Golkov et al., 2016b].
- A slight remedy to the first problem is to pad only the input rather than the feature maps in each layer [Golkov et al., 2016b]. Then the fake data that the network would have to learn to deal with is only in the input, and not different in every layer.
  - If all padding values are equal, explicitly redundantly storing them in memory is a waste of memory. This can be avoided by storing unique values only once. More specifically, features that are obtained from real data or a mixture of real and fake data can be stored explicitly, whereas features that are obtained from fake data are equal in all pixels and can be stored only once (one value per channel), rather than redundantly in many padded pixels.

Nonlinearities Many layer types, for example most of the fully connected layers and convolutional layers, often consist of an affine mapping with trainable parameters, followed by an elementwise nonlinearity, i.e. a simple nonlinear $\mathbb{R} \to \mathbb{R}$ function applied independently to each entry of the array.

Historically, nonlinearities were used that are differentiable everywhere, for fear of incompatibility with gradient-based optimization. However, nonlinearities that are differentiable *almost everywhere* (but not everywhere) can actually be successfully used as well, because reaching the null set on which they are not differentiable has probability zero in case of infinite precision, and probability almost zero in case of finite precision. Moreover, even if that null set is reached, some value like zero or a subderivative can be used in place of the derivative, or the sample can be skipped. After a training step on this or other samples, the differentiable regions are almost certainly attained again.

Another historical misconception was to use nonlinearities whose output was limited to a small interval, for example $]0, 1[$ in the case of the sigmoid nonlinearity or $]-1, 1[$ in the case of the hyperbolic tangent. These functions *saturate* without reaching the value 1 nor above. This restriction was motivated by an overreliance on the inspiration from biological neural networks. In artificial neural networks, monotonic nonlinearities with limited output intervals in some cases caused gradients to be too small for effective gradient-based optimization, and limited the expressivity of the neural network (see the argument around Eq. (9) below).

Letting go of these two historical assumptions (differentiablity everywhere and limited output values) paved the way for the currently most successful nonlinearities, such as the rectified linear unit (ReLU) [Fukushima, 1980, Hahnloser et al., 2000, Jarrett et al., 2009, Nair and Hinton, 2010, Glorot et al., 2011]

$$\mathrm{ReLU}(z) = \max\{0, z\} \tag{8}$$

and similar ones. The ReLU is one of the simplest nonlinear functions imaginable, and it works remarkably well.

An important difference between saturating nonlinearities (such as the sigmoid) and non-saturating ones (such as the ReLU) is that the latter extrapolate differently (affinely, without saturating) into regions of feature space that are not covered by training samples, which appears to contribute to better results in practice. Moreover, the ReLU (and similar nonlinearities) can approximate a larger variety of functions well. This is evidenced by the fact that a few ReLUs can approximate a sigmoid well [Kukačka et al., 2017] (with small integrated squared error and small integrated absolute error), for example

$$\text{sigm}(x) \approx \text{ReLU}(x + 0.5) - \text{ReLU}(x - 0.5), \tag{9}$$

and thus ReLUs can also approximate functions that sigmoids can approximate, whereas a finite number of sigmoids cannot approximate a ReLU with finite error.

2.3.3 Appropriate Representations for Data Types

The goal of neural networks is to "disentangle" the information that is "entangled" (present, but not obvious) in the (input) features. If we can do a part of this task well, we usually should. For example, there are many ways to represent the same information as inputs to a neural network, and we should choose representations from which it is easy for the given network architecture to learn to extract relevant information, and difficult to learn to extract irrelevant information. So the choices of data representations and of network architecture are important for good results and should match each other well. The similarity of the high-level meanings of variable values should be ideally reflected by a proportional similarity of their numerical values in the chosen representation.

For example, categorical variables are usually one-hot encoded. This encoding indicates that no two categories are a priori more similar to each other than any other two.

Another example is the representation of angles. To avoid the problem that 0° and 359° have a very similar meaning in some given application but very different values, the sine and cosine of the angle (i.e. two features) can be used as the representation instead of the number of degrees. At the neural network input, this representation is partially redundant, but redundancy at inputs is not a problem, especially when no new dimensions of overfittable irrelevant features/noise are introduced (as would for example be the case when the resolution of images is too high for the goals, so that the neural network might overfit unimportant fine details). At the neural network output, this representation of angles is ambiguous, because a neural network imperfectly approximates the optimal input-output mapping, i.e. the predicted sine and cosine will not correspond to the same angle. This, however, can be easily solved by projecting the output onto the set of valid representations of angles, e.g. by normalizing the output vector to unit length. An important aspect here is that this projection is simple, i.e. the neural network can approximately "foresee" it in the sense that the predicted sine and cosine values will be only slightly modified by

the projection if the network has been trained to try to produce sine and cosine values that are approximately compatible with each other.

Thus, in cases where inputs with a similar meaning should produce outputs with a similar meaning (as is often the case), neither the aforementioned input representation nor the output representation and projection would be the cause of problems such as sudden jumps in the input-output mapping.

2.3.4 How to Design Neural-Network Architectures

When designing a neural-network architecture for solving some problem, inspiration can be taken from architectures that perform well on similar problems/datasets. For example, if appropriate, an existing successful architecture can be tried first, and then hyperparameter search can start from the optimum that has been determined for that similar dataset.

Another approach, particularly if the problem at hand has not yet been solved by neural networks, is to design the architecture from scratch. To do this, we found the following steps to be useful:

1. Formulate (hypothetical) features that might be possible and meaningful to extract from data in words.

2. Formulate (hypothetical) meaningful/possible information flow between the aforementioned features, i.e. which latent features can be extracted from which input features or from which other latent features, and which output features can be extracted from which latent features.

3. Choose data structures/representations that are appropriate (see Section 2.3.3) for storing the aforementioned features.

4. Define additional application-specific constraints that the neural network should fulfill, for example equivariance under certain transformations and usual success ingredients such as using layers consisting of a learned affine mapping followed by a simple nonlinearity. Many successful neural networks work like this, for example fully connected layers are affine followed by a simple nonlinearity, without any constraints; convolutional layers are affine followed by a simple nonlinearity, with the additional constraints of translation-equivariance and locality of feature extraction.

5. Design a neural-network architecture that offers the information flow from step 2 between the data structures from step 3 (and not too much other information flow) and fulfills the properties from step 4.

For examples of this approach, see [Golkov et al., 2016b] (included here as Section 3.1) and [Schuchardt et al., 2019].

2.3.5 Machine-Learning Tasks and Loss Functions

Formally, training a neural network for the ML task of supervised learning, i.e. tuning the trainable parameters w of a neural network f such that for each training input x_i the output of f is similar (in terms of some dissimilarity metric d) to the corresponding output target t_i, corresponds to the following optimization problem:

$$\underset{w}{\text{minimize}} \frac{1}{|\mathcal{D}|} \sum_{(x_i,t_i)\in\mathcal{D}} d\left(f_w(x_i), t_i\right) + R(\dots), \tag{10}$$

where $\mathcal{D}$ is the training set consisting of training samples (x_i, t_i), and R is an additional *regularization term* that may depend on w, the output of f, some derivatives thereof, and/or similar things [Kukačka et al., 2017]. This optimization problem is also referred to as *empirical risk minimization.*

While supervised learning (as described in Section 2.3.1) is a very popular and straightforward ML task, it is by far not the only one. Many other ML tasks (i.e. abstract goals) exist, for example:

- Semi-supervised learning: like supervised learning, but additional samples are available that are neither labeled (i.e. do not have known output targets) nor are in the test set. Valuable information about the distribution of data in input-feature space X can be used. For example, entire "islands" of data in input-feature space (all with the same label) can be classified correctly based on only one labeled sample, if the unlabeled samples provide sufficient information about the shape of the "island".

- Clustering with an unknown number k of clusters, i.e. assigning each sample from a dataset to one of k *clusters* (i.e. unlabeled classes freely defined by the algorithm) while also determining an optimal number k. Typical goals some or all of which are formulated in one way or another within the algorithm include:

  - Samples within each cluster should be similar to each other in some way, for example in terms of some similarity metric in some latent feature space.
  - Samples from different clusters should be dissimilar from each other.
  - Each cluster should contain neither too few nor too many samples.
  - The number k of clusters should be neither too small nor too large.

- Clustering with a known number k of clusters: like above, but without the need to find an optimal k automatically. See for example our survey and new methods in [Aljalbout et al., 2018].

- Novelty detection: A dataset labeled as the "normal" class is given. In the test dataset, samples should be classified as "normal" and "abnormal" (different from "normal"). See for example our new methods based on variational autoencoders in [Vasilev et al., 2019].

- Outlier detection: None of the samples are labeled. An "abnormal" minority (the outliers) and a "normal" majority should be distinguished, usually based on the outliers having more extreme feature values or rare combinations of features.
- Domain adaptation without target-domain labels: similar to supervised learning, but the test set has a different distribution in input-feature space than the training set, for example the "source domain" (training set inputs) are drawings, whereas the "target domain" (test set inputs) are photographs.
- Similarity learning: Labels are given for pairs of samples rather than for individual samples. Usually some form of (symmetric) (dis-)similarity measure should be learned for these pairs.
- Implicit density estimation: A model is trained to randomly produce samples that are not necessarily the same as in a given dataset, but from the same overall distribution.
- Explicit density estimation: As above, but also with the ability to explicitly formulate the entire distribution and not only draw individual samples from it.

These are only a few examples. Each setting can have a combination of non-default properties, such as labels missing for some training samples or entire classes, input features partially missing for some samples or for entire classes, the requirement to continuously produce results before all training samples have been processed (online learning), or the ability to actively query labels for certain samples (active learning).

Each ML task has a variety of specific algorithmic formulations, often as closed-form optimization problems in the form of training a neural network end-to-end with special cost functions. Each formulation of an ML task has its own advantages and disadvantages. None of the formulations is universally perfect for all possible datasets.

2.3.6 Training Procedure

The loss function in deep learning can have the form of Eq. (10) for supervised learning or other forms tailored to other machine-learning tasks such as the ones listed in Section 2.3.5 and/or to specific datasets (for example by using intermediate features in certain application-specific ways for goals such as optimizing perceptual similarity between images). Such loss functions are usually optimized using stochastic gradient descent, i.e. gradient descent on alternating random subsets (*batches*) of the training dataset $\mathcal{D}$.

"Something in between minimization and maximization" is also possible. In gradient-based optimization, a stop-gradient operator, see [van den Oord et al., 2017, Golkov et al., 2020a, Chen and He, 2020], or more generally a gradient-reversal layer [Ganin and Lempitsky, 2015] with some λ value can be used. This

does not strictly correspond to solving an optimization problem, but rather creates a dynamical system [Jin et al., 2020] or variational inequality [Gidel et al., 2019], as for example in some formulations of generative adversarial networks.

2.3.7 Regularization

Using the example of supervised learning and similar machine-learning tasks that take the form of Eq. (10), several components of a deep learning algorithm can be identified that influence its results:

- the data set $\mathcal{D}$,
- the neural network architecture f,
- the dissimilarity metric d,
- the regularization term $R(\dots)$,
- the optimization algorithm.

Each of these components can be modified to improve the results. This can be referred to as *regularization* in its general sense. For an overview of regularization techniques in deep learning, see our survey [Kukačka et al., 2017].

It is interesting to note that some regularization techniques can be considered from the perspective of more than one of the five aforementioned categories. For example, dropout [Srivastava et al., 2014] (i.e. randomly setting some features to zero) can be considered a stochastic building block of the network architecture f, or alternatively as a modification of the optimization algorithm such that the current weights are projected to a subspace of weight space, an optimization step is performed in that subspace, and then the weights previously discarded by the projection are restored [Kukačka et al., 2017].

Overfitting can be considered as the extraction of irrelevant features and wrongful assignment of meaning to them based on spurious (coincidental) correlations present in the training set. *Early stopping*, i.e. stopping the training after some time, for example when the performance on a *validation dataset* (dataset with given labels, but excluded from the training set) starts decreasing, can prevent overfitting. This can be attributed to some spurious correlations being more rare in the training set than relevant ones, which requires the learning of the spurious ones to take more time, and/or to a choice of an appropriate network architecture which learns to extract irrelevant features more slowly or not at all.

2.3.8 Inspecting What the Network has Learned

During iterations of improving the neural-network architecture, it might be helpful to inspect what the current architecture learns in practice and compare that to the hypothetical features that were expected as per Section 2.3.4. Moreover, applications such as computer vision that are visually accessible can provide

valuable general intuition about neural networks for work with applications that are visually less accessible.

A large neural network learns a high-dimensional and complex mapping. This mapping is difficult to fully visualize and interpret. Instead, certain aspects of this mapping can be visualized individually. Examples include (see also [Grün et al., 2016, Seifert et al., 2017]):

- Visualizing all latent feature maps that the network produces for one given test sample, as well as weights [Harley, 2015].
- Visualizing samples from a dataset that cause the strongest activations of a certain neuron [Zeiler and Fergus, 2014, Bau et al., 2017].
- As above, with additional interactive visualization of the neurons that have the strongest-weighted connections to a certain neuron[1].
- Generating input samples that maximize the activation of a certain neuron [Simonyan et al., 2013].
- Given an input sample, generating other input samples that produce similar latent features in a certain layer. This partially reveals what information about input samples is maintained in those layers, and what information is lost. There are several approaches that produce quite different results due to the different ways how they formulate the goals (a comparison can be found in [Dosovitskiy and Brox, 2016]):
  - formulation as an inverse problem using data-consistency in latent feature space [Mahendran and Vedaldi, 2015],
  - formulation as another neural network that is trained to achieve data-consistency in input feature space [Dosovitskiy and Brox, 2016],
  - formulation as another neural network that is trained to achieve data-consistency in input feature space and in latent space, as well as to mimic the probability distribution of realistic samples [Dosovitskiy and Brox, 2016].
- Visualizing how strongly which input features influenced the output for a given test sample. The influence of each feature on the output depends on the current values of the other features, and features are not independent in the dataset. There are various ways how these relationships between input features can be taken into account, for example:
  - "neutralizing" (replacing by some placeholder value) several features which (due to their neighborhood in space) might strongly correlate, and visualizing how strongly that influences a certain output neuron [Zeiler and Fergus, 2014],

[1] `http://people.csail.mit.edu/torralba/research/drawCNN/drawNet.html`

- visualizing which (infinitesimal) modifications of all input features would most strongly influence a certain output neuron [Zeiler and Fergus, 2014, Springenberg et al., 2014],
- choosing a network architecture such that the prediction is a sum of contributions from all image regions, so that the contributions can be mapped in space [Zhou et al., 2016] and similar techniques [Selvaraju et al., 2017].

Apart from visualizations, also statistical analysis is possible, for example of correlations between features or between quality metrics, also across different networks [Nguyen et al., 2020].

2.4 Riemannian Manifolds and Variational Analysis on Them

Many successful image-processing methods consider images as defined on a spatially continuous domain and not only their discretization to pixels. These methods often yield better results than entirely discrete methods, as shown for example by [Klodt et al., 2008, Nieuwenhuis et al., 2013]. In [Golkov et al., 2015b] (included here as Section 3.4), we use the advantages of spatially continuous treatment of images as well as synergies between all six data dimensions to improve the reconstruction of diffusion MRI data from raw measurements.

In diffusion MRI, in many cases each 2D diffusion-weighted image is reconstructed from measured data individually. However, the measurements contain noise, causing the results of the reconstruction to be imperfect. Additional information about the true noise-free image intensities is partially contained in the measurements performed for neighboring diffusion weightings (q-space coordinates) as well as neighboring z-slices, which have different random noise values but similar true intensity values.

The true image intensities are similar at points with the same physical-space coordinate, same diffusion weighting, and a similar (not same) diffusion direction. In other words, the derivatives of the true image along spherical shells in q-space are mostly small.

Similarly, the true image intensities are similar at points with the same q-space coordinate and a similar (not same) physical-space coordinate. In other words, the derivatives of the true image along the physical space are mostly small. This fact is used more often for derivatives along the x- and y-dimension (because they are entangled in the measurements via the Fourier transform) than along the z-direction.

Similarly, in each 3D spatial voxel, the values of the so-called *orientation distribution function* (ODF, a function in each voxel that maps each possible diffusion direction to a real-valued relative strength of diffusion in that direction, see for example [Lin et al., 2003]) are similar for similar diffusion directions. In other words, the derivatives of the ODF along the sphere on which it is defined are small.

In [Golkov et al., 2015b], we therefore propose regularization terms that keep all aforementioned derivatives small, ensuring noise reduction and synergies along all six data dimensions, while a data-consistency term ensures that the reconstructed image is consistent with the raw measurements.

The resulting overall cost function can be optimized using a primal-dual hybrid gradient optimization method for nonlinear operators [Valkonen, 2014], which is a modification of a popular method for linear operators [Pock et al., 2009].

In the following, we describe central basic mathematical constructs used in our work. For details, see [Golkov et al., 2015b] (included here as Section 3.4).

2.4.1 Derivatives and Integrals on Manifolds

An n-dimensional *manifold* is a topological space that in a neighborhood around each point is homeomorphic to the n-dimensional Euclidean space. For details about manifolds and the topics mentioned below, see for example [Kühnel, 2015, do Carmo, 2015, Spivak, 1999].

A *scalar field* on a manifold M is a function $M \to K$, where K is a field (not to be confused with the term "scalar field") like $\mathbb{R}$ or $\mathbb{C}$.

A *differentiable manifold* is a manifold equipped with a so-called *differentiable atlas*, i.e. a collection of so-called *charts* (local coordinate systems) that cover the entire manifold such that the *transition maps* $\varphi_j \circ \varphi_i^{-1}$ between all pairs (φ_i, φ_j) of charts are differentiable.

A *tangent space* T_pM at a point p of an n-dimensional differentiable manifold M is an n-dimensional real vector space with a certain correspondence between objects on the manifold and objects in the tangent space. One of the ways to formalize this correspondence is to define T_pM as the vector space of all possible velocity vectors at p of differentiable curves on M that pass through p.

A *vector field* on a manifold M is a function that maps each point $p \in M$ to a vector from the respective tangent space T_pM.

A *Riemannian manifold* is a differentiable manifold equipped with an inner product (the *Riemannian metric*) on the tangent space at each point. The Riemannian metric induces geometric notions on manifolds such as angles, lengths, and curvature. Angles and lengths induce the notion of volume, i.e. measure. The measure can in turn can be used to compute integrals of scalar functions on the manifold.

The *gradient* of a scalar function f on a Riemannian manifold M is a certain vector field on that manifold. Each vector of this vector field points in the direction of steepest ascent on f. The vector length gives the rate of change of f for a unit length in that direction. Note that the definition of lengths uses the Riemannian metric.

Without a Riemannian metric, there is no notion of lengths of vectors, i.e. no unit-length vectors for comparing the rate of change of f in different directions, so there is no gradient. However, there is still a *total differential*, i.e. a covector field on M in which the covector at each point p outputs the rate of change of f along each vector. With a Riemannian metric, a covector can be associated with the direction of the unit-length vector with which it produces the largest output, but without a Riemannian metric, a covector cannot be associated with such a direction.

Given a Riemannian manifold M, the *Sobolev space* $\mathbb{H}^{m,p}(M)$ is, intuitively, the vector space of functions on M whose derivatives up to the m-th one (including the "0-th derivative", i.e. the function itself) exist and are p-integrable. For $p = 2$, we write $\mathbb{H}^m(M) = \mathbb{H}^{m,p}(M)$. For details about this generalization of Sobolev spaces to Riemannian manifolds, see [Hebey, 1996]. In [Golkov et al., 2015b] (included here as Section 3.4), our images are from $\mathbb{H}^2(\mathbb{R}^3 \times S^2)$. We denote the coordinate along the first three dimensions of $\mathbb{R}^3 \times S^2$, i.e. along $\mathbb{R}^3$, by $y \in \mathbb{R}^3$, and the space of the derivatives $\nabla_y U$ along the first three dimen-

sions of our aforementioned images U by $\mathbb{H}^1(\mathbb{R}^3 \times S^2, \mathbb{R}^3)$, where the latter $\mathbb{R}^3$ indicates that $\nabla_y U$ consists of three partial derivatives: one along each of the three first dimensions.

The formal definitions of some of the aforementioned mathematical objects use some charts (local coordinate systems) on the manifold for constructing the objects, but eventually yield objects that are independent of the choice of charts and can be dealt with (reasoned about, visualized) without any particular charts in mind. Our descriptions above therefore focused on their final form and the intuition and intentions behind it, without the temporary charts used during their construction.

2.4.2 Convex Conjugate and Convex Biconjugate

In [Golkov et al., 2015b] (included as Section 3.4 here), we use the so-called *convex biconjugate* and a technique called *completing the square* to rewrite $\|\hat{x}\|^2$ as follows:

$$\|\hat{x}\|^2 = \sup_{\hat{y}} \langle \hat{x}, \hat{y} \rangle - \frac{1}{4} \|\hat{y}\|^2 . \tag{11}$$

In the following, we show the details of this reformulation.

The *convex conjugate* f^* (also known as the *Legendre–Fenchel transform*) of a real-valued function $f(\hat{x})$ is defined as

$$f^*(\hat{y}) = \sup_{\hat{x}} \langle \hat{y}, \hat{x} \rangle - f(\hat{x}). \tag{12}$$

In our case, we choose $f(\hat{x}) = \|\hat{x}\|^2$, and therefore its convex conjugate is

$$f^*(\hat{y}) = \sup_{\hat{x}} \langle \hat{y}, \hat{x} \rangle - f(\hat{x}) = \sup_{\hat{x}} \langle \hat{y}, \hat{x} \rangle - \|\hat{x}\|^2 . \tag{13}$$

This can be rewritten using a technique called *completing the square*, i.e. bringing a part of the expression into the form $\|a\|^2 - 2\langle b, a \rangle$ and adding $+\|b\|^2 - \|b\|^2$:

$$\begin{aligned} f^*(\hat{y}) &= \sup_{\hat{x}} \langle \hat{y}, \hat{x} \rangle - \|\hat{x}\|^2 \\ &= \sup_{\hat{x}} -\Big(\|\hat{x}\|^2 - \langle \hat{y}, \hat{x} \rangle \Big) \\ &= \sup_{\hat{x}} -\left(\|\hat{x}\|^2 - 2 \left\langle \frac{\hat{y}}{2}, \hat{x} \right\rangle \right) \\ &= \sup_{\hat{x}} -\left(\|\hat{x}\|^2 - 2 \left\langle \frac{\hat{y}}{2}, \hat{x} \right\rangle + \left\| \frac{\hat{y}}{2} \right\|^2 - \left\| \frac{\hat{y}}{2} \right\|^2 \right). \end{aligned} \tag{14}$$

Note that $\|a-b\|^2 = \|a\|^2 - 2\langle b, a\rangle + \|b\|^2$, i.e. in our case

$$\begin{aligned}\left\|\hat{x} - \frac{\hat{y}}{2}\right\|^2 &= \left\langle \hat{x} - \frac{\hat{y}}{2}, \hat{x} - \frac{\hat{y}}{2}\right\rangle \\ &= \left\langle \hat{x} - \frac{\hat{y}}{2}, \hat{x}\right\rangle + \left\langle \hat{x} - \frac{\hat{y}}{2}, -\frac{\hat{y}}{2}\right\rangle \\ &= \langle \hat{x}, \hat{x}\rangle + \left\langle -\frac{\hat{y}}{2}, \hat{x}\right\rangle + \left\langle \hat{x}, -\frac{\hat{y}}{2}\right\rangle + \left\langle -\frac{\hat{y}}{2}, -\frac{\hat{y}}{2}\right\rangle \\ &= \|\hat{x}\|^2 - 2\left\langle \frac{\hat{y}}{2}, \hat{x}\right\rangle + \left\|\frac{\hat{y}}{2}\right\|^2,\end{aligned} \tag{15}$$

which allows to rewrite Eq. (14) as follows:

$$\begin{aligned}f^*(\hat{y}) &= \sup_{\hat{x}} -\left(\|\hat{x}\|^2 - 2\left\langle \frac{\hat{y}}{2}, \hat{x}\right\rangle + \left\|\frac{\hat{y}}{2}\right\|^2 - \left\|\frac{\hat{y}}{2}\right\|^2\right) \\ &= \sup_{\hat{x}} -\left(\left\|\hat{x} - \frac{\hat{y}}{2}\right\|^2 - \left\|\frac{\hat{y}}{2}\right\|^2\right).\end{aligned} \tag{16}$$

The term $-\left\|\hat{x} - \frac{\hat{y}}{2}\right\|^2$ is nonpositive, its supremum 0 is attained at $\hat{x} = \frac{\hat{y}}{2}$. Therefore,

$$\left(\|\hat{x}\|^2\right)^* = f^*(\hat{y}) = \sup_{\hat{x}} \left\|\frac{\hat{y}}{2}\right\|^2 = \left\|\frac{\hat{y}}{2}\right\|^2. \tag{17}$$

According to the Fenchel–Moreau theorem, if a function f is convex and lower semi-continuous, then it is equal to its own *convex biconjugate* f^{**} (i.e. the convex conjugate of its convex conjugate). In our case, $f(\hat{x}) = \|\hat{x}\|^2$ is continuous (hence lower semi-continuous) and convex, hence equal to its convex biconjugate:

$$\|\hat{x}\|^2 = \left(\|\hat{x}\|^2\right)^{**}. \tag{18}$$

By combining Eqs. (18), (17), and (12), we get

$$\|\hat{x}\|^2 \overset{(18)}{=} \left(\|\hat{x}\|^2\right)^{**} \overset{(17)}{=} \left(\left\|\frac{\hat{y}}{2}\right\|^2\right)^* \overset{(12)}{=} \sup_{\hat{y}} \langle \hat{y}, \hat{x}\rangle - \left\|\frac{\hat{y}}{2}\right\|^2. \qquad \square$$

3 Papers

3.1 Protein Contact Prediction from Amino Acid Co-Evolution Using Convolutional Networks for Graph-Valued Images

In the following paper, we use deep learning for predicting information about the structure of proteins from a rich representation of the information we can extract about their evolutionary history.

Some mutations of proteins leave their structure and function largely unaffected. Therefore, many different versions (*homologs*) of each protein exist in different organisms, and, in the case of gene duplication, several homologs exist in the same organism. For a given protein sequence, so-called homology search algorithms can identify its homologs in protein-sequence databases and align their sequences into a *multiple sequence alignment*.

Some amino acids of a protein are in physical contact with each other, with van der Waals forces acting between them and maintaining the overall structure of the protein. If one of two amino acids residues that form a contact mutates and this mutation makes the contact less stable, there is an evolutionary pressure on the other amino acid residue to mutate such that the contact becomes more stable again. Thus, the latter mutations are favored by natural selection and are likely to be encountered. In other words, the two amino acid residues *co-evolve*. The statistics of this amino acid co-evolution can be inferred using *direct coupling analysis* algorithms.

Due to this relationship between protein structure and amino acid co-evolution statistics, it is possible to predict the former from the latter. We propose using rich direct co-evolution statistics (without handcrafted suboptimal preprocessing) and deep learning for this purpose. A *contact map*, i.e. a matrix that for all pairs of protein positions indicates whether there is a physical contact, is used as the output target for each protein.

We design the neural network architecture based on our domain knowledge about the requirements: detecting protein motifs equally well regardless of their location in the sequence and of motifs at other locations; detecting secondary-structure elements and physical contacts between them (including the relative orientation between secondary-structure elements that are in physical contact).

Our processing pipeline, which is optimized on a training set rather than handcrafted suboptimally, outperforms existing state-of-the-art methods.

The author of this dissertation contributed substantially to the content of the paper, in particular concerning parts of the idea, parts of the code, neural network training, and writing parts of the paper.

Protein contact prediction from amino acid co-evolution using convolutional networks for graph-valued images

Vladimir Golkov[1], **Marcin J. Skwark**[2], **Antonij Golkov**[3], **Alexey Dosovitskiy**[4], **Thomas Brox**[4], **Jens Meiler**[2], **and Daniel Cremers**[1]
[1] Technical University of Munich, Germany
[2] Vanderbilt University, Nashville, TN, USA
[3] University of Augsburg, Germany
[4] University of Freiburg, Germany
golkov@cs.tum.edu, marcin@skwark.pl, antonij.golkov@student.uni-augsburg.de,
{dosovits,brox}@cs.uni-freiburg.de, jens.meiler@vanderbilt.edu, cremers@tum.de

Abstract

Proteins are responsible for most of the functions in life, and thus are the central focus of many areas of biomedicine. Protein structure is strongly related to protein function, but is difficult to elucidate experimentally, therefore computational structure prediction is a crucial task on the way to solve many biological questions. A contact map is a compact representation of the three-dimensional structure of a protein via the pairwise contacts between the amino acids constituting the protein. We use a convolutional network to calculate protein contact maps from detailed evolutionary coupling statistics between positions in the protein sequence. The input to the network has an image-like structure amenable to convolutions, but every "pixel" instead of color channels contains a bipartite undirected edge-weighted graph. We propose several methods for treating such "graph-valued images" in a convolutional network. The proposed method outperforms state-of-the-art methods by a considerable margin.

1 Introduction

Proteins perform most of the functions in the cells of living organisms, acting as enzymes to perform complex chemical reactions, recognizing foreign particles, conducting signals, and building cell scaffolds – to name just a few. Their function is dictated by their three-dimensional structure, which can be quite involved, despite the fact that proteins are linear polymers composed of only 20 different types of amino acids. The sequence of amino acids dictates the three-dimensional structure and related proteins share both structure and function. Predicting protein structure from amino acid sequence remains a problem that is still largely unsolved.

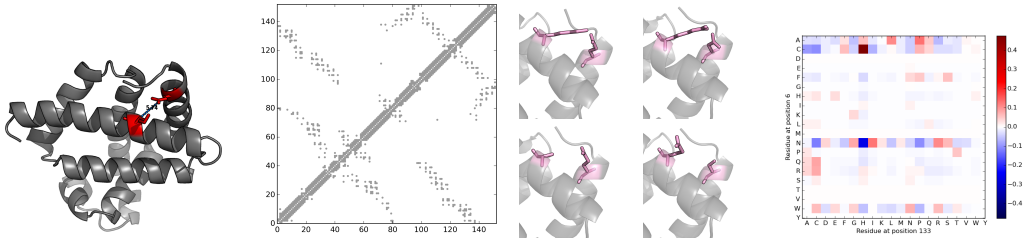
1.1 Protein structure and contact maps

The *primary structure* of a protein refers to the linear sequence of the amino acid residues that constitute the protein, as encoded by the corresponding gene. During or after its biosynthesis, a protein spatially folds into an energetically favourable conformation. Locally it folds into so-called *secondary structure* (α-helices and β-strands). The global three-dimensional structure into which the entire protein folds is referred to as the *tertiary structure*. Fig. 1a depicts the tertiary structure of a protein consisting of several α-helices.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Protein structure is mediated and stabilized by series of weak interactions (physical contacts) between pairs of its amino acids. Let L be the length of the sequence of a protein (i.e. the number of its amino acids). The tertiary structure can be partially summarized as a so-called *contact map* – a sparse $L \times L$ matrix C encoding the presence or absence of physical contact between all pairs of L amino acid residues of a protein. The entry $C_{i,j}$ is equal to 1 if residues i and j are in contact and 0 if they are not. Intermediate values may encode different levels of contact likeliness.

We use these intermediate values without rounding where possible because they hold additional information. The "contact likeliness" is a knowledge-based function derived from Protein Data Bank, dependent on the distance between $C\beta$ atoms of involved amino acids and their type. It has been parametrized based on the amino acids' heavy atoms making biophysically feasible contact in experimentally determined structures.



(a) Tertiary structure (b) Contact map (c) Variants of contact (d) Co-evolution statistics

Figure 1: Oxymyoglobin (a) and its contact between amino acid residue 6 and 133. Helix–helix contacts correspond to "checkerboard" patterns in the contact map (b). Various variants of the contact 6/133 encountered in nature (native pose in upper left, remaining poses are theoretical models) (c) are reflected in the co-evolution statistics (d).

2 Methods

The proposed method is based on inferring direct co-evolutionary couplings between pairs of amino acids of a protein, and predicting the contact map from them using a convolutional neural network.

2.1 Multiple sequence alignments

As of today the UniProt Archive (UniParc [1]) consists of approximately 130 million different protein sequences. This is only a small fraction of all the protein sequences existing on Earth, whose number is estimated to be on the order of 10^{10} to 10^{12} [2]. Despite this abundance, there exist only about 10^5 sequence families, which in turn adopt one of about 10^4 folds [2]. This is due to the fact that homologous proteins (proteins originating from common ancestors) are similar in terms of their structure and function. Homologs are under evolutionary pressure to maintain the structure and function of the ancestral protein, while at the same time adapting to the changes in the environment.

Evolutionarily related proteins can be identified by means of homology search using dynamic programming, hidden Markov models, and other statistical models, which group homologous proteins into so-called *multiple sequence alignments*. A multiple sequence alignment consists of sequences of related proteins, aligned such that corresponding amino acids share the same position (column). The 20 amino acid types are represented by the letters A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y. Besides, a "gap" (represented as "–") is used as a 21st character to account for insertions and deletions.

For the purpose of this work, all the input alignments have been generated with jackhmmer, part of HMMER package (version 3.1b2, http://hmmer.org) run against the UniParc database released in summer 2015. The alignment has been constructed with the E-value inclusion threshold of 1, allowing for inclusion of distant homologs, at a risk of contaminating the alignment with potentially evolutionarily unrelated sequences. The resultant multiple sequence alignments have not been modified in any way, except for removal of inserts (positions that were not present in the protein sequence of interest). Notably, contrary to many evolutionary approaches, we did *not* remove columns that (a) contained many gaps, (b) were too diverse or (c) were too conserved. In so doing, we emulated a fully automated prediction regime.

2.2 Potts model for co-evolution of amino acid residues

Protein structure is stabilized by series of contacts: weak, favourable interactions between amino acids adjacent in space (but not necessarily in sequence). If an amino acid becomes mutated in the course of evolution, breaking a favourable contact, there is an evolutionary pressure for a compensating mutation to occur in the interacting partner(s) to restore the protein to an unfrustrated state. These pressures lead to amino acid pairs varying in tandem in the multiple sequence alignments. The observed covariances can subsequently be used to predict which of the positions in the protein sequence are close together in space.

The directly observed covariances are by themselves a poor predictor of inter-residue contact. This is due to transitivity of correlations in multiple sequence alignments. When an amino acid *A* that is in contact with amino acids *B* and *C* mutates to *A'*, it exerts a pressure for *B* and *C* to adopt to this mutation, leading to a spurious, indirect correlation between *B* and *C*. Oftentimes these spurious correlations are more prominent than the actual, direct ones. This problem can be modelled in terms of one- and two-body interactions, analogous to the Ising model of statistical mechanics (or its generalization – the Potts model). Solving an inverse Ising/Potts problem (inferring direct causes from a set of observations), while not feasible analytically, can be accomplished by approximate, numerical algorithms. Such approaches have been recently successfully applied to the problem of protein contact prediction [3, 4].

One of the most widely-adopted approaches to this problem is pseudolikelihood maximization for inferring an inverse Potts model (plmDCA [3, 5]). It results in an $L \times L \times 21 \times 21$ array of inferred evolutionary couplings between pairs of the L positions in the protein, described in terms of 21×21 coupling matrices. These coupling matrices depict the strength of evolutionary pressure at particular amino acid type pairs (e.g. histidine–threonine) to be present at this position pair – the higher the value, the more pressure there is. These values are not directly interpretable, as they depend on the environment the amino acids are in, their propensity to mutate and many other factors. So far, the best approach to obtain scores corresponding to contact propensities was to compute the Frobenius norm of individual coupling matrices rendering a contact matrix, which then has been subject to average product correction [6]. Average product correction scales the value of contact propensity based on the mean values for involved positions and a mean value for the entire contact matrix.

As there is insufficient data to conclusively infer all the parameters, and coupling inference is inherently ill-posed, regularization is required [3, 5]. Here we used l_2 regularization with $\lambda = 0.01$.

These approaches to reduce each 21×21 coupling matrix to only one value discard valuable information encoded in matrices, consequently leading to a reduction in expected predictive capability. In this work we use the entire $L \times L \times 21 \times 21$ coupling data $\mathbf{J}$ in their unmodified form. The value $\mathbf{J}_{i,j,k,l}$ quantifies the co-evolution of residue type k at location i with residue type l at location j. The $L \times L \times 21 \times 21$ array $\mathbf{J}$ serves as the main input to the convolutional network to predict the $L \times L$ contact map C.

The following symmetries hold: $C_{i,j} = C_{j,i}$ and $\mathbf{J}_{i,j,k,l} = \mathbf{J}_{j,i,l,k} \forall i, j, k, l$.

2.3 Convolutional neural network for contact prediction

The goal of this work is to predict the contact $C_{i,j}$ between residues i and j from the co-evolution statistics $\mathbf{J}_{i,j,k,l}$ obtained from pseudolikelihood maximization [3]. Not only the local statistics $(\mathbf{J}_{i,j,k,l})_{k,l}$ for fixed (i, j) but also the neighborhood around (i, j) is informative for contact determination. Particularly, contacts between different secondary structure elements are reflected both in the spatial contact pattern, such as the "checkerboard" pattern typical for helix–helix contacts, cf. Fig. 1b (the "i" and "j" dimensions), as well as in the residue types (the "k" and "l" dimensions) at (i, j) and in its neighborhood. Thus, a convolutional neural network [7] with convolutions over (i, j), i.e. learning the transformation to be applied to all $w \times w \times 21 \times 21$ windows of $(\mathbf{J}_{i,j,k,l})$, is a highly appropriate method for prediction of $C_{i,j}$.

The features in each "pixel" (i, j) are the entries of the 21×21 co-evolution statistics $(\mathbf{J}_{i,j,k,l})_{k,l \in \{1,\dots,21\}}$ between amino acid residues i and j. Fig. 1d shows the co-evolution statistics of residues 6 and 133, i.e. $(\mathbf{J}_{6,133,k,l})_{k,l \in \{1,\dots,21\}}$, of oxymyoglobin. These $21 \cdot 21$ entries can be vectorized to constitute the feature vector of length 441 at the respective "pixel".

The neural network input $\mathbf{J}$ and at its output C should have the same size along the convolution dimensions "i" and "j". In order to achieve this, the input boundaries are padded accordingly (i.e. by the receptive window size) along these dimensions. In order to help the network distinguish the padding values (e.g. zeros) from valid co-evolution values, the indicator function of the valid region (1 in the valid $L \times L$ region and 0 in the padded region) is introduced as an additional feature channel.

Our method is based on pseudolikelihood maximization [3] and convolutional networks, *plmConv* for short.

2.4 Convolutional neural network for bipartite-graph-valued images

The fixed order of the 441 features can be considered acceptable since any input–output mapping can in principle be learned, assuming we have sufficient training data (and an appropriate network architecture). However, if the amount of training data is limited then a better-structured, more compact representation might be of great advantage as opposed to requiring to see most of the possible configurations of co-evolution. Such more compact representations can be obtained by relaxing the knowledge of the identities of the amino acid residues, as described in the following.

The features at "pixel" (i, j) correspond to the weights of a (complete) bipartite undirected edge-weighted graph $K_{21,21}$ with $21 + 21$ vertices, with the first disjoint set of 21 vertices representing the 21 amino acid types at position i, the second set representing the 21 amino acid types at position j, and the edge weights representing co-evolution of the respective variants. Thus, $B = (\mathbf{J}_{i,j,k,l})_{k,l \in \{1,\ldots,21\}}$ is the biadjacency matrix of this graph, i.e. $A = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}$ is its adjacency matrix. The edge weights (i.e. entries of B) are different at each "pixel" (i, j).

There are different possibilities of passing these features (the entries of B) to a convolutional network. We propose and evaluate the following possibilities to construct the feature vector at pixel (i, j):

1. Vectorize B, maintaining the order of the amino acid types;
2. Sort the vectorized matrix B;
3. Sort the rows of B by their row-wise norm, then vectorize;
4. Construct a histogram of the entries of B.

While the first method maintains the order of amino acid types, all others produce feature vectors that are invariant to permutations of the amino acid types.

2.5 Generalization to arbitrary graphs

In other applications to graph-valued images with general (not necessarily bipartite) graphs, similar transformations as above can be applied to the adjacency matrix A. An additional useful property is the special role of the diagonal of A. Node weights can be included as additional features, and accordingly reordered.

There has been work on neural networks which can process functions defined on graphs [8, 9, 10, 11]. In contrast to these approaches, in our case the input is defined on a regular grid, but the value of the input at each location is a graph.

2.6 Data sets

The Critical Assessment of Techniques for Protein Structure Prediction (CASP) is a bi-annual community-wide experiment in blind prediction of previously unknown protein structures. The prediction targets vary in difficulty, with some having a structure of homologous proteins already deposited in the Protein Data Bank (PDB), considered easy targets, some having no detectable homologs in PDB (hard targets), and some having entirely new folds (free modelling targets). The protein targets vary also in terms of available sequence homologs, which can range from only a few sequences to hundreds of thousands.

We posit that the method we propose is robust and general. To illustrate its performance, we have intentionally trained it on a limited set of proteins originating from CASP9 and CASP10 experiments

and tested it on CASP11 proteins. In so doing, we emulated the conditions of a real-life structure prediction experiment.

The proteins from these experiments form a suitable data set for this analysis, as they (a) are varied in terms of structure and "difficulty", (b) have previously unknown structures, which have been subsequently made public, (c) are timestamped and (d) they have been subject to contact prediction attempts by other groups whose results are publicly available. Therefore, training on CASP9 and CASP10 data sets allowed us to avoid cross-contamination. We are reasonably confident that any performance of the method originates from the method's strengths and is not a result of overfitting.

The training has been conducted on a subset of 231 proteins from CASP9 and CASP10, while the test set consisted of 89 proteins from CASP11 (all non-cancelled targets). Several proteins have been excluded from the training set for technical reasons: lack of any detectable homologs, too many homologs detected, or lack of structure known at the time of publishing of CASP sets. The problems with the number of sequences can be alleviated by attempting different homology detection strategies, which we did not do, as we wanted to keep the analysis homogeneous.

2.7 Neural network architecture

Deep learning has strong advantages over handcrafted processing pipelines and is setting new performance records and bringing new insights in the biomedical community [12, 13]. However, parts of the community are adopting deep learning with certain hesitation, even in areas where it is essential for scientific progress. One of the main objections is a belief that the craft of network architecture design and the network internals cannot be scientifically comprehended and lack theoretical underpinnings. This is a false belief. There are scientific results to the contrary, concerning the loss function [14] and network internals [15].

In the present work, we design the network architecture based on our knowledge of which features might be meaningful for the network to extract, and how.

The first layer learns 128 filters of size 1×1. Thus, 441 input features are compressed to 128 learned features. This compression enforces the grouping of similar amino acids by their properties. Examples of important properties are hydrophobicity, polarity, and size. Some of the most relevant parts of the input information "cysteine (C) at position i has a strongly positive evolutionary coupling with histidine (H) at position j" (cf. Fig. 1d) is that the amino acids co-evolving have certain hydrophilicity properties; that both are polar; and that the one at position i is rather small and the one at position j is rather large; etc. One layer is sufficient to perform such a transformation. Note that we do not handcraft these features; the network learns feature extractors that are optimal in terms of the training data. Besides, compressing the inputs in this optimal way also reduces the number of weights of the subsequent layer, thus regularizing the model in a natural way, and reducing the run time and memory requirements.

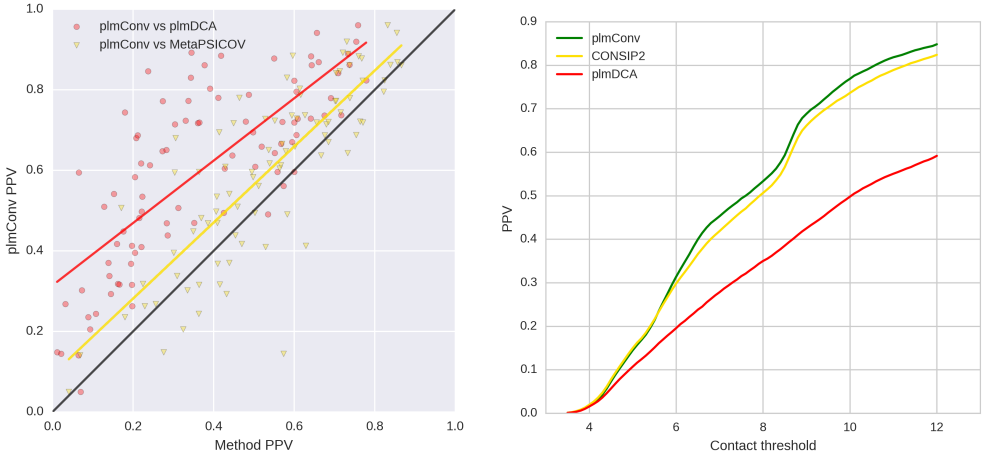
The second layer learns 64 filters of size 7×7. This allows to see the context (and end) of the contact between two secondary structure elements (e.g. a contact between two β-strands). In other words, this choice of the window size and number of filters is motivated by the fact that information such as "(i, j) is a contact between a β-strand at i and a β-strand at j, the arrangement is antiparallel, the contact ends two residues after i (and before j)" can be captured from a 7×7 window of the data, and well encoded in about 64 filters.

The third and final layer learns one filter (returning the predicted contact map) with the window size 9×9. Thus, the overall receptive window of the convolutional network is 15×15, which provides the required amount of context of the co-evolution data to predict the contacts. Particularly, the relative position (including the angle) between two contacting α-helices can be well captured at this window size. At the same time, this deep architecture is different from having, say, a network with a single 15×15 convolutional layer because a non-deep network would require seeing many possible 15×15 configurations in a non-abstract manner, and would tend to generalize badly and overfit. In contrast, abstraction to higher-level features is provided by preceding layers in our architecture.

We used mean squared error loss, dropout 0.2 after input layer, 0.5 after each hidden layer, one pixel stride, no pooling. The network is trained in Lasagne (https://github.com/Lasagne) using the Adam algorithm [16] with learning rate 0.0001 for 100 epochs.

3 Results

To assess the performance of protein contact prediction methods, we have used the contact likeliness criterion for Cβ distances (cf. Introduction), but the qualitative results are not dependent on the criterion chosen. We have evaluated predictions both in terms of Top 10 pairs that are predicted most likely to be in contact. It is estimated that in a protein one can observe L to $3L$ contacts, where L is the length of the amino acid chain. Thus we have also evaluated greater numbers of predicted contacts. We have assessed the predictions with respect to the sequence separation. It is widely accepted that it is more difficult to predict long-range contacts than the ones separated by few amino acids in the sequence space. At the same time, it is the long-range contacts that are most useful for restraining the protein structure prediction simulations [17]. Maintaining the order of amino acid types (feature vector construction method #1) yielded the best results in our case, which we focus on exclusively in the following.



(a) PPV for our approach vs. plmDCA20 and MetaPSICOV

(b) PPV for discussed methods as a function of contact definition

Figure 2: Method performance. Panel (a): prediction accuracy of plmConv (Y-axis) vs plmDCA and MetaPSICOV (X-axis, in red and yellow, respectively); lines: least square fit, circles: individual comparisons. Panel (b): prediction accuracy, depending on contact definition. X-axis: Cβ distance threshold for amino acid pair to be in contact.

plmConv yields more accurate predictions than plmDCA. We compared the predictive performance of the proposed plmConv method to plmDCA in terms of positive predictive value (PPV) at different prediction counts and different sequence separations (see Table 1 and Fig. 2a). Regardless of the chosen threshold, plmConv yields considerably higher accuracy. This effect is particularly important in context of long-range contacts, which tend to be underpredicted by plmDCA and related methods, but are readily recovered by plmConv. The notable improvement in predictive power is important, given that both plmDCA and plmConv use exactly the same data and same inference algorithm, but differ in the processing of the inferred co-evolution matrices. We posit that this may have longstanding implications for evolutionary coupling analysis, some of which we discuss below.

plmConv is more accurate than MetaPSICOV, while remaining more flexible. We compared our method to MetaPSICOV [18, 19], a method that performed best in the CASP11 experiment. We observed that plmConv results in overall higher prediction accuracy than MetaPSICOV (see Table 1 and Fig. 2a). This holds for all the criteria, except for the top-ranked short contacts. MetaPSICOV performs slightly better at the top-ranked short-range contacts, but they are easier to predict, and less useful for protein folding [17]. It is worth noting that MetaPSICOV achieves its high prediction accuracy by combining multiple sources of co-evolution data (including methods functionally identical to plmDCA) with predicted biophysical properties of a protein (e.g. secondary structure) and a feed-forward neural network. In plmConv we are able to achieve higher performance, by using (a) an arbitrary alignment and (b) a single co-evolution result, which potentially allows for tuning the hyperparameters of (a) and (b) to answer relevant biological questions.

Separation	Method	Top 10	$L/10$	$L/5$	$L/2$	L
All	MetaPSICOV	0.797	0.761	0.717	0.615	0.516
	plmDCA	0.598	0.570	0.525	0.435	0.356
	plmConv	**0.807**	**0.768**	**0.729**	**0.663**	**0.573**
Short	MetaPSICOV	**0.754**	**0.683**	**0.583**	0.415	0.294
	plmDCA	0.497	0.415	0.318	0.229	0.178
	plmConv	0.724	0.654	0.581	**0.438**	**0.320**
Medium	MetaPSICOV	0.710	0.645	0.559	0.419	0.302
	plmDCA	0.506	0.438	0.355	0.253	0.180
	plmConv	**0.744**	**0.673**	**0.583**	**0.428**	**0.304**
Long	MetaPSICOV	0.594	0.562	0.522	0.436	0.339
	plmDCA	0.536	0.516	0.455	0.372	0.285
	plmConv	**0.686**	**0.651**	**0.616**	**0.531**	**0.430**

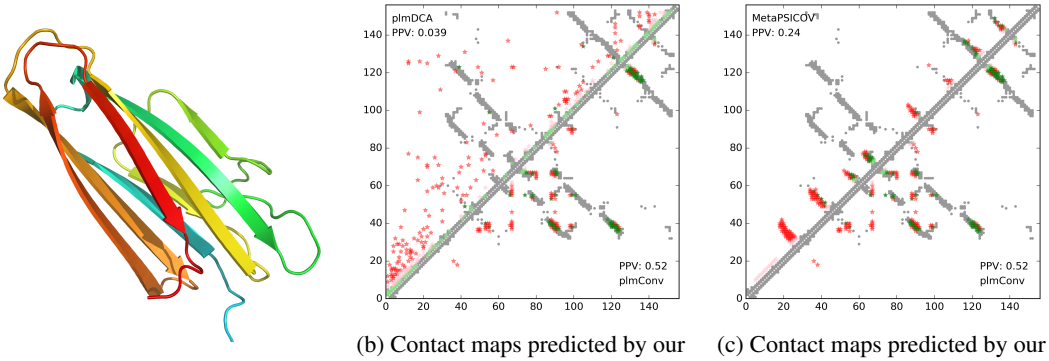
Table 1: Positive predictive value for all non-local (separation 6^+ positions), short-range, mid-range and long-range ($6-11$, $12-23$ and 24^+ positions) contacts. We demonstrate results for Top 10 contacts per protein, as well as customary thresholds of $L/10$, $L/5$, $L/2$ and L contacts per protein, where L is the length of the amino acid chain.

Figure 3: Positive predictive value for described methods at L contacts considered as a function of the information content of the alignment. Scatter plot: observed raw values. Line plot: rolling average with window size 15.

plmConv pushes the boundaries of inference with few sequences. One of the major drawbacks of statistical inference for evolutionary analysis is its dependence on availability of high amounts of homologous sequences in multiple sequence alignments. Our method to a large extent alleviates this problem. As illustrated in Fig. 3, plmConv outperforms plmDCA accross all the range. MetaPSICOV appears to be slightly better at the low-count end of the spectrum, which we believe is due to the way MetaPSICOV augments the prediction process with additional data – a technique known to improve the prediction, that we have expressly *not* used in this work.

plmConv predicts long-range contacts more accurately. As mentioned above, it is the long-range contacts which are of most utility for protein structure prediction experiments. Table 1 demonstrates that plmConv is highly suitable for predicting long range contacts, yielding better performance across all the contact count thresholds.

T0784: a success story. One of the targets in CASP11 (target ID: T0784) was a DUF4425 family protein (BACOVA_05332) from *Bacteroides ovatus* (PDB ID: 4qey). The number of identifiable sequence homologs for this protein was relatively low, which resulted in uninterpretable contact map obtained by plmDCA. The same co-evolution statistics used as input to plmConv yielded a contact map which not only was devoid of the noise present in plmDCA's contact map, but also uncovered numerous long-range contacts that were not identifiable previously. The contact map produced by plmConv for this target is also of much higher utility than the one returned by MetaPSICOV. Note in Fig. 4c how MetaPSICOV prediction lacks nearly all the long-range contacts, which are present in the plmConv prediction.



(a) Structure

(b) Contact maps predicted by our method vs. plmDCA

(c) Contact maps predicted by our method vs. MetaPSICOV

Figure 4: An example of one of CASP11 proteins (T0784), where plmConv is able to recover the contact map, which other methods cannot. True contacts (ground truth) marked in gray. Predictions of respective methods are marked in color, with true positives in green and false positives in red. Predictions along the diagonal with separation of 5 amino acids or less have not been considered in computing positive predictive value and have been marked in lighter colors in the plots.

4 Discussion and Conclusions

In this work we proposed an entirely new way to handle the outputs of the co-evolutionary analyses of multiple sequence alignments of homologous proteins. We demonstrated that this method is considerably superior to the current ways of handling the co-evolution data, able to extract more information from them, and consequently greatly aid protein contact prediction based on these data. Contact prediction with our method is more accurate and 2 to 3 times faster than with MetaPSICOV.

Relevance to the field. Until now, the utility of co-evolution-based contact prediction was limited because most of the proteins that had sufficiently high amount of sequence homologs had also their structures determined and available for comparative modelling. As plmConv is able to predict high-accuracy contact maps from as few as 100 sequences, it opens a whole new avenue of possibilities for the field. While there are only a few protein families that have thousands of known homologs but no known structure, there are hundreds which are potentially within the scope of this method. We postulate that this method should allow for computational elucidation of more structures, be it by means of pure computational simulation, or simulation guided by predicted contacts and sparse experimental restraints.

plmConv allows for varying prediction parameters. One of the strengths of the proposed method is that it is agnostic to the input data, in particular to the way input alignments are constructed and to the inference parameters (regularization strength). Therefore, one could envision using alignments of close homologs to elucidate the co-evolution of a variable region in the protein (e.g. variable regions of antibodies, extracellular loops of G protein–coupled receptors etc.), or distant homologs to yield structural insights into the overall fold of the protein. In the same way, one could vary the regularization strength of the inference, with stronger regularization allowing for more precise elucidation of the few couplings (and consequently contacts) that are most significant for protein stability or structure from the evolutionary point of view. Conversely, it is possible to relax the regularization strength and let the data speak for itself, which could potentially result in a better picture of the overall contact map and give a holistic insight into the evolutionary constraints on the structure of the protein in question.

The method we propose is directly applicable to a vast array of biological problems, being both accurate and flexible. It can use arbitrary input data and prediction parameters, which allows the end user to tailor it to answer pertinent biological questions. Most importantly, though, even if trained on the heavily constrained data set, it is able to produce results exceeding in predictive capabilities those of the state-of-the-art methods in protein contact prediction at a fraction of computational effort, making it perfectly suitable for large-scale analyses. We expect that the performance of the method will further improve when trained on a larger, more representative set of proteins.

Acknowledgments Grant support: Deutsche Telekom Foundation, ERC Consolidator Grant "3DReloaded", ERC Starting Grant "VideoLearn".

References

[1] Rasko Leinonen, Federico Garcia Diez, David Binns, Wolfgang Fleischmann, Rodrigo Lopez, and Rolf Apweiler. UniProt archive. *Bioinformatics*, 20(17):3236–3237, 2004.

[2] In-Geol Choi and Sung-Hou Kim. Evolution of protein structural classes and protein sequence families. *Proceedings of the National Academy of Sciences of the United States of America*, 103(38):14056–61, 2006.

[3] Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 87(1):1–19, 2013.

[4] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences of the United States of America*, 108(49):E1293–301, 2011.

[5] Christoph Feinauer, Marcin J. Skwark, Andrea Pagnani, and Erik Aurell. Improving contact prediction along three dimensions. *PLOS Computational Biology*, 10(10):e1003847, 2014.

[6] S. D. Dunn, L. M. Wahl, and G. B. Gloor. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340, 2008.

[7] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten ZIP code recognition. *Neural Computation*, 1(4):541–551, 1989.

[8] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[9] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*, 2014.

[10] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv:1506.05163*, 2015.

[11] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. *Advances in Neural Information Processing Systems 28*, pages 2215–2223, 2015.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: convolutional networks for medical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[13] Vladimir Golkov, Alexey Dosovitskiy, Jonathan Sperl, Marion Menzel, Michael Czisch, Philipp Samann, Thomas Brox, and Daniel Cremers. q-Space deep learning: twelve-fold shorter and model-free diffusion MRI scans. *IEEE Transactions on Medical Imaging*, 35(5):1344–1351, 2016.

[14] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 38:192–204, 2015.

[15] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[16] Diederik P. Kingma and Jimmy Lei Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[17] M Michael Gromiha and Samuel Selvaraj. Importance of long-range interactions in protein folding. *Biophysical Chemistry*, 77(1):49–68, 1999.

[18] David T. Jones, Tanya Singh, Tomasz Kosciolek, and Stuart Tetchner. MetaPSICOV: Combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics*, 31(7):999–1006, 2015.

[19] Tomasz Kosciolek and David T. Jones. Accurate contact predictions using covariation techniques and machine learning. *Proteins: Structure, Function and Bioinformatics*, 84(Suppl 1):145–151, 2016.

3.2 3D Deep Learning for Biological Function Prediction from Physical Fields

Determining the biological function of molecules such as proteins or small drug-like compounds is a central question in biology and pharmacology. Knowing the physical processes and chemical reactions in which molecules are involved helps to understand biological and pathological processes and to develop cures.

The three-dimensional structure of a molecule dictates how the molecule can interact with other molecules, i.e. its function. Therefore, our aim is to predict the biological function of molecules from their structure.

More specifically, the electron density and electrostatic potential field of a molecule dictate the attractive and repulsive forces (and thus the interactions) between the molecule in question and other molecules. Therefore, we estimate these two physical fields on a voxel grid and use them as input to a deep neural network to predict the function of the respective molecule. Thus, unlike previous methods, which use handcrafted features, our method uses the raw physical information and lets the neural network learn to extract features that are optimal for the task at hand.

Due to the quantum-mechanical intractability of the exact electron density of molecules, we use an approximation of the electron density, namely Gaussian kernels around the atom centers with the van der Waals radii as the bandwidth [Bernstein and Craig, 2010]. This is a quite good approximation especially for the electron-density values around the molecular surface, which dictate interactions with other molecules.

For proteins, we propose a new representation of three-dimensional structure, namely the usage of separate image channels for the electron density of different amino acid types and atom types. This allows the neural network to easily distinguish the amino acid types and atom types without having to learn to infer them from nuances of the structure.

To compute an estimation of the electrostatic potential field of small molecules, we estimate the partial charges using the Gasteiger–Marsili PEOE algorithm [Gasteiger and Marsili, 1978]. For proteins, we do not use the electrostatic potential field, because algorithms for computing partial charges of large molecules are error-prone, and because amino acid types largely dictate the partial charges and can be inferred from the electron density.

We propose various computational speed-ups for different sizes of molecules and datasets, for example keeping the dataset in GPU memory and computing randomly rotated physical fields on the fly with GPU acceleration, or precomputing the physical fields into files and randomly rotating them with GPU acceleration.

Our convolutional networks achieve results comparable to the state of the art on small molecules, and promising results on proteins. This indicates that neural networks are indeed capable of extracting valuable information contained in electron density and electrostatic potential fields.

Our input representations are also compatible with modern rotation-invariant neural networks.

The author of this dissertation contributed substantially to the content of the paper, in particular concerning parts of the idea, parts of the code, some of the experiments, and writing parts of the paper.

Accepted manuscript for the International Conference on 3D Vision (3DV). Published version: doi: 10.1109/3DV50981.2020.00103.

3D Deep Learning for Biological Function Prediction from Physical Fields

Vladimir Golkov [1], Marcin J. Skwark [2,3], Atanas Mirchev [1], Georgi Dikov [1,4],
Alexander R. Geanes [2], Jeffrey Mendenhall [2], Jens Meiler [2,5], and Daniel Cremers [1]
[1] Technical University of Munich [2] Vanderbilt University [3] InstaDeep [4] TomTom [5] Leipzig University

Abstract

Predicting the biological function of molecules, be it proteins or drug-like compounds, from their atomic structure is an important and long-standing problem. The electron density field and electrostatic potential field of a molecule contain the "raw fingerprint" of how this molecule can fit to binding partners. In this paper, we show that deep learning can predict biological function of molecules directly from their raw 3D approximated electron density and electrostatic potential fields. *Protein function based on Enzyme Commission numbers is predicted from the approximated electron density field. In another experiment, the activity of small molecules is predicted with quality comparable to state-of-the-art descriptor-based methods. We propose several alternative computational models for the GPU with different memory and runtime requirements for different sizes of molecules and of databases. We also propose application-specific multi-channel data representations.*

1. Introduction

Recent developments in experimental techniques for life sciences allow for studying a vast array of properties and characteristics of biologically relevant molecules. We can elucidate structures of proteins and small molecules, their composition (in terms of amino acid sequences and atoms), abundance and localization in the cells, to name just a few such traits. Most of these experimental efforts serve one purpose though – uncovering the *function* the molecule carries out in the living organism. Both for proteins and small molecules, the function can be described in terms of the effect that the molecule has on its interaction partners. This can in turn be expressed in terms of spatial interactions, such as the lock-and-key model of ligand affinity or enzyme specificity (based on spatial complementarity).

While elucidating the structures of biomolecules becomes easier, experimental function annotation remains elusive. According to UniProtKB, out of over 74 million proteins in the database just 89 thousand have experimentally determined function, 393 thousand have been labelled with a function by a human expert and only $\sim 12\%$ (9 million) have been annotated in any form, be it by human or by a computer algorithm.

Analogously, PubChem (one of the largest databases of drug-like molecules and their bioactivity assays) contains over 93 million compounds, but only 1.2 million experimental assays in which one or more of the compounds have been tested against one of ~ 10 thousand protein targets or ~ 20 thousand gene targets. Bearing in mind that a single compound can act on multiple targets and a single protein can be a target of many compounds, it is evident that this database is far from being comprehensive.

1.1. Non-structure-based function prediction

While the function is dictated by structure, it is not always necessary to know the full, atomic structure of the compound to be able to infer the function. For example, protein function prediction methods use numerous sources of information in combination. In addition to the amino acid sequence (primary structure), one can use evolutionary information (homologs of known function), sequence information inferred from genome (genomic context), gene co-expression, proteomic assays (including protein-protein interaction), as well as data from genetic assays and clinical observations [Jiang *et al.*, 2016].

1.2. Structure-based function prediction

1.2.1 Related work

There are many possibilities to represent information about molecule structure, and to feed it into a function prediction method.

For quantitative structure-activity relationship (QSAR) modeling, chemical structures are often numerically encoded with hand-made descriptors that describe chemical properties, topology or atomic connectivity, and spatial geometry of the molecule [Sliwoski *et al.*, 2014]. Scalar descriptors include molecular properties, such as molecular weight and the octanol-water partition coefficient (LogP). Topological descriptors encode the connectivity of the molecule, examples of which include substructure-matching schemes and bond distance histograms such as

2D autocorrelation functions [Sliwoski *et al*., 2016]. Geometrical descriptors include radial distribution and 3D autocorrelation functions which calculate histograms of interatomic distances within a molecule, or encoding coefficients of the spherical harmonics which best describe the shape of the molecule [Baumann, 2002, Wang *et al*., 2011]. Topological and geometrical descriptors are often weighted by atomic properties such as partial charge or polarizability to describe the spatial and topological distributions of these properties as well. Most of descriptors either do not require a three-dimensional conformation of the molecule, or else a single low-energy conformation is used for their calculation. Four-dimensional descriptors have also been described which aim to encode some dynamical properties of molecules, such as multiple conformations [Andrade *et al*., 2010], in addition to the properties described above.

For proteins, on the other hand, approaches to representing the structure are two-fold, either coordinate-based or topology-based. The first ones denote the positions of all the amino acids in Cartesian space, either by coordinates of atoms or of pseudoatoms (larger entities representing a group of atoms). Coordinate-based representations are immediately interpretable, as they comprise sufficient information to easily position all the objects in three-dimensional space. The functional relationships within the protein, though, are better captured by representations taking into account the mutual proximity of the objects (amino acids, atoms…). Such approaches have been widely adopted in the field, ranging from directly enumerating distances between bodies (often within a certain cutoff), through enumerating the bodies that are in spatial proximity (in contact, within a certain distance threshold) according to a certain metric, to purely neighborhood-based measures (such as the ones dictated by Voronoi tesselation) [Dupuis *et al*., 2005]. Through these measures it is straightforward to tell which bodies (atoms, amino acids…) interact, but exceedingly difficult to reconstruct the original structure, if the original distances have not been preserved.

The method most similar to ours – using 3D representations directly as inputs to the neural network – is AtomNet [Wallach *et al*., 2015]. Its details and differences to our method are described below in Section 1.2.3.

1.2.2 Motivation for proposed structure-based method

At a microscopic level, the electromagnetic force governs interactions between molecules of any shape or size and in particular it is responsible for the binding affinities of small molecules to proteins or for enzyme function. A classical description of molecular structure usually differentiates two major subsets of electromagnetic interactions, namely electrostatic forces (often described by an electrostatic potential) and van der Waals or steric interactions [Israelachvili,

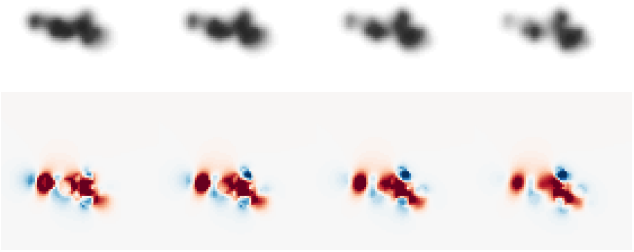


Figure 1: Four of 70 z-slices from the two-channel $70\times70\times70\times2$ representation of an active M_1 muscarinic receptor agonist (from dataset PubChem SAID 1798): approximated electron density field (top; darker means denser) and electrostatic potential field (bottom; positive potential in blue, negative in red, darker means higher magnitude). These physical fields characterize how this molecule can spatially fit to other molecules (binding partners). We thus propose using these fields directly as input to the 3D convolutional network.

2011]. Electrostatic forces are longer-range attractive or repulsive effects which are a result of charge imbalances between atoms. They establish partial positive and negative charges in different regions of the molecular structure. Van der Waals interactions are a shorter-range effect which may be either attractive, due to transient dipoles in the electron clouds, or repulsive, due to an overlap of the electron clouds of molecules. Van der Waals effects are therefore determined by the electron-dense regions around a molecule which effectively determine its shape and play an important role in determining binding interactions. Variations of electrostatic potential and electron densities are often used to computationally describe molecular interactions at a classical level, such as in molecular dynamics calculations [Salomon-Ferrer *et al*. (2013), Brooks *et al*., 1983, Alper and Levy, 1989]. Together these two properties make up a majority of what two interaction partners "see" of each other. In other words, electron density (or its estimate) and electrostatic potential are major determining factors of the molecular function. This is why we propose using these fields directly as inputs to the function prediction method.

The common theme in existing methods that predict function from structure is that they extract handcrafted features from structural information. Such transformations discard part of the information contained in the original data, very likely to the detriment of subsequent analysis. Lessons learned from the success of deep learning in numerous areas of application [Krizhevsky *et al*., 2012, Golkov *et al*., 2016, Wang *et al*., 2017] consistently indicate that deep learning can deal with the entire known raw information and learn the data transformation that is optimal for the task at

Figure 2: Eight z-slices of approximate protein electron density field in a 21-channel representation. The backbone and each residue type have individual channels (shown using hues from RasMol's `shapely` color scheme for residues and black for backbone) for the electron density (darker means denser) of respective atoms. This novel protein-specific multi-channel representation helps the neural network to distinguish the amino acid types directly.

hand. The multi-layer (deep) data transformations applied to the raw data are optimized jointly in view of the final goal (such as classification), formulated as the cost function of the neural network. The output error is back-propagated through all neural network layers, i.e. optimizes the transformations at all layers. In most cases, such automatically optimized transformations strongly outperform handcrafted ones. We therefore explore the deep learning approach to molecular function prediction.

1.2.3 Differences between AtomNet and proposed method

A similar method is AtomNet [Wallach *et al.*, 2015]. It uses various 3D grid representations of small molecules for bioactivity prediction. The main differences between AtomNet and our approach are the following:

- AtomNet requires a 3D co-complex of the molecule in question with its binding target, whereas our method uses the shape of the molecule alone.
- The version of AtomNet that uses the enumeration of atom types in a 3D grid only implicitly provides the information about the possibilities for physical interactions with other molecules, whereas our representation of the molecule via its approximate electron density and electrostatic potential 3D grids is a rawer, more direct representation of how other molecules "perceive" the molecule in question, and in what ways they can physically interact. Besides, the enumeration of atom types in a discrete grid without anti-aliasing introduces imprecisions, partially discarding information about the exact relative positions of the atoms, whereas the usage of anti-aliasing for atom enumeration or the voxel-wise estimation of electron density and electrostatic potential are unaffected by discretization-based imprecisions. Our preliminary experiments indicated that our precise representation yields better results than the discretized one.
- The version of AtomNet that uses handcrafted chemical descriptors such as SPLIF [Da and Kireev, 2014], SIFt [Deng *et al.*, 2004], or APIF [Pérez-Nueno *et al.*, 2009] brings along the aforementioned disadvantages of handcrafted features, whereas we provide the entire physical information required to extract relevant chemical and physical properties.
- Besides small molecules, we also present *protein* function prediction using 3D deep learning, demonstrating the robustness of our approach.
- The network architecture of our approach differs from the one of AtomNet in the following ways: (a) it contains max-pooling layers, thus encouraging the learning of invariances (cf. below); (b) our network has more convolutional layers. The overall architecture is based on the work of [Simonyan and Zisserman, 2015].

2. Methods

Input representation Electron density of a molecule cannot be easily computed based on the coordinates alone, but can be approximated based on the positions of atoms and their van der Waals radii. Approximate electron density and electrostatic potential are calculated on a Cartesian grid. For small-molecule experiments, the field of view is $35\,\text{Å} \times 35\,\text{Å} \times 35\,\text{Å}$ at a resolution of $0.5\,\text{Å}$, i.e. $71 \times 71 \times 71$ voxels. An example is shown in Fig. 1. For protein experiments, the field of view is $127\,\text{Å} \times 127\,\text{Å} \times 127\,\text{Å}$ at a resolution of $2\,\text{Å}$, i.e. $64 \times 64 \times 64$ voxels, for low-resolution experiments, and a resolution of $1\,\text{Å}$, i.e. $128 \times 128 \times 128$ voxels, for high-resolution experiments.

The approximate electron density used herein is emphatically *not* the true, noisy one measured by X-ray crystallography or electron microscopy. The experimental data can be used directly, but this remains to be the subject of future research. Instead, we use an estimate of the idealized electron density obtained from the atom coordinates. The electron density is estimated using a Gaussian kernel around the atom center with the van der Waals radius as its bandwidth [Bernstein & Craig, 2010].

The electrostatic potential of small molecules is estimated by computing the partial charges of the atoms using the Gasteiger–Marsili PEOE (partial equalization of orbital electronegativities) algorithm [Gasteiger and Marsili, 1978].

Experiments with proteins on the other hand were performed using approximated electron density only, without

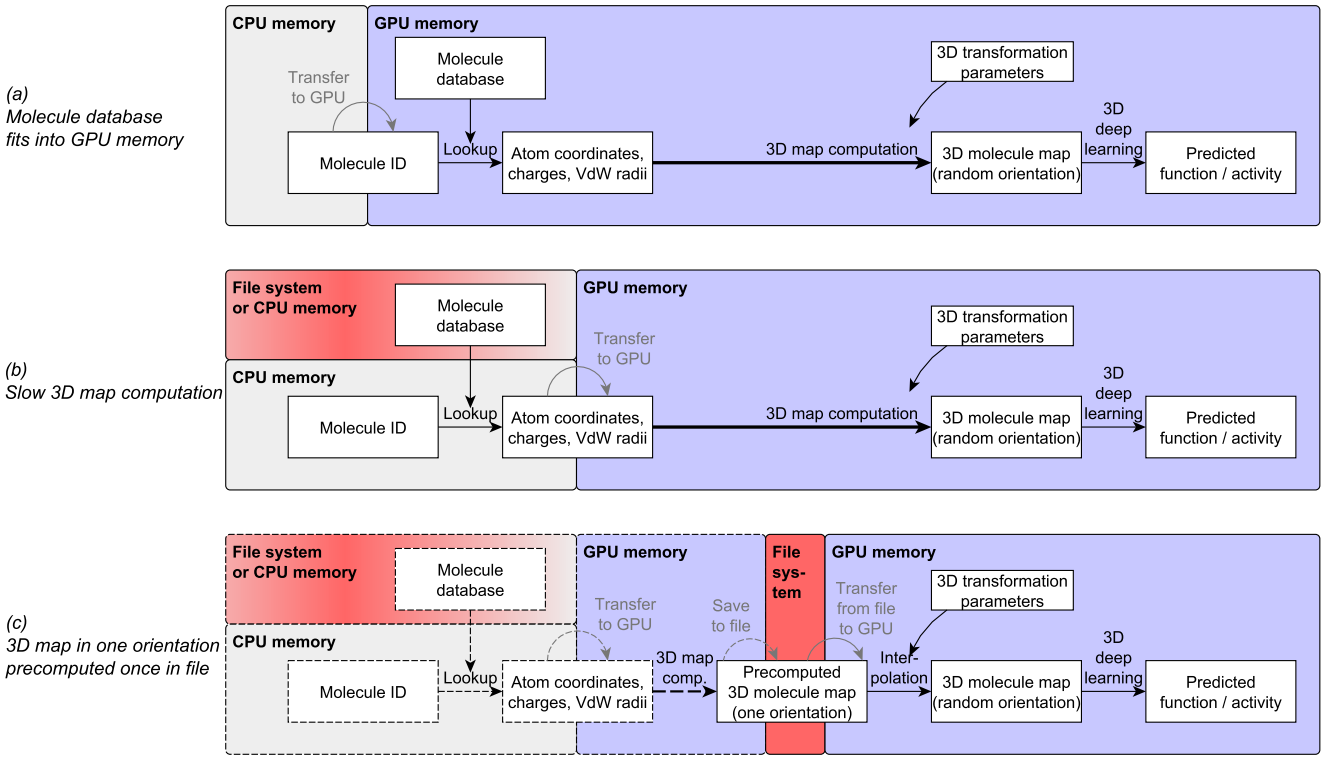


Figure 3: We propose three versions of the pipeline, addressing different needs. The all-GPU pipeline (a) is appropriate if the molecule database fits into GPU memory, for example in the case of small molecules. The large-database pipeline (b) can be applied to the study of proteins; its bottleneck is the slow computation of 3D maps. To circumvent this bottleneck, the *fast* large-database pipeline (c) precomputes the 3D maps only once (dashed lines).

electrostatic potential, for several reasons. Firstly, the computation of partial charges for proteins requires employing one of several non-trivial algorithms, all of which are fraught with substantial degree of error. Secondly, individual amino acid types have a paramount effect on the partial charges, and amino acid types can be inferred from the approximated electron density alone. Thus, due to the limited vocabulary of structural motifs in proteins, "electrostatic motifs" can be learned from "electron density motifs".

To additionally simplify the recognition of structural motifs and to make use of the limited vocabulary of amino acid residues, we propose an alternative **multi-channel input representation**. Instead of computing the approximated electron density of the entire protein in a Cartesian $64 \times 64 \times 64$ voxel grid (or $128 \times 128 \times 128$ for high-resolution experiments), we separate the atoms by the 20 amino acid residue types they belong to and calculate the approximated electron density (Gaussian kernel) 3D maps for each residue type separately, yielding 20 channels, i.e. a $64 \times 64 \times 64 \times 20$ array. An additional channel is used for the electron density of backbone atoms, the 21 channels thus summing up to the overall approximated electron density. This representation can be considered a generalization of one-hot encoding of categorical variables. These 21 channels are shown in Fig. 2. Furthermore, one additional channel is used for hydrogen atoms due to their role in hydrogen bonds, influencing molecular function, and a complementary channel for heavy-atom electron density, the two latter channels also summing up to the overall approximated electron density. Finally, another channel holds the approximated electron density for all atoms, providing three-fold redundancy, but also a disentangled information representation amenable to learning relevant deep feature extractors. The overall input size is thus $64 \times 64 \times 64 \times 24$. The three spatial dimensions are used for 3D convolutional layers, and the fourth dimension represents the channels, i.e. the voxel-wise features.

We encourage rotation-invariance by training, i.e. by using *data augmentation*, in this case **random rotations and translations** of the molecule when creating the Cartesian grid of the physical fields. This trains the neural network to produce similar output for a certain molecule regardless of its position and orientation in space. Furthermore, data augmentation prevents overfitting and facilitates generalization, since unimportant ("overfittable") features such as a specific orientation of the molecule (and associated local

"voxel value motifs") are never repeated during training, whereas structural invariants relevant for predicting molecular function are maintained.

Input computation pipeline Where possible, we perform computations on the GPU – not only the deep learning training, but also the computation of the inputs (3D maps). In our implementation with Lasagne and Theano software libraries, data *generation* and data *augmentation* are seamlessly integrated into the processing as "layers" of the neural network.

In cases of large molecules and/or large numbers of molecules, the database may not fit into GPU memory. We therefore propose different pipelines with the molecule database in the GPU memory (Fig. 3a) or CPU memory (Fig. 3b–c). Moreover, the computation of 3D maps is a bottleneck in case of large molecules. Thus, we also propose a pipeline where the 3D maps are pre-computed in one orientation, stored in a file, and rotated for purposes of data augmentation during training (Fig. 3c), resulting in slight interpolation artifacts, but retaining the important physical information. We use the all-GPU pipeline (Fig. 3a) for QSAR and the fast large-molecule pipeline (Fig. 3c) for protein function prediction.

Neural network architecture Protein function is dictated by the shape of the active site (as represented in the physical 3D maps), as well as the folds (evolutionary families) and relative positions of the domains of the protein. The overall structure of active sites and domains can be inferred from local structural motifs and their higher-level global composition. The method should therefore have the following properties:

1. Translation-covariance of low-level feature extraction
2. Locality of low-level feature extraction
3. Hierarchical feature extraction from localized and simple to larger and more abstract
4. Rotation-invariance

The first three points are ensured by employing convolutional neural networks. The fourth point is taken care of by random rotations during training.

The neural network architecture is closely based on a design practice popularized by [Simonyan and Zisserman, 2015], proposing the usage of very small convolutional filters, achieving certain receptive window sizes through increased depth of the network (allowing more elaborate data transformations) rather than large filters. Pooling layers increase the receptive window size; they encourage (but do not enforce) the learning of invariance to slight distance changes, slight translations and slight rotations; and they contribute to a higher level of feature abstraction, model regularity (generalizability) and computational tractability. We use the VoxNet implementation [Maturana and Scherer,

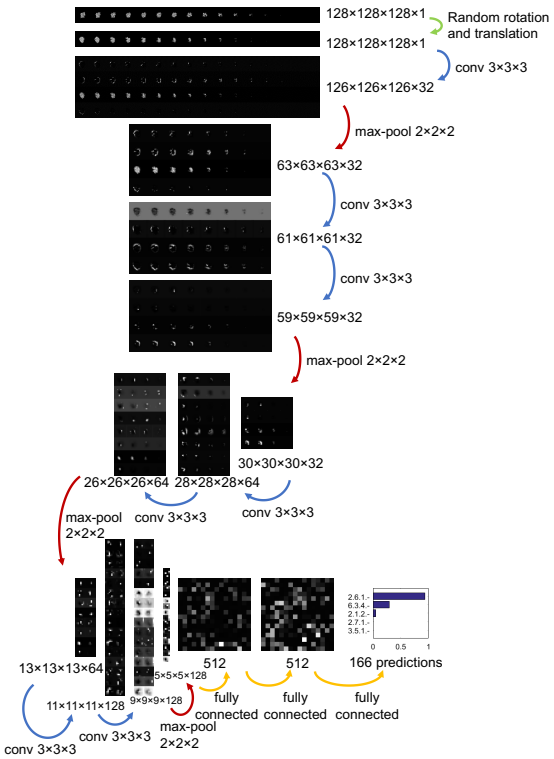


Figure 4: Schematic of the neural network architecture used in most experiments, with visualized activations (feature maps) after training. Only a fraction of slices and channels is shown (with consistency across layers). The outputs do not have to sum up to 1 because one sample can belong to several classes (and therefore sigmoid rather than softmax was used as the nonlinearity in the output layer).

2015a, Maturana and Scherer, 2015b] of 3D convolutional filters of size $3 \times 3 \times 3$, interspersed with 3D pooling layers of size $2 \times 2 \times 2$, analogously to 2D operations in the work of [Simonyan and Zisserman, 2015]. The network architecture and activation maps are shown in Fig. 4. In hidden layers, we use the leaky rectified linear unit [Maas *et al.*, 2013] defined as $\mathrm{LReLU}(z) = \max\{0.01z, z\}$. The network has an output unit for each predicted class. The output nonlinearity is the sigmoid $\sigma(z) = 1/(1+\exp(-z))$ (rather than the softmax function common in classification), so that molecules with several functions can be modeled in a straightforward manner by setting *several* output targets to 1 for the same sample (where 0 or 1 represents class membership). The corresponding loss (objective) function for C classes is the binary cross-entropy between predictions $\mathbf{y}$ and respective targets $\mathbf{t}$, summed for all classes: $\sum_{c=1}^{C} -t_c \log y_c - (1-t_c)\log(1-y_c)$. Training is performed using Adam [Kingma and Ba, 2015], learning rate 10^{-4}, mini-batch size 2, 4 or 8 for different resolutions and GPUs, and early stopping. Results were similar (not shown)

with dropout, batch normalization, skip-connections.

Data For protein experiments, we use the Enzyme Structures Database,[1] which lists PDB structures for enzymes classified by the Enzyme Commission (EC) number hierarchy, designating enzyme specificity and mode of action.

In the first experiment, we perform classification on two classes of enzymes acting on a peptide bond (peptidases): serine proteases (EC 3.4.21.-) and cysteine proteases (EC 3.4.24.-). Proteins in both classes perform the same task of cleaving a peptide bond, but differ in terms of catalytic mechanism. Both classes contain proteins that do not necessarily share evolutionary history (i.e. are not necessarily homologous) and therefore do not always share the same structure (fold). The characteristic that proteins within either of the classes share is the same functional mechanism, which may have emerged in the course of convergent evolution.

In another larger-scale experiment, we perform classification of all 166 third-level EC classes against each other, i.e. EC 1.1.1.- vs. EC 1.1.2.- vs. ... vs. EC 6.6.1.-. We experiment with two different methods of splitting samples into training and test sets. By *random split*, we refer to randomly assigning samples (PDB entries) to the training or test set. A more challenging task is function prediction with a *strict split* at the fourth EC level, meaning that if e.g. a sample in EC 3.4.21.1 (chymotrypsin) gets randomly assigned to the test set, then all other entries in this fourth-level class also get assigned to the test set and none of them to the training set. Thus, it is tested whether the neural network can correctly predict chymotrypsins (EC 3.4.21.1) to belong to the class of serine proteases (EC 3.4.21.-), based solely on information inferred from *other* subclasses of EC 3.4.21.-, such as subtilisin, thrombin or trypsin, but no samples from chymotrypsin class.

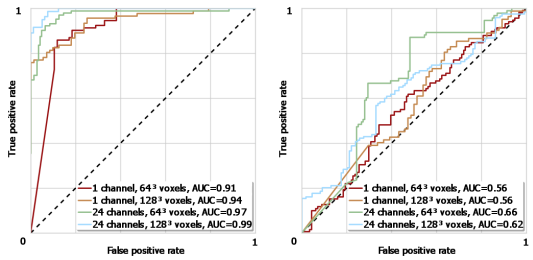
In each training/test split, we select 25% of the samples for the test set (consistently across experiments of the same splitting method). Additional 25% are picked randomly as a validation set to perform early stopping during training in order to prevent overfitting. The remaining 50% of the data are used for training.

For the small-molecule QSAR task, we use a dataset of M_1 muscarinic receptor agonists and inactive componds (PubChem SAID 1798) that were annotated in a respective assay [Butkiewicz *et al.*, 2013]. We attempt to classify the molecules into the active/inactive categories, reserving 20% of the data for testing, and training on the other 80%.

3. Results

The receiver operating characteristic (ROC) for discriminating between serine and cysteine proteases, i.e. classifying EC 3.4.21.- against EC 3.4.24.- with the random training/test split is shown in Fig. 5a. For low-resolution experiments, the area under the ROC curve (AUC) of 0.91 with single-channel inputs and 0.97 with multi-channel inputs indicates a high quality of prediction and suggests that *high accuracy protein function prediction from 3D maps is feasible*. Moreover, doubling the resolution in each dimension, i.e. using $128 \times 128 \times 128$ voxels in lieu of $64 \times 64 \times 64$ voxels, further increases the AUC to 0.94 for single-channel and 0.99 for multi-channel experiments. Thus, both of the high-resolution settings outperform the corresponding low-resolution settings; and both of the multi-channel settings outperform the respective single-channel setting. This indicates that the high-resolution data representation and the multi-channel data representation (and both together) contain valuable information for this task. However, the random split entails that both test and training set most probably contain proteins from the same evolutionary family, thus making the learning task substantially easier.

ROC for the same classes with a strict training/test split is shown in Fig. 5b. The AUC=0.56 for single-channel inputs is much lower than for the random split, confirming that the strict split – inferring protein function only based on *other* protein families sharing same function – is a considerably harder problem. However, the AUC=0.66 for multi-channel inputs also shows that the *multi-channel representation of protein structure is beneficial* for facilitating the extraction of information about protein function. With the strict split, increasing the resolution of the data does not strongly influence the AUC, and in case of the multi-channel inputs leads to decrease in predictive power. It is however important to note that increasing the resolution notably improves the expected number of true positives before encountering a false positive (left part of ROC curve; for



(a) Random split (b) Strict split

Figure 5: ROC for protein function prediction: EC 3.4.21.- (vs. EC 3.4.24.-) with random split (a) and strict split (b). The multi-channel input representation outperforms the respective single-channel settings. The proposed 3D input representation provides meaningful information about the molecules under the random split and even under the challenging strict split.

[1] https://www.ebi.ac.uk/thornton-srv/databases/enzymes/

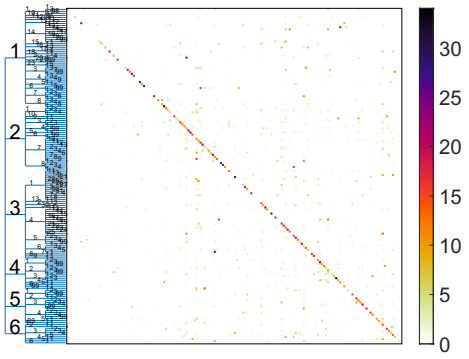


Figure 6: Confusion matrix for classification of the 166 third-level EC classes. The pronounced diagonal means correct prediction for the majority of test samples.

both single- and multi-channel methods with random split, as well as for single-channel for strict split). This signifies that the network given improved resolution learns to recognize crucial structural features of individual classes. The subsequent, mid-range drop in accuracy can, in our opinion, be attributed to insufficient amount of data in the training set, which prevents the network from generalizing properly. This is also the reason for sub-par performance of the method in "high resolution, multi-channel" regime.

It is however evident from Figures 5a and 5b that introduction of additional channels denoting the amino acid types substantially improves the expected prediction accuracy. In our opinion, this allows the network to easier attain the knowledge on structural and functional properties of amino acids, which it would have to learn in training. By introducing additional channels, we alleviate this need. It is evident that analogous effect could have been achieved by pretraining the networks with a larger data corpus, which we will demonstrate in subsequent work. Therefore, we postulate that in the limit of infinite (or at least sufficiently abundant) data, performance of "single-channel" and "multi-channel" approaches would converge.

The confusion matrix for the classification of the 166 third-level EC classes with random train/test split is shown in Fig. 6. The diagonal (correct predictions) is highly pronounced, indicating that our deep learning approach stably distinguishes the correct class from the 165 other possible classes of proteins in many cases, which is a challenging task requiring the representation of numerous function-specific structural features within the neural network. Each class had a slightly different AUC in the test data; the AUC of the individual classes averaged across all classes was 0.87 ± 0.13 on the test data.

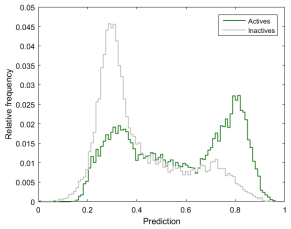
While protein function prediction results are proof of concept rather than competitive, small-molecule QSAR results are at par with state-of-the-art methods in terms of AUC. The ROC curve for biological activity classification of M_1 muscarinic receptor antagonists is shown in Figure 7b. Models trained on this dataset achieved an AUC=0.70. This value indicates that the models were capable of differentiating molecules which exhibited biological activity from those that did not at a rate substantially higher than random chance. The AUC value found here is *approximately equal to that of state-of-the-art methods based on hand-crafted descriptors* applied to the same dataset [Mendenhall and Meiler, 2016]. A further demonstration of the performance of the model can be seen in Figure 7a, which highlights a substantial gap between mode scores for the two classes of compounds.

These results are both interesting and encouraging given that state-of-the-art models utilize descriptors and network architectures that are specifically optimized for biological activity prediction. The performance of the models reported here indicates that the chosen approximations for describing molecular structure are already capable of encoding information that is equally valuable to what is found in the hand-crafted descriptors. Further refinement of both molecular structure description and network parameters are likely to boost the performance above what is seen here. Additionally, these models illustrate that deep learning can be a powerful tool even in domains where the data sets are much smaller and more heavily unbalanced than those seen in more traditional applications of deep learning.

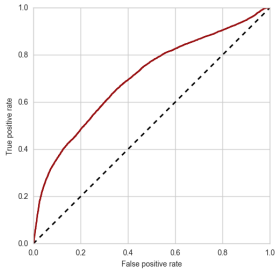
As noted in [Mendenhall and Meiler, 2016], the goal of biological activity prediction is often to prioritize a large set of compounds in order to select a small subset for physical testing, thereby making those compounds with the highest scores the most important from a practical standpoint. These models were trained with a loss function designed to optimize the overall AUC of the ROC curves (i.e. from false positive rate (FPR) values of 0 to 1) which effectively aims to best separate the two classes from each other as a whole. However, this metric does not consider the performance of the highest scoring samples, and as seen in Figure 7a, the highest-scoring active compound is separated from the highest-scoring inactive compound by a small fraction of the score range. Given these promising early results, approaching this problem with a loss function designed to optimize compound scores at low FPR values (e.g. the logAUC metric described in [Mendenhall and Meiler, 2016]) would provide a straightforward way to improve the performance of these models in a manner beneficial for practical QSAR application.

4. Discussion and Conclusions

In this work we have demonstrated the utility of 3D convolutional neural networks for discriminating between molecules in terms of their function in living organisms. They allow for accurate classification of proteins, without relying on domain (expert) knowledge or evolutionary



(a) Normalized histogram of predictions (neural network outputs) for active and inactive small molecules



(b) ROC (AUC=0.70)

Figure 7: Results for small-molecule QSAR. Predictions (a) demonstrate some separability of active and inactive small molecules in the test set. The ROC (b) has an AUC=0.70 approximately equal to that of state-of-the-art descriptor-based methods applied to this dataset (PubChem SAID 1798).

data. Additionally, the use of the 3D convolutional neural networks for classifying small molecules permits achieving comparable quality as state-of-the-art QSAR methods, while obviating the need for handcrafted descriptors.

The data on protein classification presented above were based purely on enzymes, but it is highly plausible that these results generalize to the entire protein space. The success in discriminating between a wide variety of enzyme classes strongly indicates that this approach allows for high-confidence discrimination between different functional classes (transporters, structural proteins, immunoglobulins...). Discriminating within a class (e.g. dopamine receptor from a glucagon receptor) should be possible as well, as demonstrated by our serine vs. cysteine protease experiment.

The classification experiments presented above are to be treated as proof-of-concept. While we demonstrated the applicability of convolutional neural networks for these purposes, we do acknowledge that the predictive power of these methods can only increase when supplemented with additional features, used by the other methods in the respective fields. For purposes of protein function prediction, one could easily improve expected prediction accuracy by extending the feature set by such ones as presence of common sequence motifs, structural classification (annotated or predicted) and homology-derived information. The small-molecule classification can be augmented by experimentally determined features (e.g. logP, polarizability...) and ones derived from the structure (e.g. constitutional descriptors, fragment counts...). While the latter can be learned, it may prove beneficial to provide them explicitly.

Notably, this work does not consider the fact that the vast majority of molecules in living organisms is conformationally flexible. It is possible to generate multiple conformers of small molecules, for the QSAR use case, but it is not intuitively obvious what effect will it have on the training of the method. By using generated conformations as actives one would inadvertently introduce false positives in the training set (i.e. conformations in which the ligand does not bind would be labeled as positives), but on the other hand it would allow the network to recognize also potentially active ligands, even if they were in an unsuitable conformation. For protein function prediction, conformational flexibility plays a less major role, but distinction between apo (without bound ligand) and holo (with ligand bound) structures in the training process could potentially play a role for the expected predictive power.

While the protein function prediction part of this work relied on experimental structural data, these methods can also be applied to *predicted* protein structures. It could be advisable to limit the prediction to the active site only, thus allowing for much faster training and predictions, and enabling meta-prediction using multiple variant active sites.

The other potential use case is to use experimental electron density directly, without the need for fitting atoms within. Recent developments in the area of direct imaging, especially electron microscopy, make such methods particularly relevant. As in its current form our method does not depend on any sequence-related data, it is immediately applicable to such problems. This could enable high-confidence function annotation of proteins recovered from environmental samples.

These are just a few of potential application domains for the methods we propose. By avoiding human-derived, handcrafted descriptors they allow to capture the features of the studied molecules that are truly important for functional considerations. On contrary to these descriptors, they will only increase in accuracy with the growing amount of data. In contrast to methods based on structural comparison, the methods we proposed do not require superposition. We postulate therefore that deep learning methods of inferring functional information from raw spatial 3D data will increase in importance, with the growing amounts of spatial biological information and increased resolutions of direct imaging methods.

Another promising direction is to combine our physically expressive representations with rotation-invariant deep learning [Della Libera *et al.*, 2019]. Methods with hardwired invariance tend to perform better and require less training data and no rotational data augmentation.

Acknowledgments

This work was supported by the Munich Center for Machine Learning [BMBF grant 01IS18036B], the European Research Council [Consolidator Grant "3DReloaded"], and by the Deutsche Telekom Foundation.

References

[Alper and Levy, 1989] Alper, H.E. and Levy, R.M. (1989) Computer simulations of the dielectric properties of water: Studies of the simple point charge and transferrable intermolecular potential models. *J. Chem. Phys.*, **91(2)**, 1242-1251.

[Andrade *et al*., 2010] Andrade, C.H., Pasqualoto, K.F., Ferreira, E.I. and Hopfinger, A.J. (2010) 4D-QSAR: perspectives in drug design. *Molecules*, **15(5)**, 3281-3294.

[Baumann, 2002] Baumann, K. (2002) Distance Profiles (DiP): A translationally and rotationally invariant 3D structure descriptor capturing steric properties of molecules. *Mol. Informatics*, **21(5)**, 507-519.

[Bernstein & Craig, 2010] Bernstein, H.J. and Craig, P.A. (2010) Efficient molecular surface rendering by linear-time pseudo-Gaussian approximation to Lee–Richards surfaces (PGALRS). *J. Appl. Crystallography*, **43(2)**, 356-361.

[Brooks *et al*., 1983] Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S. and Karplus, M. (1983) CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, **4(2)**, 187-217.

[Butkiewicz *et al*., 2013] Butkiewicz, M., Lowe, E.W., Mueller, R., Mendenhall, J.L., Teixeira, P.L., Weaver, C.D., and Meiler, J. (2013) Benchmarking ligand-based virtual high-throughput screening with the PubChem database. *Molecules*, **18(1)**, 735756.

[Da and Kireev, 2014] Da, C. and Kireev, D. (2014) Structural protein–ligand interaction fingerprints (SPLIF) for structure-based virtual screening: Method and benchmark study. *J. Chem. Inf. Model.*, **54**(9), 2555-2561.

[Della Libera *et al*., 2019] Della Libera, L., Golkov, V., Zhu, Y., Mielke, A., and Cremers, D. (2019) Deep Learning for 2D and 3D Rotatable Data: An Overview of Methods. *arXiv.org*, arXiv:1910.14594.

[Deng *et al*., 2004] Deng, Z., Chuaqui, C., and Singh, J. (2014) Structural interaction fingerprint (SIFt): a novel method for analyzing three-dimensional protein-ligand binding interactions. *J. Med. Chem.*, **47**(2), 337-344.

[Dupuis *et al*., 2005] Dupuis, F., Sadoc, J.F., Jullien, R., Angelov, B., and Mornon, J.P. (2005) Voro3D: 3D Voronoi tessellations applied to protein structures. *Bioinformatics*, **21(8)**, 1715-1716.

[Gasteiger and Marsili, 1978] Gasteiger, J. and Marsili, M. (1978) A new model for calculating atomic charges in molecules. *Tetrahedron Lett.*, **34**, 3181-3184.

[Golkov *et al*., 2016] Golkov, V., Skwark, M.J., Golkov, A., Dosovitskiy, A., Brox, T., Meiler, J., and Cremers, D. (2016) Protein contact prediction from amino acid co-evolution using convolutional networks for graph-valued images. *Adv. Neural Inf. Process. Syst. (NIPS)*, **29**, 6488.

[Israelachvili, 2011] Israelachvili, J.N. (2011) Intermolecular and surface forces. Academic Press.

[Jiang *et al*., 2016] Jiang, Y. *et al*. (2016) An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.*, **17**, 184.

[Kingma and Ba, 2015] Kingma, D.P. and Ba, J.L. (2015) Adam: a method for stochastic optimization. *Int. Conf. on Learning Representations (ICLR)*.

[Krizhevsky *et al*., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012) ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst. (NIPS)*, **25**, 4824.

[Maas *et al*., 2013] Maas, A.L., Hannun, A.Y., and Ng, A.Y. (2013) Rectifier nonlinearities improve neural network acoustic models. *Int. Conf. on Machine Learning (ICML)*, **30**.

[Maturana and Scherer, 2015a] Maturana, D. and Scherer, S. (2015a) 3D convolutional neural networks for landing zone detection from LiDAR. *IEEE Int. Conf. Robot. Autom. (ICRA)*, 3471-3478.

[Maturana and Scherer, 2015b] Maturana, D. and Scherer, S. (2015b) VoxNet: A 3D convolutional neural network for real-time object recognition. *IEEE Int. Conf. Robots and Systems (IROS)*, 922-928.

[Mendenhall and Meiler, 2016] Mendenhall, J. and Meiler, J. (2016) Improving quantitative structureactivity relationship models using Artificial Neural Networks trained with dropout. *J. Comput. Aided Mol. Design*, **30**, 177-189.

[Pérez-Nueno *et al*., 2009] Pérez-Nueno, V.I., Rabal, O., Borrell, J.I., and Teixidó, J. (2009) APIF: A new interaction fingerprint based on atom pairs and its application to virtual screening. *J. Chem. Inf. Model.*, **49**(5), 1245-1260.

[Salomon-Ferrer *et al*. (2013)] Salomon-Ferrer, R., Case, D.A., and Walker, R.C. (2013) An overview of the Amber biomolecular simulation package. *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, **3(2)**, 198-210.

[Simonyan and Zisserman, 2015] Simonyan, K. and Zisserman, A. (2015) Very deep convolutional networks for large-scale image recognition. *Int. Conf. on Learning Representations (ICLR)*.

[Sliwoski *et al*., 2014] Sliwoski, G., Kothiwale, S., Meiler J., and Lowe, E.W. Jr. (2014) Computational methods in drug discovery. *Pharmacol. Rev.* **66.1**, 334-395.

[Sliwoski *et al*., 2016] Sliwoski, G., Mendenhall, J., and Meiler, J. (2016) Autocorrelation descriptor improvements for QSAR: 2DA_Sign and 3DA_Sign. *J. Comput. Aided Mol. Des.*, 30(3), 209-217.

[Wallach *et al*., 2015] Wallach, I., Dzamba, M., and Heifets, A. (2015) AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv.org*, arXiv:1510.02855.

[Wang *et al*., 2011] Wang, Q., Birod, K., Angioni, C., Grösch, S., Geppert, T., Schneider, P., Rupp, M., and Schneider, G. (2011) Spherical harmonics coefficients for ligand-based virtual screening of cyclooxygenase inhibitors. *PLoS One*, **6(7)**, e21554.

[Wang *et al*., 2017] Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. (2017) Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Comp Bio*, **13(1)**, e1005324.

3.3 q-Space Deep Learning: Twelve-Fold Shorter and Model-Free Diffusion MRI Scans

Usual data processing pipelines for diffusion MRI consist of several steps each of which is handcrafted and does not optimally preserve valuable information. Moreover, the classical pipelines process each data sample separately without using valuable information about the typical statistics of the data contained in databases.

These two properties of the usual algorithms strongly indicated that the usage of end-to-end deep learning, i.e. the joint optimization of all processing steps and the usage of available training data, might be beneficial for diffusion MRI.

And indeed, by proposing to apply deep learning to diffusion MRI data, we achieved a twelve-fold reduction in scan time (for example from 24 minutes to 2 minutes), which allows to strongly reduce costs and make the unique sensitivity of diffusion-MRI-based diagnostic methods available for patients who cannot hold perfectly still for a long time in the loud and narrow MRI scanner. Another achievement of our proposed methods is the possibility to become independent of handcrafted models and representations that suboptimally discard parts of the valuable information contained in the measurements. But even if model-based processing is desired for some reason, our methods provide a twelve-fold reduction of scan time.

More specifically, some of the main issues of previous methods are the following. Previous methods are based on handcrafted models that discard some of the information present in the measurements. Moreover, fitting those models to the noisy measurements is an ill-posed/ill-conditioned problem. Possible solutions could be disambiguated by using prior knowledge about statistics of real data, but previous methods do not attempt to do that. Instead, they rely on large numbers of measurements, thus requiring very long scans.

We show that deep learning can be successfully applied to diffusion MRI and achieves a multitude of goals. Our methods can inpaint missing measurements in diffusion-encoding space (q-space), estimate model parameters, and most importantly directly identify tissue types from raw q-space measurements without using any handcrafted suboptimal models nor representations.

We use q-space measurements as inputs to neural networks. The output targets are chosen depending on the goals in each respective setting.

The author of this dissertation contributed substantially to the content of the paper, in particular concerning the idea, parts of the code, neural network training, and writing parts of the paper.

Accepted manuscript for IEEE Transactions on Medical Imaging. Published version: doi: 10.1109/TMI.2016.2551324.

q-Space Deep Learning: Twelve-Fold Shorter and Model-Free Diffusion MRI Scans

Vladimir Golkov*, Alexey Dosovitskiy, Jonathan I. Sperl, Marion I. Menzel, Michael Czisch, Philipp Sämann, Thomas Brox, and Daniel Cremers

***Abstract*—Numerous scientific fields rely on elaborate but partly suboptimal data processing pipelines. An example is diffusion magnetic resonance imaging (diffusion MRI), a non-invasive microstructure assessment method with a prominent application in neuroimaging. Advanced diffusion models providing accurate microstructural characterization so far have required long acquisition times and thus have been inapplicable for children and adults who are uncooperative, uncomfortable, or unwell. We show that the long scan time requirements are mainly due to disadvantages of classical data processing. We demonstrate how deep learning, a group of algorithms based on recent advances in the field of artificial neural networks, can be applied to reduce diffusion MRI data processing to a single optimized step. This modification allows obtaining scalar measures from advanced models at twelve-fold reduced scan time and detecting abnormalities without using diffusion models. We set a new state of the art by estimating diffusion kurtosis measures from only 12 data points and neurite orientation dispersion and density measures from only 8 data points. This allows unprecedentedly fast and robust protocols facilitating clinical routine and demonstrates how classical data processing can be streamlined by means of deep learning.**

***Index Terms*—Diffusion magnetic resonance imaging (diffusion MRI), artificial neural networks, diffusion kurtosis imaging (DKI), neurite orientation dispersion and density imaging (NODDI).**

I. Introduction

Over the past three decades, diffusion magnetic resonance imaging (diffusion MRI) [1]–[4] has taken on an important role in assessing microstructural tissue and material properties non-invasively based on the diffusion of gases and liquids, primarily water. In radiology, diffusion MRI is a powerful technique, mainly due to its sensitivity to diffusion restriction (e.g. caused by brain ischemia), yet also any other microstructural tissue rebuilding as found in neoplasms or inflammatory lesions. Its potential as a basis for diagnostic and treatment monitoring markers has been established over the last years [5]–[8]. Advanced diffusion MRI models such as diffusion kurtosis imaging [2], [3] (DKI) and neurite orientation dispersion and density imaging [4] (NODDI) provide more accurate characterization of tissue microstructure [2], [4], [9]–[11] but require long acquisition time. This has so far led to high scan costs and has made advanced diffusion models inapplicable for patients who are uncooperative, uncomfortable or unwell.

A. Model Fitting, Analytical Solutions, Approximation

In diffusion MRI, a number of diffusion-weighted images (DWIs) for different diffusion weightings[1] and directions (constituting the so-called three-dimensional q-space) are acquired [1]. Signal intensity in these images contains information regarding diffusion properties. The task in quantitative diffusion MRI is to find a mapping from a limited number of noisy signal samples to rotationally invariant scalar measures that quantify microstructural tissue properties. This inverse problem is solved in each image voxel. Currently, this problem is addressed by three approaches.

The classical approach of estimating scalar measures is model fitting. Its data processing pipeline consists of fitting [12] a diffusion model and calculating rotationally invariant measures from the fitted model parameters. Prior to model fitting, the q-space data can be obtained by regular acquisition, or using advanced methods such as compressed sensing or dictionary learning (cf. below).

Another approach can be taken if closed-form analytical solutions exist. For the diffusion model of DKI [2], [3] – which requires approximately 150 DWIs [3], [13], [14] – it has recently been shown [15], [16] that for certain DKI-based measures much fewer DWIs (e.g. 13 or 19 DWIs) are sufficient, and that these measures can be analytically calculated from the data in a single step. This has led us to the assumption that for many other scalar measures and tissue properties the most relevant information might as well be recovered from only a few DWIs.

The third approach of calculating scalar measures is approximation, particularly machine learning. Simulations of simplified tissue models with extensive sets of diffusion weightings [17], [18] indicate that standard model fitting

This work was supported by the Deutsche Telekom Foundation. *Asterisk indicates corresponding author.*

*V. Golkov is with the Department of Computer Science, Technical University of Munich, Garching, Germany (e-mail: golkov@cs.tum.edu).

A. Dosovitskiy and T. Brox are with the Department of Computer Science, University of Freiburg, Freiburg, Germany.

P. Sämann and M. Czisch are with the Max Planck Institute of Psychiatry, Munich, Germany.

J. I. Sperl and M. Menzel are with GE Global Research, Munich, Germany.

D. Cremers is with the Department of Computer Science, Technical University of Munich, Garching, Germany.

[1] For simplicity, we also include images with diffusion weighting zero into the definition of "DWIs".

procedures can be replaced by approximation methods. It was also mentioned [18] that feature selection methods could be applied to identify the most relevant DWIs in order to reduce these extensive sets of diffusion weightings. On the basis of these observations, we apply deep learning [19]–[23] for accurate approximation and present a deep learning framework for different inputs (full and subsampled sets of regular DWIs, non-diffusion contrasts) and different outputs (denoising, missing DWI reconstruction, scalar measure estimation, tissue segmentation). Scalar measure estimation from twelve-fold accelerated acquisition is demonstrated on two advanced models: DKI [2] (using radial kurtosis and fractional kurtosis anisotropy) and NODDI [4] (using orientation dispersion index and intracellular volume fraction). In comparison to most of the well-established models (e.g. diffusion tensor imaging [1]), DKI and NODDI are more elaborate and thus can provide improved sensitivity [2], [4], [9]–[11]; however, they also require considerably longer acquisition times. By shortening the acquisition duration of advanced models by an order of magnitude, we strongly improve their potential for clinical use, and reduce scan costs and motion artifacts caused by long scan durations.

B. Advantages of Deep Learning

Deep learning [19]–[23] is a family of algorithms for efficient learning of complicated dependencies between input data and outputs by propagating a training dataset through several layers of hidden units (artificial neurons). Each layer is a data transformation step. The classical diffusion MRI pipeline involving model fitting also consists of several steps. In the example of DKI, approximately 150 measurements [3], [13], [14] are reduced to 22 model parameters in the classical pipeline, then to a few rotationally invariant measures, and finally (implicitly or explicitly) to one parameter, i.e. the tissue property of interest such as the amount of disease-based microstructural change. (For NODDI, rotationally invariant measures are estimated during model fitting rather than in an additional step, see Fig. 1a.) In every step, information is partly lost by reducing the degrees of freedom. However, the classical pipeline does not provide feedback from the later steps to the earlier steps with regard to what part of the information should be retained or discarded and which transformations should be applied. Thus, the pipeline relies on handcrafting and fixing each step, i.e. the diffusion model and derived scalar measures. Deep learning takes a more flexible approach: the effects of each layer on the final result are propagated back to adjust preceding layers, such that all layers are optimized jointly in terms of the final objective, namely minimizing the output error. This prevents the loss of information during intermediate steps. Advantages of deep learning over handcrafted features have been shown in numerous other applications [23].

The main novelties introduced herein are:

- Using subsampled DWIs as machine learning input *directly*,
- Unprecedented scan time reduction for DKI and NODDI,
- Segmentation without using diffusion models.

Preliminary results presented at a conference [24] are herein extended by additional evaluation, including the influence of neural network parameters, and more[2]. Related applications of machine learning are tractography [25] and non-diffusion MRI [26].

II. Materials and Methods

The relationship between the diffusion-weighted signal and microstructural tissue properties is non-trivial. However, an appropriately chosen, tuned and trained machine learning algorithm can theoretically represent any relationship between inputs and outputs [27] if such a relationship exists. We make use of this fact in order to leverage information contained in very limited numbers of input DWIs. In all experiments presented in this work, training datasets originate from a different human subject than the test datasets. The proposed family of methods is termed "q-space deep learning" (q-DL). In q-DL, we treat each image voxel individually as a data sample.

The task of estimating the vector m of scalar measures from the vector S of signal measurements can be formalized as follows. The analytical solution is as simple as calculating $H(S)$, where H is the closed-form function that maps S to m. Such closed-form solutions are available only for certain measures and certain diffusion weightings [15], [16]. In model fitting, m is estimated as $g(f(S))$, where $\theta = f(S)$ are the estimated diffusion model parameters obtained through model fitting f by solving an optimization problem, e.g. least squares [12], and g calculates rotationally invariant scalar measures from θ. In DKI, the steps of applying f and g are independent and not optimized jointly with respect to the accuracy of estimation of m; in NODDI, f and g are one joint step; in all cases, fitting is susceptible to noise. In contrast, q-DL adjusts the parameters of a multilayer neural network such that the outputs of the network well approximate the target measures m. The measures m are obtained for the training dataset by model fitting, but model fitting is not required for the datasets to which the trained network is subsequently applied.

A. Feed-Forward Neural Networks

A so-called multilayer perceptron is a multilayer artificial neural network that performs a nonlinear data transformation in each layer. Layer 0 is called the input layer, layer L the output layer, intermediate layers are called hidden layers. The transformation in layer $i \in \{1, \dots, L\}$ follows the rule

$$a_j^{(i)} = s_i\left(W^{(i)} a_j^{(i-1)} + b^{(i)}\right), \qquad (1)$$

where $a_j^{(i)}$ is the output vector of layer i for data sample j, the vector $a_j^{(0)}$ is the input of the network, $W^{(i)}$ is called the weight

[2] This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This includes additional methods (denoising and reconstruction of missing DWIs), formal algorithms, results for additional scalar measures, the q-space subsampling schemes, stability to different random initializations, reproducibility with respect to different choices of training and test datasets, quantitative comparison to compressed sensing, stability to head rotation.

matrix, $b^{(i)}$ the bias vector, and s_i are nonlinearities (see below). The length of the vector $a_j^{(i)}$ corresponds to the number of artificial neurons (hidden units) in layer i. During training, all weight matrices and bias terms are jointly adjusted such that the output vectors $a_j^{(L)}$ for each training sample j (in our case: each image voxel j) well approximate the target output vectors y_j. This adjustment is achieved by using the backpropagation algorithm (implemented in the deep learning toolbox [28]) to solve the optimization problem

$$\underset{W,b}{\operatorname{argmin}} \sum_j \left\| a_j^{(L)} - y_j \right\|^2, \qquad (2)$$

where the sum of errors is taken over all training samples j, and the outputs $a_j^{(L)}$ recursively depend on the parameters $W^{(i)}$ and $b^{(i)}$ according to the aforementioned recursive transformation rule for the $a_j^{(i)}$ for $i \in \{1, \dots, L\}$. Once trained, such a neural network works in a deterministic manner.

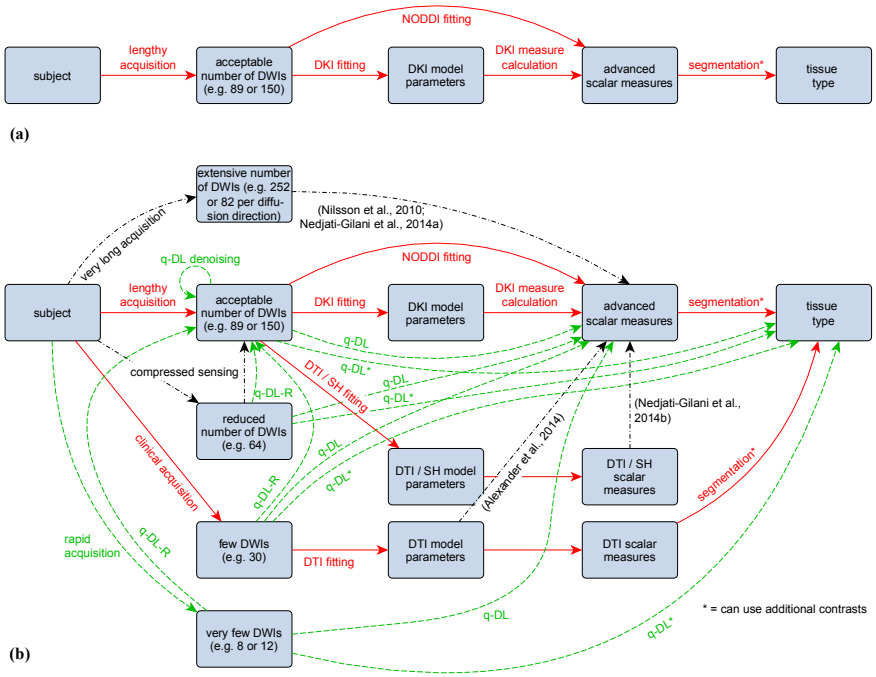


Fig. 1. Possible steps of data processing from scanning a real-world subject (left) to the determination of the tissue properties (right). Standard DKI/NODDI pipeline is shown individually (a) and in comparison to advanced methods (b). Arrows designate possible data processing in the standard pipelines (solid red), state-of-the-art methods based on compressed sensing and machine learning (dash-dotted black) and novel processing possibilities introduced with q-space deep learning (dashed green), see also Ref. [24].

B. q-Space Deep Learning

The proposed pipelines based on q-space deep learning reduce scan duration and perform the data processing as directly as possible without discarding information at intermediate steps. This is reflected in the comparison of q-DL to the standard pipeline and to other state-of-the-art methods in terms of possible steps of data processing (Fig. 1). Previous methods based on machine learning rely either on extensive acquisitions or on intermediate steps involving model fitting based on diffusion tensor imaging (DTI) and spherical harmonics (SH), whereas q-space deep learning provides the fastest acquisitions and the most direct data processing steps.

In all experiments, training data originate from different human subjects than test data. The neural networks thus do not "know" the true output vectors of the test data but rather estimate them based on the input-output-mapping learned from training data. Each voxel j is treated individually as a data sample. The algorithm does not know its location in the image. We introduce several input-output-mapping tasks. Different deep networks are trained for different tasks:

1) Estimation of Scalar Measures

A network is trained to predict microstructure-characterizing scalar measures m_j directly from the (reduced set of) DWIs $S_{j,\alpha}$ where α is a pseudorandom subsampling multi-index (such that the q-space sampling is consistent across training and test data). In other words, inputs are $a_j^{(0)} = S_{j,\alpha}$ with length $|\alpha|$, and targets are $y_j = m_j$. The length of the output vector is the number of considered scalar measures. Training targets $y_j = m_j$ are obtained from a fully sampled training dataset S_j (consisting of $|S_j|=n$ DWIs) by model fitting; however, a neural network is trained to predict m_j from the subsampled data $S_{j,\alpha}$. As a consequence, the neural network is able to estimate m_j from α-subsampled datasets. This allows an estimation of m_j at a scan time reduction factor of $n/|\alpha|$ for all subsequent datasets. In our experiments, we use scan time reduction factors of up to $n/|\alpha|=148/12\approx12.3$ for DKI and up to $n/|\alpha|=99/8\approx12.4$ for NODDI.

2) Model-Free Segmentation

Tissue segmentation is achieved by training a neural network to discriminate between several tissue types. We propose modifying the approach [29] of multi-parametric MRI tissue characterization by artificial neural networks such that the DWIs are directly used as inputs rather than using scalar measures obtained from model fitting. Our approach thus allows using the unique information provided by diffusion MRI directly without the information reduction imposed by models. State-of-the-art automatic segmentation [30], [31] (based on non-diffusion images with spatial priors) into healthy white matter (WM), grey matter (GM), cerebrospinal fluid (CSF) and multiple sclerosis lesions was used as ground truth for our proof-of-concept model-free segmentation (based on diffusion images without spatial priors). The q-DL framework allows incorporating additional contrasts other than DWIs as inputs to the learning algorithm. We used fluid-attenuated inversion recovery (FLAIR) signal as an additional input. The length of the output vector is the number of tissue classes (with each

output representing a relative class membership "likeliness" using softmax, see below).

C. Details of the Neural Networks

The deep learning toolbox [28] was used for deep learning experiments. The artificial neural network used is a multilayer perceptron with three hidden layers, each consisting of 150 hidden units with a nonlinearity known as the rectified linear unit [19], [20], i.e. $s_i(z)=\max(0,z)$. This layout, applied to each image voxel independently, can be considered a convolutional neural network with window size 1×1 in each layer, masking out the loss for non-brain voxels. Linear units $s_L(z)=z$ are used in the output layer L for fitting tasks and softmax outputs $s_L(z)=\exp(z)/\|\exp(z)\|_1$ for classification tasks. Each input and output of the neural network is independently scaled to the interval [0,1] and the same affine transformation parameters are reused for the test datasets. The network is initialized with orthogonal random weights [22]. We use a dropout [21] fraction of 0.1, stochastic gradient descent with momentum 0.9, minibatch size 128, learning rate 0.01 with a warm-up learning rate of 0.001 for the first 10 epochs. The learning rate was decreased by factor 0.9 whenever the training set error stagnated (averaged over 5 epochs) compared to the previous 5 epochs. To prevent overfitting, 10% of the voxels in the training data set were used as a validation set and early stopping was employed when the validation set error (averaged over 10 epochs) increased compared to the average over the previous 10 epochs. These choices of the neural network parameters are based on practical considerations as described in Ref. [32]. We use a multilayer perceptron because it is a straightforward and powerful method. Three hidden layers provide acceptable results and short runtime for our purposes. Other network settings are evaluated in Fig. 6. In all experiments, training data originate from different human subjects than test data (except Fig. 12, panels (e,k,q,v)). For different q-space sampling schemes, the values of the network inputs (signal intensities) have a different meaning (and length), therefore a different network must be trained independently for every q-space scheme.

D. Data

Approval by the local ethics board for the *in vivo* study protocols and prior informed consent were obtained. In the multiple sclerosis data, datasets from five patients were used for training, and the dataset of the respective sixth patient was used for testing (in all combinations). In all other datasets, data from one healthy volunteer was used for training, and data from another healthy volunteer for testing.

1) Five-Shell and Cartesian Healthy Volunteer Data

Data sets of a total of two healthy volunteers were acquired using the common radial q-space scheme with 30 directions sampled on five shells (b=600, 1200, 1800, 2400, 3000s/mm²) and eight b=0 images. Ten repetitions of this scheme were acquired for each volunteer. Besides, Cartesian sampling [33] (515 points, $b_{\max}$=3000s/mm²) was also performed. Echo-planar imaging was performed using a 3T GE MR750 MR scanner (GE Healthcare, Waukesha, WI, USA) equipped with a 32-channel head coil (TE = 80.7ms, TR = 2s, FOV = 24cm × 24cm × 4cm, isotropic voxel size 2.5mm, ASSET factor 2). All data underwent FSL topup distortion correction [34], [35]. All DWIs were registered using an affine transformation [36] to compensate for motion. Advanced treatment of motion is subject of future work. Each volunteer data set contained approximately 40,000 brain voxels (i.e. training/test samples).

2) Three-Shell Healthy Volunteer Data

Data sets of a total of four healthy volunteers were acquired using a scheme optimized [13], [14] for DKI and suitable for NODDI [4]: three shells (b=750, 1070, 3000s/mm²) with 25, 40, 75 directions, respectively, and eight b=0 images. Acquisition parameters and postprocessing were the same as for the five-shell and Cartesian acquisitions.

3) Human Connectome Project Data

To demonstrate feasibility on a different scanner with different acquisition parameters, we used data sets of a total of two healthy volunteers from the Human Connectome Project (HCP) [37]–[44].

4) Multiple Sclerosis Data

For tissue segmentation and lesion detection, six multiple sclerosis patients were scanned using a diffusion spectrum [33] random subsampling pattern with 167 DWIs ($b_{\max}$ = 3000s/mm², TE = 80.3ms, TR = 5.4s, FOV = 24cm × 24cm × 12cm, isotropic voxel size 2.5mm, ASSET factor 2).

E. Experiments

In all experiments, training data originate from different human subjects than test data. Estimation of scalar measures based on q-DL was performed on the five-shell, three-shell and HCP data for all subsampling sizes $|\alpha|$ from n down to 8 (as well as down to 1 for error evaluation). DKI-based radial kurtosis [45] was estimated for HCP data and five-shell data. Different networks were trained for these different q-space sampling schemes. NODDI-based neurite orientation dispersion index [4] was estimated for three-shell data. State-of-the-art model fitting [4], [12] (own implementation for DKI; NODDI Matlab toolbox for NODDI) and compressed sensing (CS) for Cartesian schemes based on dictionary learning [46] (followed by model fitting) were performed for comparison because they are the currently used approaches to estimate model-based measures (CS was applied to registered Cartesian data of the same volunteer). Model fitting of one fully sampled scan was used on the training set to generate output targets for q-DL training. The quality of the methods on the test data was evaluated in terms of root-mean-squared error:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{j=1}^{J}\left(\hat{m}_j - m_{j,\mathrm{gt}}\right)^2}{J}}, \qquad (3)$$

where the sum is taken over all J voxels, the $\hat{m}_j$ are the results being evaluated, and the model fit of the nine additional independent repetitions of the scan was used for ground truth $m_{j,\mathrm{gt}}$ ("reference standard"). The five-shell data were used for this evaluation. The fraction of voxels for which the q-DL value was close to the reference standard value was calculated for the different scalar measures. In addition to the neural network settings described above, different numbers of units per hidden

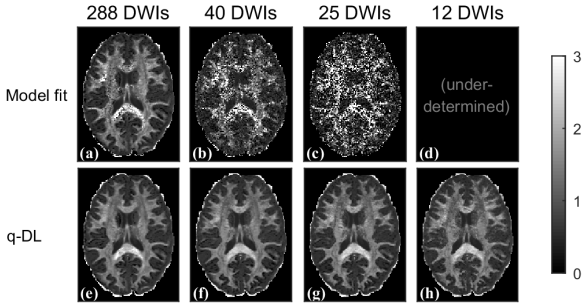


Fig. 2. Maps of radial kurtosis in the human brain for various methods and MRI scan acceleration factors. From left to right: 288, 40, 25 and 12 randomly selected DWIs are used. Model fitting followed by radial kurtosis calculation (a–d), and q-DL for radial kurtosis approximation (e–h) are compared. Model fitting is outperformed by the proposed method.

layer (between 50 and 750 in steps of 100) and different dropout fractions (between 0 and 0.5 in steps of 0.05) were compared. Using the three-shell datasets of four volunteers, the influence of three different training datasets on the same test dataset was compared, with reference standard obtained from fully-sampled model fitting.

Model-free segmentation was applied to the multiple sclerosis data. State-of-the-art automatic segmentation [30], [31] into lesions, healthy WM, GM and CSF based on non-diffusion images with spatial priors (see supporting information for details) was used as ground truth for our proof-of-concept model-free segmentation including diffusion images without spatial priors. The ground truth of the training data was used as output targets during training; the ground truth of the test data was used for segmentation quality evaluation. Segmentation quality was evaluated using the area under the curve (AUC) of the receiver operating characteristic (ROC). The deep learning models presented here cannot be more knowledgeable than the technique used to generate the labels.

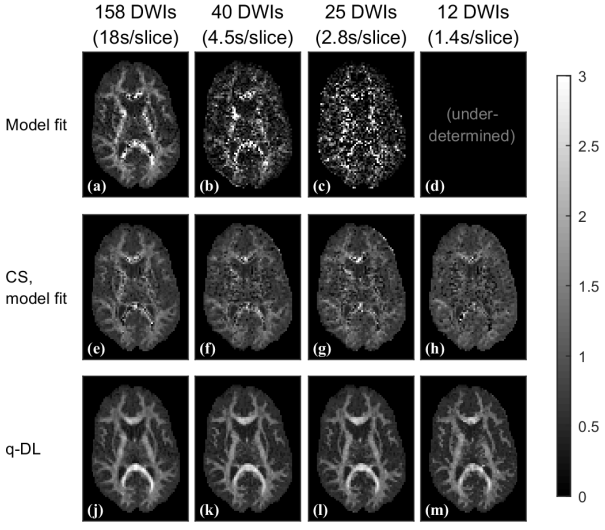


Fig. 3. Same as Fig. 2 (different scanner, different volunteer), including a comparison to compressed sensing (e–h). Required scan time for each sampling scheme is shown in seconds per slice. Model fitting and compressed sensing are outperformed by the proposed methods.

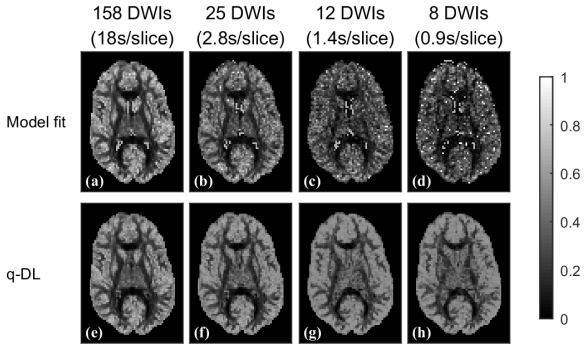


Fig. 4. Same as Fig. 2 for neurite orientation dispersion index based on NODDI. The proposed method better preserves contrast at short scan times.

III. Results and Discussion

Figs. 2–5 compare the estimation of scalar measures produced by different methods. We show DKI-based radial kurtosis [45] of HCP data in Fig. 2 and of five-shell data in Fig. 3 (with compressed sensing (CS) [46] applied to Cartesian data of the same volunteer in Fig. 3e–h) as well as NODDI-based neurite orientation dispersion index [4] of three-shell data in Fig. 4. State-of-the-art model fitting [4], [12] (Figs. 2a–d, 3a–d, 4a–d), CS (Fig. 3e–h), and q-DL (Figs. 2e–h, 3j–m, 4e–h) are compared. Several numbers of used DWIs are compared, ranging from full sampling to 12-fold reduced scan time (scan time is shown in seconds per image slice).

Compared with the standard pipeline, results of q-DL exhibit feasibility of scan time reduction by a factor of twelve. Thus, protocols lasting about 30 minutes (Figs. 2–4 panel a) can be reduced to 2.5 minutes, strongly improving clinical feasibility.

Fig. 5 compares the methods in terms of root-mean-squared error. This represents a quantitative evaluation of the results presented in Figs. 2–4. For DKI measures, q-DL always outperforms model fitting (Fig. 5a,b). Model fitting of 158 DWIs (error: 0.306 (Fig. 5a), 0.195 (Fig. 5b)) is even outperformed by q-DL of 12 DWIs (error: 0.272 (Fig. 5a),

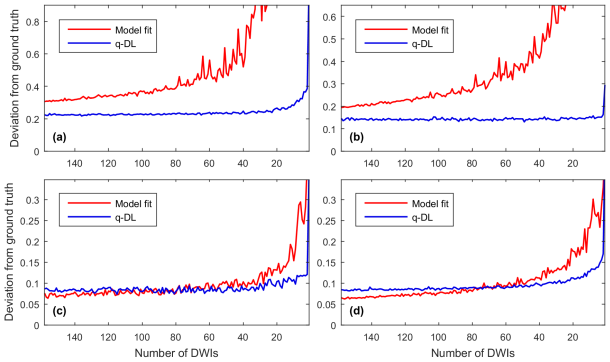


Fig. 5. Root-mean-squared error for different methods and different numbers of DWIs; estimation of radial kurtosis (a), fractional kurtosis anisotropy (b), intra-cellular volume fraction (c), and neurite orientation dispersion index (d); comparison of two different methods: model fitting (red) and q-DL (blue). Reference standard is model fit of nine independent repetitions, i.e. 1422 DWIs, registered to the test data. For DKI measures (a,b), model fitting is always outperformed by q-DL. For NODDI measures (c,d), model fitting is outperformed by q-DL if less than 70 DWIs are used.

TABLE I
ERROR REPRODUCIBILITY

Sampling Scheme	Training Dataset	Initialization 1	Initialization 2	Initialization 3
1	2	0.331	0.329	0.331
1	3	0.321	0.320	0.321
1	4	0.329	0.332	0.330
2	2	0.337	0.345	0.332
2	3	0.332	0.335	0.334
2	4	0.340	0.340	0.340
3	2	0.334	0.343	0.341
3	3	0.327	0.326	0.329
3	4	0.343	0.341	0.342

Root-mean-squared error of radial kurtosis estimated by q-DL from 12 DWIs of test dataset 1 for 27 experiments with different random subsampling schemes, different volunteer training datasets and different neural network initializations.

0.150 (Fig. 5b)). For NODDI measures, q-DL outperforms model fitting when less than 70 DWIs are used (Fig. 5c,d).

These curves demonstrate the trade-off between scan duration and quality provided by q-DL. Particularly, twelve-fold reduced scan time provides an error magnitude similar to that of model fitting at full scan time (and for DKI-based measures even lower than that of model fitting at full scan time).

For each number of subsampled DWIs, the subsampling was performed randomly and completely independently (but equally for the three compared methods). Thus, oscillations (amplitude of fluctuation) of the curves in Fig. 5 demonstrate the impact of random subsampling. Not all random subsamplings are equally useful. Among the compared methods, q-DL is most stable with respect to the choice of the samples, whereas model fitting decreases in stability (from very stable to unstable) with decreasing number of DWIs. Analogous variation was observed for repetitions of random subsampling instantiations when the number of DWIs was held constant (not shown).

For model fitting of 158 DWIs, 95.0% of all voxels had a value within the interval $m_{gt}\pm0.5$ (where m_{gt} is the reference standard value) for radial kurtosis. The ratio was comparably high at 94.7% for q-DL of only 12 DWIs. For fractional kurtosis anisotropy, 81.0% of all voxels in model fit of 158 DWIs had a value in the interval $m_{gt}\pm0.3$, whereas for q-DL of only 12 DWIs the ratio was as high as 95.9%. Intracellular volume fraction was estimated within $m_{gt}\pm0.3$ by model fit of 158 DWIs in

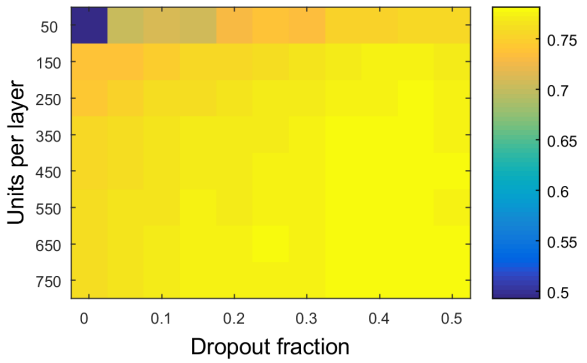


Fig. 6. Correlation of radial kurtosis estimations using different dropout fractions and layer sizes for q-DL from 12 DWIs with radial kurtosis from fully sampled (148 DWIs) model fitting.

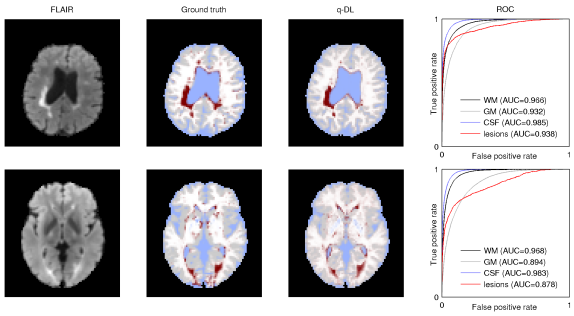


Fig. 7. Direct model-free tissue segmentation and lesion detection. When learning to discriminate multiple sclerosis lesions (red), healthy WM, GM and CSF based on DWIs and FLAIR, the proposed method segments the tissue types well and reliably detects lesions without using any diffusion model. Slices from datasets with the best (upper row, 0.938) and worst lesion AUC (lower row, 0.878) are shown.

88.9% of all voxels, and by q-DL of 12 DWIs in 90.6%. For neurite orientation dispersion index, the ratios were 79.3% and 85.3%, respectively. Thus, q-DL of as few as 12 DWIs provides comparable, and often a better, proximity to the true value compared to model fitting of as many as 158 DWIs.

Table I shows the effects of different random subsampling schemes, training datasets and neural network initializations on the error. All results are very similar; each training dataset leads to good results. Accidental generation of a degenerate subsampling scheme or degenerate network initialization is extremely improbable, has not been encountered in practice, and can be easily checked for (using any qualitative or quantitative experiment).

Fig. 6 shows the effect of neural network settings on the test set quality, indicating that using at least 150 hidden units per layer or a dropout [21] fraction of at least 0.05 improves the performance of q-DL. Results of other quality measures such as root-mean-squared deviation are analogous (not shown). Note that we merely compare the effect of different parameters on the test set, rather than performing definitive hyper-parameter fitting on a validation set.

The final application of q-DL presented here is tissue segmentation and lesion detection. This task is achieved by training the neural network to discriminate between several tissue types based on the diffusion-weighted signal from the DWIs. In a proof-of-concept experiment, we used segmentation

TABLE II
COMPARISON OF REQUIRED NUMBER OF DWIS

Method	Number of DWIs required for DKI	Number of DWIs required for NODDI	References
Standard Pipeline	150	99	[3], [4]
Compressed Sensing	64	–	[50]
Machine Learning *with* model fitting	–	30	[48]
Analytical Solutions	13-19 (specific measures only)	–	[15], [16]
q-Space Deep Learning	**12**	**8**	proposed

Comparison of suggested protocols and scan time for scalar measure estimation using different methods. q-Space Deep Learning provides the highest scan time reduction for both DKI and NODDI.

into WM, GM, CSF and multiple sclerosis lesions. Segmentation results from q-DL are shown in Fig. 7. The AUC of the ROC for lesions ranged between 0.878 and 0.938 for six different patients. AUC for WM, GM and CSF was consistently above 0.894 for all patients. Thus, DWIs can be used directly for segmentation without a diffusion model, i.e. without the intermediate information loss detailed in section I.B. Tailoring the protocol to optimal results in specific applications is subject of future research.

Other previously proposed methods, including machine learning methods [17], [18], [47]–[49] as well as state-of-the-art compressed sensing [50] require more DWIs and several intermediate steps (see Fig. 1b). For the number of DWIs suggested for different methods, see Table II. Most notably, compressed sensing and machine learning publications suggest using 64 DWIs for DKI [50] and 30 DWIs for NODDI [48], whereas our methods work with only 12 DWIs for DKI and 8 DWIs for NODDI. Previous work that uses the DWIs directly as inputs to machine learning for tissue characterization [17], [18] does not only use large numbers of DWIs but is also limited so far to Monte Carlo simulations only, rather than *in vivo* experiments. A related idea is the use of DWIs directly as inputs to machine learning for tractography [25].

When switching to another scanner such that the DWI intensities are not the same anymore, the intensities should either be normalized or the network should be retrained. The same holds for changes in acquisition parameters such as echo time. A network that is able to understand data from different settings is subject of future research.

In all presented applications, neural network training takes about one minute on a desktop computer. The network needs to be trained only once and can be applied to any number of datasets, taking 0.03 seconds per dataset, as opposed to several minutes per dataset required by most model fitting methods. Analytical solutions [15], [16] of scalar measure estimation provide acceleration of acquisition and processing comparable to q-DL, but are limited to specific scalar measures and acquisition schemes. With q-DL, the acceleration factor can be freely chosen and all scalar measures can be obtained simultaneously. There is also freedom in the choice of the sampling; in particular, random sampling yields robust results.

IV. Conclusions

The presented scan acceleration factor twelve sets a new state of the art in DKI and NODDI and thus opens new perspectives for clinical protocols. The results indicate that a considerable amount of information is contained in a limited number of DWIs, and that this information can be better retrieved by deep learning than by model fitting. The number of used DWIs can be freely chosen and represents a better trade-off between scan duration and quality than provided by conventional methods.

Our framework for model-free diffusion MRI can be used to estimate arbitrary tissue properties in various settings where ground truth training datasets are available. Future research may focus on creating ground truth training data from simulations, scanned phantoms and histologically validated data. Moreover, q-DL is the first model-free diffusion MRI segmentation method, meaning that it uses q-space data directly and does not partly discard information at intermediate steps.

Recent work [51] indicates that the complexity of state-of-the-art diffusion models is at the limit of allowing a stable model fit to the noisy diffusion MRI data obtained in an acceptable scan duration. Herein we demonstrate the fact that omitting model fitting allows considerably more stable measure estimation at short scan durations; this might circumvent the fitting stability "bottleneck" when balancing scan duration against model complexity.

Classical quantitative diffusion MRI requires creating a diffusion model that well captures disease-related tissue changes via its associated scalar measures. Subsequently, a set of MRI contrasts needs to be chosen (diffusion-weighted gradient strengths and durations, single-pulsed or other gradient forms, non-diffusion sequences) that allow estimating all parameters of the model. The presented segmentation and abnormality detection method on the other hand is concerned with finding a set of contrasts whose signal "vector" (signal values from all contrasts) is most strongly affected by disease[3]. Simulational tissue models can still drive the design of meaningful gradient forms, but subsequent experiments do not rely on any model – particularly, model parameters do not have to be estimated. This allows future research to explore experiment design using elaborate simulational tissue models with large numbers of microstructural parameters. In this framework, model complexity is not limited by ill-posedness of subsequent model parameter estimation.

A combination of q-DL (requiring twelve times less DWIs than standard methods for estimation of arbitrary scalar measures) with simultaneous multi-slice imaging [39] (three-fold accelerated acquisition of the DWIs) in future applications is straightforward, yielding an unprecedented 36-fold scan time reduction.

Our recommendation in the short term is to use short acquisitions with q-DL instead of long acquisitions with fitting. In the long term, we recommend creating complex tissue models that are not limited by fitting instabilities and using model-free q-DL tissue characterization.

The capability of q-DL to accelerate the acquisition by an order of magnitude and detect tissue changes without a diffusion model opens new perspectives for research in quantitative diffusion MRI and demonstrates the benefits of deep learning for multi-step data processing pipelines.

Acknowledgment

We thank Sebastian Pölsterl and Björn Menze (TU Munich) for discussions. Data for Fig. 2 were provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers

[3] In other words, we search for a set of contrasts that well captures disease-related variation of the data; as opposed to principle component analysis and related methods, deep learning allows capturing arbitrary non-linear data distributions.

that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

REFERENCES

[1] D. Le Bihan, "Looking into the functional architecture of the brain with diffusion MRI," *Nat. Rev. Neurosci.*, vol. 4, no. 6, pp. 469–80, 2003.

[2] J. H. Jensen, J. A. Helpern, A. Ramani, H. Lu, and K. Kaczynski, "Diffusional kurtosis imaging: the quantification of non-Gaussian water diffusion by means of magnetic resonance imaging," *Magn. Reson. Med.*, vol. 53, no. 6, pp. 1432–40, 2005.

[3] H. Lu, J. H. Jensen, A. Ramani, and J. A. Helpern, "Three-dimensional characterization of non-Gaussian water diffusion in humans using diffusion kurtosis imaging," *NMR Biomed.*, vol. 19, no. 2, pp. 236–47, 2006.

[4] H. Zhang, T. Schneider, C. A. Wheeler-Kingshott, and D. C. Alexander, "NODDI: practical in vivo neurite orientation dispersion and density imaging of the human brain," *NeuroImage*, vol. 61, no. 4, pp. 1000–16, 2012.

[5] A. R. Padhani, G. Liu, D. Mu-Koh, T. L. Chenevert, H. C. Thoeny, B. D. Ross, M. Van Cauteren, D. Collins, D. A. Hammoud, G. J. S. Rustin, and B. Taouli, "Diffusion-weighted magnetic resonance imaging as a cancer biomarker: consensus and recommendations," *Neoplasia*, vol. 11, no. 2, pp. 102–125, 2009.

[6] M. Hori, I. Fukunaga, Y. Masutani, T. Taoka, K. Kamagata, Y. Suzuki, and S. Aoki, "Visualizing non-Gaussian diffusion: clinical application of q-space imaging and diffusional kurtosis imaging of the brain and spine," *Magn. Reson. Med. Sci.*, vol. 11, no. 4, pp. 221–33, 2012.

[7] A. Luna, R. Ribes, and J. A. Soto, *Diffusion MRI Outside the Brain*. Springer Berlin Heidelberg, 2012.

[8] J. H. Gillard, A. D. Waldman, and P. B. Barker, Eds., *Clinical MR neuroimaging: physiological and functional techniques*, 2nd ed. Cambridge: Cambridge University Press, 2010.

[9] M. M. Cheung, E. S. Hui, K. C. Chan, J. A. Helpern, L. Qi, and E. X. Wu, "Does diffusion kurtosis imaging lead to better neural tissue characterization? A rodent brain maturation study," *NeuroImage*, vol. 45, no. 2, pp. 386–92, 2009.

[10] G. P. Winston, "The physical and biological basis of quantitative parameters derived from diffusion MRI," *Quant. Imaging Med. Surg.*, vol. 2, no. 4, pp. 254–65, 2012.

[11] L. Qi, Y. Wang, and E. X. Wu, "D-eigenvalues of diffusion kurtosis tensors," *J. Comput. Appl. Math.*, vol. 221, no. 1, pp. 150–157, 2008.

[12] J. Veraart, J. Sijbers, S. Sunaert, A. Leemans, and B. Jeurissen, "Weighted linear least squares estimation of diffusion MRI parameters: strengths, limitations, and pitfalls," *NeuroImage*, vol. 81, pp. 335–46, 2013.

[13] D. H. J. Poot, A. J. den Dekker, E. Achten, M. Verhoye, and J. Sijbers, "Optimal experimental design for diffusion kurtosis imaging," *IEEE Trans. Med. Imaging*, vol. 29, no. 3, pp. 819–29, 2010.

[14] J. Veraart, W. Van Hecke, and J. Sijbers, "Constrained maximum likelihood estimation of the diffusion kurtosis tensor using a Rician noise model," *Magn. Reson. Med.*, vol. 66, no. 3, pp. 678–86, 2011.

[15] B. Hansen, T. E. Lund, R. Sangill, and S. N. Jespersen, "Experimentally and computationally fast method for estimation of a mean kurtosis," *Magn. Reson. Med.*, vol. 69, no. 6, pp. 1754–60, 2013.

[16] B. Hansen, T. E. Lund, R. Sangill, and S. N. Jespersen, "A fast and robust method for simultaneous estimation of mean diffusivity and mean tensor kurtosis," in *Proc. Joint Annual Meeting ISMRM-ESMRMB*, 2014, p. 2602.

[17] M. Nilsson, E. Alerstam, R. Wirestam, F. Ståhlberg, S. Brockstedt, and J. Lätt, "Evaluating the accuracy and precision of a two-compartment Kärger model using Monte Carlo simulations," *J. Magn. Reson.*, vol. 206, pp. 59–67, 2010.

[18] G. Nedjati-Gilani, M. G. Hall, C. A. M. Wheeler-Kingshott, and D. C. Alexander, "Learning microstructure parameters from diffusion-weighted MRI using random forests," in *Joint Annual Meeting ISMRM-ESMRMB*, 2014, p. 2626.

[19] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *IEEE 12th International Conference on Computer Vision*, 2009, pp. 2146–2153.

[20] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th International Conference on Machine Learning*, 2010, no. 3, p. 432.

[21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv:1207.0580*, 2012.

[22] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *Proc. International Conference on Learning Representations*, 2014.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[24] V. Golkov, A. Dosovitskiy, P. Sämann, J. I. Sperl, T. Sprenger, M. Czisch, M. I. Menzel, P. A. Gómez, A. Haase, T. Brox, and D. Cremers, "q-Space deep learning for twelve-fold shorter and model-free diffusion MRI scans," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 37–44.

[25] P. F. Neher, M. Götz, T. Norajitra, C. Weber, and K. H. Maier-Hein, "A machine learning based approach to fiber tractography using classifier voting," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 45–52.

[26] S. M. Plis, D. R. Hjelm, R. Slakhutdinov, E. A. Allen, H. J. Bockholt, J. D. Long, H. Johnson, J. Paulsen, J. Turner, and V. D. Calhoun, "Deep learning for neuroimaging: a validation study," *Front. Neurosci.*, vol. 8:229, 2014.

[27] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in Neural Information Processing Systems*, 2014, vol. 27, pp. 2924–2932.

[28] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," Master's thesis, Technical University of Denmark, 2012.
[29] H. Bagher-Ebadian, K. Jafari-Khouzani, P. D. Mitsias, M. Lu, H. Soltanian-Zadeh, M. Chopp, and J. R. Ewing, "Predicting final extent of ischemic infarction using artificial neural network analysis of multi-parametric MRI in patients with stroke," *PLoS One*, vol. 6, no. 8, 2011.
[30] J. Ashburner and K. J. Friston, "Unified segmentation," *NeuroImage*, vol. 26, pp. 839–851, 2005.
[31] K. Van Leemput, F. Maes, D. Vandermeulen, A. Colchester, and P. Suetens, "Automated segmentation of multiple sclerosis lesions by model outlier detection," *IEEE Trans. Med. Imaging*, vol. 20, pp. 677–688, 2001.
[32] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, 2nd ed., G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Springer, 2012.
[33] V. J. Wedeen, P. Hagmann, W.-Y. I. Tseng, T. G. Reese, and R. M. Weisskoff, "Mapping complex tissue architecture with diffusion spectrum magnetic resonance imaging," *Magn. Reson. Med.*, vol. 54, no. 6, pp. 1377–86, 2005.
[34] J. L. R. Andersson, S. Skare, and J. Ashburner, "How to correct susceptibility distortions in spin-echo echo-planar images: application to diffusion tensor imaging," *NeuroImage*, vol. 20, no. 2, pp. 870–88, 2003.
[35] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. J. Behrens, H. Johansen-Berg, P. R. Bannister, M. De Luca, I. Drobnjak, D. E. Flitney, R. K. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. De Stefano, J. M. Brady, and P. M. Matthews, "Advances in functional and structural MR image analysis and implementation as FSL," *NeuroImage*, vol. 23 Suppl 1, pp. S208–19, 2004.
[36] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. W. Pluim, "elastix: a toolbox for intensity based medical image registration," *IEEE Trans. Med. Imaging*, vol. 29, no. 1, pp. 196 – 205, 2010.
[37] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. J. Behrens, E. Yacoub, and K. Ugurbil, "The WU-Minn Human Connectome Project: An overview," *NeuroImage*, vol. 80, pp. 62–79, 2013.
[38] D. A. Feinberg, S. Moeller, S. M. Smith, E. Auerbach, S. Ramanna, M. F. Glasser, K. L. Miller, K. Ugurbil, and E. Yacoub, "Multiplexed echo planar imaging for sub-second whole brain fmri and fast diffusion imaging," *PLoS One*, vol. 5, no. 12, 2010.
[39] K. Setsompop, J. Cohen-Adad, B. A. Gagoski, T. Raij, A. Yendiki, B. Keil, V. J. Wedeen, and L. L. Wald, "Improving diffusion MRI using simultaneous multi-slice echo planar imaging," *NeuroImage*, vol. 63, no. 1, pp. 569–80, 2012.
[40] J. Xu, K. Li, R. A. Smith, J. C. Waterton, P. Zhao, H. Chen, M. D. Does, H. C. Manning, and J. C. Gore, "Characterizing tumor response to chemotherapy at various length scales using temporal diffusion spectroscopy," *PLoS One*, vol. 7, no. 7, e41714, 2012.
[41] S. N. Sotiropoulos, S. Jbabdi, J. Xu, J. L. Andersson, S. Moeller, E. J. Auerbach, M. F. Glasser, M. Hernandez, G. Sapiro, M. Jenkinson, D. a. Feinberg, E. Yacoub, C. Lenglet, D. C. Van Essen, K. Ugurbil, and T. E. J. Behrens, "Advances in diffusion MRI acquisition and processing in the Human Connectome Project," *NeuroImage*, vol. 80, pp. 125–143, 2013.
[42] M. F. Glasser, S. N. Sotiropoulos, J. A. Wilson, T. S. Coalson, B. Fischl, J. L. Andersson, J. Xu, S. Jbabdi, M. Webster, J. R. Polimeni, D. C. Van Essen, and M. Jenkinson, "The minimal preprocessing pipelines for the Human Connectome Project," *NeuroImage*, vol. 80, pp. 105–124, 2013.
[43] M. Jenkinson, C. F. Beckmann, T. E. J. Behrens, M. W. Woolrich, and S. M. Smith, "FSL," *NeuroImage*, vol. 62, no. 2, pp. 782–790, 2012.
[44] B. Fischl, "FreeSurfer," *NeuroImage*, vol. 62, no. 2, pp. 774–781, 2012.
[45] E. S. Hui, M. M. Cheung, L. Qi, and E. X. Wu, "Towards better MR characterization of neural tissues using directional diffusion kurtosis analysis," *NeuroImage*, vol. 42, no. 1, pp. 122–34, 2008.
[46] B. Bilgic, K. Setsompop, J. Cohen-Adad, A. Yendiki, L. L. Wald, and E. Adalsteinsson, "Accelerated diffusion spectrum imaging with compressed sensing using adaptive dictionaries," *Magn. Reson. Med.*, vol. 68, no. 6, pp. 1747–54, 2012.
[47] G. Nedjati-Gilani, T. Schneider, M. G. Hall, C. A. M. Wheeler-Kingshott, and D. C. Alexander, "Machine learning based compartment models with permeability for white matter microstructure imaging," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2014, pp. 257–264.
[48] D. C. Alexander, D. Zikic, J. Zhang, H. Zhang, and A. Criminisi, "Image quality transfer via random forest regression: applications in diffusion MRI," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2014, pp. 225–232.
[49] T. Schultz, "Learning a reliable estimate of the number of fiber directions in diffusion MRI," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2012, pp. 493–500.
[50] M. Paquette, S. Merlet, G. Gilbert, R. Deriche, and M. Descoteaux, "Comparison of sampling strategies and sparsifying transforms to improve compressed sensing diffusion spectrum imaging," *Magn. Reson. Med.*, vol. 73, pp. 401–416, 2015.
[51] U. Ferizi, T. Schneider, T. Witzel, L. L. Wald, H. Zhang, C. A. M. Wheeler-Kingshott, and D. C. Alexander, "White matter compartment models for in vivo diffusion MRI at 300mT/m," *NeuroImage*, vol. 118, pp. 468–483, 2015.

q-Space Deep Learning (Golkov et al.)
Supplementary Materials

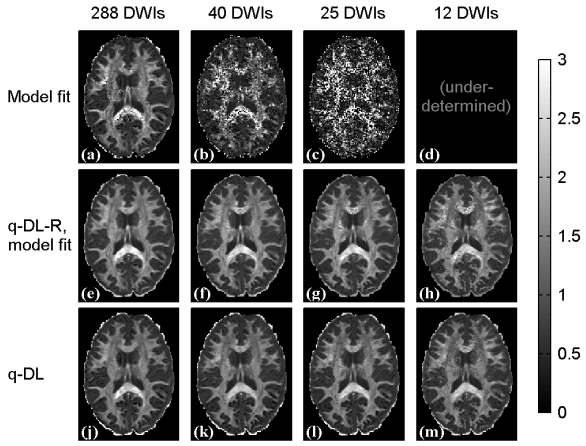


Fig. 8. Same as Fig. 2, including a comparison with q-DL-R followed by model fitting (e–h).

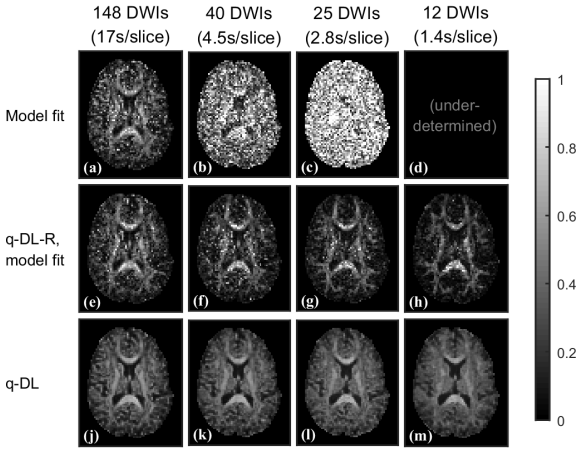


Fig. 9. Same as Fig. 8 for fractional kurtosis anisotropy based on DKI. The proposed methods better cope with noisy data despite noisy training data and strongly outperform model fitting.

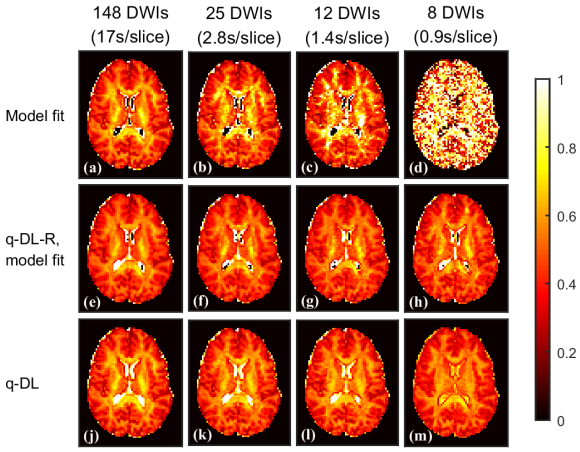


Fig. 10. Same as Fig. 8 for intra-cellular volume fraction based on NODDI. The original protocol for this model [4] consists of 99 DWIs, and, more recently, an acceleration to 30 DWIs by means of machine learning was achieved with some contrast loss [48], whereas the methods proposed herein require only 8 DWIs and preserve contrast (panels (h) and (m)).

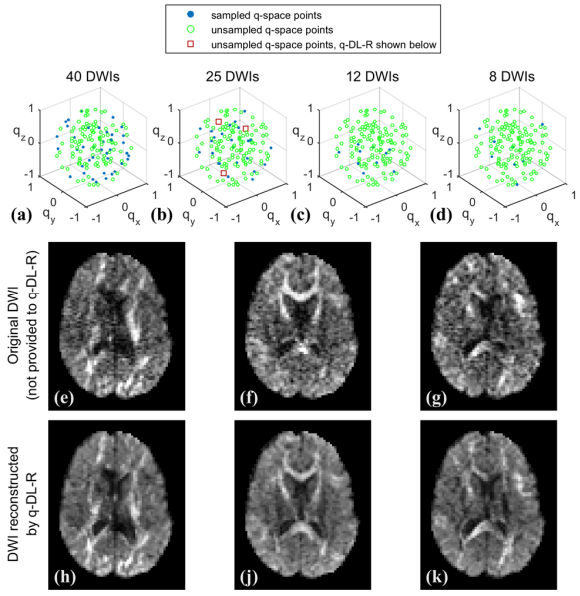


Fig. 11. Sampling schemes and DWIs reconstructed by q-DL-R. Random subsampling of the optimized DKI sampling scheme [13], [14] used in our experiments (a–d), three original DWIs acquired but not provided to q-DL-R (e–g) and three corresponding DWIs reconstructed by q-DL-R from 25 other DWIs (h–k) are shown.

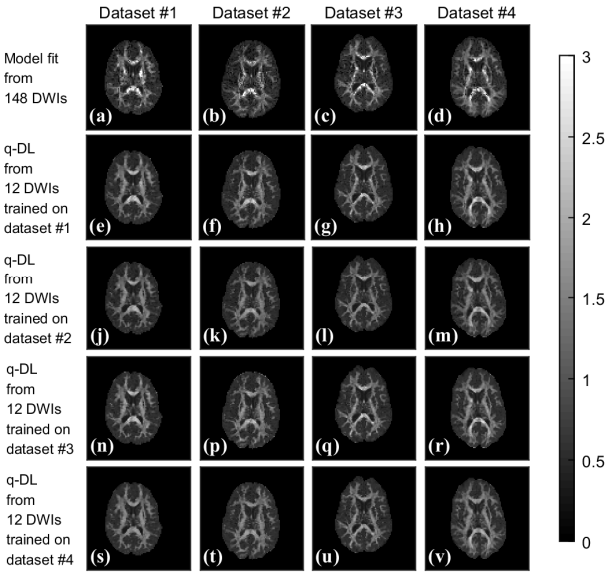


Fig. 12. Reproducibility of twelve-fold accelerated q-DL for four healthy volunteers. Radial kurtosis maps are shown. Standard pipeline from 148 DWIs (a–d) and twelve-fold accelerated q-DL for all combinations of training and test dataset choices (e–v) are shown.

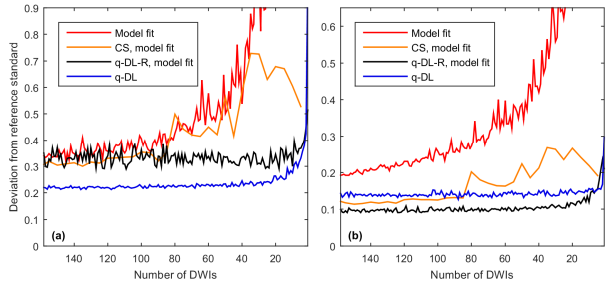


Fig. 13. Same as Fig. 5a–b (error of radial kurtosis (a) and fractional kurtosis anisotropy (b) for five-shell data) but also with evaluation of compressed sensing [46] (orange curve; evaluated only in one image slice as opposed to Fig. 5a–b, and for CS only every five abscissa positions due to long execution time of CS) and of q-DL-R (black curve). Model fitting is outperformed by CS nearly everywhere. For radial kurtosis, CS is outperformed by q-DL everywhere, and by q-DL-R if less than 80 DWIs are used (a). For fractional kurtosis anisotropy, CS is outperformed by q-DL-R everywhere, and by q-DL if less than 80 DWIs are used (b).

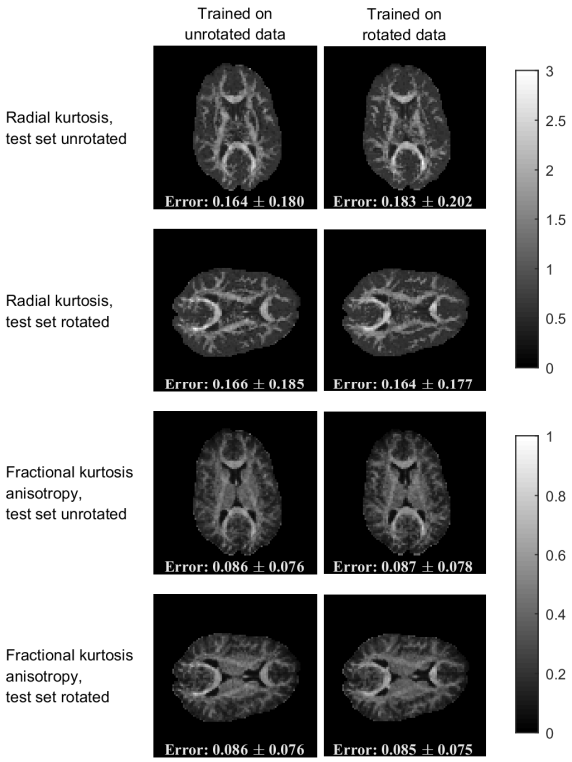


Fig. 14. The effect of a tilted head in the training set and/or the test set for q-DL of 12 DWIs. Fully-sampled Cartesian data was jointly rotated in image space and in q-space by 90°, without rotating the retrospective q-space subsampling mask of 12 DWIs. This allowed to study the effect of the data being rotated (jointly rotated in image space and q-space) or not rotated (neither in image space nor q-space). Mean and standard deviation of the error magnitude throughout all voxels is given. The method works well in all four cases. If training and test set have the same orientation (either both rotated, or both unrotated), the error is marginally lower.

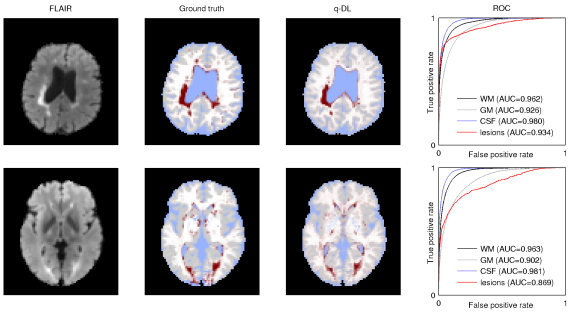


Fig. 15. Same as Fig. 6 without using FLAIR, i.e. using DWIs only.

TABLE III
SEGMENTATION QUALITY

Inputs	Dataset	WM	GM	CSF	Lesions
DWIs and FLAIR	1	0.976	0.934	0.978	0.880
DWIs and FLAIR	2	0.966	0.931	0.984	0.900
DWIs and FLAIR	3	0.968	0.894	0.983	0.878
DWIs and FLAIR	4	0.966	0.932	0.985	0.938
DWIs and FLAIR	5	0.969	0.932	0.986	0.930
DWIs and FLAIR	6	0.973	0.906	0.989	0.868
DWIs only	1	0.975	0.931	0.981	0.878
DWIs only	2	0.962	0.917	0.975	0.898
DWIs only	3	0.963	0.902	0.981	0.869
DWIs only	4	0.962	0.926	0.980	0.934
DWIs only	5	0.967	0.925	0.981	0.928
DWIs only	6	0.970	0.892	0.985	0.859
FLAIR only	1	0.641	0.599	0.838	0.561
FLAIR only	2	0.674	0.649	0.935	0.659
FLAIR only	3	0.638	0.574	0.901	0.673
FLAIR only	4	0.692	0.657	0.948	0.753
FLAIR only	5	0.711	0.599	0.953	0.845
FLAIR only	6	0.744	0.601	0.977	0.627

Area under the curve of the receiver operating characteristic for each patient.

APPENDIX

A. Additional Tasks and Formal Algorithms

1) Denoising

The first and most simple application of our framework is denoising of DWIs, introduced here merely for the sake of completeness. For denoising, the voxel-wise signal from all DWIs is used as both the input and output of the neural network. A network trained to reconstruct its own inputs is known as an autoencoder [52]. Its approximate nature and dropout-based training [21] prevent overfitting and thus reduce noise. For q-DL denoising, the signal vector S_j of length n (from all n DWIs) in voxel j is used as both the input $a_j^{(0)}$ and target y_j of the neural network. The number of network inputs is the number n of used DWIs, i.e. each input vector $a_j^{(0)} = S_j$ has length n (for every j). The length of the output vector $a_j^{(L)} = S_j$ is also n. This is a special case of reconstruction of DWIs (see below). In pseudo-code, q-DL denoising can be represented as follows:

Algorithm 1: q-DL denoising
inputs: $\text{dataset}_{\text{training}}$, $\text{dataset}_{\text{test}}$
// load training DWIs
$S_{\text{training}} \leftarrow$ load_many_DWIs($\text{dataset}_{\text{training}}$)
// train neural network to predict targets from inputs:
nn $\leftarrow$ deep_learning_training(nn_inputs=S_{training},
nn_targets=S_{training})
$S_{\text{test}} \leftarrow$ load_many_DWIs($\text{dataset}_{\text{test}}$)
$S_{\text{test,denoised}} \leftarrow$ get_neural_network_outputs(network=nn,
nn_inputs=S_{test})
output: $S_{\text{test,denoised}}$

Note that the denoising effect in this preliminary study is due to the inherent regularity of the mapping. Specific distributions (e.g. Rician) can be accounted for in future work by using tailored denoising-based training [53].

2) Reconstruction of Missing DWIs

For q-DL-based Reconstruction of missing DWIs (q-DL-R), a neural network is trained to predict the signal S_j in all DWIs (voxel j) from a reduced subset $S_{j,\alpha}$ where α is a pseudorandom subsampling multi-index (such that the q-space sampling is consistent across training and test data). The input vector consists of the intensities of the subsampled DWIs, i.e. $a_j^{(0)} = S_{j,\alpha}$, and its length is $|\alpha|$. The output vector consists of the intensities of all the DWIs that are being reconstructed, i.e. $a_j^{(L)} = S_j$ with length n. Due to partial data redundancy in q-space, missing DWIs can be reconstructed from a reduced subset. By "all DWIs", we refer to all the DWIs available in a fully sampled training set (in the different datasets used herein, the number is between 148 and 288). For each length of α, the DWIs were selected uniformly randomly, and completely independently for different lengths of α. One b=0 image was always used, the other b=0 images were treated equally to DWIs in the random selection process.

Algorithm 2: q-DL-R
inputs: $\text{dataset}_{\text{training}}$, $\text{dataset}_{\text{test}}$
$S_{\text{training}} \leftarrow$ load_many_DWIs($\text{dataset}_{\text{training}}$)
$S_{\text{training,subsampled}} \leftarrow$ subsample(S_{training})
nn $\leftarrow$ deep_learning_training(nn_inputs=$S_{\text{training,subsampled}}$,
nn_targets=S_{training})
// test dataset may have only few DWIs
$S_{\text{test,subsampled}} \leftarrow$ load_few_DWIs($\text{dataset}_{\text{test}}$)
$S_{\text{test,reconstructed}} \leftarrow$ get_neural_network_outputs(network=nn,
nn_inputs=$S_{\text{test,subsampled}}$)
output: $S_{\text{test,reconstructed}}$

3) Estimation of Scalar Measures

Algorithm 3: q-DL
inputs: $\text{dataset}_{\text{training}}$, $\text{dataset}_{\text{test}}$
$S_{\text{training}} \leftarrow$ load_many_DWIs($\text{dataset}_{\text{training}}$)
// model fitting f and scalar measure calculation g
$m_{\text{training}} \leftarrow g(f(S_{\text{training}}))$
$S_{\text{training,subsampled}} \leftarrow$ subsample(S_{training})
nn $\leftarrow$ deep_learning_training(nn_inputs=$S_{\text{training,subsampled}}$,
nn_targets=m_{training})
// test dataset may have only few DWIs
$S_{\text{test,subsampled}} \leftarrow$ load_few_DWIs($\text{dataset}_{\text{test}}$)
$m_{\text{test,approximated}} \leftarrow$ get_neural_network_outputs(network=nn,
nn_inputs=$S_{\text{test,subsampled}}$)
output: $m_{\text{test,approximated}}$

4) Model-Free Segmentation

Algorithm 4: q-DL segmentation
inputs: $\text{dataset}_{\text{training}}$, $\text{dataset}_{\text{test}}$
$S_{\text{training}} \leftarrow$ load_many_DWIs($\text{dataset}_{\text{training}}$)
// gold standard labels for training data
$l_{\text{training}} \leftarrow$ load_tissue_labels($\text{dataset}_{\text{training}}$)
$S_{\text{training,subsampled}} \leftarrow$ subsample(S_{training})
nn $\leftarrow$ deep_learning_training(nn_inputs=$S_{\text{training,subsampled}}$,
nn_targets=l_{training})
// test dataset may have only few DWIs
$S_{\text{test,subsampled}} \leftarrow$ load_few_DWIs($\text{dataset}_{\text{test}}$)
$l_{\text{test,approximated}} \leftarrow$ get_neural_network_outputs(network=nn,
nn_inputs=$S_{\text{test,subsampled}}$)
output: $l_{\text{test,approximated}}$

B. Differences of q-DL(-R) to Other Methods

1) Differences to Compressed Sensing

One of the several tasks presented herein is indeed the reconstruction of missing q-space data, and this is similar to compressed sensing in q-space [46], [50]. However, there are major differences that separate our work from compressed sensing methods. While an extensive comparison of state-of-the-art compressed sensing methods [50] has revealed that about 64 DWIs are required for DKI compressed sensing, we require only 12 DWIs. Another difference between compressed sensing and q-DL-R is that most compressed sensing methods require data subsampled on a specific q-space grid, whereas q-DL-R does not require the q-space coordinates as an input and can be applied in a straightforward manner to arbitrary (i.e. non-Cartesian, non-radial) sampling schemes. Moreover, while we do explain q-DL denoising as well as q-DL-R *for the sake of completeness* (rather than competitiveness), the focus of our work and the main contributions are somewhat different. We introduce direct estimation of arbitrary scalar measures in one single step *without model fitting* from few DWIs. This is different from compressed sensing methods which rely on model fitting as an additional processing step. Moreover, in contrast to state-of-the-art methods, we introduce segmentation *without using diffusion models*. These differences to compressed sensing are shown in Fig. 1 in terms of data processing, in Table II in terms of number of required DWIs, and evaluated in Figs. 3 and 13. Besides, compressed sensing is image-model-driven (even if some model parameters can be learned) and even the most elaborate image models (regularizers) introduce artifacts [54], whereas deep learning is data-driven.

2) Differences to Other Machine Learning Methods

Previous machine learning methods for diffusion MRI rely either on using an extensive set of DWIs [17], [18] or on fitting diffusion models as an intermediate data processing step and using either the fitted model parameters [48] or scalar measures calculated from model parameters [47], [49] as inputs to pattern recognition and machine learning algorithms in the next step. In contrast to these methods, we use *a reduced set of DWIs directly* as input for deep learning. To our knowledge, using less DWIs in conjunction with machine learning is performed only by [48]; therein, the authors propose performing model fitting first, followed by machine learning, requiring 30 DWIs for NODDI, whereas we use the DWIs as inputs directly, requiring only 8 DWIs. Differences are shown in Fig. 1 in terms of data processing and in Table II in terms of number of required DWIs. Previous work that uses the DWIs directly as inputs to machine learning for tissue characterization [17], [18] does not only use large numbers of DWIs but is also limited so far to Monte Carlo simulations only, rather than *in vivo* experiments, whereas we demonstrate *in vivo* experiments and reduced numbers of DWIs. In our research, we tried random forests on the subsampled DWIs directly (this also is novel), but the results (not shown) were inferior to deep learning.

3) Differences to Dictionary Learning

Dictionary learning methods [46], [55] can be considered single-layer models, and lack the capability of learning more powerful multilayer hierarchical representations, whereas we propose learning multilayer representations (three hidden layers in our experiments). Dictionary learning methods require about 40 DWIs [55]. The compressed sensing method used in Figs. 3 and 13 is based on dictionary learning [46].

C. Additional Remarks

Fig. 12 demonstrates reproducibility of twelve-fold accelerated processing using q-DL with respect to different choices of training and test datasets.

The quality of solutions obtained by deep learning – despite random initialization and randomized stochastic training – was empirically shown to be high in numerous tasks [23]. Recent work [27], [56]–[59] has revealed new theoretical underpinnings of the success of deep learning.

Supplementary References

[52] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1798 – 1828, 2013.

[53] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[54] M. Benning, C. Brune, M. Burger, and J. Müller, "Higher-order TV methods—enhancement via Bregman iteration," *J. Sci. Comp.*, vol. 54, no. 2–3, 2012.

[55] A. Gramfort, C. Poupon, and M. Descoteaux, "Denoising and fast diffusion imaging with physically constrained sparse dictionary learning," *Med. Image Anal.*, vol. 18, no. 1, pp. 36–49, 2014.

[56] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in Neural Information Processing Systems*, 2014, vol. 27, pp. 2933–2941.

[57] A. Choromanska, M. Henaff, M. Mathieu, G. Ben Arous, and Y. LeCun, "The loss surfaces of multilayer networks," *J. Mach. Learn. Res. Work. Conf. Proc.*, vol. 38, pp. 192–204, 2015.

[58] M. Janzamin, H. Sedghi, and A. Anandkumar, "Generalization bounds for neural networks through tensor factorization," *arXiv:1506.08473*, 2015.

[59] B. D. Haeffele and R. Vidal, "Global optimality in tensor factorization, deep learning, and beyond," *arXiv:1506.07540*, 2015.

3.4 Holistic Image Reconstruction for Diffusion MRI

Diffusion MR images are six-dimensional: they are defined over 3D physical space and 3D diffusion-encoding space (q-space). Raw measurements are in most cases a version of the six-dimensional image that is Fourier-transformed along two of the three dimensions of physical space. Thus, image reconstruction is necessary to obtain the image from the raw measurements.

The number of sampled points and the signal-to-noise ratio are limited. Thus, the measurements have ambiguities, i.e. image reconstruction is an ill-posed problem. To resolve ambiguities, image priors are used. For example, out of several images that all correspond to raw measurements equally well, the image with fewer fluctuations (less noise) is preferred. Noise is measured and penalized via regularization terms by computing spatial derivatives of the reconstructed image.

Existing diffusion MRI reconstruction methods reconstruct each 2D image slice for each q-space coordinate separately. Thus, they do not use valuable information that is present in neighborhoods in the third spatial dimension and in 3D q-space.

We propose reconstructing the entire six-dimensional image jointly, and using regularization terms that denoise the image along all six dimensions, thus allowing to average out the noise at points that are neighbors in the six-dimensional space and to achieve better image quality.

Specifically, our regularization terms are based on derivatives in 3D Euclidean physical space, on 2D spherical shells in q-space, and on the 2D spherical orientation distribution function (see Section 2.4).

The operator in MRI reconstruction is nonlinear [Valkonen, 2014]. Therefore, we use an adaptation to nonlinear operators [Valkonen, 2014] of a primal-dual hybrid gradient optimization method [Pock et al., 2009].

To evaluate our approach, we use use data from the Human Connectome Project acquired with a custom-built high-resolution scanner as a reference and undersample it retrospectively to emulate a normal scanner. Results show that our approach yields images of superior quality compared to existing methods. Resolution is considerably higher in physical space and in q-space. This indicates that valuable information is contained in neighborhoods in the six-dimensional image space, and that it can be effectively used to improve image quality by employing appropriate regularization terms.

In other words, considering all six data dimensions jointly improves image reconstruction results in diffusion MRI.

The author of this dissertation contributed substantially to the content of the paper, in particular concerning parts of the idea, the code, experiments, and writing parts of the paper.

Accepted manuscript for the MICCAI Workshop on Computational Diffusion MRI. Published version: doi: 10.1007/978-3-319-28588-7_3.

Holistic Image Reconstruction for Diffusion MRI

Vladimir Golkov[1], Jorg M. Portegies[2], Antonij Golkov[3], Remco Duits[2,4], and Daniel Cremers[1]

[1] Department of Informatics, Technische Universität München, Garching, Germany
golkov@cs.tum.edu, cremers@tum.de
[2] Department of Mathematics and Computer Science,
Eindhoven University of Technology, Eindhoven, The Netherlands
r.duits@tue.nl, j.m.portegies@tue.nl
[3] Department of Mathematics, Augsburg University, Augsburg, Germany
antonij.golkov@student.uni-augsburg.de
[4] Department of Biomedical Engineering, Eindhoven University of Technology,
Eindhoven, The Netherlands

Abstract. Diffusion MRI provides unique information on the microarchitecture of biological tissues. One of the major challenges is finding a balance between image resolution, acquisition duration, noise level and image artifacts. Recent methods tackle this challenge by performing super-resolution reconstruction in image space or in diffusion space, regularization of the image data or of postprocessed data (such as the orientation distribution function, ODF) along different dimensions, and/or impose data-consistency in the original acquisition space. Each of these techniques has its own advantages; however, it is rare that even a few of them are combined. Here we present a holistic framework for diffusion MRI reconstruction that allows combining the advantages of all these techniques in a single reconstruction step. In proof-of-concept experiments, we demonstrate super-resolution on HARDI shells and in image space, regularization of the ODF and of the images in spatial and angular dimensions, and data consistency in the original acquisition space. Reconstruction quality is superior to standard reconstruction, demonstrating the feasibility of combining advanced techniques into one step.

1 Introduction

Among the main problems in diffusion MRI are scan duration limits (thus a limited amount of data), image resolution limits, noise, and image artifacts. In recent years, a host of methods [1,2,3,4,5,6,7,8,9] have been developed to tackle these issues. These methods use (simplified) assumptions about the data, such as specific types of smoothness / transform-domain sparsity / low-rankedness, specific types of data similarity between different coordinates in the 3-D space of diffusion directions and weightings (q-space), accurate or simplified image acquisition models, in some cases combined with a tailored acquisition strategy.

Super-resolution in diffusion MRI allows increasing the resolution beyond the hardware limits. In the original super-resolution techniques for diffusion

MRI [10,11], there is no coupling of different q-space coordinates, *i.e.* each q-space coordinate is treated independently without taking advantage of common structure. It is performed from image space to image space, independently of the image reconstruction step. Recent methods [12,13,14] couple q-space coordinates and use the original data-acquisition space but regularize only in the reconstruction space – not in additional spaces.

The proposed method allows leveraging complementary information by coupling in q-space, while imposing data consistency in the original space and balancing regularization in several arbitrary representations simultaneously.

The rest of the paper is organized as follows. In Section 2.1, we describe the data formation model. In Section 2.2, we introduce holistic reconstruction (raw data consistency, several regularization spaces, super-resolution reconstruction in image and diffusion space) and give details on sampling in acquisition and reconstruction spaces, the regularizers, the optimization procedure and its implementation. We show results of holistic super-resolution reconstruction after artifical subsampling of Human Connectome Project data in Section 3 and conclude with a discussion in Section 4.

2 Methods

2.1 Image Acquisition Model

The image is modeled on a domain $\Omega \times \mathbb{R}^3$, where $\Omega \subset \mathbb{R}^3$ represents the domain in image space, and dimensions four to six of $\Omega \times \mathbb{R}^3$ represent the space consisting of three-dimensional diffusion directions and diffusion weightings (*q-space*) for which discrete samples are acquired. A complex-valued diffusion MRI image ρ is a mapping

$$\rho : \Omega \times \mathbb{R}^3 \to \mathbb{C} \text{ given by} \tag{1}$$

$$(y, q) \mapsto \rho(y, q) = r(y, q) \exp(i\varphi(r, q)), \tag{2}$$

where r is the image magnitude and φ is the image phase at spatial coordinate $y \in \Omega$ and q-space coordinate $q \in \mathbb{R}^3$. Magnitude r and phase φ are mappings

$$r \colon \Omega \times \mathbb{R}^3 \to \mathbb{R}, \tag{3}$$

$$\varphi \colon \Omega \times \mathbb{R}^3 \to S^1. \tag{4}$$

These images are not acquired directly. Acquisition is performed in *k-space* (more precisely: in the joint six-dimensional (k, q)-*space*), after Fourier transform $\mathcal{F}_{1,2}$ along the spatial dimensions 1 and 2 of Ω. When sampled at N data points, the resulting data $d \in \mathbb{C}^N$ forms from r and φ according to

$$d = T(r, \varphi) + \varepsilon, \tag{5}$$

where ε is complex-valued i.i.d. Gaussian noise (thermal noise) and T is the encoding operator. The operator T composes r and φ pointwise into a complex-valued image via $C(r, \varphi) = r \odot \exp(i\varphi)$ where "$\odot$" is the pointwise product,

followed by a Fourier transform into (k, q)-space and discrete sampling S:

$$T(r, \varphi) = S\mathcal{F}_{1,2}C(r, \varphi), \text{ with} \tag{6}$$

$$S : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{C}^N \text{ given by} \tag{7}$$

$$(S\hat{\rho})_n = \int_{[-0.5,0.5]^3} \hat{\rho}(k_n + v, q_n)\, \mathrm{d}v, \tag{8}$$

where the $((k_n, q_n))_{n\in\{1,\ldots,N\}}$ are the sampling points in (k, q)-space. Details can be found in refs. [15,16].

2.2 Holistic Reconstruction

Our goal is to reconstruct the image magnitude r and phase φ from the acquired data d. In order to improve image quality, such a reconstruction should include state-of-the-art image processing methods, such as denoising, super-resolution reconstruction and orientation distribution function[5] (ODF) enhancement. Rather than performing this in a classical manner, where each step is performed separately, we couple all transformations and regularizers into a single optimization problem. This allows performing the entire reconstruction in a single step, while having full control over the balance between all regularizers simultaneously. Furthermore, this avoids data-consistency formulations in intermediate spaces, where the noise distribution is difficult to model correctly (*e.g.* Rician signal distribution and other cases) – our least squares data term penalizes deviation from k-space measurements, where noise is Gaussian, while still reconstructing and regularizing in arbitrary spaces. Finally, a holistic formulation allows regularizing in additional spaces other than the acquisition and the reconstruction space. This allows for example using information from the ODF (otherwise calculated independently at a later step) to inform the super-resolution reconstruction in image space.

In our proof-of-concept holistic reconstruction experiments, we treat the entire six-dimensional data jointly (rather than treating each q-space coordinate independently during image space reconstruction, followed by treating each image coordinate y independently during q-space-based processing) and combine the following concepts into a single optimization problem:

- Data consistency in the original (k, q)-space,
- Reconstruction into (y, q)-space with super-resolution in both the spatial and diffusional dimensions,
- Spatial regularization of (y, q)-space data,
- Angular regularization of (y, q)-space data by treating each q-space shell independently as functions on the (uncoupled) space $\mathbb{R}^3 \times S^2$ of positions and orientations,

[5] The ODF is a formalism that characterizes the strength of diffusion in different directions. It is defined formally below in Eq. (10).

– Spatial and angular regularization of the ODFs which implicitly correspond to the reconstructed (y, q)-space data by treating them as functions on the (uncoupled) space $\mathbb{R}^3 \times S^2$ of positions and orientations.

The general form of holistic reconstruction into (y, q)-space is

$$\arg\min_{r,\varphi} \frac{1}{2}\|T(r,\varphi) - d\|^2 + R(r), \tag{9}$$

where $R(r)$ is a sum of regularization terms which may or may not transform the image magnitude r into another space, such as ODFs, prior to penalizing non-regularity[6].

The "codomain" of our pipeline, *i.e.* the reconstruction space, can be extended into diffusion models, as in refs. [17,18]. These model-based methods can be complemented by our regularizers in additional spaces to yield a holistic framework.

Sampling Scheme in (k, q)-Space In order to verify the super-resolution reconstruction capability of our holistic reconstruction, we use data of uniquely high resolution from the Human Connectome Project [19,20,21,22,23,24,25,26], assuming it to be the ground truth underlying image data, and simulate a low-resolution k-space sampling of these ground truth images. In order to leverage complementarity of data in q-space, we employ a low-resolution (k, q)-space sampling scheme [13] in which high resolution components are left out alternatingly in vertical or horizontal image directions for different q-space coordinates. The q-space coordinates and the respective alternating vertical/horizontal k-space subsampling are shown in Figure 1, left. Both acquisition and reconstruction (see next paragraph) use the set of b-values $\mathcal{B} = \{0, 1000, 2000, 3000\}\,\mathrm{s/mm^2}$.

Super-Resolution Sampling Scheme in Reconstruction Space While data are artificially subsampled in k-space for the experiments, the reconstruction space is discretized such that the original high image resolution is reconstructed. While 270 q-space coordinates are sampled (Figure 1, left), 486 are reconstructed (Figure 1, right). This scheme achieves a super-resolution reconstruction in image and diffusion space.

Regularization We will regularize several images of the type $U \in \mathbb{H}^2(\mathbb{R}^3 \times S^2)$, namely the ODF and the spherical shells in q-space.

The ODF [27] for image r at image location $y \in \Omega$ and direction $n \in S^2$ can be calculated as

$$\mathrm{ODF}(r)(y, n) = \frac{1}{Z_\kappa} \int_0^\infty (\mathcal{F}_{4,5,6} r)(y, pn) p^\kappa \,\mathrm{d}p \tag{10}$$

[6] The precise formula that we use for $R(r)$ will follow later in Eq. (12).

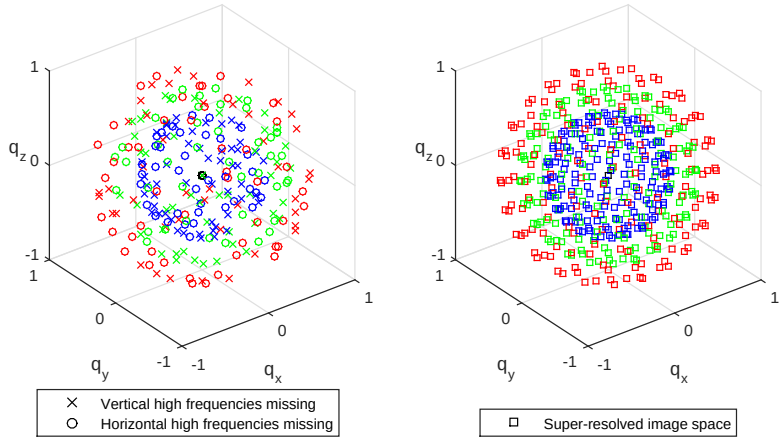


Fig. 1. Sampling scheme in q-space during acquisition (left) and reconstruction (right). The acquired data have alternating artificial subsampling in vertical/horizontal high frequencies in k-space. All high frequencies for all images are reconstructed. Colors encode the b-value: $\mathcal{B} = \{0, 1000, 2000, 3000\}\,\mathrm{s/mm^2}$.

with the usual choice $\kappa = 2$, where Z_κ is a normalization constant and $\mathcal{F}_{4,5,6}$ is the Fourier transform along the diffusion dimensions four to six that calculates the diffusion propagator from q-space data in an idealized setting [28].

Let G_b be the linear operator that extracts a spherical q-space shell at a given b-value (diffusion weighting) from r:

$$\left(G_b(r)\right)(y,n) = r(y, \sqrt{b}n). \tag{11}$$

In a proof-of-concept holistic reconstruction, the shells and the ODFs are regularized in the uncoupled space $\mathbb{R}^3 \times S^2$ of positions and orientations as follows:

$$\begin{aligned} R(r) = \sum_{b\in\mathcal{B}} \int_{\mathbb{R}^3\times S^2} & \alpha_1 \|\nabla_y G_b(r)(y,n)\|^2 \\ - \alpha_2 \big\langle G_b(r)(y,n), \Delta_{S^2} G_b(r)(y,n)\big\rangle + \alpha_3 |\Delta_{S^2} G_b(r)(y,n)|^2 \,\mathrm{d}y\,\mathrm{d}\sigma(n) \\ + \int_{\mathbb{R}^3\times S^2} & \alpha_4 \|\nabla_y \mathrm{ODF}(r)(y,n)\|^2 \\ - \alpha_5 \big\langle \mathrm{ODF}(r)(y,n), \Delta_{S^2}\mathrm{ODF}(r)(y,n)\big\rangle + \alpha_6 |\Delta_{S^2}\mathrm{ODF}(r)(y,n)|^2 \,\mathrm{d}y\,\mathrm{d}\sigma(n), \end{aligned} \tag{12}$$

where $\mathcal{B}$ is the set of reconstructed b-values, the α_i are regularization parameters, σ is the usual surface measure on S^2, Δ_{S^2} is the Laplace–Beltrami operator on the sphere and the negative inner products correspond to first-order regularization according to $\int -\langle U, \Delta U\rangle = \int \|\nabla U\|^2$ (*i.e.* Green's identity with vanishing boundary conditions as we assume our functions U to vanish at the boundary).

Defining appropriate inner products on the space $\mathbb{H}^2(\mathbb{R}^3 \times S^2) \ni U, V$ and on $\mathbb{H}^1(\mathbb{R}^3 \times S^2, \mathbb{R}^3) \ni \nabla_y U, \nabla_y V$ as

$$\langle U, V \rangle = \int_{\mathbb{R}^3 \times S^2} U(y,n) V(y,n) \,\mathrm{d}y \,\mathrm{d}\sigma(n), \tag{13}$$

$$\langle \nabla_y U, \nabla_y V \rangle = \sum_{i \in \{1,2,3\}} \int_{\mathbb{R}^3 \times S^2} \left(\nabla_y U(y,n)\right)_i \left(\nabla_y V(y,n)\right)_i \,\mathrm{d}y \,\mathrm{d}\sigma(n), \tag{14}$$

and using the induced norms, we can rewrite the problem (9,12) as follows:

$$\begin{aligned} &\min_{r,\varphi} \frac{1}{2}\|T(r,\varphi) - d\|^2 \\ &+ \sum_{b \in \mathcal{B}} \alpha_1 \|\nabla_y G_b(r)\|^2 - \alpha_2 \langle G_b(r), \Delta_{S^2} G_b(r) \rangle + \alpha_3 \|\Delta_{S^2} G_b(r)\|^2 \\ &\quad + \alpha_4 \|\nabla_y \mathrm{ODF}(r)\|^2 - \alpha_5 \langle \mathrm{ODF}(r), \Delta_{S^2} \mathrm{ODF}(r) \rangle + \alpha_6 \|\Delta_{S^2} \mathrm{ODF}(r)\|^2. \end{aligned} \tag{15}$$

Reformulations To obtain a convenient min-max form with simpler expressions within the norms, we shall use the identity:

$$\|\hat{x}\|^2 = \sup_{\hat{y}} \langle \hat{x}, \hat{y} \rangle - \frac{1}{4}\|\hat{y}\|^2, \tag{16}$$

obtained by taking the convex biconjugate and completing the square. This reformulation introduces dual variables $\hat{y}$.

Optimization Procedure Our optimization problem (15) can be rewritten as a min-max problem of the form

$$\min_x \max_y G(x) + \langle K(x), y \rangle - F^*(y) \tag{17}$$

with convex G, F^* and a nonlinear K, which can be solved with the modified primal-dual hybrid gradient method for nonlinear K [29,30,15]:

$$x^{i+1} := (I + \tau \partial G)^{-1}(x^i - \tau [\nabla K(x^i)]^* y^i), \tag{18a}$$

$$x_\omega^{i+1} := x^{i+1} + \omega(x^{i+1} - x^i), \tag{18b}$$

$$y^{i+1} := (I + \sigma \partial F^*)^{-1}(y^i + \sigma K(x_\omega^{i+1})), \tag{18c}$$

where ∂f represents the subdifferential of a function f, defined as

$$\partial f(x_0) = \left\{ v \mid f(x) - f(x_0) \geq \langle v, x - x_0 \rangle \ \forall x \in \mathrm{dom} f \right\}, \tag{19}$$

and $(I + \lambda \partial f)^{-1}$ is the resolvent of the subdifferential, corresponding to the proximal operator [31]:

$$(I + \lambda \partial f)^{-1} x = \mathrm{prox}_{\lambda f}(x) = \arg\min_z f(z) + \frac{1}{2\lambda}\|x - z\|^2. \tag{20}$$

The algorithm (18) has been applied [15] with the operator $T(r,\varphi)$ to non-diffusion MRI, and with another operator to diffusion MRI. The author announces combining $T(r,\varphi)$ with direct reconstruction of the diffusion tensor in a future study, while we present an application of $T(r,\varphi)$ to reconstruction in image×diffusion space.

By rewriting all five norms in our problem (15) using the identity (16), we obtain the min-max form

$$\begin{aligned}\min_{r,\varphi}\max_{\lambda,(\zeta_b)_{b\in\mathcal{B}},(\eta_b)_{b\in\mathcal{B}},\xi,\nu} &\big\langle T(r,\varphi),\lambda\big\rangle - \big\langle d,\lambda\big\rangle - \frac{1}{2}\|\lambda\|^2 \\ &+\sum_{b\in\mathcal{B}}\alpha_1\left(\big\langle\nabla_y G_b(r),\zeta_b\big\rangle - \frac{1}{4}\|\zeta_b\|^2\right) \\ -\alpha_2\big\langle G_b(r),\Delta_{S^2}G_b(r)\big\rangle &+\alpha_3\left(\big\langle\Delta_{S^2}G_b(r),\eta_b\big\rangle - \frac{1}{4}\|\eta_b\|^2\right) \\ &+\alpha_4\left(\big\langle\nabla_y \mathrm{ODF}(r),\xi\big\rangle - \frac{1}{4}\|\xi\|^2\right) \\ -\alpha_5\big\langle \mathrm{ODF}(r),\Delta_{S^2}\mathrm{ODF}(r)\big\rangle &+\alpha_6\left(\big\langle\Delta_{S^2}\mathrm{ODF}(r),\nu\big\rangle - \frac{1}{4}\|\nu\|^2\right).\end{aligned} \tag{21}$$

The primal variables are $x=(r,\varphi)$ and the dual ones are $y=(\lambda,(\zeta_b)_{b\in\mathcal{B}},(\eta_b)_{b\in\mathcal{B}},\xi,\nu)$, where for example η_b denotes the dual variable associated to $\|\Delta_{S^2}G_b(r)\|^2$. This can be regrouped into the standard form (17) as follows:

$$\begin{aligned} G(x) &= \sum_{b\in\mathcal{B}} -\alpha_2\big\langle G_b(r),\Delta_{S^2}G_b(r)\big\rangle - \alpha_5\big\langle \mathrm{ODF}(r),\Delta_{S^2}\mathrm{ODF}(r)\big\rangle, \\ \big\langle K(x),y\big\rangle &= \big\langle T(r,\varphi),\lambda\big\rangle + \sum_{b\in\mathcal{B}}\alpha_1\big\langle\nabla_y G_b(r),\zeta_b\big\rangle + \alpha_3\big\langle\Delta_{S^2}G_b(r),\eta_b\big\rangle \\ &\quad + \alpha_4\big\langle\nabla_y\mathrm{ODF}(r),\xi\big\rangle + \alpha_6\big\langle\Delta_{S^2}\mathrm{ODF}(r),\nu\big\rangle, \\ \pm F^*(y) &= \pm\big\langle d,\lambda\big\rangle \pm \frac{1}{2}\|\lambda\|^2 \\ &\quad \pm\frac{1}{4}\left(\sum_{b\in\mathcal{B}}\alpha_1\|\zeta_b\|^2+\alpha_3\|\eta_b\|^2+\alpha_4\|\xi\|^2+\alpha_6\|\nu\|^2\right). \end{aligned} \tag{22}$$

For the implementation of algorithm (18), we calculate the proximal operators [31]:

$$(I+\tau\partial G)^{-1}x = (I+\tau(Q+Q^*))^{-1}x, \tag{23}$$

$$Q = \sum_{b\in\mathcal{B}} G_b^*\Delta_{S^2}G_b + \mathrm{ODF}^*\Delta_{S^2}\mathrm{ODF}, \tag{24}$$

$$(I+\sigma\partial F^*)^{-1}y = \begin{pmatrix} (\lambda-\sigma d)/(\sigma+1) \\ (\zeta_b/(1+\alpha_1\sigma/2))_{b\in\mathcal{B}} \\ (\eta_b/(1+\alpha_3\sigma/2))_{b\in\mathcal{B}} \\ \xi/(1+\alpha_4\sigma/2) \\ \nu/(1+\alpha_6\sigma/2) \end{pmatrix}. \tag{25}$$

Calculating $[\nabla K(x^i)]^*$ (18) for the nonlinear part $T(r,\varphi)$ (22) yields

$$[\nabla T(r,\varphi)]^* = (S\mathcal{F}_{1,2}[\nabla C(r,\varphi)])^* = [\nabla C(r,\varphi)]^* \mathcal{F}_{1,2}^* S^*, \tag{26}$$

$$[\nabla C(r,\varphi)]^* \hat{\lambda} = \begin{pmatrix} \Re(\hat{\lambda})\cos(\varphi) + \Im(\hat{\lambda})\sin(\varphi) \\ r(\Im(\hat{\lambda})\cos(\varphi) - \Re(\hat{\lambda})\sin(\varphi)) \end{pmatrix}. \tag{27}$$

Unbounded ODF Operator When writing out the Fourier transform $\mathcal{F}_{4,5,6}$ over $Q \in \mathbb{R}^3$, the ODF (10) contains the diverging term $\exp(-i\langle pn, Q\rangle)p^2$. Thus, the ODF operator is unbounded. Since an adjoint is required for the algorithm (18), the operator can be made bounded in the infinite-dimensional setting by including a Gaussian damping factor $\exp(-p^2/\varsigma^2)$ as a mollifier. The operator bound of the discrete operator depends on the discretization, and in our discretization scheme no mollifier was needed in practice.

Implementation Details The operators $\mathcal{F}_{1,2}$, S (6), ODF (10), G_b (11), ∇_y and Δ_{S^2} are linear. In the implementation, the spaces in which acquisition, regularization and reconstruction take place are discretized and thus the operators can be written as matrices. We obtain these matrices explicitly. Where not evident, an operator matrix is computed by applying the operator to all standard basis vectors of the discretized space, yielding the columns of the matrix. For pointwise operators, we compute and store repeating coefficients only once. When computing $[\nabla K(\cdot)]^*$ and $K(\cdot)$ in the algorithm (18), having the operator matrices explicitly has the advantages of rapid computation by matrix multiplication and easy computation of the adjoint operators. Besides, in the discretized setting, the ODF operator is not unbounded anymore and thus has an adjoint, as required by the algorithm. The norm $\|[\nabla K(\cdot)]^*\|$ of the operator $[\nabla K(\cdot)]^*$ explodes as the discretization becomes finer, but in our discretization settings there was no need to include a Gaussian mollifier in (10).

3 Results

Figure 2 shows the high-resolution "ground truth" image data from the Human Connectome Project (Figure 2, left) alongside the results of two reconstruction methods applied to the same data that has been artificially subsampled according to the sampling scheme in (k,q)-space described in section 2.2 and illustrated in Figure 1, left. This artificial subsampling procedure emulates a clinical setting where resolution is considerably lower than in the Human Connectome Project, and enables a comparison to this exceptionally high-resolution ground truth data. The two compared reconstruction methods are standard reconstruction ($\mathcal{F}_{1,2}$-transformed subsampled data; Figure 2, middle) and holistic image reconstruction (as described above, with super-resolution sampling as in Figure 1, right; results in Figure 2, right).

The employed parameters were $\alpha_1 = 0.3, \alpha_2 = 0.1, \alpha_3 = 0.1, \alpha_4 = 0.01, \alpha_5 = 0.3, \alpha_6 = 0.01$.

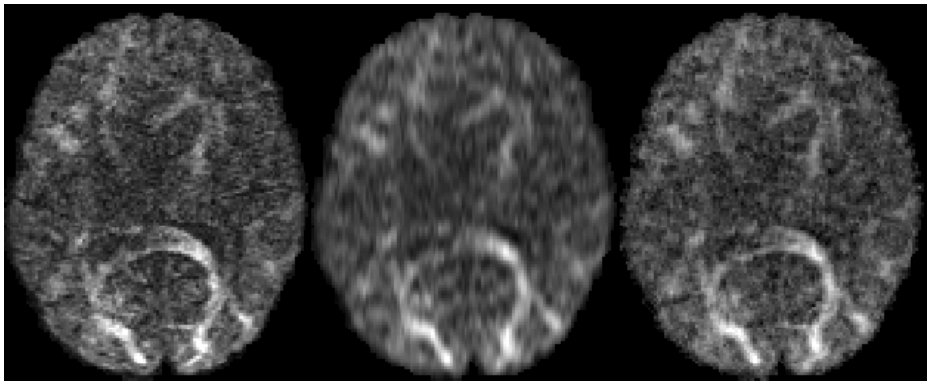


Fig. 2. High-resolution ground truth (left), standard reconstruction (middle), holistic super-resolution reconstruction (right).

Holistic image reconstruction demonstrates considerably more detail than standard reconstruction. While standard reconstruction results have a visibly lower resolution, holistic reconstruction retrieves details that are present in the ground truth data due to its super-resolution scheme and regularization in image and diffusion space.

4 Discussion

The results of holistic reconstruction demonstrate considerably more detail than the standard reconstruction.

Among the numerous advanced diffusion MRI reconstruction methods existing in literature, many methods perform denoising, missing data reconstruction (q-space compressed sensing), enhancement, etc. as an intermediate post-processing step *after* image-space reconstruction. However, standard-reconstructed images can contain artifacts, intensity bias (*e.g.* Rician or more complicated), and irretrievably discard some parts of information present in the raw k-space data. Imposing data consistency in reconstructed image space can lead to these errors being propagated on into subsequent data processing steps, and/or introduce less tractable bias-correction terms. There is strong evidence that one-step pipelines are better than multi-step pipelines due to information loss in intermediate steps [32]. Particularly, imposing data consistency on the original raw data in k-space yields improved results compared to multi-step processing [13]. The holistic reconstruction framework presented herein allows imposing data consistency in the original data acquisition space, while also including regularization in several spaces (such as (y, q)-space and "(y,ODF)-space"), and reconstructing into an arbitrary space, including super-resolution reconstruction sampling.

Super-resolution methods are beneficial for diffusion MRI due to their capability to exceed hardware limitations on resolution. In the presented holistic

reconstruction framework, super-resolution is performed in image space *and* diffusion space simultaneously, cf. Figure 1. At the same time, data consistency in the original space and regularizations in additional spaces are incorporated in a straightforward manner.

Many competing regularizers in different spaces exist in recent literature. Each of them incorporates certain assumptions and improves data quality at certain intermediate regularization strengths. Regularizations in different spaces can be combined into one procedure (including true data consistency and super-resolution) using holistic image reconstruction.

Reconstruction can be performed jointly with motion and distortion correction [5] in the future.

Finally, our choice of priors in (15) was based on isotropic Laplacians over the spatial and angular part, and as such defined on $\mathbb{R}^3 \times S^2$. Including anisotropies and alignment modeling in a crossing-preserving way via the *coupled* space $\mathbb{R}^3 \rtimes S^2 = SE(3)/(\{\mathbf{0} \times SO(2)\})$, see [1] Thm. 2, and [33], is expected to give better results in future work.

Acknowledgements We thank Evgeny Strekalovskiy (TU München) for helpful discussions. V.G. is supported by the Deutsche Telekom Foundation. The research leading to the results of this article has received funding from the European Research Council under the ECs 7th Framework Programme (FP7/2007-2014) / ERC grant agr. no. 335555. Data were provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

References

1. Duits, R., Franken, E.: Left-invariant diffusions on the space of positions and orientations and their application to crossing-preserving smoothing of HARDI images. International Journal of Computer Vision 92(3), 231–264 (2010) 1, 10
2. Lustig, M., Donoho, D., Pauly, J.M.: Sparse MRI: The application of compressed sensing for rapid MR imaging. Magnetic Resonance in Medicine 58(6), 1182–95 (2007) 1
3. Khare, K., Hardy, C.J., King, K.F., Turski, P.A., Marinelli, L.: Accelerated MR imaging using compressive sensing with no free parameters. Magnetic Resonance in Medicine 68(5), 1450–7 (2012) 1
4. Paquette, M., Merlet, S., Gilbert, G., Deriche, R., Descoteaux, M.: Comparison of sampling strategies and sparsifying transforms to improve compressed sensing diffusion spectrum imaging. Magnetic Resonance in Medicine 73, 401–416 (2015) 1
5. Tao, S., Trzasko, J.D., Shu, Y., Huston, J., Bernstein, M.A.: Integrated image reconstruction and gradient nonlinearity correction. Magnetic Resonance in Medicine early view, (2014) 1, 10

6. Feng, L., Grimm, R., Block, K.T., Chandarana, H., Kim, S., Xu, J., Axel, L., Sodickson, D.K., Otazo, R.: Golden-angle radial sparse parallel MRI: Combination of compressed sensing, parallel imaging, and golden-angle radial sampling for fast and flexible dynamic volumetric MRI. Magnetic Resonance in Medicine 72, 707–717 (2014) 1
7. Cauley, S.F., Xi, Y., Bilgic, B., Xia, J., Adalsteinsson, E., Balakrishnan, V., Wald, L.L., Setsompop, K.: Fast reconstruction for multichannel compressed sensing using a hierarchically semiseparable solver. Magnetic Resonance in Medicine 73, 1034–1040 (2015) 1
8. Mani, M., Jacob, M., Guidon, A., Magnotta, V., Zhong, J.: Acceleration of high angular and spatial resolution diffusion imaging using compressed sensing with multichannel spiral data. Magnetic Resonance in Medicine 73, 126–138 (2015) 1
9. Rathi, Y., Michailovich, O., Laun, F., Setsompop, K., Grant, P.E., Westin, C.F.: Multi-shell diffusion signal recovery from sparse measurements. Medical Image Analysis 18(7), 1143–1156 (2014) 1
10. Scherrer, B., Gholipour, A., Warfield, S.K.: Super-resolution reconstruction to increase the spatial resolution of diffusion weighted images from orthogonal anisotropic acquisitions. Medical Image Analysis 16(7), 1465–76 (2012) 2
11. Poot, D.H.J., Jeurissen, B., Bastiaensen, Y., Veraart, J., Van Hecke, W., Parizel, P.M., Sijbers, J.: Super-resolution for multislice diffusion tensor imaging. Magnetic Resonance in Medicine 69(1), 103–13 (2013) 2
12. Tobisch, A., Neher, P.F., Rowe, M.C., Maier-Hein, K.H., Zhang, H.: Model-based super-resolution of diffusion MRI. In Schultz, T., Nedjati-Gilani, G., Venkataraman, A., O'Donnell, L., Panagiotaki, E., eds.: Computational Diffusion MRI and Brain Connectivity, MICCAI Workshops 2013. Mathematics and Visualization. Springer International Publishing (2014) 25–34 2
13. Golkov, V., Sperl, J.I., Menzel, M.I., Sprenger, T., Tan, E.T., Marinelli, L., Hardy, C.J., Haase, A., Cremers, D.: Joint super-resolution using only one anisotropic low-resolution image per q-space coordinate. In O'Donnell, L., Nedjati-Gilani, G., Rathi, Y., Reisert, M., Schneider, T., eds.: Computational Diffusion MRI, MICCAI Workshop 2014. Springer International Publishing (2015) 181–191 2, 4, 9
14. Van Steenkiste, G., Jeurissen, B., Veraart, J., den Dekker, A.J., Parizel, P.M., Poot, D.H.J., Sijbers, J.: Super-resolution reconstruction of diffusion parameters from diffusion-weighted images with different slice orientations. Magnetic Resonance in Medicine early view, (2015) 2
15. Valkonen, T.: A primal-dual hybrid gradient method for non-linear operators with applications to MRI. Inverse Problems 30(5), 055012 (2014) 3, 6, 7
16. Brown, R.W., Cheng, Y.C.N., Haacke, E.M., Thompson, M.R., Venkatesan, R.: Magnetic Resonance Imaging: Physical Principles and Sequence Design. 2nd edn. Wiley-Blackwell (2014) 3
17. Welsh, C.L., Dibella, E.V.R., Adluru, G., Hsu, E.W.: Model-based reconstruction of undersampled diffusion tensor k-space data. Magnetic Resonance in Medicine 70(2), 429–40 (2013) 4
18. Valkonen, T., Bredies, K., Knoll, F.: TGV for diffusion tensors: A comparison of fidelity functions. Journal of Inverse and Ill-Posed Problems 21(3), 355–377 (2013) 4
19. Van Essen, D.C., Smith, S.M., Barch, D.M., Behrens, T.E.J., Yacoub, E., Ugurbil, K.: The WU-Minn Human Connectome Project: An overview. NeuroImage 80, 62–79 (2013) 4

20. Feinberg, D.A., Moeller, S., Smith, S.M., Auerbach, E., Ramanna, S., Glasser, M.F., Miller, K.L., Ugurbil, K., Yacoub, E.: Multiplexed echo planar imaging for sub-second whole brain fmri and fast diffusion imaging. PLoS ONE 5(12), (2010) 4
21. Setsompop, K., Cohen-Adad, J., Gagoski, B.A., Raij, T., Yendiki, A., Keil, B., Wedeen, V.J., Wald, L.L.: Improving diffusion MRI using simultaneous multi-slice echo planar imaging. NeuroImage 63(1), 569–80 (2012) 4
22. Xu, J., Li, K., Smith, R.A., Waterton, J.C., Zhao, P., Chen, H., Does, M.D., Manning, H.C., Gore, J.C.: Characterizing tumor response to chemotherapy at various length scales using temporal diffusion spectroscopy. PloS one 7(7), e41714 (2012) 4
23. Sotiropoulos, S.N., Jbabdi, S., Xu, J., Andersson, J.L., Moeller, S., Auerbach, E.J., Glasser, M.F., Hernandez, M., Sapiro, G., Jenkinson, M., Feinberg, D.a., Yacoub, E., Lenglet, C., Van Essen, D.C., Ugurbil, K., Behrens, T.E.J.: Advances in diffusion MRI acquisition and processing in the Human Connectome Project. NeuroImage 80, 125–43 (2013) 4
24. Glasser, M.F., Sotiropoulos, S.N., Wilson, J.A., Coalson, T.S., Fischl, B., Andersson, J.L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J.R., Van Essen, D.C., Jenkinson, M.: The minimal preprocessing pipelines for the Human Connectome Project. NeuroImage 80, 105–124 (2013) 4
25. Jenkinson, M., Beckmann, C.F., Behrens, T.E.J., Woolrich, M.W., Smith, S.M.: FSL. NeuroImage 62(2), 782–790 (2012) 4
26. Fischl, B.: FreeSurfer. NeuroImage 62(2), 774–781 (2012) 4
27. Lin, C.P., Wedeen, V.J., Chen, J.H., Yao, C., Tseng, W.Y.I.: Validation of diffusion spectrum magnetic resonance imaging with manganese-enhanced rat optic tracts and ex vivo phantoms. NeuroImage 19, 482–495 (2003) 4
28. Stejskal, E.O.: Use of spin echoes in a pulsed magnetic-field gradient to study anisotropic, restricted diffusion and flow. The Journal of Chemical Physics 43(10), 3597–3603 (1965) 5
29. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the MumfordShah functional. In: 2009 IEEE 12th International Conference on Computer Vision (ICCV). Number 813396, IEEE (2009) 1133–1140 6
30. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision 40(1), 120–145 (2011) 6
31. Parikh, N., Boyd, S.: Proximal Algorithms. Foundations and Trends in Optimization 1, 123–231 (2014) 6, 7
32. Golkov, V., Dosovitskiy, A., Sämann, P., Sperl, J.I., Sprenger, T., Czisch, M., Menzel, M.I., Gómez, P.A., Haase, A., Brox, T., Cremers, D.: q-Space deep learning for twelve-fold shorter and model-free diffusion MRI scans. In: MICCAI. (2015) 9
33. Portegies, J.M., Fick, R.H.J., Sanguinetti, G.R., Meesters, S.P.L., Girard, G., Duits, R.: Improving fiber alignment in HARDI by combining contextual PDE flow with constrained spherical deconvolution. Submitted to PLOS ONE. See http://bmia.bmt.tue.nl/people/RDuits/mainJorg.pdf, available on ArXiv 2015, (2015) 10

4 Conclusions

We used the techniques described above in Section 1.3.4 to analyze [Golkov et al., 2013b] existing state-of-the-art processing methods for diffusion MRI and identify important gaps in the set of existing methods, cf. Fig. 1 in [Golkov et al., 2016a] (included above as Section 3.3). We used emerging families of methods to fill these gaps. On one hand, we used convolutional neural networks and obtained results that are twelve-fold better [Golkov et al., 2015a, Golkov et al., 2016a] than in previous state-of-the-art methods for diffusion MRI. On the other hand, we used state-of-the-art primal-dual optimization algorithms to successfully leverage synergies along all six dimensions of diffusion MRI scans to improve image quality [Golkov et al., 2014b, Golkov et al., 2014c, Golkov et al., 2014d, Golkov et al., 2015b, Naeyaert et al., 2020, Naeyaert et al., 2021].

Subsequently, when another family of neural-network architectures – namely rotation-equivariant deep learning for 2D and 3D data – emerged and started outperforming other methods, we systematized it [Della Libera et al., 2019] and adapted it to 6D diffusion MRI data [Müller et al., 2021a, Müller et al., 2021b], outperforming previous methods on diffusion MRI.

We also summarized important properties of methods in a tabular form [Aljalbout et al., 2018, Swazinna et al., 2019] as described in Section 1.3.5. This allowed us to create novel general-purpose clustering methods [Aljalbout et al., 2018], also by using [Haeusser et al., 2018] patterns of neural network architecture design from other ML tasks; and to address specific use cases for diffusion MRI in a targeted manner, such as weakly supervised localization [Golkov et al., 2018a] and various kinds of anomaly detection [Golkov et al., 2016c, Golkov et al., 2018b, Vasilev et al., 2019, Swazinna et al., 2019]. Thus, we explored various ML tasks, as described in Section 2.3.5.

Moreover, we advanced the research in important areas of biology, particularly in relationship to geometry and deep learning. We proposed novel representations of information about the three-dimensional structure of molecules such as proteins [Golkov et al., 2020b] and RNA [Do et al., 2018]. Our representations make it easy for neural networks to learn to extract relevant features. Particularly, our representations directly expose physical properties of the molecule (for example the electron density and the electrostatic potential field) that are relevant for the task at hand; they explicitly distinguish different amino acid types via a highly untangled three-dimensional multi-channel encoding, so that the neural network does not have to learn to seek subtle cues in order to discover that information; and they collect all the information about each structural motif in one place in a way that allows to easily use deep learning that is equivariant under displacements of structural motifs, i.e. detects them and uses information about them in a consistent, reliable manner. Some of our representations of molecular structure [Golkov et al., 2020b] are directly compatible with state-of-the-art neural networks that are invariant under rotations of the molecules, whereas our other representations [Do et al., 2018] are themselves invariant under such rotations.

We proposed the usage of rich information about the evolution of proteins

as input to neural networks, so that the network training ensures optimal extraction of relevant information [Golkov et al., 2016b]. We also analyzed hypothetical biologically meaningful extracted features, which allowed us to directly choose an according neural-network architecture that outperformed state-of-the-art methods [Golkov et al., 2016b]. One method of ours predicts the structure of proteins from their sequence and evolutionary history [Golkov et al., 2016b]. Our other methods predict the function of RNA from its secondary structure (base pairing) [Do et al., 2018], and the function of small molecules and proteins from their three-dimensional structure [Golkov et al., 2020b]. Overall, the usage of appropriate datasets and data representations and end-to-end training of appropriate network architectures ensured optimal results. Results were promising in proof-of-concept experiments, and outperformed the state of the art in our full experiments.

Cost functions based on the receiver operating characteristic are rarely used in deep learning. However, we identified various theoretical reasons for the superiority of such cost functions over the default ones when dealing with virtual screening, i.e. the prediction of the function of small molecules in pharmacology [Golkov et al., 2020a]. These reasons are related to the special circumstances present in virtual screening: severe class imbalance, high decision thresholds, and a lack of ground truth labels in some datasets. We also developed new cost functions from this family of cost functions to further address the problem of high decision thresholds, as well as new training schemes. Our methods outperform standard deep learning approaches. These theoretical and empirical insights demonstrate that a careful, application-specific choice of loss functions can be worthwhile even in common tasks such as classification.

We also successfully applied deep learning to optical flow estimation [Dosovitskiy et al., 2015], where pairs of images are mapped to vector fields, speech synthesis [Fabbro et al., 2020], X-ray image analysis [Pasa et al., 2019], evolutionary algorithms [Schuchardt et al., 2019], and we systematized regularization methods in deep learning [Kukačka et al., 2017].

Overall, by designing methods as described in Sections 1.3 and 2.3.4, adapting existing state-of-the-art methods to new problems, or exploring new ML tasks as described in Section 2.3.5, good results can be achieved in many cases.

The successes and methods so far open new avenues for addressing other unsolved problems and complex challenges in the years to come.

References

[Adler and Öktem, 2017] Adler, J. and Öktem, O. (2017). Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007. (ˆ9)

[Aja-Fernández et al., 2009] Aja-Fernández, S., Tristán-Vega, A., and Alberola-López, C. (2009). Noise estimation in single-and multiple-coil magnetic resonance data based on statistical models. *Magnetic resonance imaging*, 27(10):1397–1409. (ˆ11)

[Aljalbout et al., 2018] Aljalbout, E., Golkov, V., Siddiqui, Y., Strobel, M., and Cremers, D. (2018). Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648*. (ˆ12, 32, 91)

[Amos and Kolter, 2017] Amos, B. and Kolter, J. Z. (2017). OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR. (ˆ9)

[Andrychowicz et al., 2016] Andrychowicz, M., Denil, M., Colmenarejo, S. G., Hoffman, M. W., Pfau, D., Schaul, T., and de Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3981–3989. (ˆ9)

[Bau et al., 2017] Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549. (ˆ35)

[Bernstein and Craig, 2010] Bernstein, H. J. and Craig, P. A. (2010). Efficient molecular surface rendering by linear-time pseudo-gaussian approximation to lee–richards surfaces (pgalrs). *Journal of applied crystallography*, 43(2):356–361. (ˆ51)

[Bernstein et al., 2004] Bernstein, M. A., King, K. F., and Zhou, X. J. (2004). *Handbook of MRI pulse sequences*. Elsevier. (ˆ17)

[Bock et al., 2013] Bock, L. V., Blau, C., Schröder, G. F., Davydov, I. I., Fischer, N., Stark, H., Rodnina, M. V., Vaiana, A. C., and Grubmüller, H. (2013). Energy barriers and driving forces in trna translocation through the ribosome. *Nature structural & molecular biology*, 20(12):1390–1396. (ˆ19)

[Bredies and Lorenz, 2011] Bredies, K. and Lorenz, D. (2011). *Mathematische Bildverarbeitung*. Springer. (ˆ1, 27)

[Brown et al., 2014] Brown, R. W., Cheng, Y.-C. N., Haacke, E. M., Thompson, M. R., and Venkatesan, R. (2014). *Magnetic resonance imaging: physical principles and sequence design*. John Wiley & Sons. (ˆ15)

[Cech and Steitz, 2014] Cech, T. R. and Steitz, J. A. (2014). The noncoding rna revolution—trashing old rules to forge new ones. *Cell*, 157(1):77–94. (ˆ19)

[Chen and He, 2020] Chen, X. and He, K. (2020). Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*. (ˆ33)

[Cohen et al., 2019] Cohen, T. S., Geiger, M., and Weiler, M. (2019). A general theory of equivariant CNNs on homogeneous spaces. *Advances In Neural Information Processing Systems 32 (Nips 2019)*, 32(CONF). (ˆ26)

[Couprie, 2011] Couprie, C. (2011). *Graph-based variational optimization and applications in computer vision*. PhD thesis, Paris Est. (ˆ11)

[Della Libera et al., 2019] Della Libera, L., Golkov, V., Zhu, Y., Mielke, A., and Cremers, D. (2019). Deep learning for 2D and 3D rotatable data: An overview of methods. *arXiv preprint arXiv:1910.14594*. (ˆ11, 12, 26, 91)

[Dieleman et al., 2016] Dieleman, S., De Fauw, J., and Kavukcuoglu, K. (2016). Exploiting cyclic symmetry in convolutional neural networks. In *ICML*, pages 1889–1898. (ˆ25)

[Do et al., 2018] Do, B. T., Golkov, V., Gürel, G. E., and Cremers, D. (2018). Precursor microRNA identification using deep convolutional neural networks. In *bioRxiv preprint*. (ˆ91, 92)

[do Carmo, 2015] do Carmo, M. P. (2015). *Geometria riemanniana*. Instituto de Matemática Pura e Aplicada, 5 edition. (ˆ38)

[Dosovitskiy and Brox, 2016] Dosovitskiy, A. and Brox, T. (2016). Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4829–4837. (ˆ35)

[Dosovitskiy et al., 2015] Dosovitskiy, A., Fischer, P., Ilg, E., Haeusser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet: Learning Optical Flow with Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*. (ˆ92)

[Du et al., 2020] Du, M., Yang, F., Zou, N., and Hu, X. (2020). Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems*. (ˆ1)

[Duran et al., 2016] Duran, J., Moeller, M., Sbert, C., and Cremers, D. (2016). Collaborative total variation: a general framework for vectorial tv models. *SIAM Journal on Imaging Sciences*, 9(1):116–151. (ˆ11)

[Fabbro et al., 2020] Fabbro, G., Golkov, V., Kemp, T., and Cremers, D. (2020). Speech synthesis and control using differentiable DSP. *arXiv preprint arXiv:2010.15084*. (ˆ92)

[Fukushima, 1980] Fukushima, K. (1980). Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202. (ˆ29)

[Ganin and Lempitsky, 2015] Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR. (ˆ33)

[Gasteiger and Marsili, 1978] Gasteiger, J. and Marsili, M. (1978). A new model for calculating atomic charges in molecules. *Tetrahedron Letters*, 19(34):3181–3184. (ˆ51)

[Gidel et al., 2019] Gidel, G., Berard, H., Vignoud, G., Vincent, P., and Lacoste-Julien, S. (2019). A variational inequality perspective on generative adversarial networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. (ˆ34)

[Glorot et al., 2011] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings. (ˆ29)

[Goldluecke et al., 2012] Goldluecke, B., Strekalovskiy, E., and Cremers, D. (2012). The natural vectorial total variation which arises from geometric measure theory. *SIAM Journal on Imaging Sciences*, 5(2):537–563. (ˆ11)

[Golkov et al., 2020a] Golkov, V., Becker, A., Plop, D. T., Čuturilo, D., Davoudi, N., Mendenhall, J., Moretti, R., Meiler, J., and Cremers, D. (2020a). Deep learning for virtual screening: Five reasons to use ROC cost functions. *arXiv preprint arXiv:2007.07029*. (ˆ9, 33, 92)

[Golkov et al., 2016a] Golkov, V., Dosovitskiy, A., Sperl, J. I., Menzel, M. I., Czisch, M., Sämann, P., Brox, T., and Cremers, D. (2016a). q-Space deep learning: Twelve-fold shorter and model-free diffusion MRI scans. *IEEE Transactions on Medical Imaging*, 35. © 2016 IEEE. Reprinted with permission. (ˆ8, 9, 11, 91)

[Golkov et al., 2015a] Golkov, V., Dosovitskiy, A., Sämann, P., Sperl, J. I., Sprenger, T., Czisch, M., Menzel, M. I., Gómez, P. A., Haase, A., Brox, T., and Cremers, D. (2015a). q-Space deep learning for twelve-fold shorter and model-free diffusion MRI scans. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Munich, Germany. (ˆ91)

[Golkov et al., 2014a] Golkov, V., Menzel, M., Sprenger, T., Haase, A., Cremers, D., and Sperl, J. (2014a). Semi-joint reconstruction for diffusion MRI denoising imposing similarity of edges in similar diffusion-weighted images. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Golkov et al., 2013a] Golkov, V., Menzel, M., Sprenger, T., Menini, A., Cremers, D., and Sperl, J. (2013a). Corrected joint SENSE reconstruction, low-rank constraints, and compressed-sensing-accelerated diffusion spectrum imaging in denoising and kurtosis tensor estimation. In *ISMRM Workshop on Diffusion as a Probe of Neural Tissue Microstructure*.

[Golkov et al., 2013b] Golkov, V., Menzel, M., Sprenger, T., Menini, A., Cremers, D., and Sperl, J. (2013b). Reconstruction, regularization, and quality in diffusion MRI using the example of accelerated diffusion spectrum imaging. In *16th Annual Meeting of the German Chapter of the ISMRM*. (ˆ10, 11, 91)

[Golkov et al., 2014b] Golkov, V., Menzel, M., Sprenger, T., Souiai, M., Haase, A., Cremers, D., and Sperl, J. (2014b). Direct reconstruction of the average diffusion propagator with simultaneous compressed-sensing-accelerated diffusion spectrum imaging and image denoising by means of total generalized variation regularization. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*. (ˆ91)

[Golkov et al., 2014c] Golkov, V., Menzel, M., Sprenger, T., Souiai, M., Haase, A., Cremers, D., and Sperl, J. (2014c). Improved diffusion kurtosis imaging and direct propagator estimation using 6-D compressed sensing. In *Organization for Human Brain Mapping (OHBM) Annual Meeting*. (ˆ91)

[Golkov et al., 2015b] Golkov, V., Portegies, J. M., Golkov, A., Duits, R., and Cremers, D. (2015b). Holistic image reconstruction for diffusion MRI. In *Computational Diffusion MRI*. Springer, Munich, Germany. Reprinted/adapted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature: Computational Diffusion MRI (Proceedings of the 2015 MICCAI Workshop), Editors: Fuster, A., Ghosh, A., Kaden, E., Rathi, Y., Reisert, M. COPYRIGHT 2016. (ˆ8, 11, 15, 37, 38, 39, 91)

[Golkov et al., 2016b] Golkov, V., Skwark, M. J., Golkov, A., Dosovitskiy, A., Brox, T., Meiler, J., and Cremers, D. (2016b). Protein contact prediction from amino acid co-evolution using convolutional networks for graph-valued images. In *Annual Conference on Neural Information Processing Systems (NIPS)*, Barcelona, Spain. (ˆ7, 29, 31, 92)

[Golkov et al., 2020b] Golkov, V., Skwark, M. J., Mirchev, A., Dikov, G., Geanes, A. R., Mendenhall, J., Meiler, J., and Cremers, D. (2020b). 3D deep learning for biological function prediction from physical fields. In *International Conference on 3D Vision (3DV)*. © 2020 IEEE. Reprinted with permission. (ˆ91, 92)

[Golkov et al., 2014d] Golkov, V., Sperl, J., Menzel, M., Sprenger, T., Tan, E., Marinelli, L., Hardy, C., Haase, A., and Cremers, D. (2014d). Joint super-resolution using only one anisotropic low-resolution image per q-space coordinate. In *Computational Diffusion MRI*. Springer. (ˆ8, 10, 11, 91)

[Golkov et al., 2013c] Golkov, V., Sprenger, T., Menini, A., Menzel, M., Cremers, D., and Sperl, J. (2013c). Effects of low-rank constraints, line-process denoising, and q-space compressed sensing on diffusion MR image reconstruction and kurtosis tensor estimation. In *European Society for Magnetic Resonance in Medicine and Biology (ESMRMB) Annual Meeting*.

[Golkov et al., 2013d] Golkov, V., Sprenger, T., Menzel, M., Cremers, D., and Sperl, J. (2013d). Line-process-based joint SENSE reconstruction of diffusion images with intensity inhomogeneity correction and noise non-stationarity correction. In *European Society for Magnetic Resonance in Medicine and Biology (ESMRMB) Annual Meeting*.

[Golkov et al., 2013e] Golkov, V., Sprenger, T., Menzel, M., Tan, E., King, K., Hardy, C., Marinelli, L., Cremers, D., and Sperl, J. (2013e). Noise reduction in accelerated diffusion spectrum imaging through integration of SENSE reconstruction into joint reconstruction in combination with q-space compressed sensing. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Golkov et al., 2016c] Golkov, V., Sprenger, T., Sperl, J. I., Menzel, M. I., Czisch, M., Sämann, P., and Cremers, D. (2016c). Model-free novelty-based diffusion MRI. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, Prague, Czech Republic. (ˆ8, 91)

[Golkov et al., 2018a] Golkov, V., Swazinna, P., Schmitt, M. M., Khan, Q. A., Tax, C. M. W., Serahlazau, M., Pasa, F., Pfeiffer, F., Biessels, G. J., Leemans, A., and Cremers, D. (2018a). q-Space deep learning for Alzheimer's disease diagnosis: Global prediction and weakly-supervised localization. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*. (ˆ8, 11, 91)

[Golkov et al., 2018b] Golkov, V., Vasilev, A., Pasa, F., Lipp, I., Boubaker, W., Sgarlata, E., Pfeiffer, F., Tomassini, V., Jones, D. K., and Cremers, D. (2018b). q-Space novelty detection in short diffusion MRI scans of multiple sclerosis. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*. (ˆ8, 91)

[Grün et al., 2016] Grün, F., Rupprecht, C., Navab, N., and Tombari, F. (2016). A taxonomy and library for visualizing learned features in convolutional neural networks. *arXiv preprint arXiv:1606.07757*. (ˆ35)

[Gómez et al., 2015] Gómez, P., Sprenger, T., López, A., Sperl, J., Fernandez, B., Molina-Romero, M., Liu, X., Golkov, V., Czisch, M., Saemann, P., Menzel, M., and Menze, B. (2015). Using diffusion and structural MRI for the

automated segmentation of multiple sclerosis lesions. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Haeusser et al., 2018] Haeusser, P., Plapp, J., Golkov, V., Aljalbout, E., and Cremers, D. (2018). Associative deep clustering - training a classification network with no labels. In *Proc. of the German Conference on Pattern Recognition (GCPR)*. (ˆ91)

[Hahnloser et al., 2000] Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., and Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951. (ˆ29)

[Hanson, 2008] Hanson, L. G. (2008). Is quantum mechanics necessary for understanding magnetic resonance? *Concepts in Magnetic Resonance Part A*, 32A(5):329–340. (ˆ14)

[Harley, 2015] Harley, A. W. (2015). An interactive node-link visualization of convolutional neural networks. In *ISVC*, pages 867–877. (ˆ35)

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. (ˆ23)

[Hebey, 1996] Hebey, E. (1996). *Sobolev spaces on Riemannian manifolds*, volume 1635. Springer Science & Business Media. (ˆ38)

[Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708. (ˆ23)

[Jarrett et al., 2009] Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, pages 2146–2153. IEEE Computer Society. (ˆ29)

[Jin et al., 2020] Jin, C., Netrapalli, P., and Jordan, M. I. (2020). What is local optimality in nonconvex-nonconcave minimax optimization? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4880–4889. PMLR. (ˆ34)

[Johansen-Berg and Behrens, 2013] Johansen-Berg, H. and Behrens, T. E. J., editors (2013). *Diffusion MRI: from quantitative measurement to in vivo neuroanatomy*. Academic Press, 2nd edition. (ˆ16)

[Jones, 2010] Jones, D. K., editor (2010). *Diffusion MRI*. Oxford University Press. (ˆ16)

[Justesen et al., 2019] Justesen, N., Bontrager, P., Togelius, J., and Risi, S. (2019). Deep learning for video game playing. *IEEE Transactions on Games*, 12(1):1–20. (ˆ11, 12)

[Klodt et al., 2008] Klodt, M., Schoenemann, T., Kolev, K., Schikora, M., and Cremers, D. (2008). An experimental comparison of discrete and continuous shape optimization methods. In *European Conference on Computer Vision (ECCV)*, Marseille, France. (ˆ37)

[Knoll et al., 2011] Knoll, F., Bredies, K., Pock, T., and Stollberger, R. (2011). Second order total generalized variation (TGV) for MRI. *Magnetic resonance in medicine*, 65(2):480–491. (ˆ15)

[Kondor and Trivedi, 2018] Kondor, R. and Trivedi, S. (2018). On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *ICML*, pages 2747–2755. (ˆ26)

[Kühnel, 2015] Kühnel, W. (2015). *Differential geometry*, volume 77. American Mathematical Soc. (ˆ38)

[Kukačka et al., 2017] Kukačka, J., Golkov, V., and Cremers, D. (2017). Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*. (ˆ11, 21, 22, 27, 30, 32, 34, 92)

[Li and Malik, 2017] Li, K. and Malik, J. (2017). Learning to optimize. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. (ˆ9)

[Lin et al., 2003] Lin, C.-P., Wedeen, V. J., Chen, J.-H., Yao, C., and Tseng, W.-Y. I. (2003). Validation of diffusion spectrum magnetic resonance imaging with manganese-enhanced rat optic tracts and ex vivo phantoms. *Neuroimage*, 19(3):482–495. (ˆ37)

[Ma et al., 2013] Ma, D., Gulani, V., Seiberlich, N., Liu, K., Sunshine, J. L., Duerk, J. L., and Griswold, M. A. (2013). Magnetic resonance fingerprinting. *Nature*, 495(7440):187–192. (ˆ17)

[Ma et al., 2016] Ma, D., Pierre, E. Y., Jiang, Y., Schluchter, M. D., Setsompop, K., Gulani, V., and Griswold, M. A. (2016). Music-based magnetic resonance fingerprinting to improve patient comfort during mri examinations. *Magnetic resonance in medicine*, 75(6):2303–2314. (ˆ17)

[Mahendran and Vedaldi, 2015] Mahendran, A. and Vedaldi, A. (2015). Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196. (ˆ35)

[Mehrabi et al., 2019] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2019). A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*. (ˆ1)

[Menini et al., 2015] Menini, A., Golkov, V., and Wiesinger, F. (2015). Free-breathing, self-navigated RUFIS lung imaging with motion compensated image reconstruction. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Mensch and Blondel, 2018] Mensch, A. and Blondel, M. (2018). Differentiable dynamic programming for structured prediction and attention. In *International Conference on Machine Learning*, pages 3462–3471. PMLR. (ˆ9)

[Menzel et al., 2015] Menzel, M., Sprenger, T., Tan, E., Golkov, V., Hardy, C., Marinelli, L., and Sperl, J. (2015). Robustness of phase sensitive reconstruction in diffusion spectrum imaging. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Moeller et al., 2019] Moeller, M., Mollenhoff, T., and Cremers, D. (2019). Controlling neural networks via energy dissipation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3256–3265. (ˆ9)

[Mori, 2007] Mori, S. (2007). *Introduction to diffusion tensor imaging*. Elsevier. (ˆ16)

[Müller et al., 2021a] Müller, P., Golkov, V., Tomassini, V., and Cremers, D. (2021a). Rotation-equivariant deep learning for diffusion MRI. *arXiv preprint*. (ˆ8, 91)

[Müller et al., 2021b] Müller, P., Golkov, V., Tomassini, V., and Cremers, D. (2021b). Rotation-equivariant deep learning for diffusion MRI (short version). In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*. (ˆ91)

[Naeyaert et al., 2020] Naeyaert, M., Aelterman, J., Audekerke, J. V., Golkov, V., Cremers, D., Pižurica, A., Sijbers, J., and Verhoye, M. (2020). Accelerating in vivo fast spin echo high angular resolution diffusion imaging with an isotropic resolution in mice through compressed sensing. *Magnetic Resonance in Medicine*, 85(3):1397–1413. (ˆ91)

[Naeyaert et al., 2021] Naeyaert, M., Golkov, V., Cremers, D., Sijbers, J., and Verhoye, M. (2021). Faster and better HARDI using FSE and holistic reconstruction. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*. (ˆ91)

[Nair and Hinton, 2010] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA. Omnipress. (ˆ29)

[Neves et al., 2004] Neves, G., Zucker, J., Daly, M., and Chess, A. (2004). Stochastic yet biased expression of multiple dscam splice variants by individual cells. *Nature genetics*, 36(3):240–246. (ˆ20)

[Nguyen et al., 2020] Nguyen, T., Raghu, M., and Kornblith, S. (2020). Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*. (ˆ36)

[Nieuwenhuis et al., 2013] Nieuwenhuis, C., Toeppe, E., and Cremers, D. (2013). A survey and comparison of discrete and continuous multi-label optimization approaches for the potts model. *International Journal of Computer Vision*, 104(3):223–240. (ˆ37)

[Pasa et al., 2019] Pasa, F., Golkov, V., Pfeiffer, F., Cremers, D., and Pfeiffer, D. (2019). Efficient deep network architectures for fast chest X-ray tuberculosis screening and visualization. *Scientific Reports*, 9(1):6268. (ˆ92)

[Peeken et al., 2017] Peeken, J., Knie, C., Golkov, V., Kessel, K., Pasa, F., Khan, Q., Seroglazov, M., Kukačka, J., Goldberg, T., Richter, L., Reeb, J., Rost, B., Pfeiffer, F., Cremers, D., Nüsslin, F., and Combs, S. (2017). Establishment of an interdisciplinary workflow of machine learning-based radiomics in sarcoma patients. In *23. Jahrestagung der Deutschen Gesellschaft für Radioonkologie (DEGRO)*.

[Pham et al., 2020] Pham, H., Xie, Q., Dai, Z., and Le, Q. V. (2020). Meta pseudo labels. *arXiv preprint arXiv:2003.10580*. (ˆ1)

[Pipe, 2014] Pipe, J. (2014). Chapter 2 - pulse sequences for diffusion-weighted MRI. In Johansen-Berg, H. and Behrens, T. E., editors, *Diffusion MRI*, pages 11–34. Academic Press, San Diego, second edition edition. (ˆ16)

[Pock et al., 2009] Pock, T., Cremers, D., Bischof, H., and Chambolle, A. (2009). An algorithm for minimizing the piecewise smooth mumford-shah functional. In *IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan. (ˆ37, 77)

[Poorman et al., 2020] Poorman, M. E., Martin, M. N., Ma, D., McGivney, D. F., Gulani, V., Griswold, M. A., and Keenan, K. E. (2020). Magnetic resonance fingerprinting part 1: Potential uses, current challenges, and recommendations. *Journal of Magnetic Resonance Imaging*, 51(3):675–692. (ˆ11)

[Porter and Looger, 2018] Porter, L. L. and Looger, L. L. (2018). Extant fold-switching proteins are widespread. *Proceedings of the National Academy of Sciences*, 115(23):5968–5973. (ˆ19)

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer. (ˆ23)

[Schmucker et al., 2000] Schmucker, D., Clemens, J. C., Shu, H., Worby, C. A., Xiao, J., Muda, M., Dixon, J. E., and Zipursky, S. L. (2000). Drosophila dscam is an axon guidance receptor exhibiting extraordinary molecular diversity. *Cell*, 101(6):671–684. (ˆ20)

[Schuchardt et al., 2019] Schuchardt, J., Golkov, V., and Cremers, D. (2019). Learning to evolve. *arXiv preprint arXiv:1905.03389*. (ˆ7, 31, 92)

[Seifert et al., 2017] Seifert, C., Aamir, A., Balagopalan, A., Jain, D., Sharma, A., Grottel, S., and Gumhold, S. (2017). Visualizations of deep neural networks in computer vision: A survey. In *Transparent data mining for big and small data*, pages 123–144. Springer. (ˆ35)

[Selvaraju et al., 2017] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626. (ˆ36)

[Simonyan et al., 2013] Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*. (ˆ35)

[Sperl et al., 2014] Sperl, J., Sprenger, T., Tan, E., Golkov, V., Menzel, M., Hardy, C., and Marinelli, L. (2014). Total variation-regularized compressed sensing reconstruction for multi-shell diffusion kurtosis imaging. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Sperl et al., 2013] Sperl, J., Tan, E., Sprenger, T., Golkov, V., King, K., Hardy, C., Marinelli, L., and Menzel, M. (2013). Phase sensitive reconstruction in diffusion spectrum imaging enabling velocity encoding and unbiased noise distribution. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Spivak, 1999] Spivak, M. D. (1999). *A comprehensive introduction to differential geometry*. Publish or perish, 3 edition. (ˆ38)

[Sprenger et al., 2013] Sprenger, T., Fernandez, B., Sperl, J., Golkov, V., Bach, M., Tan, E., King, K., Hardy, C., Marinelli, L., Czisch, M., Sämann, P., Haase, A., and Menzel, M. (2013). Snr-dependent quality assessment of compressed-sensing-accelerated diffusion spectrum imaging using a fiber crossing phantom. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Sprenger et al., 2014] Sprenger, T., Sperl, J., Fernandez, B., Golkov, V., Tan, E., Hardy, C., Marinelli, L., Czisch, M., Sämann, P., Haase, A., and Menzel, M. (2014). Novel acquisition scheme for diffusion kurtosis imaging based on compressed-sensing accelerated DSI yielding superior image quality. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*.

[Sprenger et al., 2016] Sprenger, T., Sperl, J. I., Fernandez, B., Golkov, V., Eidner, I., Sämann, P. G., Czisch, M., Tan, E. T., Hardy, C. J., Marinelli, L., Haase, A., and Menzel, M. I. (2016). Bias and precision analysis of diffusional kurtosis imaging for different acquisition schemes. *Magnetic Resonance in Medicine*.

[Springenberg et al., 2014] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*. (ˆ36)

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958. (ˆ22, 34)

[Swazinna et al., 2019] Swazinna, P., Golkov, V., Lipp, I., Sgarlata, E., Tomassini, V., Jones, D. K., and Cremers, D. (2019). Negative-unlabeled learning for diffusion MRI. In *International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting*. (ˆ8, 91)

[Szegedy et al., 2017] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31. (ˆ23)

[Szegedy et al., 2016] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826. (ˆ23)

[Tay et al., 2020] Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. (2020). Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*. (ˆ12)

[Tewari et al., 2020] Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Sunkavalli, K., Martin-Brualla, R., Simon, T., Saragih, J., Nießner, M., et al. (2020). State of the art on neural rendering. In *Computer Graphics Forum*, volume 39, pages 701–727. Wiley Online Library. (ˆ11)

[Tzeng et al., 2017] Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176. (ˆ12)

[Valkonen, 2014] Valkonen, T. (2014). A primal–dual hybrid gradient method for nonlinear operators with applications to mri. *Inverse Problems*, 30(5):055012. (ˆ37, 77)

[van den Oord et al., 2017] van den Oord, A., Vinyals, O., and Kavukcuoglu, K. (2017). Neural discrete representation learning. In *Proceedings of the 31st*

International Conference on Neural Information Processing Systems, pages 6309–6318. (ˆ33)

[Vasilev et al., 2019] Vasilev, A., Golkov, V., Meissner, M., Lipp, I., Sgarlata, E., Tomassini, V., Jones, D. K., and Cremers, D. (2019). q-Space novelty detection with variational autoencoders. In *MICCAI 2019 International Workshop on Computational Diffusion MRI*. (ˆ8, 32, 91)

[Vegas et al., 2009] Vegas, S., Juristo, N., and Basili, V. R. (2009). Maturing software engineering knowledge through classifications: A case study on unit testing techniques. *IEEE Transactions on Software Engineering*, 35(4):551–565. (ˆ12)

[Vese and Le Guyader, 2015] Vese, L. A. and Le Guyader, C. (2015). *Variational methods in image processing*. CRC Press. (ˆ1)

[Weiler and Cesa, 2019] Weiler, M. and Cesa, G. (2019). General E(2)-equivariant steerable CNNs. In *NeurIPS*, pages 14334–45. (ˆ26)

[Yan et al., 2003] Yan, L., Dodier, R. H., Mozer, M., and Wolniewicz, R. H. (2003). Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 848–855. (ˆ9)

[Zeiler and Fergus, 2014] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer. (ˆ35, 36)

[Zhou et al., 2016] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929. (ˆ36)

[Zhu et al., 2018] Zhu, B., Liu, J. Z., Cauley, S. F., Rosen, B. R., and Rosen, M. S. (2018). Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492. (ˆ15)