Sparsity in Continuous-Depth Neural Networks

Hananeh Aliee
Helmholtz Munich

Till Richter
Helmholtz Munich

Mikhail Solonin*
Technical University of Munich

Ignacio Ibarra
Helmholtz Munich

Fabian Theis
Technical University of Munich
Helmholtz Munich
[1]

Niki Kilbertus
Technical University of Munich
Helmholtz AI, Munich

{hananeh.aliee, till.richter, ignacio.ibarra, fabian.theis, niki.kilbertus}
@helmholtz-muenchen.de

Abstract

Neural Ordinary Differential Equations (NODEs) have proven successful in learning dynamical systems in terms of accurately recovering the observed trajectories. While different types of sparsity have been proposed to improve robustness, the generalization properties of NODEs for dynamical systems beyond the observed data are underexplored. We systematically study the influence of weight and feature sparsity on forecasting as well as on identifying the underlying dynamical laws. Besides assessing existing methods, we propose a regularization technique to sparsify "input-output connections" and extract relevant features during training. Moreover, we curate real-world datasets consisting of human motion capture and human hematopoiesis single-cell RNA-seq data to realistically analyze different levels of out-of-distribution (OOD) generalization in forecasting and dynamics identification respectively. Our extensive empirical evaluation on these challenging benchmarks suggests that weight sparsity improves generalization in the presence of noise or irregular sampling. However, it does not prevent learning spurious feature dependencies in the inferred dynamics, rendering them impractical for predictions under interventions, or for inferring the true underlying dynamics. Instead, feature sparsity can indeed help with recovering sparse ground-truth dynamics compared to unregularized NODEs.

*Work done while at TUM. MS is currently employed by J.P. Morgan Chase Co.; mikhail.solonin@jpmorgan.com

36th Conference on Neural Information Processing Systems (NeurIPS 2022)

1 Introduction

This paper explores the role of sparsity in Neural Ordinary Differential Equations (NODEs) for modeling dynamical systems. While extreme overparameterization contributes to the success of deep neural networks, it can lead to increased computational costs. Sparse neural networks offer benefits like imitating human learning, enhancing computational efficiency, and improving interpretability. The study focuses on two types of sparsity: weight sparsity, which reduces computational needs, and feature sparsity, which can enhance the identification of underlying dynamical laws. The authors introduce a novel regularization technique, PathReg2, which promotes both weight and feature sparsity. They also evaluate the performance of various sparsity methods using real-world datasets, aiming to assess their impact on out-of-distribution generalization for prediction and dynamical law inference.

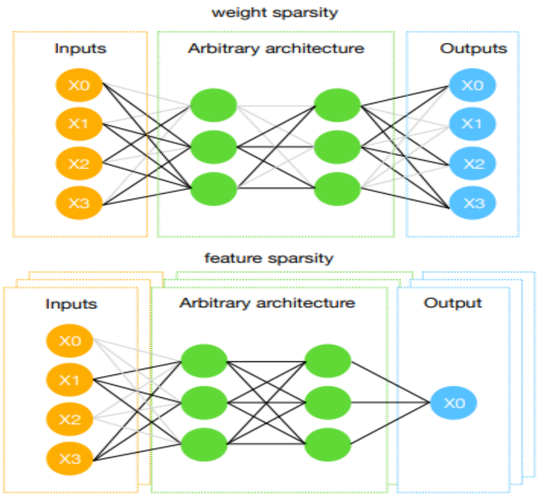

Figure 1: Model vs feature sparsity

2 Background

2.1 Continuous-depth neural nets

Among the plethora of deep learning based methods to estimate dynamical systems from data [4],[3],[1],[8], we focus on Neural Ordinary Differential Equations (NODEs). In NODEs, a neural network with parameters θ is used to learn the function $f_\theta \approx f$ from data, where f defines a (first order) ODE in its explicit representation $\dot{X} = f(\mathrm{X}, t)$ [2]. Starting from the initial observation $X(a)$ at some time $t = a$, an explicit iterative ODE solver is deployed to predict $X(t)$ for $t \in (a,\ b]$ using the current derivative estimates from f_θ. The parameters θ are then updated via backpropagation on the mean squared error (MSE) between predictions and observations. As discussed extensively in the literature, NODEs can outperform traditional ODE parameter inference techniques in terms of reconstruction error, especially for non-linear dynamics f [6],[5]. In particular, one advantage of NODEs over previous methods for inferring non-linear dynamics such as SINDy [20] is that no dictionary of non-linear basis functions has to be pre-specified. A variant of NODEs for second order systems called SONODE exploits the common reduction of higher-order ODEs to first order systems [7]. The second required initial condition, the initial velocity $\dot{X}(a)$, is simply learned from $X(a)$ via another neural network in an end-to-end fashion. Our experiments build on the public NODE and SONODE implementations. However, our framework is readily applicable to most other continuous-depth neural nets including Augmented NODE [8], latent NODEs [1], and neural stochastic DEs [10],[31]

2.2 Sparsity and generalization

Two prominent motivations for enforcing sparsity in over-parameterized deep learning models are (a) pruning large networks for efficiency (speed, memory), (b) as a regularizer that can improve interpretability or prevent overfitting. In both cases, one aims at preserving (as much as possible) i.i.d. generalization performance, i.e., performance on unseen data from the same distribution as the training data, compared to a non-sparse model [14]. Another line of research has explored the link between generalizability of neural nets and causal learning [28], where generalization outside the i.i.d. setting, is conjectured to require an underlying *causal model*. Deducing true laws of nature purely from observational data could be considered an instance of inferring a causal model of the world. Learning the correct causal model enables accurate predictions not only on the observed data (next observation), but also under distribution shifts for example under interventions [18]. A common assumption in causal modeling is that each variable depends on (or is a function of) few other variables [7],[27],[29]. In the ODE context, we interpret the variables (or their derivatives) that enter a specific component f_i as causal parents, such that we can write $\dot{X}_i = f(pa(X_i), t)$ [24]. Thus, feature sparsity translates to each variable X_i having only few parents $pa(X_i)$, which can also be interpreted as asking for "simple" dynamics. Since weight sparsity as well as regularizing the number of function evaluations in NODEs can also be considered to bias them towards "simpler dynamics", these notions are not strictly disjoint, raising the question whether one implies the other. In terms of feature sparsity, Aliee et al.[15], Bellot et al. [18] study system identification and causal structure inference from time-series data using NODEs. They suggest that enforcing sparsity in the number of causal interactions improves parameter estimation as well as predictions under in terventions. Let us write out f_θ as a fully connected net with L hidden layers parameterized by $\theta := (W^l, b^l)_{l=1}^{L+1}$ as

$$f_\theta(X) = W^{L+1}\sigma\left(\ldots\sigma\left(W^2\sigma\left(W^1X + b^1\right) + b^2\right)\ldots\right) \tag{1}$$

with element-wise activation function σ, l-th layer weights W^1 , and biases b^l . Aliee et al. [4] then seek to reduce the overall number of parents of all variables by attempting to cancel all contributions of a given input on a given output node through the neural net. In a linear setting, where *σ(x) = x*, the regularization term is defined *by*[3]

$$\|A\|_{1,1} = \|W^{L+1}\cdots W^1\|_{1,1} \tag{2}$$

where $A_{ij} = 0$ if and only if the i-th output is constant in the j-th input. In the non-linear setting, for certain *σ(x) ≠ x*, the regularizer $\|A\|_{1,1} = \|W^{L+1}\cdots W^1\|_{1,1}$ (with entry wise absolute values on all W^l) is an upper bound on the number of input-output dependencies, i.e., for each output i summing up all the inputs j it is not constant in. Regularizing input gradients [17],[11],[9]is another alternative to train neural networks that depend on fewer inputs, however it is not scalable to high-dimensional regression tasks. Bellot et al. [13] instead train a separate neural net $f_{\theta,i} : \mathbb{R}^n \to \mathbb{R}$ for each variable X_i and penalize NODEs using GroupLasso on the inputs via

$$\sum_{k,i=1}^{n} \left\| \left[W_i^1\right]_{\cdot,k} \right\|_2 \tag{3}$$

where W_i^1 is the weight matrix in the input layer of f_i and $[W_i^1]_{\cdot,k}$] refers to the k th column of W_i^1 that should simultaneously (as a group) be set to zero or not. While enforcing strict feature sparsity (instead of regularizing an upper bound), parallel training of multiple NODE networks can be computationally expensive (Figure 1, bottom). While this work suggests that sparsity of causal interactions helps system identification, its empirical evaluation predominantly focuses on synthetic data settings, leaving performance on real data underexplored. Another recent work suggests that standard weight or neuron pruning improves generalization for NODEs [15]. The authors show that pruning lowers empirical risk in density estimation tasks and decreases Hessian's eigenvalues, thus obtaining better generalization (flat minima). However, the effect of sparsity on identifying the underlying dynamics as well as the generalization properties of NODEs to forecast future values are not assessed.

3 Sparsification of neural ODEs

In an attempt to combine the strengths of both weight and feature sparsity, we propose a new regularization technique, called PathReg, which compares favorably to both C-NODE [1] and GroupLasso [19]. Before introducing PathReg, we describe how to extend existing methods to NODEs for an exhaustive empirical evaluation.

3.1 Methods

L0 regularization. Inspired by [31], we use a *differentiable* L0 norm regularization method that can be incorporated in the objective function and optimized via stochastic gradient descent. The L0 regularizer prunes the network during training by encouraging weights to get *exactly zero* using a set of non-negative stochastic gates z. For an efficient gradient-based optimization, Louizos et al. [25] propose to use a continuous random variable s with distribution *q(s)* and parameters ϕ, where z is then given by

$$s \sim q(s \mid \Phi), \quad z = \min(1, \max(0, s)) \tag{4}$$

Gate z is a hard-sigmoid rectification of s^2 that allows the gate to be exactly zero. While we have the freedom to choose any smoothing distribution *q(s)*, we use binary concrete distribution [16],[12] as suggested by the original work [30]. The regularization term is then defined as the probability of s being positive

$$q(z \neq 0 \mid \phi) = 1 - Q(s \leq 0 \mid \phi) \tag{5}$$

where Q is the cumulative distribution function of s. Minimizing the regularizer pushes many gates to be zero, which implies weight sparsity as these gates are multiplied with model weights. L0 regularization can be added directly to the NODE network f_θ.

LassoNet is a feature selection method [26] that uses an input-to-output skip (residual) connection that allows a feature to participate in a hidden unit only if its skip connection is active. LassoNet can be thought of as a residual feed-forward neural net $Y = S^T X + h_\theta(X)$, where h_θ denotes another feed-forward network with parameters $0, S \in \mathbb{R}^{n \times n}$ refers to the weights in the residual layer, and Y are the responses. To enforce feature sparsity, L1 regularization is applied to the weights of the skip connection, defined as $\|S\|_{1,1}$ (where $\|.\|_{1,1}$ denotes the element-wise L1 norm). A constraint term with factor p is then added to the objective to budget the non-linearity involved for feature k relative to the importance of X_k

$$\min_{\theta,S} \mathcal{L}_{\mathcal{D}}(\theta, S) + \lambda \|S\|_{1,1} \text{subject to } \left\| [W^1]_{\cdot,k} \right\|_\infty \leq \rho \left\| [S]_{\cdot,k} \right\|_2 \quad \text{for } k \in \{1, \ldots, n\}, \tag{6}$$

where $[W^1]_{\cdot,k}$ denotes the k th column of the first layer weights of h_θ, and $[S]_{,k}$ represents the k th column of S. When $\rho = 0$, only the skip connection remains (standard linear Lasso), while $\rho \to \infty$ corresponds to unregularized network training.

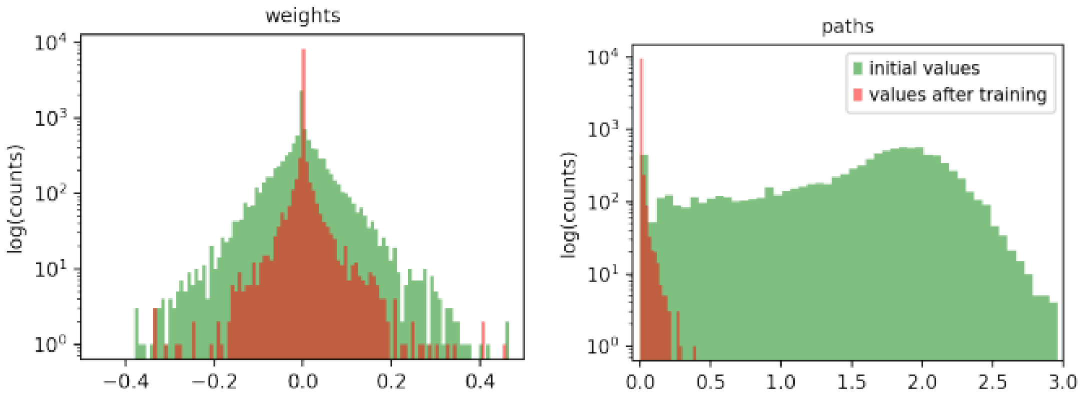

Figure 2: The distributions of network weights and paths weights (over the entries of the matrix in Eq. 13) using PathReg applied to single-cell data in Section 4.3. PathReg increases both model and feature sparsity

LassoNet regularization can be extended to NODEs by adding the skip connection either before or after the integration (the ODE solver). If added before the integration, which we call Inner LassoNet, a linear function of X is added to its time derivative

$$\dot{X} = S^T X + f_\theta(X, t), \quad X(0) = x_0 \tag{7}$$

Adding the skip connection after the integration (and the predictor o), called Outer LassoNet, yields

$$X_t = S^T X_t + b(\text{ODESolver}(f_\theta, x_0, t_0, t)) \tag{8}$$

PathReg (ours). While L0 regularization minimizes the number of non-zero weights in a network, it does not necessarily lead to feature sparsity meaning that the response variables can still depend on all features including spurious ones. To constrain the number of input-output paths, i.e., to enforce feature-sparsity, we regularize the probability of any path throughout the entire network contributing to an output. To this end, we use non-negative stochastic gates $z = g(s)$ similar to Eq. 4, where the probability of an input-output path P being non-zero is given by

$$q(P \neq 0) = \prod_{i=1}^{n} q(z_i \neq 0 \mid \phi) \tag{9}$$

and we constrain

$$\sum_{i=1}^{\#\text{paths}} \prod_{z \in P_i} q(z \neq 0 \mid \phi) \tag{10}$$

to minimize the number of paths that yield non-zero contributions from inputs to outputs. This is equivalent to regularizing the *gate adjacency matrix* $A_z = G^{L+1}...G^1$. Where G^l is a probability matrix corresponding to the probability of the l-th layer gates being positive. Then, A_{zij} represents the sum of the probabilities of all paths between the i-th input and the j-th output. Ultimately, we thus obtain our PathReg regularization term

$$\|A_z\|_{1,1} = \|G^{L+1} \cdots G^1\|_{1,1} \text{ with } G^l_{ij} = q(z_{ij} \neq 0 \mid \phi_i), \tag{11}$$

where $q^l(z_{ij} \neq 0)$ with parameters $\phi_{i,j}$ is the probability of the l-th layer gate z_{ij} being nonzero. Regularizing $\|A_z\|_{1,1}$, minimizes the number of paths between inputs and outputs and induces no shrinkage on the actual values of the weights. Therefore, we can utilize other regularizers on θ such as $\|A\|_{1,1}$ in conjunction with PathReg similar to Eq. 2. In this work, we consider the following overall loss function

$$\begin{aligned} \mathcal{R}(\theta, \Phi) &= \mathbb{E}_{q(s|\Phi)} \left[-\frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\sigma(f(x_i;\theta) + s), y_i) \right] + \lambda_0 \|A_z\|_{1,1} + \lambda_1 \|A\|_{1,1} \\ &= \mathcal{L}_{\mathcal{E}}(\theta, \Phi) + \lambda_0 \|A_z\|_{1,1} + \lambda_1 \|A\|_{1,1} \end{aligned} \tag{12}$$

Table 1: Results for the synthetic second-order ODE

	MSE		SPARSITY (%)	
MODEL	TRAIN	EXTRAPOLATION	FEATURES	WEIGHTS
BASE	1.3×10^{-4}	3.2×10^{-3}	47.9	11.1
LO	1.9×10^{-2}	3.3×10^{-2}	0.0	**49.6**
C-NODE	6.6×10^{-5}	4.1×10^{-4}	**75.5**	16.7
PATHREG	**6.3** $\times 10^{-5}$	**3.9** $\times 10^{-4}$	**75.5**	35.2
LASSONET	8.4×10^{-3}	4.0×10^{-2}	0.0	0.0
GROUPLASSO	8.8×10^{-5}	1.5×10^{-3}	61.1	6.0

fits the current dataset. While A1,1 shrinks the actual values of weights θ in an attempt to zero out entire paths, $\|A_z\|_{1,1}$ enforces exact zeros for entire paths at once. When $\|A_z\|_{1,1}$= 0 the output i is constant in input j. In practice, during training the expectation in Eq. 12 is estimated as usual via Monte Carlo sampling. Unlike GroupLasso [26], PathReg does not require training of multiple networks in parallel. Moreover, unlike C-NODE [21], PathReg leads to exact zeros in the weight matrices and requires no choice of threshold for deciding which paths are considered as zeros (see Figure 2)

4 Results

In this section, we present empirical results on several datasets from different domains including human motion capture data and large-scale single-cell RNA-seq data. We demonstrate the effect of the different sparsity methods including L0 [22], C-NODE [30], LassoNet [23], GroupLasso [18], and our PathReg on the generalization capability of continuous-depth neural nets. We address different aspects of generalization by assessing the accuracy of these models for time-series forecasting and the identification of the governing dynamical laws. In all examples, we assume that the observed motion can be (approximately) described by a system of ODEs where variables correspond directly to the observables of the trajectories.

4.2 Sparsity improves time-series forecasting

We next demonstrate the robustness of sparse models for both reconstructing and extrapolating human movements using real motion capture data from mocap.cs.cmu.edu. Each datapoint is 93-dimensional (31 joint locations with three dimensions each) captured over time. We select three different movements including walking, waving, and golfing, and use 100 frames each for training. After training, we query all models to extrapolate the next 100 frames. For walking and golfing, where multiple trials are available, we also test the model on unseen data in the sense that it has not seen any subset of those specific sequences (despite having trained on other sequences depicting the same type of movement). train data extrapolation PathReg Base PathReg Base test data extrapolation.

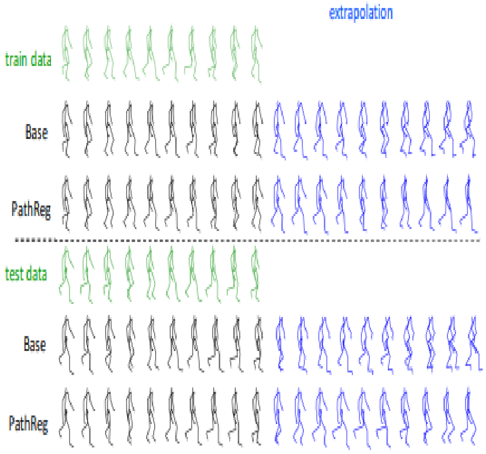

Figure 4: Human-movement reconstructions and extrapolation

Generally, for time-series forecasting, it is not straight forward to give precise definitions of in-, and out-of-distribution. In Table 7 in Appendix C, we show a grid of plots illustrating loss and sparsity as a function of the regularization parameter of each method on this dataset. We show detailed results in Table 3 in Appendix C and summarize them concisely in Figure 3. We did not manage to scale GroupLasso to this 93-dimensional dataset. While all other models show comparative performance (in terms of MSE error), only PathReg achieves strong levels of both weight and feature sparsity (quantitative comparison is in Table 3). The results show that PathReg outperforms other meth

ods with respect to MSE for both train and test datasets while resulting in the highest model and feature sparsity. We further examine the extrapolation (in time) capability of these models. We observe that all sparsity regularization techniques improve extrapolation over unregularized NODEs. Figure 4 qualitatively illustrates the reconstruction and extrapolation performance of both the baseline and PathReg for the walking dataset (visually, the results are similar to PathReg for other regularizers). While these models perform well for extrapolation and generalization to unseen data, we show that only PathReg and C-NODE are able to avoid using spurious features for forecasting. Figures 7 and 8 in Appendix C show that PathReg and C-NODE learn sensible and arguably correct adjacency matrices (derived from joint connectedness), whereas L0 and LassoNet fail to sparsify the input-output interactions appropriately (more details in Appendix C.2). Since we use second-order NODEs for this dataset, the adjacency matrix described in Section 3.3 represents how the acceleration predicted by f_θ of each joint point (e.g., X_i) depends on the positions as well as the velocities of other joint points (e.g., X_j). In case the dependency is non-constant, we infer that the joint point X_i interacts with X_j

5 Conclusion and discussion

We have shown the efficacy of various sparsity enforcing techniques on generalization of continuousdepth NODEs and proposed PathReg, a regularizer acting directly on entire paths throughout a neural network and achieving exact zeros. We curate real-world datasets from different domains consisting of human motion

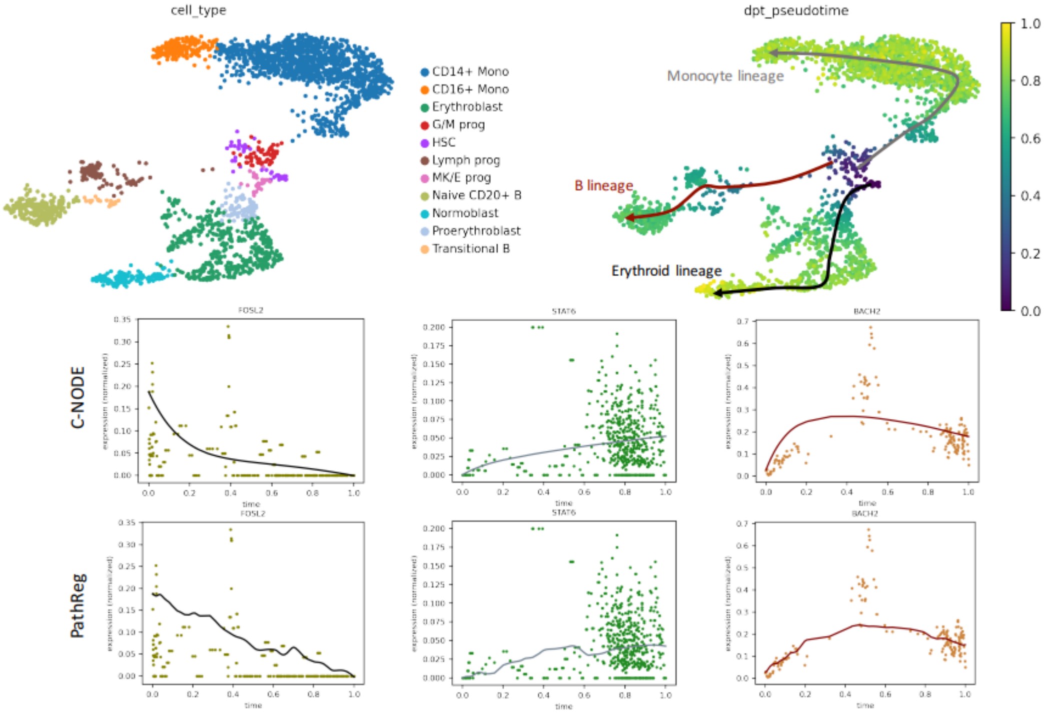

Figure 5: Single-cell RNA-seq Data. Shown are 2D UMAP visualisation of training data colored by cell types and pseudotime (top) where each point is a cell, and the expressions of three genes and their predictions over pseudo time using C-NODE (top) and PathReg (bottom). The predictions across all genes are presented in Figure 9.

capture and single-cell RNA-seq data. Our findings demonstrate that sparsity improves out-of-distribution generalization (for the types of OOD considered) of NODEs when our objective is system identification from observational data, extrapolation of a trajectory, or real-world generalization to unseen biological datasets collected from diverse donors. Finally, we show that unlike weight sparsity, feature sparsity as enforced by PathReg can indeed help in identifying the underlying dynamical laws instead of merely achieving strong in-distribution predictive performance. This is particularly relevant for applications like gene-regulatory network inference, where the ultimate goal is not prediction and forecasting, but revealing the true underlying regulatory interactions between genes of interest. We hope that our empirical findings as well as curated datasets can serve as useful benchmarks to a broader community and expect that extensions of our framework to incorporate both observational and experimental data will further improve practical system identification for ODEs. Finally, our path-based regularization technique may be of interest to other communities that aim at enforcing various types of shape constraints or allowed dependencies into deep learning based models

Acknowledgments and Disclosure of Funding

We thank Ronan Le Gleut from the Core Facility Statistical Consulting at Helmholtz Munich for the statistical support. HA and FJT acknowledge support by the BMBF (grant 01IS18036B) and the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI (grant ZT-I-PF-5-01) and CausalCell-Dynamics (grant Interlabs-0029). TR is supported by the Helmholtz Association under the joint research school "Munich School for Data Science" - MUDS. NK was funded by Helmholtz Association's Initiative and Networking Fund through Helmholtz AI. FJT reports receiving consulting fees from Roche Diagnostics GmbH and ImmunAI, and ownership interest in Cellarity, Inc. and Dermagnostix. The remaining authors

declare no competing interests.

References

[1] Atte Aalto et al. "Gene regulatory network inference from sparsely sampled noisy data". In: *Nature Communications* 11.1 (2020), p. 3493. ISSN: 2041-1723.

[2] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. "A Convergence Theory for Deep Learning via Over-Parameterization". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 242–252.

[3] Brian Bartoldson et al. "The Generalization-Stability Tradeoff In Neural Network Pruning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. 2020, pp. 20852–20864.

[4] A Bolstad, B D Van Veen, and R Nowak. "Causal Network Inference Via Group Sparse Regularization". In: *IEEE Transactions on Signal Processing* 59.6 (2011), pp. 2628–2641.

[5] Nutan Chen et al. *Learning Flat Latent Manifolds with VAEs*. 2020. arXiv: `2002.04881 [stat.ML]`.

[6] Darya Deen et al. "Identification of the transcription factor MAZ as a regulator of erythropoiesis". In: *Blood Advances* 5 (15 2021). ISSN: 24739537.

[7] Simon S. Du et al. "Gradient Descent Provably Optimizes Over-parameterized Neural Networks". In: *International Conference on Learning Representations*. 2019.

[8] Arnab Ghosh et al. "STEER : Simple Temporal Regularization For Neural ODE". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. 2020, pp. 14831–14843.

[9] Ramin Hasani et al. "Liquid Time-constant Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.9 (May 2021), pp. 7657–7666. URL: `https://ojs.aaai.org/index.php/AAAI/article/view/16936`.

[10] Torsten Hoefler et al. "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks". In: *Journal of Machine Learning Research* 22.241 (2021), pp. 1–124.

[11] Jacob Kelly et al. "Learning Differential Equations that are Easy to Solve". In: *Neural Information Processing Systems*. 2020.

[12] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. 2017.

[14] Samuel A Lambert et al. "The Human Transcription Factors". In: *Cell* 172.4 (Feb. 2018), pp. 650–665.

[15] Yijia Liu, Lexin Li, and Xiao Wang. "A nonlinear sparse neural ordinary differential equation model for multiple functional processes". In: *Canadian Journal of Statistics* 50 (1 2022). ISSN: 1708945X. DOI: `10.1002/cjs.11666`.

[16] Christos Louizos, Max Welling, and Diederik P. Kingma. "Learning Sparse Neural Networks through L_0 Regularization". In: *International Conference on Learning Representations*. 2018.

[17] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables". In: *International Conference on Learning Representations*. 2017. URL: `https://openreview.net/forum?id=S1jE5L5gl`.

[18] F. Micci et al. "High-throughput sequencing identifies an NFIA/CBFA2T3 fusion gene in acute erythroid leukemia with t(1;16)(p31;q24)". In: *Leukemia* 27 (4 2013). ISSN: 08876924.

[19] Grégoire Montavon et al. "Layer-Wise Relevance Propagation: An Overview". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Cham: Springer International Publishing, 2019, pp. 193–209.

[20] Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. "From Ordinary Differential Equations to Structural Causal Models: The Deterministic Case". In: *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*. UAI'13. Bellevue, WA: AUAI Press, 2013, pp. 440–448.

[21] Norman Mu and Justin Gilmer. *MNIST-C: A Robustness Benchmark for Computer Vision*. 2019. arXiv: `1906.02337 [cs.CV]`.

[22] Behnam Neyshabur et al. *Towards Understanding the Role of Over-Parametrization in Generalization of Neural Networks*. 2018. arXiv: `1805.12076 [cs.LG]`.

[23] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. "On the Role of Sparsity and DAG Constraints for Learning Linear DAGs". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. 2020, pp. 17943–17954.

[24] Avik Pal et al. "Opening the Blackbox: Accelerating Neural Differential Equations by Regularizing Internal Solver Heuristics". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8325–8335.

[25] Xiaojie Qiu et al. "Inferring Causal Gene Regulatory Networks from Coupled Single-Cell Expression Dynamics Using Scribe". In: *Cell Systems* 10.3 (2020), 265–274.e11. ISSN: 2405-4712.

[26] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. "Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 2017, pp. 2662–2670.

[27] F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. "SCANPY: Large-scale single-cell gene expression data analysis". In: *Genome Biology* (2018). ISSN: 1474760X.

[28] Shiyun Xu et al. "Sparse Neural Additive Model: Interpretable Deep Learning with Feature Selection via Group Sparsity". In: *ICLR 2022 Workshop on PAIRˆ2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*. 2022.

[29] Xingzi Xu et al. "Characteristic Neural Ordinary Differential Equations". In: *arXiv preprint arXiv:2111.13207* (2021).

[30] Chiyuan Zhang et al. "Understanding Deep Learning (Still) Requires Rethinking Generalization". In: *Commun. ACM* 64.3 (2021), pp. 107–115. ISSN: 0001-0782.

[31] Xun Zheng et al. *DAGs with NO TEARS: Continuous Optimization for Structure Learning*. 2018.