Supplementary Material: "Sparse Flows: Pruning Continuous-depth Models"

Lucas Liebenwein* MIT CSAIL lucas@csail.mit.edu

Ramin Hasani* MIT CSAIL rhasani@mit.edu

Alexander Amini MIT CSAIL amini@mit.edu

Daniela Rus MIT CSAIL rus@csail.mit.edu

S1 Hyperparameters

We provide the necessary hyperparameters to reproduce our experiments below. For each set of experiments (Toy, Tabular, Images) we summarize the architecture of the unpruned model and relevant hyperparameters pertaining to the training/pruning process. All experiments were repeated **three times** with separate random seeds and average results are reported.

Table S1: **Toy Dataset** Hyperparameters.

Hyperparameters		GAUSSIANS	GAUSSIANSPIRAL	SPIRAL	MOON
Architecture	Layers	2	4	4	2
	Hidden Size	128	64	64	128
	Activation	Sigmoid	Sigmoid	Sigmoid	Tanh
	Divergence	Hutchison	Hutchison	Hutchison	Hutchison
Solver	Type	Dopri	Dopri	Dopri	Dopri
	Rel. tol.	1.0e-5	1.0e-5	1.0e-5	1.0e-4
	Abs. tol	1.0e-5	1.0e-5	1.0e-5	1.0e-4
	Backprop.	Adjoint	Adjoint	Adjoint	Adjoint
(Re-)Training	Optimizer	AdamW	AdamW	AdamW	Adam
	Epochs	100	100	100	50
	Batch size	1024	1024	1024	128
	LR	5.0e-3	5.0e-2	5.0e-2	1.0e-2
	β_1	0.9	0.9	0.9	0.9
	β_2	0.999	0.999	0.999	0.999
	Weight decay	1.0e-5	1.0e-2	1.0e-6	1.0e-4
Pruning	PR	10%	10%	10%	10%

Table S2: **Tabular Datasets** Hyperparameters.

Hyperparameters		POWER	GAS	HEPMASS	MINIBOONE	BSDS300
Architecture		Please refer to Table 4, Appendix B.1 of Grathwohl et al. (2019).				
Solver		Please refer to Appendix C of Grathwohl et al. (2019).				
(Re-)Training	Optimizer	Adam	Adam	Adam	Adam	Adam
	Epochs	100	30	400	400	100
	Batch size	10000	1000	10000	1000	10000
	LR	1.0e-3	1.0e-3	1.0e-3	1.0e-3	1.0e-3
	LR step	0.1@{90, 97}	0.1@{25, 28}	0.1@{250, 295}	0.1@{300, 350}	0.1@{96, 99}
	β_1	0.9	0.9	0.9	0.9	0.9
	β_2	0.999	0.999	0.999	0.999	0.999
	Weight decay	1.0e-6	1.0e-6	1.0e-6	1.0e-6	1.0e-6
Pruning	PR	25%	25%	22%	22%	25%

Table S3: **Image Datasets** Hyperparameters.

Hyperparameters		MNIST	CIFAR-10
Architecture		Please refer to Appendix B.1 (multi-scale) of Grathwohl et al. (2019).	
Solver		Please refer to Appendix C of Grathwohl et al. (2019).	
(Re-)Training	Optimizer	Adam	Adam
	Epochs	50	50
	Batch size	200	200
	LR	1.0e-3	1.0e-3
	LR step	0.1@{45}	0.1@{45}
	β_1	0.9	0.9
	β_2	0.999	0.999
	Weight decay	0.0	0.0
Pruning	PR	22%	22%

S2 Figure 6 Clarifications

In figure below, we show the distribution and vector field of learned FFJORD networks (unpruned (top) and 70% pruned down). The area in the vector field declared with a black circle shows the vector field structure around an actual mode in the dataset. We see that the vector field (which is illustrated by black arrows) attracts samples towards the mean of this distribution in both pruned and unpruned networks.

However, there is a drastic difference between the vector field structure in-between modes (annotated by purple circles), between the unpruned and pruned network. In the unpruned network, the vector field attracts samples in-between modes. In contrast, in the pruned network, the vector field is repellent in-between modes. Correspondingly, this illustration shows how an unpruned network tends to have samples in-between modes, while the pruned network avoids this shortcoming.

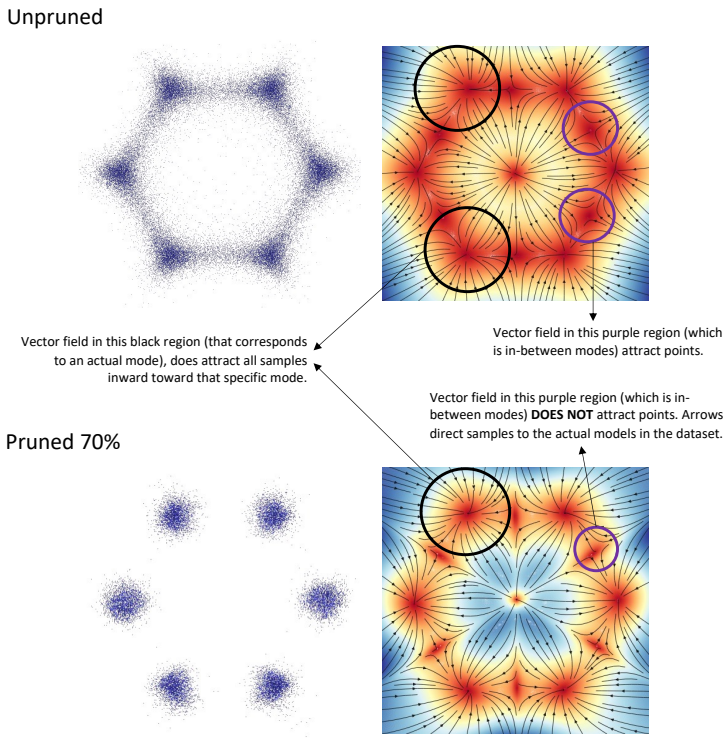


Figure S1: Clarification of how unpruned networks led to mode-collapse and pruned networks did not.

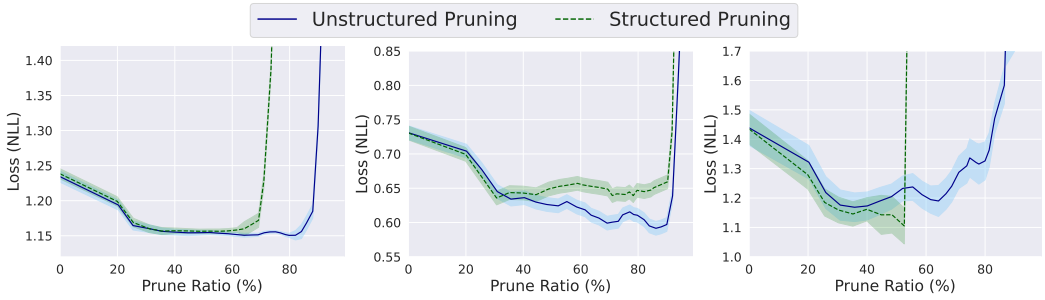
S3 Density Estimation on 2D Data with regular CNF

FFJORD is an efficient way to reparameterize continuous normalizing flows (CNFs) during the backwards pass in order to avoid computing the full Hessian, which can be prohibitively expensive in large-scale experiments. In this section, we validate whether our observations hold independent of the method to compute the Hessian.

Specifically, we train regular CNFs with full Hessian computation on a multi-modal Gaussian distribution, a multi-model set of Gaussian distributions placed orderly on a spiral as well as a spiral distribution with sparse regions following the setup from Section 4.1.

Figure S2 illustrates that densely connected flows (prune ratio = 0%) might get stuck in sharp local minima and thus exhibit unfavorable generalization performance in terms of the NLL. Once we perform pruning, we observe that the quality of the density estimation in all tasks considerably improves. If we continue sparsifying the flows, depending on the task at hand, the flows get disrupted again. Notably, we find that the straightforward Hessian computation may lead to slight improvements compared to using the FFJORD approximation, cf. results for the Spiral dataset using FFJORD and regular CNF as shown in Figure 4(c) and S2(c), respectively. This may be expected in certain cases as FFJORD replaces the exact Hessian computation with an approximation, thus potentially destabilizing training.

Overall, we can experimentally validate that our observations hold regardless of the specific method to compute the Hessian.



(a) Gaussians (b) Gaussian Spiral (c) Spiral

Figure S2: Negative log likelihood of Sparse Flow as function of prune ratio. Sparse Flows were trained using regular CNFs instead of the FFJORD approximation. The remaining experimental setup follows Section 4.1.

S4 Tabularized Results for Density Estimation on Real Data - Tabular

Table S4: Negative test log-likelihood (NLL) in nats on **POWER** tabular dataset and corresponding architecture size in number of parameters and prune ratio for Sparse Flow (based on FFJORD). Results for unstructured and structured pruning are reported.

POWER		**Loss (nats)**	
Prune ratio (%)	Number of parameters	Unstructured	Structured
0%	43.3K	-0.34	-0.34
30%	30.3K	-0.48	-0.41
47%	23.0K	-0.50	-0.48
60%	17.4K	-0.51	-0.45
70%	13.2K	-0.55	-0.45
77%	9.95K	-0.55	-0.44
82%	7.65K	-0.52	-0.39
87%	5.81K	-0.45	-0.25
90%	4.43K	-0.39	0.04

Table S5: Negative test log-likelihood (NLL) in nats on **GAS** tabular dataset and corresponding architecture size in number of parameters and prune ratio for Sparse Flow (based on FFJORD). Results for unstructured and structured pruning are reported.

GAS		**Loss (nats)**	
Prune ratio (%)	Number of parameters	Unstructured	Structured
0%	279K	-8.64	-8.64
30%	194K	-10.85	-10.63
47%	147K	-11.15	-10.93
60%	112K	-11.39	-10.69
70%	84.6K	-11.59	-10.20
77%	64.3K	-11.47	-9.95
83%	48.6K	-10.85	-9.85
87%	36.9K	-10.73	-9.16
90%	28.0K	-10.03	-7.58

Table S6: Negative test log-likelihood (NLL) in nats on **HEPMASS** tabular dataset and corresponding architecture size in number of parameters and prune ratio for Sparse Flow (based on FFJORD). Results for unstructured and structured pruning are reported.

HEPMASS		**Loss (nats)**	
Prune ratio (%)	Number of parameters	Unstructured	Structured
0%	547K	17.54	17.54
20%	437K	16.90	16.91
38%	340K	16.56	16.54
52%	264K	16.22	16.39
63%	205K	16.00	16.21
71%	160K	15.80	16.48
77%	124K	15.67	16.48
82%	96.7K	15.58	16.35
86%	75.5K	15.59	16.97
89%	59.0K	15.62	16.92
92%	46.4K	15.99	17.34
93%	36.3K	15.90	17.80
95%	28.9K	16.04	18.77

Table S7: Negative test log-likelihood (NLL) in nats on **MINIBOONE** tabular dataset and corresponding architecture size in number of parameters and prune ratio for Sparse Flow (based on FFJORD). Results for unstructured and structured pruning are reported.

MINIBOONE		**Loss (nats)**	
Prune ratio (%)	Number of parameters	Unstructured	Structured
0%	821K	10.38	10.38
20%	656K	10.83	10.87
38%	510K	11.11	10.93
52%	397K	10.50	11.09
62%	308K	10.77	11.10
71%	240K	10.51	11.40
77%	186K	10.64	11.07
82%	145K	10.46	11.50
86%	112K	10.37	11.35
89%	86.9K	10.44	11.11
92%	67.7K	10.60	10.93
94%	52.4K	10.27	10.75
95%	40.8K	10.05	10.41
96%	32.3K	9.90	11.26
97%	24.4K	10.15	12.09
98%	17.6K	10.72	14.92
99%	9.13K	13.61	39.01

Table S8: Negative test log-likelihood (NLL) in nats on **BSDS300** tabular dataset and corresponding architecture size in number of parameters and prune ratio for Sparse Flow (based on FFJORD). Results for unstructured and structured pruning are reported.

POWER		**Loss (nats)**	
Prune ratio (%)	Number of parameters	Unstructured	Structured
0%	6.70M	-128.32	-128.32
30%	4.69M	-145.54	-145.42
47%	3.55M	-148.78	-148.70
60%	2.68M	-149.96	-149.92
70%	2.03M	-150.28	-150.66
77%	1.54M	-151.11	-150.11
83%	1.16M	-151.22	-149.41
87%	878K	-151.13	-149.42
90%	662K	-150.53	-148.59

S5 Tabularized Results for Density Estimation on Real Data - Vision

Below, we report the tabularized results on MNIST and CIFAR10 for Sparse Flow using unstructured and structured pruning. Note that highly sparse networks can fail to converge beyond a certain prune ratio. Moreover, structured pruning may not converge for lower prune ratios compared to unstructured pruning since we prune away entire channels and neurons in the former.

Table S9: Negative test log-likelihood (NLL) in bits/dim on **MNIST** and corresponding architecture size in number of parameters and prune ratio for Sparse Flow (based on FFJORD). Results for unstructured and structured pruning are reported.

MNIST		**Loss (bits/dim)**	
Prune ratio (%)	Number of parameters	Unstructured	Structured
0%	801K	1.01	1.01
20%	641K	0.97	0.98
38%	499K	0.96	0.97
52%	387K	0.95	0.97
62%	302K	0.95	0.97
71%	234K	0.96	0.98
77%	182K	0.97	0.98
82%	141K	0.98	0.99
86%	109K	0.97	0.99
89%	84.5K	0.98	1.00

Table S10: Negative test log-likelihood (NLL) in bits/dim on **CIFAR10** and corresponding architecture size in number of parameters and prune ratio for Sparse Flow (based on FFJORD). Results for unstructured and structured pruning are reported. "N/A" indicates that Sparse Flow did not converge for the given prune ratio.

CIFAR10		**Loss (bits/dim)**	
Prune ratio (%)	Number of parameters	Unstructured	Structured
0%	1.36M	3.45	3.45
20%	1.09M	3.38	3.39
38%	845K	3.37	3.38
52%	657K	3.36	3.39
63%	510K	3.37	3.40
71%	395K	3.38	N/A
77%	308K	3.39	N/A
82%	239K	3.40	N/A
86%	186K	3.42	N/A
89%	144K	3.43	N/A
92%	112K	3.45	N/A
94%	86.7K	3.48	N/A
95%	67.4K	3.50	N/A

S6 More Hessian Analysis

We performed the following additional Hessian experiments. We observe that our conclusions on the behavior of the Hessian is generalizable to other datasets as well.

Table S11: Gaussians - Hessian Analysis (Structured Pruning)

Model	NLL	$\lambda_{max}(H)$	$\text{tr}(H)$	$\kappa(H)$
Unpruned FFJORD	1.173	0.0190	0.098	48.2k
Sparse Flows(PR=25%)	1.157	0.0110	0.076	2.76k
Sparse Flows(PR=67%)	1.148	0.0090	0.560	15.17k
Sparse Flows(PR=82%)	1.120	0.0065	0.058	22.75k
Sparse Flows(PR=90%)	1.136	0.0035	0.033	4.70k
Sparse Flows(PR=94%)	1.173	0.0069	0.033	3.94k
Sparse Flows(PR=96%)	1.244	0.0071	0.043	0.58k

Table S12: Gaussians-Spiral - Hessian Analysis (Structured Pruning)

Model	NLL	$\lambda_{max}(H)$	tr(H)	$\kappa(H)$
Unpruned FFJORD	0.880	0.0130	0.121	0.34k
Sparse Flows(PR=25%)	0.692	0.0076	0.058	0.76k
Sparse Flows(PR=48%)	0.634	0.0049	0.047	0.22k
Sparse Flows(PR=67%)	0.646	0.0052	0.051	0.75k
Sparse Flows(PR=82%)	0.657	0.0053	0.053	1.69k
Sparse Flows(PR=94%)	0.740	0.0086	0.070	0.11k
Sparse Flows(PR=96%)	0.986	0.0100	0.095	0.23k

S7 Ablation Study

We have prepared our systematic ablation study with results provided as part of the supplementary material. In our ablation we study different types of features of neural ODE models.

Please find the detailed description of this ablation study in the `README.txt` file. We include this file's description here as well.

Overview

We test different configurations by applying Sparse Flows (Algorithm 1) to investigate what type of network configurations are most stable and robust with respect to pruning. For each type of sweep (ablation), we highlight one key study and one key result.

Sweep over Optimization Parameters

Setup. We study the stability of different configurations for the optimizer and how the different configurations affect the generalization performance during pruning.

Key observation. We can find the most stable parameter configuration for the optimizer by considering sparsifying the flow and thus inducing additional regularization. The most stable optimizer configuration is the one for which we can achieve the most pruning.

Sweep over Model Sizes - Depth vs. Width

Setup. We study different network configurations with (approximately) the same number of parameters. The networks differ in the depth vs. width configuration. We test deep and narrow vs. shallow and wide.

Key observation. Increasing the depth of the network while reducing the width of the network, in general, does not help improve the generalization performance of the network over different prune ratios. Specifically, one should pick the minimal depth of the RHS that ensures convergence. Usually, any depth beyond that does not help improve the generalization performance of the flow.

Sweep over Activations

Setup. We study the same network configurations for the same amount of pruning and vary the activation function of the neural network on the RHS. As we prune, we hope to unearth which activation function is most robust to pruning and consequently to changes in the architecture.

Key observation. ReLU is usually not a very useful activation function. Rather, some Lipschitz continuous activation functions are most useful. Generally, we found tanh and sigmoid to be most useful, although sigmoid was probably the most robust single configuration across all experiments

Sweep over ODE Solvers

Setup. We study the same network configurations for the same amount of pruning and vary the ODE solver of the neural ODE flow. As we prune, we hope to unearth which solver is most robust to pruning and consequently to changes in the architecture.

Key observation. Generally, we found adaptive step size solvers (dopri5) superior to fixed step size solvers (rk4, Euler). Moreover, we found backpropagation through time (BPTT) to be slightly more stable than the adjoint method. Interestingly enough, we could oftentimes only observe the differences between the robustness of the different solvers after we start pruning and sparsifying the flows.

S8 More on Pruning Configurations

Iterative learning rate rewinding: We use learning rate rewinding (LRR) which is a hyperparameter schedule for training/pruning/retraining of neural nets (Renda et al., 2020) (see Algorithm 1).

Pre-defined pruning threshold: We pick a desired prune ratio and prune the weights with the smallest magnitudes until we obtain it. The largest weight that is being pruned constitutes the pre-defined threshold for pruning.

What kinds of structures are being considered in structured pruning? We consider neurons in fully-connected layers and channels with their corresponding filters in convolutional layers for structured pruning. The corresponding pruning score is the norm of the neuron/channel weights as specified in Table 1.

S9 Reproducibility Matters

All code and data which contains the details of the hyperparameters used in all experiments are openly accessible online at: https://github.com/lucaslie/torchprune For the experiments on the toy datasets, we based our code on the TorchDyn library (Poli et al., 2020a). For the experiments on the tabular datasets and image experiments, we based our code on the official code repository of FFJORD (Grathwohl et al., 2019).

References

Ryan P Adams, Jeffrey Pennington, Matthew J Johnson, Jamie Smith, Yaniv Ovadia, Brian Patton, and James Saunderson. Estimating the spectral density of large implicit matrices. *arXiv preprint arXiv:1802.03451*, 2018.

Cenk Baykal, Lucas Liebenwein, Igor Gilitschenski, Dan Feldman, and Daniela Rus. Data-dependent coresets for compressing neural networks with applications to generalization bounds. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=HJfwJ2A5KX.

Cenk Baykal, Lucas Liebenwein, Igor Gilitschenski, Dan Feldman, and Daniela Rus. Sipping neural networks: Sensitivity-informed provable pruning of neural networks. *arXiv preprint arXiv:1910.05422*, 2019b.

Rianne van den Berg, Leonard Hasenclever, Jakub M Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. *arXiv preprint arXiv:1803.05649*, 2018.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2007/file/0d3180d672e08b4c5312dcdafdf6ef36-Paper.pdf.

Ricky T. Q. Chen, Jens Behrmann, David K Duvenaud, and Joern-Henrik Jacobsen. Residual flows for invertible generative modeling. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5d0d5594d24f0f955548f0fc0ff83d10-Paper.pdf.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. In *International Conference on Learning Representations*, 2015.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. In *Advances in Neural Information Processing Systems*, pages 3134–3144, 2019.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *arXiv preprint arXiv:1906.04032*, 2019.

N. Benjamin Erichson, Omri Azencot, Alejandro Queiruga, Liam Hodgkinson, and Michael W. Mahoney. Lipschitz recurrent neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-N7PBXqOUJZ.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam M Oberman. How to train your neural ode. *arXiv preprint arXiv:2002.02798*, 2020.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 881–889, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/germain15.html.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241. PMLR, 2019.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *International Conference on Learning Representations*, 2019.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR*, abs/1510.00149, 2015a. URL http://arxiv.org/abs/1510.00149.

Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1135–1143, 2015b.

Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. A natural lottery ticket winner: Reinforcement learning with ordinary neural circuits. In *International Conference on Machine Learning*, pages 4082–4093. PMLR, 2020.

Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. Liquid time-constant networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7657–7666, May 2021.

Babak Hassibi and David G Stork. *Second order derivatives for network pruning: Optimal brain surgeon*. Morgan Kaufmann, 1993.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2078–2087. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/huang18d.html.

Chin-Wei Huang, Ricky TQ Chen, Christos Tsirigotis, and Aaron Courville. Convex potential flows: Universal probability distributions with optimal transport and convex optimization. *arXiv preprint arXiv:2012.05942*, 2020.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

Priyank Jaini, Kira A Selby, and Yaoliang Yu. Sum-of-squares polynomial flow. In *International Conference on Machine Learning*, pages 3009–3018. PMLR, 2019.

Nitish Shirish Keskar, Jorge Nocedal, Ping Tak Peter Tang, Dheevatsa Mudigere, and Mikhail Smelyanskiy. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

Diederik P Kingma and Prafulla Dhariwal. Glow: generative flow with invertible $1\times$ 1 convolutions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10236–10245, 2018.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934*, 2016.

Zhifeng Kong and Kamalika Chaudhuri. The expressive power of a class of normalizing flow models. In *International Conference on Artificial Intelligence and Statistics*, pages 3599–3609. PMLR, 2020.

Mathias Lechner and Ramin Hasani. Learning long-term dependencies in irregularly-sampled time series. *arXiv preprint arXiv:2006.04418*, 2020.

Mathias Lechner, Ramin Hasani, Alexander Amini, Thomas A Henzinger, Daniela Rus, and Radu Grosu. Neural circuit policies enabling auditable autonomy. *Nature Machine Intelligence*, 2(10): 642–652, 2020a.

Mathias Lechner, Ramin Hasani, Daniela Rus, and Radu Grosu. Gershgorin loss stabilizes the recurrent neural network compartment of an end-to-end robot learning scheme. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5446–5452. IEEE, 2020b.

Mathias Lechner, Ramin Hasani, Radu Grosu, Daniela Rus, and Thomas A Henzinger. Adversarial training is not ready for robot learning. *arXiv preprint arXiv:2103.08187*, 2021.

Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 3870–3882. PMLR, 2020.

Lucas Liebenwein, Cenk Baykal, Harry Lang, Dan Feldman, and Daniela Rus. Provable filter pruning for efficient neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJxkOlSYDH.

Lucas Liebenwein, Cenk Baykal, Brandon Carter, David Gifford, and Daniela Rus. Lost in pruning: The effects of pruning neural networks beyond test accuracy. *Proceedings of Machine Learning and Systems*, 3, 2021.

Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pages 5058–5066, 2017.

Alaa Maalouf, Harry Lang, Daniela Rus, and Dan Feldman. Deep learning meets projective clustering. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=EQfpYwF3-b.

Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asma. Dissecting neural odes. In *34th Conference on Neural Information Processing Systems, NeurIPS 2020*. The Neural Information Processing Systems, 2020.

P Molchanov, S Tyree, T Karras, T Aila, and J Kautz. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017-Conference Track Proceedings*, 2019.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics (TOG)*, 38(5):1–19, 2019.

Junier Oliva, Avinava Dubey, Manzil Zaheer, Barnabas Poczos, Ruslan Salakhutdinov, Eric Xing, and Jeff Schneider. Transformation autoregressive networks. In *International Conference on Machine Learning*, pages 3898–3907. PMLR, 2018a.

Junier Oliva, Avinava Dubey, Manzil Zaheer, Barnabas Poczos, Ruslan Salakhutdinov, Eric Xing, and Jeff Schneider. Transformation autoregressive networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3898–3907. PMLR, 10–15 Jul 2018b. URL http://proceedings.mlr.press/v80/oliva18a.html.

Derek Onken, Samy Wu Fung, Xingjian Li, and Lars Ruthotto. Ot-flow: Fast and accurate continuous normalizing flows via optimal transport. *arXiv preprint arXiv:2006.00104*, 2020.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*, 2017.

Michael Poli, Stefano Massaroli, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Torchdyn: A neural differential equations library. *arXiv preprint arXiv:2009.09346*, 2020a.

Michael Poli, Stefano Massaroli, Atsushi Yamashita, Hajime Asama, Jinkyoo Park, et al. Hypersolvers: Toward fast continuous-depth models. *Advances in Neural Information Processing Systems*, 33, 2020b.

Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. Routledge, 2018.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing fine-tuning and rewinding in neural network pruning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1gSj0NKvB.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a. URL https://proceedings.neurips.cc/paper/2017/file/44a2e0804995faf8d2e3b084a1e2db1d-Paper.pdf.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3310–3320, 2017b.

Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. *arXiv preprint arXiv:2006.11469*, 2020.

Charles Vorbach, Ramin Hasani, Alexander Amini, Mathias Lechner, and Daniela Rus. Causal navigation by continuous-time neural networks. *arXiv preprint arXiv:2106.08314*, 2021.

Antoine Wehenkel and Gilles Louppe. Unconstrained monotonic neural networks. *arXiv preprint arXiv:1908.05164*, 2019.

Liu Yang and George Em Karniadakis. Potential flow generator with l2 optimal transport regularity for generative models. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. Designing energy-efficient convolutional neural networks using energy-aware pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5687–5695, 2017.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 581–590. IEEE, 2020.