RESEARCH STATEMENT

TAN MINH NGUYEN

The recent advances in machine learning present the field of mathematical sciences and computational engineering with numerous opportunities and challenges, among them how to design models that integrate scientific mechanistic modeling, e.g. differential equations, with machine learning methodologies like deep learning. *An overarching theme of my research focuses on developing principled models for machine learning via three fundamental approaches: 1) optimization, 2) numerical analysis, and 3) statistical modeling.*

From an optimization viewpoint, I am interested in establishing the connections between deep learning models such as recurrent neural networks (RNNs) and neural ordinary differential equations (NeuralODEs) with optimization methods such as the gradient descent (GD) algorithm. I then propose new architectures given these connections. In particular, I have been developing new families of momentum-based models that take advantage of momentum methods to improve the convergence speed and the ability to capture long-range dependencies in the data of the models. From a numerical analysis standpoint, I am employing numerical methods such as the fast multipole method and diffusion process to improve deep learning models including transformers and graph neural networks for better efficiency and accuracy. Using statistical modeling as a tool, I develop new generative models that shed light on the architecture of deep neural networks, as well as the attention mechanism in transformers, and suggest new directions to improve these models. On the application side, I have been using the models I develop to solve challenging problems in computational fluid dynamics including turbulence modeling and to design efficient numerical solvers. For future work, I am exploring a new class of machine learning models that can do reasoning via fixed-point algorithms. *I believe reasoning is a key component to develop next-generation machine learning models.*

In the following, I shall outline several major directions of my research.

1 Optimization methods and momentum-based deep learning models

Deep learning models have been achieving state-of-the-art performance on a wide range of machine learning tasks including those in computer vision and natural language processing. Recently, these models have been employed for computational modeling and scientific discovery with very promising results, leading to new directions in the field of scientific machine learning [1, 2]. Despite their popularity in applications, designing deep learning models is an art that often involves an expensive search over candidate architectures.

Momentum-based recurrent neural networks. RNNs are a class of neural networks that capture the dynamics of sequences via cycles in the network of nodes. A recurrent cell in RNNs employs a cyclic connection to update the current hidden state ($\boldsymbol{h}_t$) using the past hidden state ($\boldsymbol{h}_{t-1}$) and the current input data ($\boldsymbol{x}_t$); the dependence of $\boldsymbol{h}_t$ on $\boldsymbol{h}_{t-1}$ and $\boldsymbol{x}_t$ in a recurrent cell can be written as

$$\boldsymbol{h}_t = \sigma(\mathbf{U}\boldsymbol{h}_{t-1} + \mathbf{W}\boldsymbol{x}_t + \boldsymbol{b}),\ \boldsymbol{x}_t \in \mathbb{R}^d, \text{ and } \boldsymbol{h}_{t-1}, \boldsymbol{h}_t \in \mathbb{R}^h,\ \ t = 1, 2, \cdots, T, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{h\times h}, \mathbf{W} \in \mathbb{R}^{h\times d}$, and $\boldsymbol{b} \in \mathbb{R}^h$ are trainable parameters; $\sigma(\cdot)$ is a nonlinear activation function, e.g., sigmoid or hyperbolic tangent. Error backpropagation through time is used to train RNNs, but it tends to result in exploding or vanishing gradients [3]. Thus RNNs may fail to learn long term dependencies. In my work with Prof. Richard Baraniuk, Prof. Andrea Bertozzi, Prof. Stanley Osher and Prof. Bao Wang [4], I develop a gradient descent (GD) analogy of the recurrent cell. In particular, the hidden state update in a recurrent cell in Eqn. (1) is associated with a gradient descent

step towards the optimal representation of the hidden state. I then propose to integrate momentum that used for accelerating gradient dynamics into the recurrent cell, which results in the Momentum-RNN. By choosing the appropriate momentum coefficients, MomentumRNN can alleviate vanishing gradient problem and accelerate training. My momentum framework for designing RNNs is principled with theoretical guarantees provided by the momentum-accelerated dynamical system for optimization and sampling. The design principle can be applied to many existing RNNs and generalized to other advanced momentum-based optimization methods, including Adam [5] and Nesterov accelerated gradients with a restart [6, 7]. In this direction, I also develop an adaptive strategy to compute the momentum coefficients based on the optimal momentum for quadratic optimization and extend my momentum framework to design linear attention in transformers, another popular class of deep learning models for sequential data [8].

Momentum-based neural ordinary differential equations. Apart from discrete models like RNNs, continuous models are gaining currency due to their ability to learn from irregularly-sampled sequential data and to model complex dynamical systems. Among these models are the NeuralODEs, a family of continuous-depth machine learning (ML) models whose forward and backward propagations rely on solving an ODE and its adjoint equation [9]. In particular, NeuralODEs model the dynamics of hidden features $\boldsymbol{h}_t \in \mathbb{R}^h$ using an ODE, which is parameterized by a neural network $f(\boldsymbol{h}_t, t, \theta) \in \mathbb{R}^h$ with learnable parameters θ, i.e. $d\boldsymbol{h}_t/dt = f(\boldsymbol{h}_t, t, \theta)$. Despite their advantages and popularity, the drawback of NeuralODEs is also prominent. In many machine learning and modeling tasks, NeuralODEs require a very high number of steps to solve the ODEs in both training and inference, especially in high accuracy settings where a lower tolerance is needed. As the ODE solver evaluates the function f at each step, this number of steps is often referred to as the number of function evaluations (NFEs). This NFEs increases rapidly with training; high NFEs reduces computational speed and accuracy of NeuralODEs and can lead to blow-ups in the worst-case scenario [10, 11]. Another issue is that NeuralODEs often fail to effectively learn long-term dependencies in sequential data [12].

Motivated by the fast convergence speed of momentum methods, in [13], Hedi Xia (Ph.D. student), Vai Suliafu (Ph.D. student), Prof. Stanley Osher, Prof. Bao Wang and I leverage the continuous limit of the classical momentum accelerated gradient descent and propose the heavy ball NeuralODEs to improve the efficiency of NeuralODEs training and inference. At the core of heavy ball NeuralODEs is replacing the first-order ODE in NeuralODEs with a heavy ball ODE, i.e., a second-order ODE with an appropriate damping term. Our proposed heavy ball NeuralODEs have two theoretical properties that imply practical advantages over NeuralODEs. First, the adjoint equation used for training a heavy ball NeuralODE is also a heavy ball NeuralODE, thus accelerating both forward and backward propagation. Second, the spectrum of the heavy ball NeuralODE is well-structured, alleviating the vanishing gradient issue in back-propagation and enabling the model to effectively learn long-term dependencies from sequential data.

In addition, with Nghia Nguyen (Master student), Prof. Stanley Osher and other colleagues [14], I propose the Nesterov NeuralODEs whose layers solve the second-order ordinary differential equations (ODEs) limit of Nesterov's accelerated gradient (NAG) method. Taking the advantage of the convergence rate $\mathcal{O}(k^2)$ of the NAG scheme [6], the Nesterov NeuralODEs improve over the heavy ball NeuralODEs, further speeding up training and inference.

2 Numerical analysis methods for improving deep learning models

Thus far, I have presented my optimization frameworks for designing deep learning models. Next, I shall describe the numerical analysis approaches that I take to improve machine learning models using the fast multipole method and diffusion process.

Enhancing the efficiency of transformers via the fast multipole method. Like RNNs and NeuralODEs, transformers are among the state-of-the-art models for sequential processing tasks. These models rely on the attention mechanism and particularly self-attention, an inductive bias that connects each token in the input through a relevance weighted basis of every other token, as fundamental building blocks for their modeling. This mechanism allows a token to pay attention to other tokens in the input sequence and attain a contextual representation. The main drawback of transformers is that the computational complexity and memory cost of computing attention are quadratic with respect to the sequence length [15]. In the joint work with Prof. Stanley Osher, Prof. Bao Wang [16] and other collaborators, leveraging the idea of the fast multipole method (FMM), I propose the FMMformers, a class of efficient and flexible transformer, to improve the performance and efficiency of transformers. FMM decomposes particle-particle interaction into near-field and far-field components and then performs direct and coarse-grained computation, respectively. Similarly, FMMformers decompose the attention into near-field and far-field attentions, modeling the near-field attention by a banded sparse matrix and the far-field attention by a low-rank matrix. Computing the attention matrix in FMMformers only requires linear complexity in computational time and memory footprint with respect to the sequence length. In an ongoing joint work, using hierarchical matrices, I am introducing hierarchical structures into the FMMformer to further improve the efficiency of the model.

Overcoming the oversmoothing issue in graph neural networks using diffusion process with a source term. Graph neural networks (GNNs) are the backbone for deep learning on graphs, a class of deep learning models that directly operate on graph structures. A well-known problem of GNNs is that increasing the depth of GNNs often results in a significant drop in performance on various graph learning tasks. This performance degradation has been widely interpreted as the oversmoothing issue of GNNs [17]. Moreover, the accuracy of existing GNNs drops severely when they are trained with a limited amount of labeled data. In [18], Prof. Matthew Thorpe, Hedi Xia (Ph.D. student) and I focus on developing new continuous-depth GNNs that overcome the oversmoothing issue and achieve better accuracy in low-labeling rate regimes. We first present a random walk interpretation of GNNs, revealing a potentially inevitable oversmoothing phenomenon. Based on our random walk viewpoint of GNNs, we then propose graph neural diffusion with a source term (GRAND++) that corrects the bias arising from the diffusion process underlying GNNs. GRAND++ theoretically guarantees that: (i) under GRAND++ dynamics, the graph node features do not converge to a constant vector over all nodes even as the time goes to infinity, and (ii) GRAND++ can provide accurate prediction even when it is trained with the limited number of labeled nodes. These theoretical results resonate with the practical advantages of GRAND++. In an ongoing work, I am replacing the diffusion process underlying GNNs and GRAND++ by a wave propagation to further improve the ability of the models to overcome the oversmoothing issue.

3 Statistical aspects of deep learning

The final aspect of my research is to employ statistical modeling tools to interpret and improve deep learning models. In particular, I develop generative models underlying deep learning architectures such as deep neural networks (DNNs) and self-attention mechanism in transformers.

A probabilistic framework for deep neural networks. DNNs with their great performance have been transforming many research areas including science and engineering fields. The success of deep neural network is impressive, but a fundamental question remains: Why do they work? Intuitions abound to explain their success, but a coherent theoretical framework for understanding, analyzing, and synthesizing deep learning architectures has remained elusive. In joint work with Prof. Richard Baraniuk and Prof. Ankit Patel [19], we develop a new theoretical framework that provides insights

into both the successes and shortcomings of DNNs, as well as a principled route to their design and improvement. Our framework is based on a generative probabilistic model, namely the Deep Rendering Mixture Model (DRMM), that explicitly captures variation due to latent variables. We demonstrate that max-sum inference in the DRMM yields an algorithm that exactly reproduces the operations in DNNs, providing a first principle derivation. DRMM training via the expectation-maximization (EM) algorithm is a promising alternative to back-propagation for training DNNs.

In another work with Prof. Nhat Ho, Prof. Michael Jordan and Prof. Richard Baraniuk [20], I enable the DRMM to capture the dependency between latent variables, which results in the Deconvolutional Generative Model (DGM). The dependency between latent variables in DGM yields a new regularization over the set of latent variables, named the rendering path normalization, that is useful for semi-supervised learning tasks. I also establish statistical guarantees of parameter estimation in DGM and derive a generalization bound for (semi)-supervised learning tasks based on the DGM's structure.

In addition, in joint work with Yujia Huang (Ph.D. student), Prof. Doris Tsao, Prof. Anima Anandkumar and other collaborators [21], we use the DGM as a recurrent generative feedback for DNNs to enforce the self-consistency in DNNs for robust perception. Here, self-consistency implies that given the input data, the model can infer the latent variables and vice versus.

A mixture model framework for attention mechanism in transformers. Beyond DNNs, in joint work with Tam Nguyen (undergraduate student), Prof. Richard Baraniuk, Prof. Nhat Ho, Prof. Stanley Osher and other collaborators [22], I develop a probabilistic framework underlying attention mechanism in transformers and propose a new transformer with a mixture of Gaussian keys (Transformer-MGK), that replaces redundant heads in transformers with a mixture of keys at each head. In particular, despite their impressive performance, it has been observed that many heads in transformers learn redundant representations [23]. Reducing such redundancy to improve the effectiveness and efficiency of the model is one of the main focuses of current research in transformers. However, to mitigate this limitation and diversify the learned patterns are challenging without a mathematical framework for understanding attention mechanism in transformers. To address this problem, I derive a new Gaussian mixture model (GMM) for attention queries. Each Gaussian distribution in this mixture has an attention key as its mean. The posterior distributions of attention keys given attention queries in this mixture model correspond to the attention scores in self-attention, that capture the similarity between queries and keys. As a result, the GMM that I propose provides a principled probabilistic framework to study self-attention in transformers. Given this framework, I discover that a Gaussian distribution centered around each key has limited capacity to capture the distribution of attention queries since this distribution can be asymmetric, skewed or even multimodal. Therefore, I further propose to use a mixture of Gaussian keys to increase the representation power of the model so that attention keys can explain the queries better, resulting in the Transformer-MGK. I then derive the hard E-step inference, soft E-step inference, and the learning algorithm for Transformer-MGK. I also extend this model to use with linear attention, which leads to another new model named transformer with a mixture of linear keys (Transformer-MLK).

Along this direction, with Tam Nguyen, Prof. Stanley Osher, Prof. Nhat Ho and other collaborators, in [24], I extend mixture of keys in Transformer-MGK to a new finite admixture of keys (FiAK) for pruning redundant attention scores in transformers. In another work [25], from the observation that attention matrices, which are matrices of attention scores, in transformers lie on a low-dimensional manifold [26], I propose a new finite admixture of shared heads (FiSH) that generates many local attention matrices from a small set of global attention matrices, thus reducing both computational and memory costs. In [27], I extend my mixture model framework for self-attention to a nonparametric kernel regression model. I then propose the FourierFormer, a new class of transformers in which the dot-product kernels are replaced by the novel generalized Fourier integral kernels. Different from the

dot-product kernels, where a good covariance matrix needs to be chosen to capture the dependency of the features of data, the generalized Fourier integral kernels can automatically capture such dependency and remove the need to tune the covariance matrix.

4 Applications

In addition to developing principled machine learning models, I have also employed deep learning methods for applications in scientific machine learning including turbulence modeling and developing efficient numerical solvers.

Turbulence modeling. In [28], my collaborators and I propose a new machine learning methodology that captures, de novo, underlying turbulence phenomenology without a pre-specified model form. To illustrate the approach, we consider transient modeling of the dissipation of turbulent kinetic energy–a fundamental turbulent process that is central to a wide range of turbulence models–using NeuralODE models. After presenting details of the methodology, we show that this approach outperforms the state-of-the-art methods.

Efficient numerical solvers. In [29], with Prof. Animesh Garg, Prof. Richard Baraniuk and Prof. Anima Anandkumar, I study the speed-up for NeuralODEs and their related models including the continuous normalizing flows (CNFs) by tuning the error tolerances of the ODE solvers. With carefully selected error tolerances, NeuralODEs and CNFs can gain higher speed and better performance. However, the process of manually tuning the tolerances is time-consuming and requires a large amount of computational budget. To overcome this limitation, I propose a new method to learn the error tolerances of the ODE solvers in these models from input data via the REINFORCE algorithm. In an ongoing work with Professor Tan Bui, I draw a connection between transformers and various numerical ODE/PDE solvers and develop a new class of transformers for solving ODEs/PDEs.

5 Future research plans

The principled models that I have described above can be classified into the class of systems 1, which are models that do pattern recognition. For future work and looking forward to high-impact research directions, I am going to incorporate models that can do reasoning into my research agenda. These models are classified into the class of systems 2 [30]. To develop reasoning models, I am going to employ fixed point methods to learn good representation of the data that can generalize well. These fixed point methods add recurrence loops into the model and compute the representation of the data as a fixed-point solution of an implicit problem. This process is similar to how humans think carefully over and over again before making a decision in a challenging task such as playing chess. For a robot, this recurrent process can potentially help the robot understand the environment and the interaction with other robots around it. In my ongoing work, I observe that this fixed point method for designing models helps reduce the sample complexity of a reinforcement learning model that performs a reasoning task significantly. Other recent works have also observed the great capability to do reasoning of the models that learn with recurrent connections or fixed-point methods [31, 32]. In addition to fixed point methods, I will also explore other approaches to enable a model to reason including the recently proposed Generative Flow Networks [33].

References

[1] Karen E. Wilcox. Scientific machine learning: Where physics-based modeling meets data-driven learning. JuliaCon 2020, 2020.

[2] Chuizheng Meng, Sungyong Seo, Defu Cao, Sam Griesemer, and Yan Liu. When physics meets machine learning: A survey of physics-informed machine learning. *arXiv preprint arXiv:2203.16797*, 2022.

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[4] Tan Nguyen, Richard Baraniuk, Andrea Bertozzi, Stanley Osher, and Bao Wang. Momentumrnn: Integrating momentum into recurrent neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1924–1936. Curran Associates, Inc., 2020.

[5] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[6] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. Akad. Nauk Sssr*, volume 269, pages 543–547, 1983.

[7] Bao Wang, Tan M Nguyen, Andrea L Bertozzi, Richard G Baraniuk, and Stanley J Osher. Scheduled restart momentum for accelerated stochastic gradient descent. *SIAM Journal on Imaging Sciences*, 2022.

[8] Tan M Nguyen, Richard G Baraniuk, Mike Kirby, Stanley J Osher, and Bao Wang. Momentum transformer: Closing the performance gap between self-attention and its linearization. *Under review, Mathematical and Scientific Machine Learning (MSML), 2022*, 2022.

[9] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[10] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[11] Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Dissecting neural odes. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3952–3963. Curran Associates, Inc., 2020.

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

[13] Hedi Xia, Vai Suliafu, Hangjie Ji, Tan Minh Nguyen, Andrea Bertozzi, Stanley Osher, and Bao Wang. Heavy ball neural ordinary differential equations. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[14] Nghia Nguyen, Tan M Nguyen, Huyen Vo, Stanley J Osher, and Thieu Vo. Nesterov neural differential equations. *Under review, International Conference on Machine Learning (ICML)*, 2022.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[16] Tan Nguyen, Vai Suliafu, Stanley Osher, Long Chen, and Bao Wang. Fmmformer: Efficient and flexible transformer via decomposed near-field and far-field attention. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. S. Liang, J. W. Vaughan, and Y. Dauphin, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29449–29463. Curran Associates, Inc., 2021.

[17] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*, 2018.

[18] Matthew Thorpe, Tan Minh Nguyen, Hedi Xia, Thomas Strohmer, Andrea Bertozzi, Stanley Osher, and Bao Wang. GRAND++: Graph neural diffusion with a source term. In *International Conference on Learning Representations*, 2022.

[19] Ankit B Patel, Minh T Nguyen, and Richard Baraniuk. A probabilistic framework for deep learning. *Advances in neural information processing systems*, 29, 2016.

[20] Tan Nguyen, Nhat Ho, Ankit Patel, Anima Anandkumar, Michael I Jordan, and Richard G Baraniuk. A bayesian perspective of convolutional neural networks through a deconvolutional generative model. *Under review, Journal of Machine Learning Research*, 2022.

[21] Yujia Huang, James Gornet, Sihui Dai, Zhiding Yu, Tan Nguyen, Doris Tsao, and Anima Anandkumar. Neural networks with recurrent generative feedback. *Advances in Neural Information Processing Systems*, 33:535–545, 2020.

[22] Tam Nguyen, Tan M Nguyen, Dung Le, Khuong Nguyen, Anh Tran, Richard G Baraniuk, Nhat Ho, and Stanley J Osher. Improving transformers with probabilistic attention keys. *Under review, International Conference on Machine Learning (ICML)*, 2022.

[23] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? 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.

[24] Tan M Nguyen, Tam Nguyen, Long Bui, Hai Do, Dung Le, Hung Tran-The, Khuong Nguyen, Richard G Baraniuk, Nhat Ho, and Stanley J Osher. A probabilistic framework for pruning transformers via a finite admixture of keys. *Under review, European Conference on Computer Vision (ECCV)*, 2022.

[25] Tan M Nguyen, Tam Nguyen, Hai Do, Khai Nguyen, Vishwanath Saragadam, Minh Pham, Khuong Nguyen, Nhat Ho, and Stanley J Osher. Fishformer: Transformer with a finite admixture of shared heads. *Under review, International Conference on Machine Learning (ICML)*, 2022.

[26] Srinadh Bhojanapalli, Ayan Chakrabarti, Himanshu Jain, Sanjiv Kumar, Michal Lukasik, and Andreas Veit. Eigen analysis of self-attention and its reconstruction from partial computation. *arXiv preprint arXiv:2106.08823*, 2021.

[27] Tan Nguyen, Minh Pham, Tam Nguyen, Khai Nguyen, Stanley J Osher, and Nhat Ho. Transformer with fourier integral attentions. *arXiv preprint arXiv:2206.00206*, 2022.

[28] Gavin D Portwood, Peetak P Mitra, Mateus Dias Ribeiro, Tan Minh Nguyen, Balasubramanya T Nadiga, Juan A Saenz, Michael Chertkov, Animesh Garg, Anima Anandkumar, Andreas Dengel, et al. Turbulence forecasting via neural ode. *NeurIPS Workshop on Machine Learning and the Physical Sciences*, 2019.

[29] Tan M Nguyen, Animesh Garg, Richard G Baraniuk, and Anima Anandkumar. Infocnf: An efficient conditional continuous normalizing flow with adaptive solvers. *arXiv preprint arXiv:1912.03978*, 2019.

[30] Yoshua Bengio. From system 1 deep learning to system 2 deep learning. Conference on Neural Information Processing Systems (NeurIPS), 2019, 2019.

[31] Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[32] Arpit Bansal, Avi Schwarzschild, Eitan Borgnia, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. End-to-end algorithm synthesis with recurrent networks: Logical extrapolation without overthinking. *arXiv preprint arXiv:2202.05826*, 2022.

[33] Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34, 2021.