

# Differential Equations and Continuous-Time Deep Learning

David Duvenaud<sup>\*1</sup>, Markus Heinonen<sup>\*2</sup>, Michael Tiemann<sup>\*3</sup>, and Max Welling<sup>\*4</sup>

1 University of Toronto, CA. [duvenaud@cs.toronto.edu](mailto:duvenaud@cs.toronto.edu)

2 Aalto University, FI. [markus.o.heinonen@aalto.fi](mailto:markus.o.heinonen@aalto.fi)

3 Robert Bosch GmbH – Renningen, DE. [michael.tiemann@de.bosch.com](mailto:michael.tiemann@de.bosch.com)

4 University of Amsterdam, NL. [welling.max@gmail.com](mailto:welling.max@gmail.com)

---

## Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 22332 “Differential Equations and Continuous-Time Deep Learning”. Neural ordinary-differential equations and similar continuous model architectures have gained interest in recent years, due to the existence of a vast literature in calculus and numerical analysis. Thus, continuous models might lead to architectures with finer control over prior assumptions or theoretical understanding. In this seminar, we have sought to bring together researchers from traditionally disjoint areas – machine learning, numerical analysis, dynamical systems and their “consumers” – to try and develop a joint language about this novel modeling paradigm. Through talks & group discussions, we have identified common interests and we hope that this first seminar is but the first step on a joint journey.

**Seminar** August 15–19, 2022 – <http://www.dagstuhl.de/22332>

**2012 ACM Subject Classification** Computing methodologies → Machine learning; Computing methodologies → Philosophical/theoretical foundations of artificial intelligence; Mathematics of computing → Differential equations; Mathematics of computing → Solvers

**Keywords and phrases** deep learning, differential equations

**Digital Object Identifier** 10.4230/DagRep.12.8.20

## 1 Executive Summary

*David Duvenaud*

*Markus Heinonen*

*Michael Tiemann*

*Max Welling*

**License**  Creative Commons BY 4.0 International license

© David Duvenaud, Markus Heinonen, Michael Tiemann, and Max Welling

Deep models have revolutionised machine learning due to their remarkable ability to iteratively construct more and more refined representations of data over the layers. Perhaps unsurprisingly, very deep learning architectures have recently been shown to converge to differential equation models, which are ubiquitous in sciences, but so far overlooked in machine learning. This striking connection opens new avenues of theory and practice of continuous-time machine learning inspired by physical sciences. Simultaneously, neural networks have started to emerge as powerful alternatives to cumbersome mechanistic dynamical systems. Finally, deep learning models in conjunction with stochastic gradient optimisation has been used to numerically solve high-dimensional partial differential equations. Thus, we have entered a new era of continuous-time modelling in machine learning.

---

\* Editor / Organizer



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 4.0 International license

Differential Equations and Continuous-Time Deep Learning, *Dagstuhl Reports*, Vol. 12, Issue 8, pp. 20–30  
Editors: David Duvenaud, Markus Heinonen, Michael Tiemann, and Max Welling



Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This change in perspective is currently gaining interest rapidly across domains and provides an excellent and topical opportunity to bring together experts in dynamical systems, computational science, machine learning and the relevant scientific domains to lay solid foundations of these efforts. On the other hand, as the scientific communities, events and outlets are significantly disjoint, it is key to organize an interdisciplinary event and establish novel communication channels to ensure the distribution of relevant knowledge.

Over the course of this Dagstuhl Seminar, we want to establish strong contacts, communication and collaboration of the different research communities. Let's have an exchange of each community's best practices, known pitfalls and tricks of the trade. We will try to identify the most important open questions and avenues forward to foster interdisciplinary research. To this end, this seminar will feature not only individual contributed talks, but also general discussions and "collaboration bazaars", for which participants will have the possibility to pitch ideas for break-out project sessions to each other. In the break-out sessions, participants may discuss open problems, joint research obstacles, or community building work.

## 2 Table of Contents

### Executive Summary

*David Duvenaud, Markus Heinonen, Michael Tiemann, and Max Welling* . . . . . 20

### Overview of Talks

Differential Equations for Causal Inference in Complex Stochastic Biological Processes  
*Hananeh Aliee* . . . . . 23

Computation Theory for Continuous Time. Programming with Ordinary Differential Equations.  
*Olivier Bournez* . . . . . 23

Injecting Physics into Differential Equation based Deep Learning Models  
*Biswadip Dey* . . . . . 23

Equivariant Deep Learning via PDEs  
*Remco Duits* . . . . . 24

Putting All of Modeling into Adaptive SPDE Solvers  
*David Duvenaud* . . . . . 24

Bayesian Calibration of Computer Models & Beyond  
*Maurizio Filippone* . . . . . 25

High Order SDE Solvers in Machine Learning  
*James Foster* . . . . . 25

Interpretable Polynomial Neural Ordinary Differential Equations  
*Colby Fronk* . . . . . 25

Neural Differential Equations and Operator Learning  
*Jacob Seidman* . . . . . 26

On Practical Inference and Learning in Dynamical Systems  
*Arno Solin* . . . . . 26

Partial Differential Equations and Deep Learning  
*Nils Thuerey* . . . . . 26

Dynamical Systems Cookbook (& their solvers, & their optimization)  
*Michael Tiemann* . . . . . 27

Graph-based Differential Equations, Continuum Limits, and Merriman-Bence-Osher schemes  
*Yves van Gennip* . . . . . 27

### Working groups

Brainstorm session  
*Yves van Gennip, Olivier Bournez, Joachim M. Buhmann, Remco Duits, Sho Sonoda, and Max Welling* . . . . . 27

**Participants** . . . . . 30

### 3 Overview of Talks

#### 3.1 Differential Equations for Causal Inference in Complex Stochastic Biological Processes

*Hananeh Aliee (Helmholtz Zentrum München, DE)*

License  Creative Commons BY 4.0 International license  
© Hananeh Aliee

In my talk, I presented a sparsity-enforcing regularizer for continuous-time neural networks motivated by causality. Sparsification can help to identify the parameters of the differential equations and infer the causal interaction between variables. I also discussed an application of that in single-cell genomics for modeling gene dynamics and inferring gene regulatory networks using neural ODEs. Finally, I discussed some open problems and challenges in modeling complex stochastic biological processes and potential directions for future work.

#### 3.2 Computation Theory for Continuous Time. Programming with Ordinary Differential Equations.

*Olivier Bournez (Ecole Polytechnique – Palaiseau, FR)*

License  Creative Commons BY 4.0 International license  
© Olivier Bournez

In this talk, we will argue that computation theory for continuous time analog models did not develop at the level as the one for digital models. We will review some examples of such models, such as the General Purpose Analog Computer (GPAC) from Claude Shannon, proposed as a model of Differential Analyzers. We will show how this model can be programmed on several example. We will then discuss about how this model relates to classical models of computability such as Turing machines, both considering computability theory and complexity theory. We will show the close relation between this model and polynomial Ordinary Differential Equations (pODEs). As a side effect of our constructions, we will see that one can program with pODEs and we will discuss applications.

#### 3.3 Injecting Physics into Differential Equation based Deep Learning Models

*Biswadip Dey (Siemens – Princeton, US)*

License  Creative Commons BY 4.0 International license  
© Biswadip Dey

This talk focused on demonstrating the usefulness of using a physics-informed inductive bias in differential equation based deep learning models and highlighted some open problems on this topic. We discussed Symplectic-ODENet and its extensions which encode energy conservation into the computation graph to improve model performance, efficiency, and interpretability. However, these models typically assumes that the systems states can be directly measured. This leads to the following open questions: (i) Can we learn a suitable latent representation from high-dimensional observations and then enforce physics (e.g., energy conservation) in the learned latent space? and (ii) Can we enforce physics even when only a subset of the system states can be directly measured?

### 3.4 Equivariant Deep Learning via PDEs

*Remco Duits (TU Eindhoven, NL)*

License  Creative Commons BY 4.0 International license  
© Remco Duits

We consider PDE-based Group Convolutional Neural Networks (PDE-G-CNNs) that generalize Group equivariant Convolutional Neural Networks (G-CNNs). In PDE-G-CNNs a network layer is a set of PDE-solvers where geometrically meaningful PDE-coefficients become trainable weights. The underlying PDEs are morphological and linear scale space PDEs on the homogeneous space of positions and orientations to the roto-translation group  $SE(2)$ . The PDEs provide a geometrical and probabilistic understanding of the network. The network is implemented by morphological convolutions with approximations to kernels solving nonlinear HJB-PDEs (for morphological  $\alpha$ -scale spaces), and to linear convolutions solving linear PDEs (for linear  $\alpha$ -scale spaces). In the morphological setting, the parameter  $\alpha$  regulates soft max-pooling over Riemannian balls, whereas in the linear setting the cases  $\alpha = 1/2$  and  $\alpha = 1$  correspond to the Poisson and Gaussian semigroup. We prove that our practical analytic approximation kernels are accurate. In the morphological setting, we propose analytic approximations of (sub)-Riemannian balls on  $M(2)$  which carry the correct reflectional symmetries globally and we provide asymptotic error analysis. The analytic approximations allow for efficient, accurate training of fundamental neuro-geometrical association field models in the GPU-implementations of our PDE-G-CNNs. The equivariant PDE-G-CNN network implementation consists solely of linear and morphological convolutions with parameterized analytic kernels on  $M(d)$ . Common mystifying nonlinearities in CNNs are now obsolete and excluded. We present blood vessel segmentation experiments in medical images that show clear benefits of PDE-G-CNNs compared to state-of-the-art G-CNNs: increase of performance along with a huge reduction in network parameters and training data.

### 3.5 Putting All of Modeling into Adaptive SPDE Solvers

*David Duvenaud (University of Toronto, CA)*

License  Creative Commons BY 4.0 International license  
© David Duvenaud

My talk presented a roadmap for building spatiotemporal models which can automatically introduce auxiliary variables. These auxiliary variables can be tuned jointly with the parameters of the model to find dynamics which are easy to integrate, either by encouraging approximate spatial factorization, or fast mixing temporally. I also introduced a scheme for stateless sampling from Brownian sheets.

### 3.6 Bayesian Calibration of Computer Models & Beyond

*Maurizio Filippone (EURECOM – Biot, FR)*

License  Creative Commons BY 4.0 International license  
© Maurizio Filippone

Bayesian calibration of computationally expensive computer models offers an established framework for quantification of uncertainty of model parameters and predictions. Traditional Bayesian calibration involves the emulation of the computer model and an additive model discrepancy term using Gaussian processes; inference is then carried out using Markov chain Monte Carlo. In this talk, I present a calibration framework where limited flexibility and scalability are addressed by means of compositions of Gaussian processes into Deep Gaussian processes and scalable variational inference techniques. This formulation can be easily implemented in development environments featuring automatic differentiation and exploiting GPU-type hardware. I then discuss identifiability issues and cases where the computer model implements ODEs/PDEs/SDEs. Finally, I draw connections with other inference frameworks, such as transfer learning, gradient matching for ODEs and SDEs, and Physics-informed priors for Bayesian deep learning.

### 3.7 High Order SDE Solvers in Machine Learning

*James Foster (University of Oxford, GB)*

License  Creative Commons BY 4.0 International license  
© James Foster

From Markov Chain Monte Carlo to Neural SDEs and Score-based diffusions, there has been a recent uptick in the applications of SDEs in machine learning. However, SDEs have been studied by the mathematics community for decades and it has been well established that SDE solvers have fundamental limitations in their convergence rates. In this talk, we will review this theory and discuss how noise types influence convergence rates for SDEs solvers. This will naturally lead us to pose the following question:

“Can we construct SDEs that are easy to solve?”

By considering both kinetic Langevin and Score-based diffusions, two prominent examples of SDEs, we give a positive answer to this question and speculate that finding such “easy-to-solve” SDEs will be an area of opportunity in future research.

### 3.8 Interpretable Polynomial Neural Ordinary Differential Equations

*Colby Fronk (University of California – Santa Barbara, US)*

License  Creative Commons BY 4.0 International license  
© Colby Fronk

Neural networks have the ability to serve as universal function approximators, but they are not interpretable and don’t generalize well outside of their training region. Both of these issues are problematic when trying to apply standard neural ordinary differential equations (neural ODEs) to dynamical systems. We introduce the polynomial neural ODE, which is a deep polynomial neural network inside of the neural ODE framework. We demonstrate the capability of polynomial neural ODEs to predict outside of the training region, as well as perform direct symbolic regression without additional tools such as SINDy.

### 3.9 Neural Differential Equations and Operator Learning

*Jacob Seidman (University of Pennsylvania – Philadelphia, US)*

License  Creative Commons BY 4.0 International license  
© Jacob Seidman

My talk presented two categories of methods to learn maps between spaces of functions. The first is known as Neural PDEs/SDEs and parameterizes PDEs/SDEs to implicitly define operators through their solutions. The other category is typically known as Operator Learning and uses compositions of parameterized integral transformations, pointwise transformations, and function reconstructions from learned basis or nonlinear representations. I posed the question of which approach works better in different scenarios. This led to a discussion about the pros and cons of each approach in terms of properties such as expressivity, ability to encode prior information, and computational efficiency.

### 3.10 On Practical Inference and Learning in Dynamical Systems

*Arno Solin (Aalto University, FI)*

License  Creative Commons BY 4.0 International license  
© Arno Solin

In general spatio-temporal systems, time takes a fundamentally different role from other (spatial) dimensions as observations can be ordered over time. This talk takes interest in challenges in online inference and learning problems, where the model admits the form of a stochastic differential equation (SDE) or stochastic partial differential equation (SPDE). These types of problems occur naturally in sensor fusion applications where the dynamics borrow from first principles but also include unknown (stochastic) effects. The talk presents open problems in designing principled approximate inference methods, with non-linear continuous-discrete inertial navigation as a practical example.

### 3.11 Partial Differential Equations and Deep Learning

*Nils Thuerey (TU München, DE)*

License  Creative Commons BY 4.0 International license  
© Nils Thuerey

In my talk I focused on the combination of PDE for applications such as fluids and deep learning (DL). Despite success of integrating solvers as differentiable components in DL, many challenges for training remain. Interestingly, the regular gradient has some fundamental problems, as indicated by its mismatch in terms of units. I discussed potential avenues for alleviating these problems, such as using inverse solvers of partial inversions.

### 3.12 Dynamical Systems Cookbook (& their solvers, & their optimization)

*Michael Tiemann (Robert Bosch GmbH – Renningen, DE)*

License  Creative Commons BY 4.0 International license  
© Michael Tiemann

In many areas of science and engineering, neural ordinary differential equations seem like natural candidates for extending limited first-principles models. However, retaining an interpretability in terms of preserved quantities of interest or system properties, such as volume invariances, preserved first integrals and other conservation laws, requires an algebra of models that represent a wide variety of dynamical systems, while guaranteeing the preservation of these quantities by construction. In this call for contributions, we hope to establish a grass-roots initiative that will contribute to cookbook of building blocks that represent a wide variety of potential applications, while working reliability “out-of-the-box” for the majority of modeling problems. Furthermore, this cookbook needs to consider not only the algebra of the ODE vector fields, but also that of the numerical discretizations and finally of their identification through means of optimization or other adaptation methods.

### 3.13 Graph-based Differential Equations, Continuum Limits, and Merriman-Bence-Osher schemes

*Yves van Gennip (TU Delft, NL)*

License  Creative Commons BY 4.0 International license  
© Yves van Gennip

Ideas and methods from differential equations and variational methods on graphs can also play a role for neural networks (NN). In particular, we take a look at the Merriman–Bence–Osher (MBO) scheme and the family of semi-discrete implicit Euler (SDIE) schemes and see that they can be written as NN. We also discuss discrete-to-continuum limits at the variational and gradient flow levels and open questions.

## 4 Working groups

### 4.1 Brainstorm session

*Yves van Gennip (TU Delft, NL), Olivier Bournez (Ecole Polytechnique – Palaiseau, FR), Joachim M. Buhmann (ETH Zürich, CH), Remco Duits (TU Eindhoven, NL), Sho Sonoda (RIKEN – Tokyo, JP), and Max Welling (University of Amsterdam, NL)*

License  Creative Commons BY 4.0 International license  
© Yves van Gennip, Olivier Bournez, Joachim M. Buhmann, Remco Duits, Sho Sonoda, and Max Welling

#### 1. local-nonlocal interactions

We asked the question if the PDE models with local derivatives can be generalized to more general non-local (integral) operators. We believe this is possible and would lead to genuinely new models that would be better in modeling problems with highly nonlocal interactions.

## 2. multi-scale/renormalisation group

We asked if there is merit to introduce a scale-space into the representations. For instance, every layer can represent a full scale space, or the progression through the layers represents a coarse graining transformation. The former can be viewed as a special case of scale equivariance (a semi-group!), while the latter is more like a renormalization group transformation.

## 3. equivariance/symmetries\*/local equivariance (\*in continuum formulation and after discretisation)

We discussed if there are extensions to equivariance to non-group transformations (e.g. semi-groups see above). Also, if we formulate the NN as a PDE in the continuum limit, we can model symmetries also as a transformation with a generator that commutes with the Hamiltonian. Can we think of equivariance as simply finding a homomorphism in the hidden layers that forms a commuting diagram with the transformations in the input layer: transformation in input layer  $\rightarrow$  embedding to hidden layer = embedding to hidden layer  $\rightarrow$  transformation in hidden layer. We also discussed the role of local versus global symmetries: to what extent does a global equivariance also enforce local equivariance. Can this be formalized? Can this be generalized to diffeomorphisms?

## 4. quantum extensions (learning unitary operators in quantum computers)

The Schrodinger equation is also a PDE. We can extend the continuum PDE limit of a linear layer to a quantum layer by evolving an input quantum state using the SE. This maps to a model for optical quantum hardware. Is this beneficial, or more powerful for ML? Can we include symmetries?

## 5. conserved quantities/Noether's theorem (see also 3)

We discussed what could be at the basis of a general theory and thought the notion of conserved quantities to be a good candidate. We discussed how to apply Hamiltonian reduction by stages (Marsden) from classical mechanics to deep learning.

## 6. (geometrical flow?) interpretations of full networks (example: mean curvature flow)

We saw one particular example of a CNN that appears to be interpretable in terms of mean curvature flow. This raises a more general question regarding the possibility of interpreting NNs (not per layer, but as a whole) in terms of geometrical flows.

PDE-G-CNNs provide geometric interpretation of flows in the neural networks. PDE-G-CNNs, in contrast to CNNs, do not include ad-hoc nonlinearities, but only solutions to linear and nonlinear PDEs, both solved by equivariant convolutions over different semirings. The merging of association fields as visible in feature maps of PDE-G-CNNs requires algebraic geometry (Betti numbers). We discussed Lie group extensions of recent works of Creemers, but realized it is quite challenging.

## 7. new operators/integral operators

We discussed if we also want to consider nonlocal operators (besides classical "local" PDEs), such as non-local derivatives and fractional powers of semi-group evolutions, in the network layers we consider, or if nonlocalities are only allowed to appear as a result of interactions between many layers.

## 8. why is deep better than wide? (linear vs polynomial scaling of "influence" of neurons?)

Why do deep NNs perform better than wide ones? The initial thought is that interactions between layers scale nonlinearly in the number of neurons, whereas interactions within a layer scale linearly.

**9. how to design nonlinearities?**

We prefer HJB-PDEs that allow for morphological convolutions, not by (ad-hoc) ReLU's that are a non-optimal special case.

**10. continuum limits**

Techniques exist in the mathematical literature to find continuum limits of (loss) functionals, such as Gamma-convergence, and limits of gradient flows derived from those functionals. Can we employ those to prove relevant continuum limits for neural networks?

**11. wishlist for PDEs (equivariance; semi-group structure; homogeneity in metric tensor per "unit")**

We wish for an axiomatic approach to PDE-based equivariant deep learning with Lie-group domains and semi-ring co-domains.

We started investigations on that after the seminar and will continue to work on this thoroughly in the coming year.

**12. PDE GCNNs on graphs.**

We noted that equivariant networks on  $SE(3)$  in general require sparsification to become practical in view of memory management. The PDEs can enter by providing appropriate kernels for equivariant graph neural networks.

## Participants

- Hananeh Aliee  
Helmholtz Zentrum  
München, DE
- Jesse Bettencourt  
University of Toronto, CA
- Olivier Bournez  
Ecole Polytechnique –  
Palaiseau, FR
- Joachim M. Buhmann  
ETH Zürich, CH
- Johanne Cohen  
University Paris-Saclay –  
Orsay, FR
- Biswadip Dey  
Siemens – Princeton, US
- Remco Duits  
TU Eindhoven, NL
- David Duvenaud  
University of Toronto, CA
- Maurizio Filippone  
EURECOM – Biot, FR
- James Foster  
University of Oxford, GB
- Colby Fronk  
University of California –  
Santa Barbara, US
- Jan Hasenauer  
Universität Bonn, DE
- Markus Heinonen  
Aalto University, FI
- Patrick Kidger  
Google X – Bay Area, US
- Diederik P. Kingma  
Google – Mountain View, US
- Linda Petzold  
University of California –  
Santa Barbara, US
- Jack Richter-Powell  
University of Toronto, CA
- Lars Ruthotto  
Emory University – Atlanta, US
- Jacob Seidman  
University of Pennsylvania –  
Philadelphia, US
- Arno Solin  
Aalto University, FI
- Sho Sonoda  
RIKEN – Tokyo, JP
- Nils Thuerey  
TU München, DE
- Michael Tiemann  
Robert Bosch GmbH –  
Renningen, DE
- Filip Tronarp  
Universität Tübingen, DE
- Yves van Gennip  
TU Delft, NL
- Max Welling  
University of Amsterdam, NL
- Verena Wolf  
Universität des Saarlandes –  
Saarbrücken, DE
- Daniel Worrall  
DeepMind – London, GB

