Deep Learning in Physics

Qiang Liu and Patrick Schnell

15.10.2025

Qiang Liu
PhD candidate

qiang7.liu@tum.de

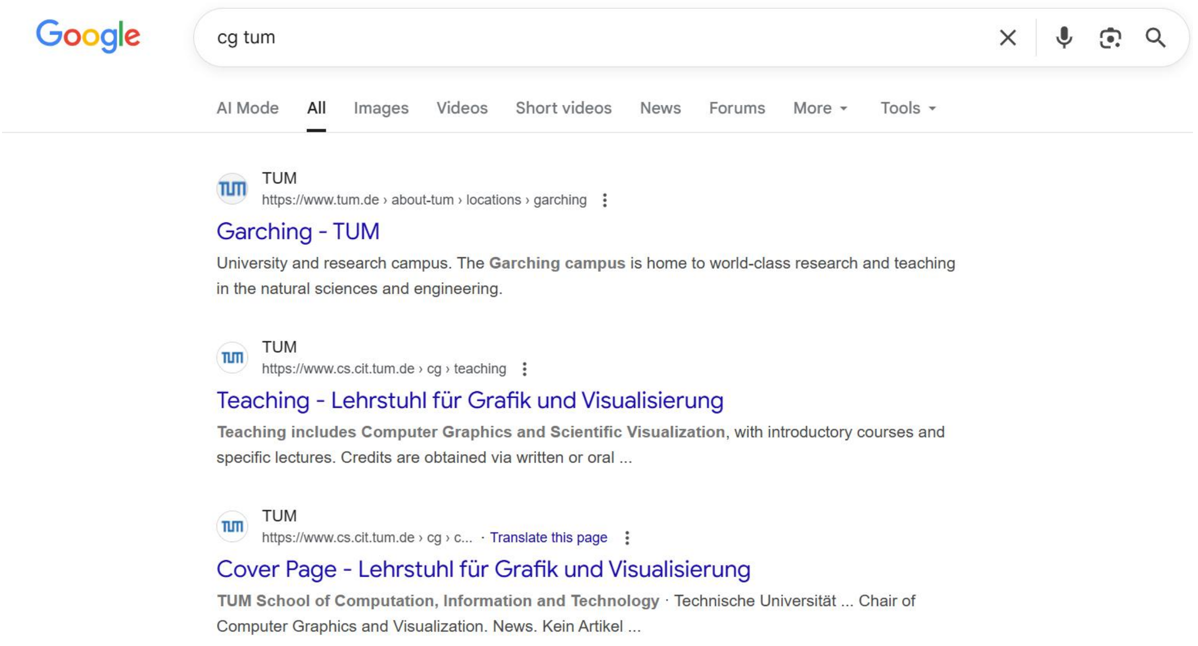
Patrick Schnell
PhD candidate

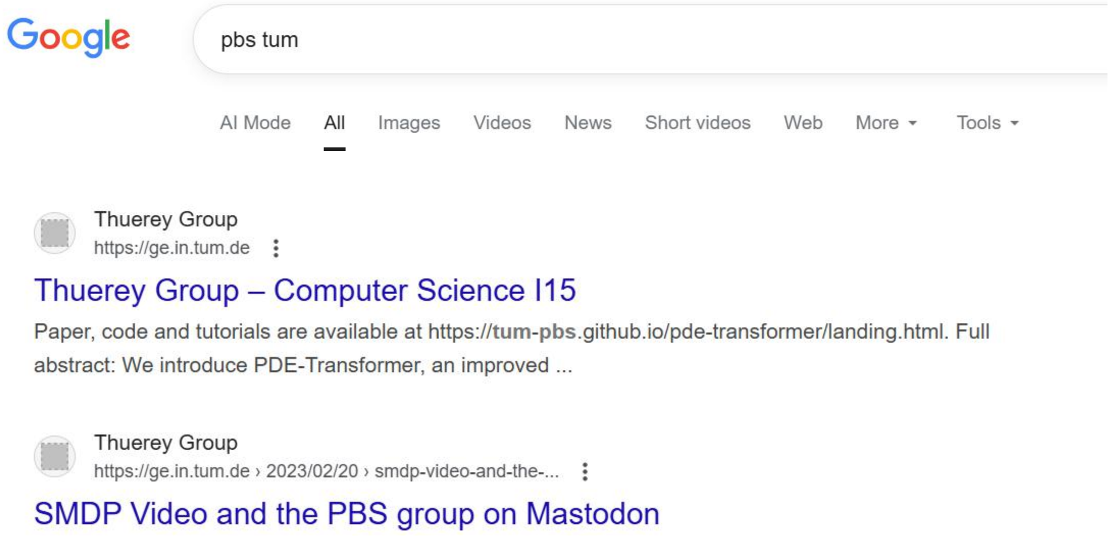
patrick.schnell@tum.de

Prof. Nils Thuerey

Course page:
https://www.cs.cit.tum.de/cg/teaching/winter-term-25-26/deep-learning-in-physics/

Group page:
https://ge.in.tum.de/





About this Seminar

- Research topics in **deep learning for physics**
  - Learning algorithms
  - Architectures
  - Applications
- Familiarize yourself with the underlying physics & ML applicability
- Students conduct independent analyses of the topic and related work
- Develop writing & presentation skills
- Submission: **Presentation slides, Report**

Deep Learning in Physics

Neural Operator

Unrolling

Differentiable Simulation

PINNs

Deep Learning in Physics

Neural Operator

Unrolling

Differentiable Simulation

PINNs

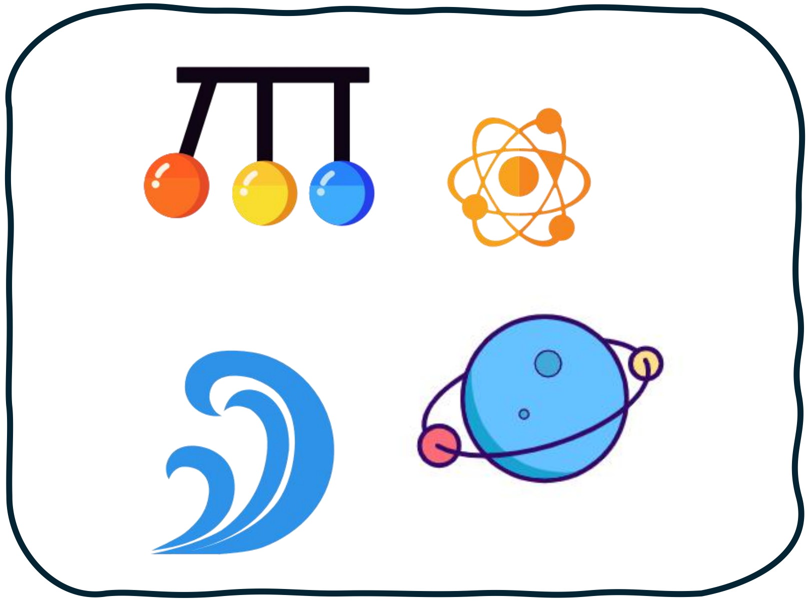
Transformers

Reinforcement Learning

Generative Models

Deep Learning in Physics

Physics is the use of Mathematics to describe the World



$$F = G\frac{Mm}{r^2}$$

$$Q = I^2 Rt$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}$$

$$l = l_0 \sqrt{1 - (\frac{v}{c})^2}$$

In this seminar, we mainly discuss PDEs.

In mathematics, a **partial differential equation** (**PDE**) is an equation which involves a multivariable function and one or more of its partial derivatives.

Partial differential equations are ubiquitous in mathematically oriented scientific fields, such as physics and engineering. For instance, they are foundational in the modern scientific understanding of sound, heat, diffusion, electrostatics, electrodynamics, thermodynamics, fluid dynamics, elasticity, general relativity, and quantum mechanics (Schrödinger equation, Pauli equation etc.). They also arise from many purely mathematical considerations, such as differential geometry and the calculus of variations; among other notable applications, they are the fundamental tool in the proof of the Poincaré conjecture from geometric topology.

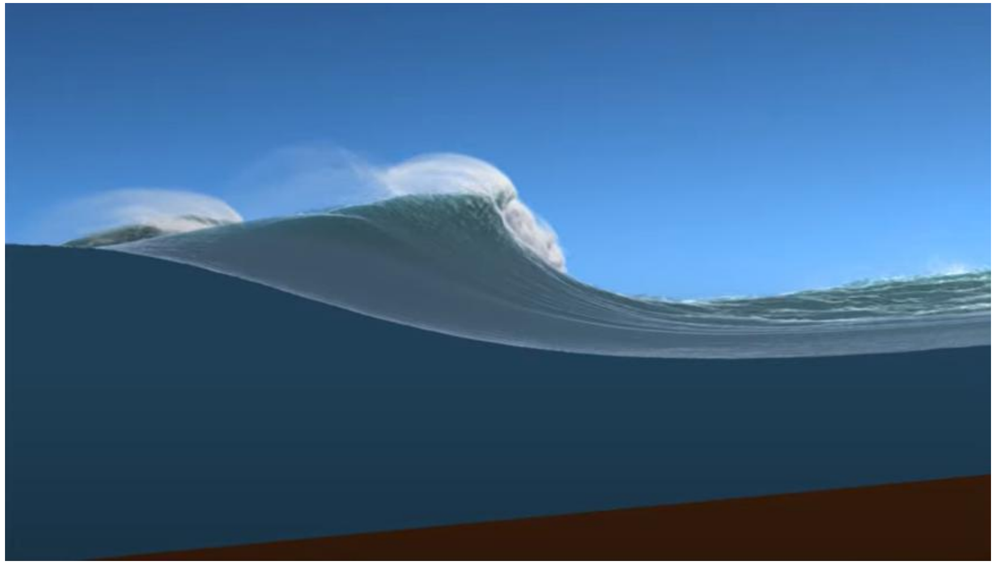
https://en.wikipedia.org/wiki/Partial_differential_equation

$$\frac{\partial \mathbf{u}}{\partial t} = f(\mathbf{u}, \frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial^2 \mathbf{u}}{\partial \mathbf{x^2}}, \cdots)$$

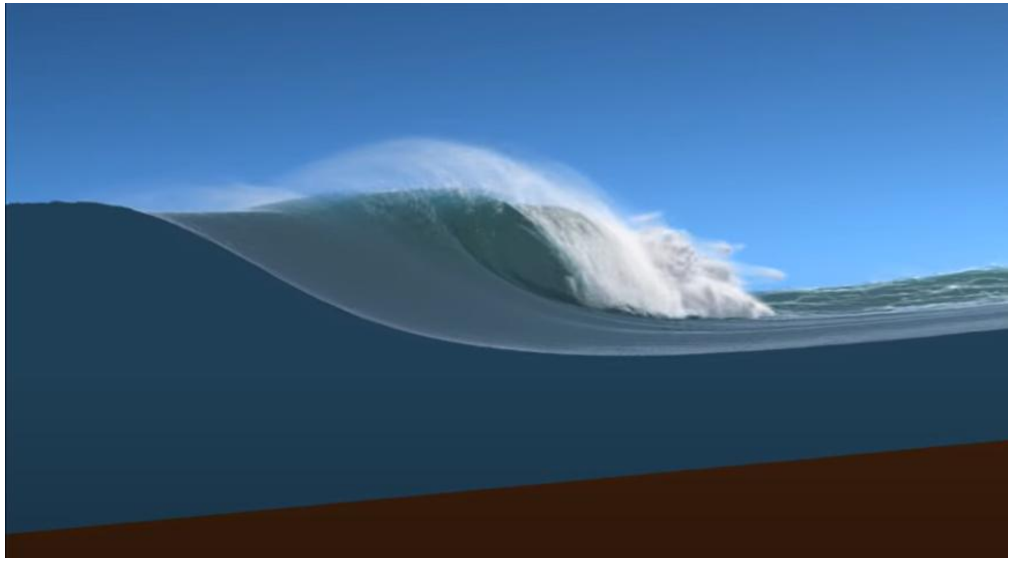
How a physical system evolves over time.

In this seminar, we mainly discuss PDEs.

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nabla \cdot (\nu \nabla \mathbf{U}) = -\nabla p$$



$\mathbf{U}_t$



$\mathbf{U}_{t+1}$

Initial value problem

Given a **state** of a physical system, and its **governing PDEs** (dynamics), how do we obtain any **future state** of this physical system?

$$\frac{\partial \mathbf{u}}{\partial t} = f(\mathbf{u}, \frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial^2 \mathbf{u}}{\partial \mathbf{x^2}}, \cdots)$$

Time integration

Euler
Runge–Kutta methods

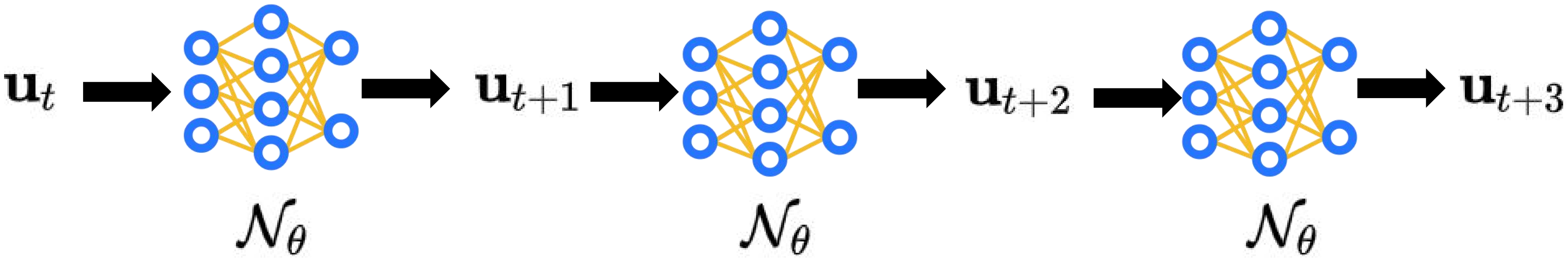
Spatial derivatives

Finite differences
Finite element
Finite volume
Spectral method
…

$$\mathbf{u}_{t+1} = \mathbf{u}_t + f(\mathbf{u}_t)\delta t$$

Neural Operators

Given a **state** of a physical system, and its **governing PDEs** (dynamics), how do we obtain any **future state** of this physical system?



$$\mathcal{L} = ||\mathcal{N}_\theta(u_t) - u_{t+1}||$$

- DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators
- Fourier Neural Operator for Parametric Partial Differential Equations

Neural Operators

- DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators
- Fourier Neural Operator for Parametric Partial Differential Equations

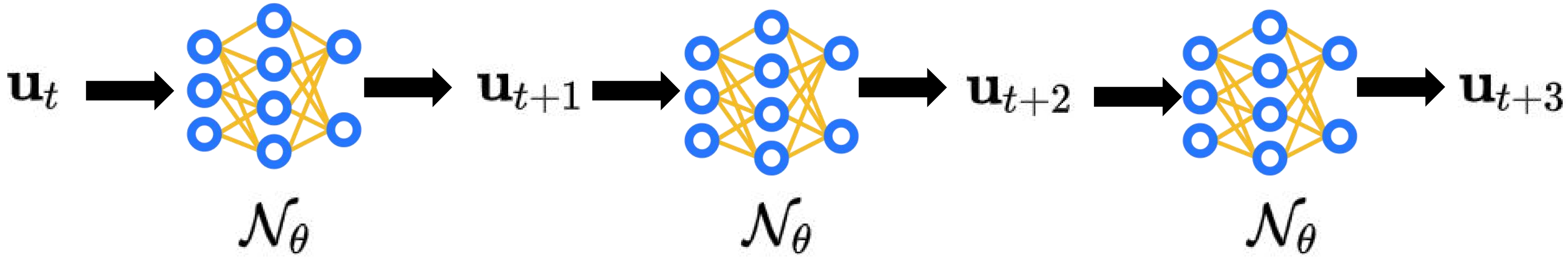
Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs

Nikola Kovachki[*,†] NKOVACHKI@NVIDIA.COM *Nvidia*
Zongyi Li[*] ZONGYILI@CALTECH.EDU *Caltech*
Burigede Liu BL377@CAM.AC.UK *Cambridge University*
Kamyar Azizzadenesheli KAMYARA@NVIDIA.COM *Nvidia*
Kaushik Bhattacharya BHATTA@CALTECH.EDU *Caltech*
Andrew Stuart ASTUART@CALTECH.EDU *Caltech*
Anima Anandkumar ANIMA@CALTECH.EDU *Caltech*

What are the **differences** between neural operators and other regression networks?

What is the **function space**? Why it is important to learn in function space?

Unrolling



$$\mathcal{L} = ||\mathcal{N}_\theta(u_t) - u_{t+1}||$$

$$\mathcal{L} = ||\mathcal{N}_\theta(\mathcal{N}_\theta(\mathcal{N}_\theta(u_t))) - u_{t+2}||_2$$

- Unrolling describes the process of **auto-regressively** evolving the learned system in a **training iteration**

What are the challenges of unrolling?

How to solve these challenges?

- Low-Variance Gradient Estimation in Unrolled Computation Graphs with ES-Single
- The curse of unrolling

Differentiable simulations

$$\frac{\partial \mathbf{u}}{\partial t} = f(\mathbf{u}, \frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial^2 \mathbf{u}}{\partial \mathbf{x}^2}, \cdots)$$

Time integration

Euler
Runge–Kutta methods

Spatial derivatives

Finite differences
Finite element
Finite volume
Spectral method
…

Integrate **simulators** into deep learning pipelines

- What does “differentiable”mean ?
- Why the simulator needs to be differentiable for deep learning tasks?
- How does a differentiable solver work with the neural networks?

- Learning to control PDEs with differentiable physics
- Do Differentiable Simulators Give Better Policy Gradients?
- Accelerated Policy Learning With Parallel Differentiable Simulation

Differentiable simulations

$$\frac{\partial \mathbf{u}}{\partial t} = f\left(\mathbf{u}, \frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial^2 \mathbf{u}}{\partial \mathbf{x^2}}, \cdots\right)$$

Time integration

Euler
Runge–Kutta methods

Spatial derivatives

Finite differences
Finite element
Finite volume
Spectral method
…

Integrate **simulators** into deep learning pipelines

- What does “differentiable”mean ?
- Why the simulator needs to be differentiable for deep learning tasks?
- How does a differentiable solver work with the neural networks?

- Learning to control PDEs with differentiable physics
- Do Differentiable Simulators Give Better Policy Gradients?
- Accelerated Policy Learning With Parallel Differentiable Simulation

PINNs

Simulation: $\mathbf{u}_{t+1} = \mathbf{u}_t + f(\mathbf{u}_t)\delta t$

Regression: $\mathbf{u}_{t+1} = \mathcal{N}_\theta(\mathbf{u}_t)$

PINNs: $\mathbf{u}_{t+1}(\mathbf{x}) = \mathcal{N}_\theta(t+1, \mathbf{x})$

$$\frac{\partial \mathbf{u}}{\partial t} = f(\mathbf{u}, \frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial^2 \mathbf{u}}{\partial \mathbf{x}^2}, \cdots)$$

$$\frac{\partial \mathcal{N}_\theta(\mathbf{x}, t)}{\partial t} = f\left(\mathcal{N}_\theta(\mathbf{x}, t), \frac{\partial \mathcal{N}_\theta(\mathbf{x}, t)}{\partial \mathbf{x}}, \frac{\partial^2 \mathcal{N}_\theta(\mathbf{x}, t)}{\partial \mathbf{x}^2}, \cdots\right)$$

- How does PINNs calculate the derivatives?
- What are the difficulties training PINNs and how to solve them?

PINNs

Simulation: $\mathbf{u}_{t+1} = \mathbf{u}_t + f(\mathbf{u}_t)\delta t$

Regression: $\mathbf{u}_{t+1} = \mathcal{N}_\theta(\mathbf{u}_t)$

PINNs: $\mathbf{u}_{t+1}(\mathbf{x}) = \mathcal{N}_\theta(t_{+1}, \mathbf{x})$

$$\frac{\partial \mathbf{u}}{\partial t} = f(\mathbf{u}, \frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial^2 \mathbf{u}}{\partial \mathbf{x}^2}, \cdots)$$

$$\frac{\partial \mathcal{N}_\theta(\mathbf{x}, t)}{\partial t} = f\left(\mathcal{N}_\theta(\mathbf{x}, t), \frac{\partial \mathcal{N}_\theta(\mathbf{x}, t)}{\partial \mathbf{x}}, \frac{\partial^2 \mathcal{N}_\theta(\mathbf{x}, t)}{\partial \mathbf{x}^2}, \cdots\right)$$

? How does PINNs calculate the derivatives?

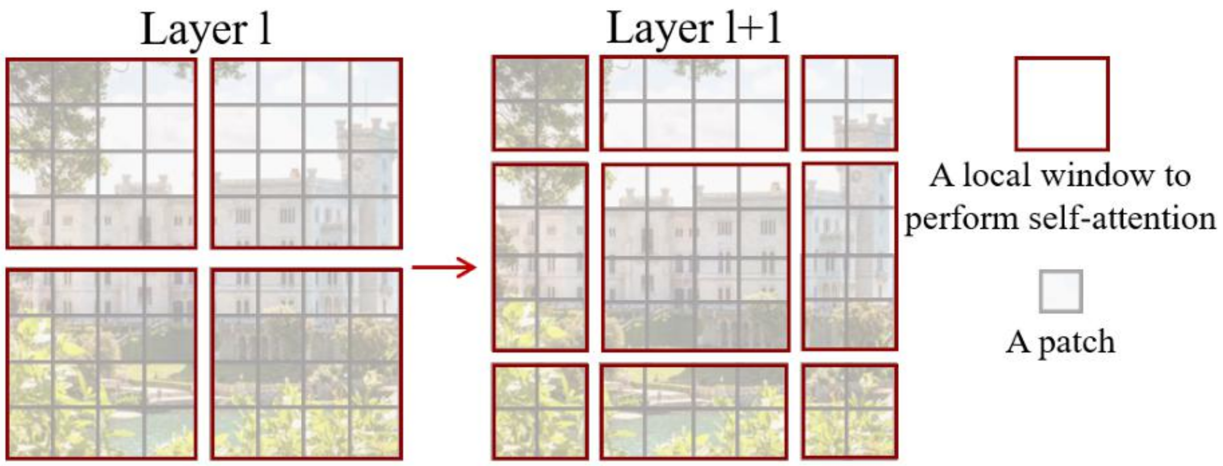
? What are the difficulties training PINNs and how to solve them?

- Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations
- PINNacle: A Comprehensive Benchmark of Physics-Informed Neural Networks for Solving PDEs
- ConFIG: Towards conflict-free training of physics informed neural networks

Fast forward topics

Transformer for Physics: efficient **spatial transformers**

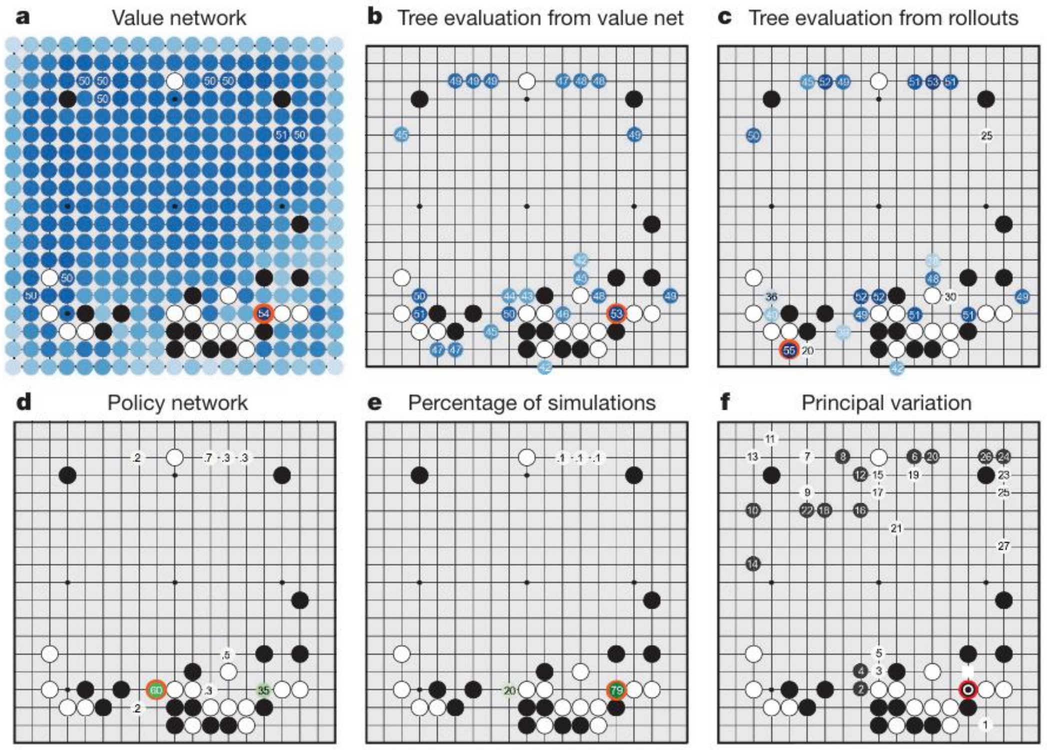
- Swin Transformer: Hierarchical Vision Transformer using Shifted Windows
- Transolver: A Fast Transformer Solver for PDEs on General Geometries
- Unisolver: PDE-Conditional Transformers Towards Universal Neural PDE Solvers
- PDE-Transformer: Efficient and Versatile Transformers for Physics Simulations



Fast forward topics

Reinforcement learning

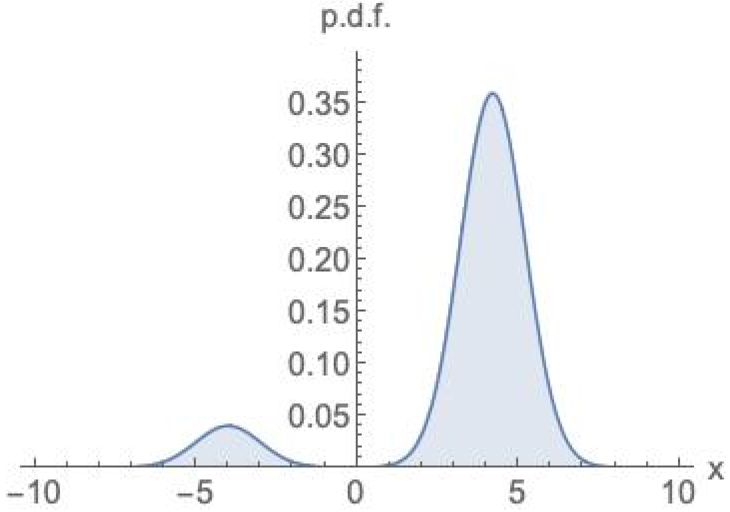
- Playing Atari with Deep Reinforcement Learning
- AlphaGo: Mastering the Game of Go with Deep Neural Networks and Tree Search

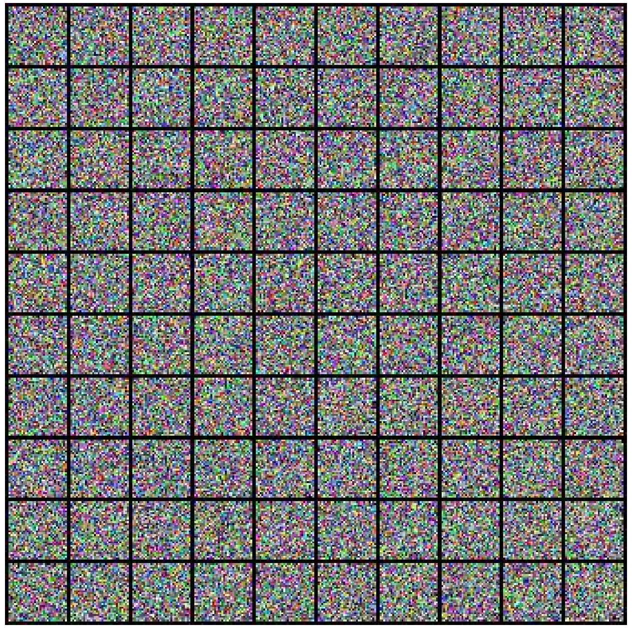


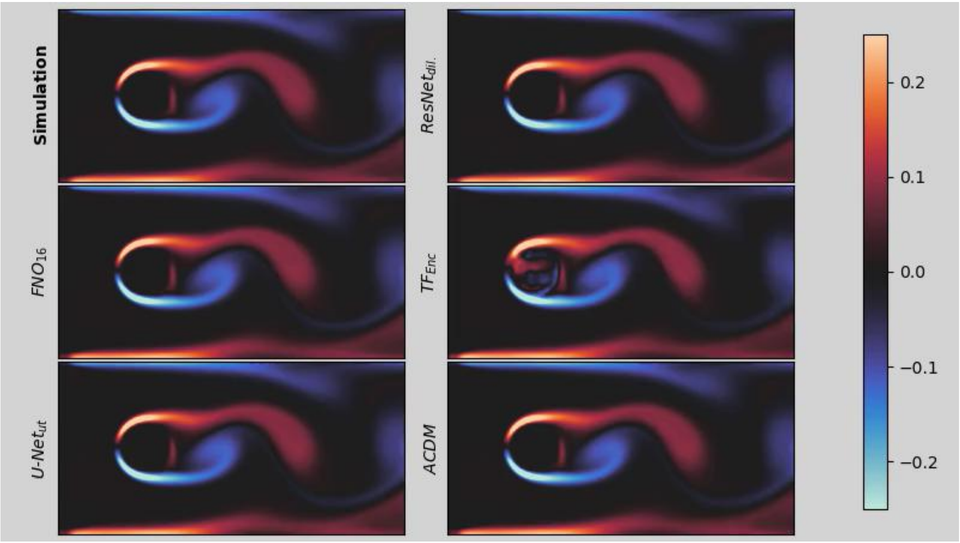
Fast forward topics

Generative models

- Generative Modeling by Estimating Gradients of the Data Distribution (blog paper)
- A Physics-informed Diffusion Model for High-fidelity Flow Field Reconstruction
- PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers
- Flow Matching Meets PDEs: A Unified Framework for Physics-Constrained Generation







Seminar regulations

Course page:
https://www.cs.cit.tum.de/cg/teaching/winter-term-25-26/deep-learning-in-physics/

Report

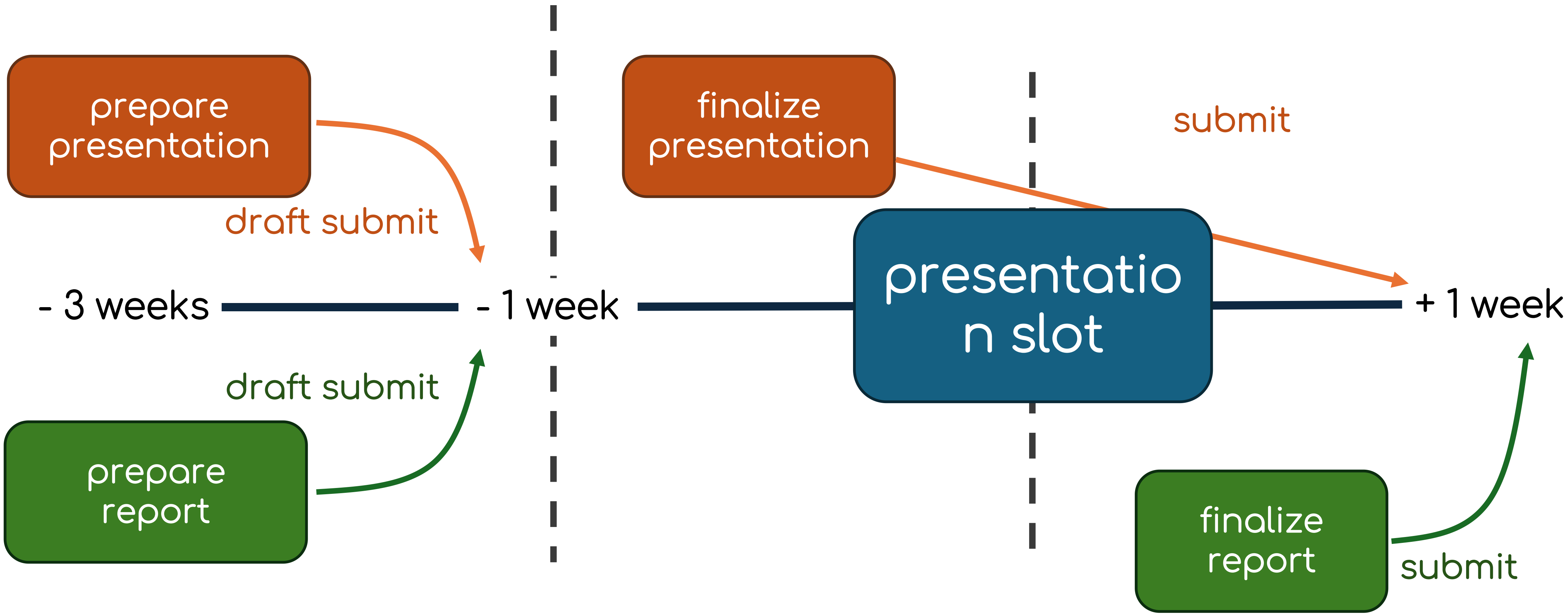
- Maximum **4** pages
- ACM SIGGRAPH TOG format (acmtog) available online
- Guideline
  - Start with a **summary of the paper (required for semi-final version!)**
  - **Own thoughts and reasoning** should be the main focus
  - Example: comparison to literature, pros & cons, future work…
- Feedback provided by advisor, final version due **after** talk

Presentation

- Slides:
  - Any style you like, **submit as PDF**.
  - **Follow guidelines** (text-balance, visualizations, highlighting etc.)
  - Feedback on semi-final slides provided by your advisor

- Presenting:
  - Present in **English**
  - Target **25 min** for presentation, **10 min** for questions
  - **Test your setup beforehand** (laptop/projector)!
  - Tips for a good presentation: DocTUM: How to give a great scientific talk

Your Timeline



Additional Resources

All information is available on the website!

Background Reading:

- Book: Hastie et al., The Elements of Statistical Learning
- Book/Online: Goodfellow et al., Deep Learning
- Online: Nielsen, Neural Networks and Deep Learning
- Online: Thuerey et al., Physics-based Deep Learning

Additional Information

- TUMonline registration is handled by us, you do **not** need to sign up

- Advisor:
  - Assigned to you in advance (see website)
  - Contact your advisor **1 week before** your presentation at the latest

- Attendance:
  - Missing **one session** is allowed, let us know in advance and write a short summary of the papers (ca. 1 page)
  - Missing **another** session means failing the seminar (special rules for severe issues as appropriate)

Grading Criteria

Presentation

- Good explanations
- Knowledgeable
- Clarity
- Stage performance

Slides

- Design, text density
- Citations
- Highlighting
- Visualizations

Report

- Base summary
- Literature review
- Own judgement

Other

- Own experiments
- Participation in discussions

Any questions?