Solution of Quantum Graphs by Physics-Informed DeepONets

TECHNISCHE UNIVERSITÄT IN DER KULTURHAUPTSTADT EUROPAS CHEMNITZ

Jan Blechschmidt, Jan-F. Pietschmann, Tom-C. Riemer, Martin Stoll, Max Winkler

Introduction

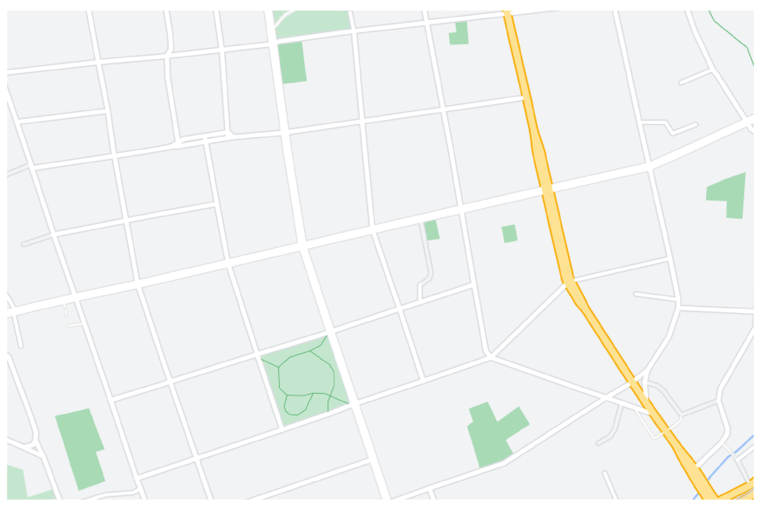
Abstract

We focus on a machine learning approach for **quantum graphs**, i.e. metric graphs with an associated differential operator.

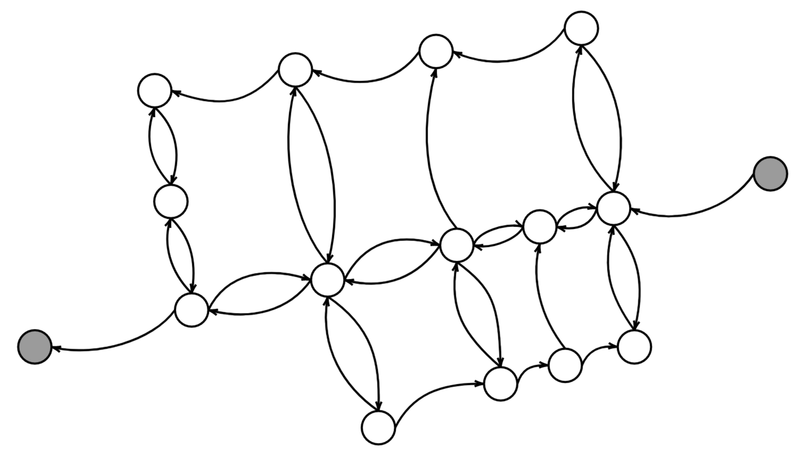
In our case the differential equation is a **nonlinear drift-diffusion equation**. Computational methods for quantum graphs require a careful discretization of the differential operator that also incorporates the node conditions, in our case **Kirchhoff-Neumann** conditions. Traditional numerical schemes are rather mature but have to be tailored manually when the differential equation becomes the constraint in an optimization problem.

We train **physics-informed DeepONet** models on a simple reference graph and show how to combine them for the solution of quantum graphs.

Example: Road network map with modeling metric graph



(a) A street network in Chemnitz, Saxony, Germany. Image from Google maps. Central point coordinates: $50.83, 12.90$.



(b) A metric graph modeling a compact road network within the left road network. Empty circles are interior vertices while filled ones depict exterior ones.

PDE Model

Quantum Graph Model: PDE, initial conditions ...

Given a graph $\Gamma = (\mathcal{V}, \mathcal{E})$ consisting of vertices $v \in \mathcal{V}$ and edges $e \in \mathcal{E}$ with associated lengths $\ell_e > 0$, we consider the non-linear drift-diffusion equation

$$\partial_t \rho_e = \partial_x(\epsilon\, \partial_x \rho_e - f(\rho_e)), \quad \text{for all } x \in (0, \ell_e), e \in \mathcal{E},$$

with $f(\rho) = \rho\,(1-\rho)$ and *initial conditions*

$$\rho_e\,(0, x) = u_e^{\text{init}}\,(x)\,, \quad \text{for all } x \in (0, \ell_e), e \in \mathcal{E}.$$

... and coupling conditions

Furthermore, there hold *homogeneous Kirchhoff-Neumann cond's and continuity*

$$\sum_{e\in\mathcal{E}_v} J_e(v)\, n_e(v) = 0 \text{ and } \rho_e(v) = \rho_{e'}(v), \quad \text{for all } e,\, e' \in \mathcal{E}_v, v \in \mathcal{V}_\mathcal{K} \subset \mathcal{V},$$

with $J_e = -\epsilon\, \partial_x \rho_e + f(\rho_e)$ and *flux boundary conditions (inflow and outflow)*

$$\sum_{e\in\mathcal{E}_v} J_e(v)\, n_e(v) = -u_v^{\text{inflow}}(t)\,(1-\rho_v) + u_v^{\text{outflow}}(t)\,\rho_v, \quad \text{for all } v \in \mathcal{V}_\mathcal{D} := \mathcal{V} \setminus \mathcal{V}_\mathcal{K}.$$

1st Step

1st step – Learning an edge surrogate model on a *reference graph* by physics-informed DeepONets

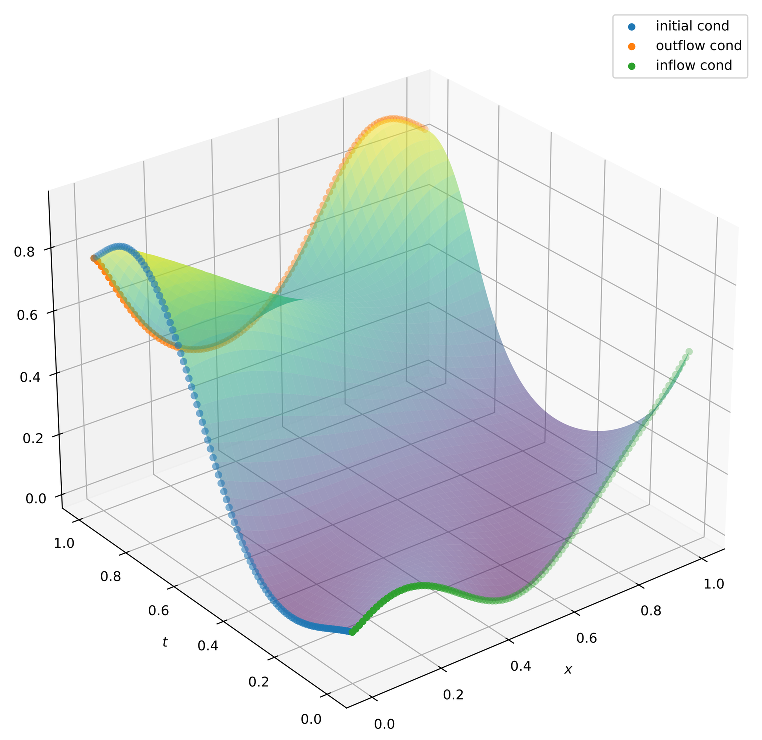


Figure: Illustration of random GP training data.

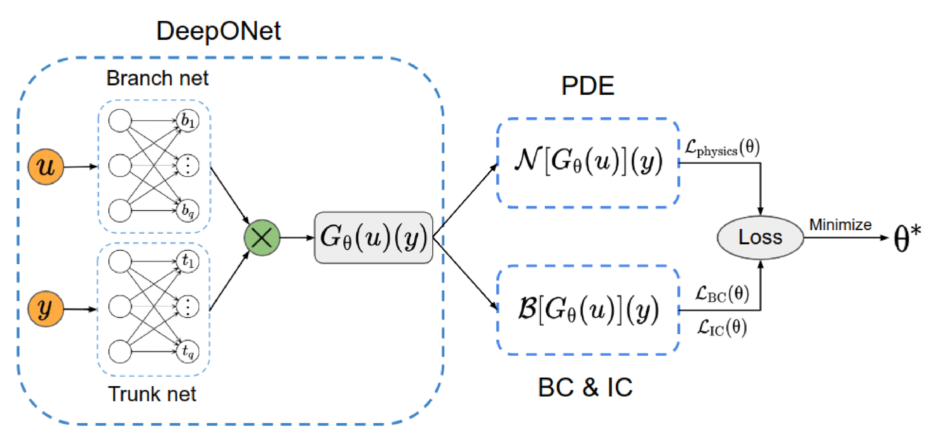


Figure: Illustration of PIDeepONet taken from [1].

Learning of an operator approximation $G_\theta(u)(y)$ by minimization of

$$\mathcal{L}_{\text{physics}}(\theta; u, y) + \mathcal{L}_{\text{init}}(\theta; u, y) + \mathcal{L}_{\text{flow}}(\theta; u, y)$$

using 5K training samples, 10K steps of ADAM, **no** Kirchhoff-Neumann and continuity cond's here.

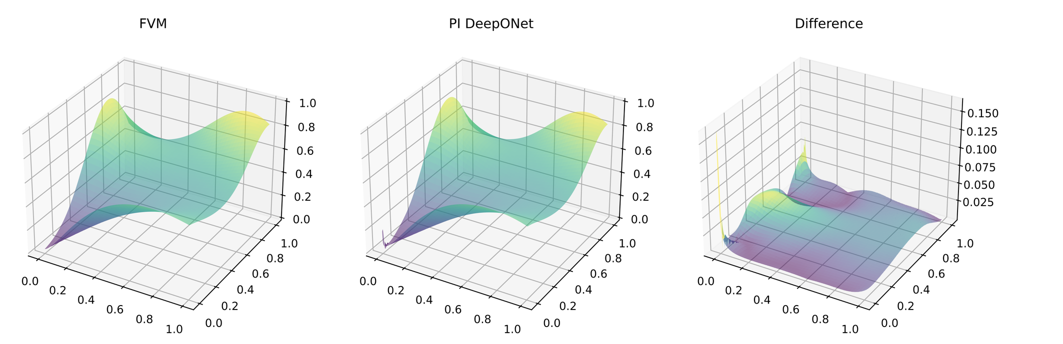


Figure: Example solution on *reference edge* by FVM (left), PI DeepONet (middle), abs. difference (right)

Result 1st step: Approximation of the nonlinear drift-diffusion operator which maps

$$(u, y) = \big((u_{\text{inflow}}, u_{\text{outflow}}, u_{\text{init}}), (t, x)\big) \mapsto \hat{\rho}_e^u(t, x)$$

for $(t, x) \in [0, 1] \times [0, 1]$, i.e., its solution on *reference edge*.

2nd Step

2nd Step – Applying the model to more complex graphs and ensuring coupling conditions (KN, continuity, in-/outflow)

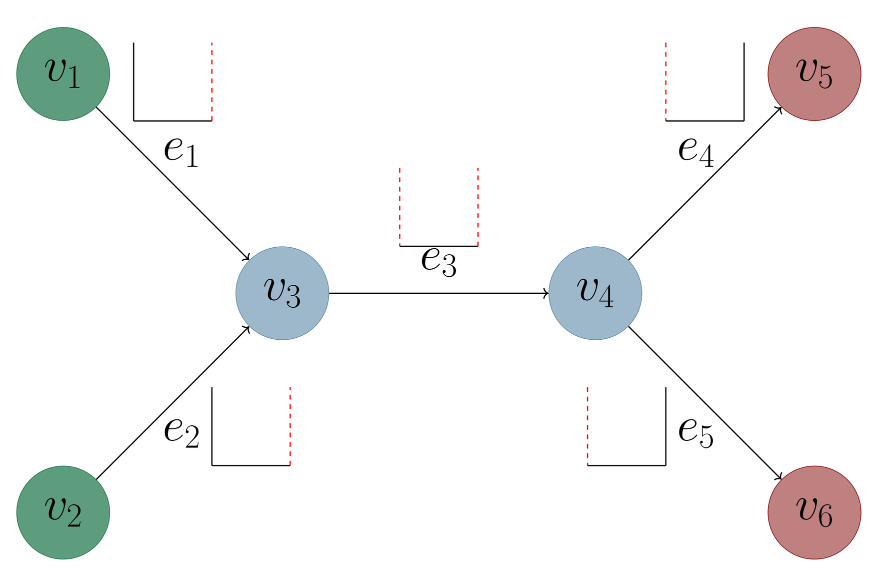


Figure: Model metric graph with known (solid black) and unknown data (dashed red).

Learning unknown flow parameters $\mathbf{z} \in \mathbb{R}^{m\times n_t}$ with $m = \sum_{v\in\mathcal{V}_\mathcal{K}} |\mathcal{E}_v|$ by minimization of

$$\underbrace{\sum_{v\in\mathcal{V}_\mathcal{K}} \sum_{e,e'\in\mathcal{E}_v} \left(\hat{\rho}_e^{u(\mathbf{z})}(v) - \hat{\rho}_{e'}^{u(\mathbf{z})}(v)\right)^2}_{\text{continuity loss}}$$

$$+ \underbrace{\sum_{v\in\mathcal{V}_\mathcal{K}} \Big(\sum_{e\in\mathcal{E}_v} (\hat{J}_e^{u(\mathbf{z})}(v)\, n_e(v)\Big)^2}_{\text{Kirchhoff-Neumann loss}} + \lambda \underbrace{\sum_{v\in\mathcal{V}_\mathcal{K}} \sum_{e\in\mathcal{E}_v} \|z_{v,e}\|_{H^1}^2}_{\text{regularization}}$$

- approach allows for solution of inverse problems by incorpation of measured data
- faster and more flexible than PINN approach considered in [3].

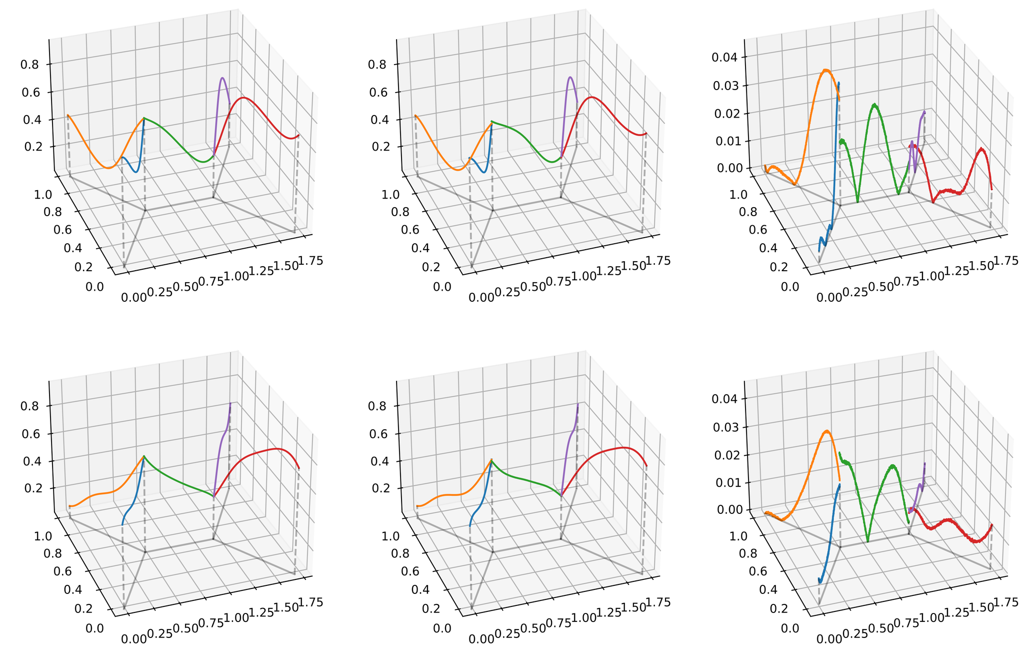


Figure: Solution on model graph at $t = 0.25$ and $t = 0.75$ by FVM (left), PI DeepONet (middle), abs. difference (right)

Ref.

[1] S. Wang, H. Wang, and P. Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science advances*, 7(40):eabi8605, 2021.

[2] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

[3] J. Blechschmidt, J.-F. Pietschmann, T.-C. Riemer, M Stoll, and M. Winkler. A comparison of PINN approaches for drift-diffusion equations on metric graphs. *arXiv:1808.04327*, 2022.