Message Passing Neural PDE Solvers

Johannes Brandstetter* [12] Daniel Worrall* [3] Max Welling[1]

[1]University of Amsterdam [2]Johannes Kepler University Linz [3]Qualcomm AI Research

Generalizing neural PDE solver

Partial Differential Equations (PDEs) of the form

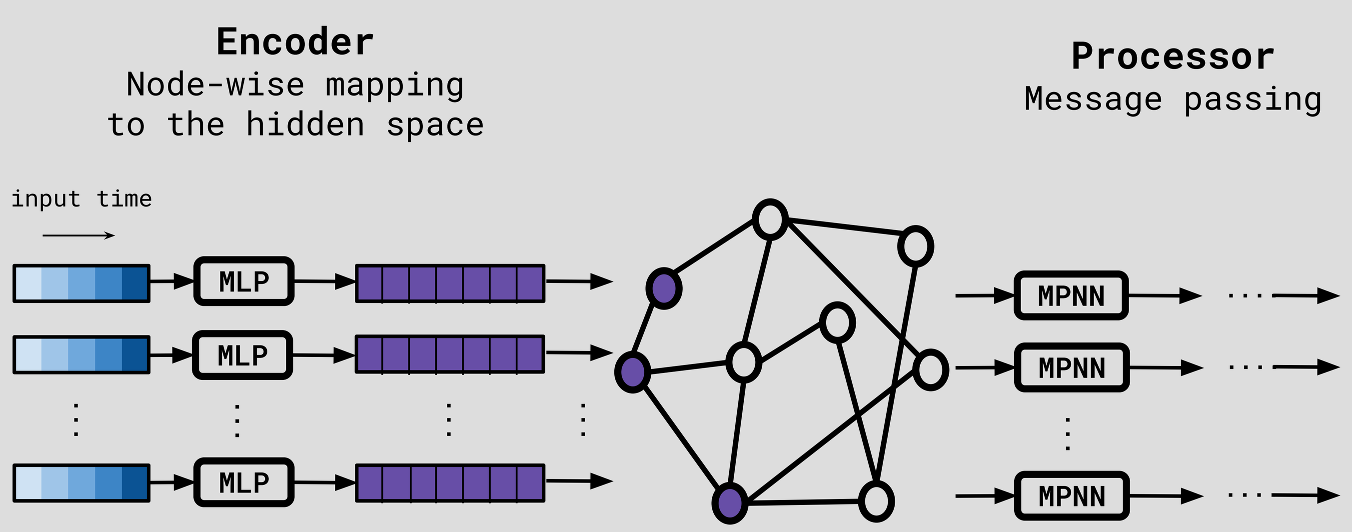
$$\partial_t \mathbf{u} = F(t, \mathbf{x}, \mathbf{u}, \partial_{\mathbf{x}}\mathbf{u}, \partial_{\mathbf{xx}}\mathbf{u}, \ldots)$$

are abundant, yet numerical PDE solving is a **splitter field**:

- Goal is to design a **fully numerical PDE solver** which offers flexibility to satisfy as many requirements as possible.
- Common way to solve PDEs is to approximate **spatial derivatives** and solve for **temporal derivatives**.
- Message passing neural network to update $\mathbf{u}(\mathbf{x}, t) \rightarrow \mathbf{u}'(\mathbf{x}, t')$

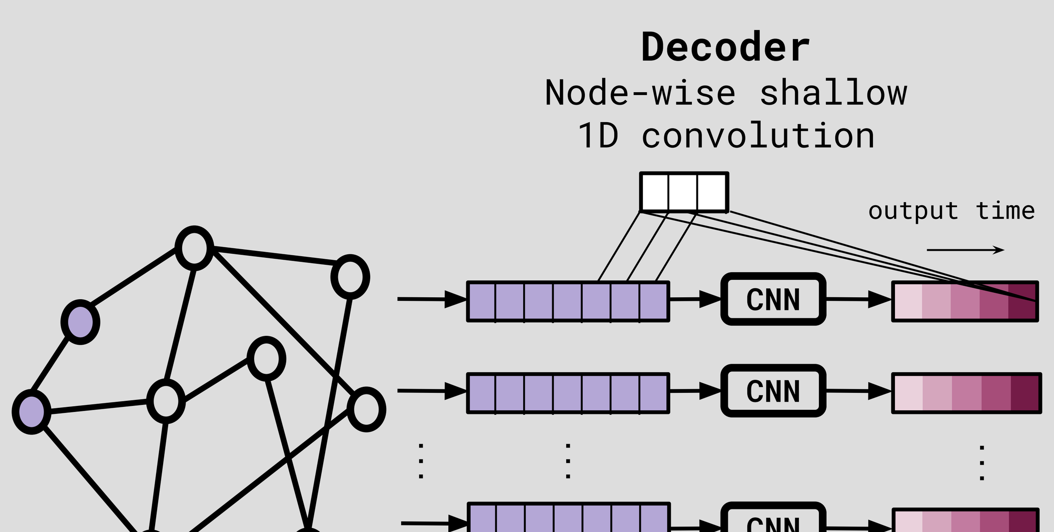
Representational containment of spatial solvers

- Message$_{\text{edge } j \rightarrow i}$: $\mathbf{m}_{ij}^m = \phi\left(\mathbf{f}_i^m, \mathbf{f}_j^m, \mathbf{u}_i^t - \mathbf{u}_j^t, \mathbf{x}_i - \mathbf{x}_j, \boldsymbol{\theta}_{\text{PDE}}\right)$.
- Generalizes **estimation of spatial derivatives**.
- Finite difference, finite volume and WENO scheme are representationally contained (if one, two, or three message passing layers are used).



Representational containment of temporal solvers

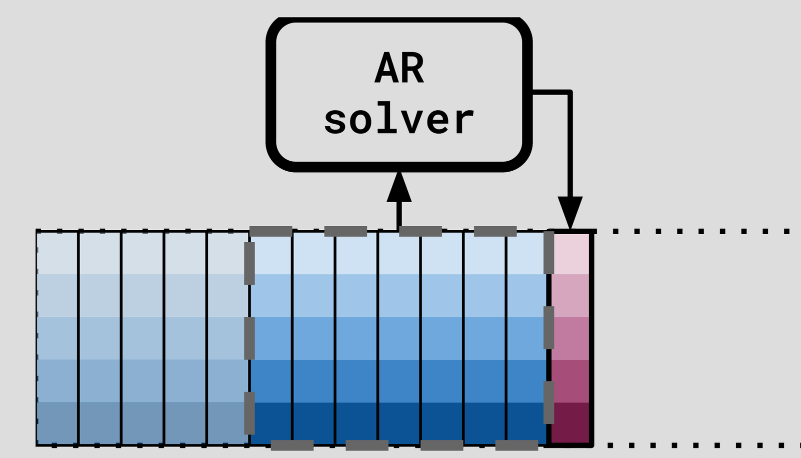
Decoder as 1D convolutional network with shared weights across spatial locations (reminiscent of **linear multistep methods**).



Challenges for autoregressive solvers

Solving PDEs iteratively gives strong physical interpretability, however:

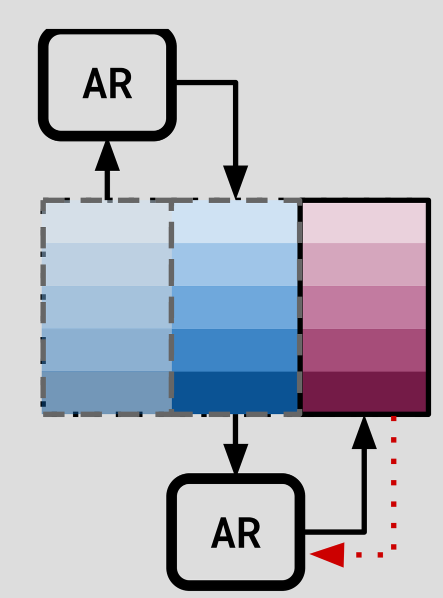
- Hard to train since errors at test time accumulate.
- How to enforce stability? How to simulate error input distribution in training?



Autoregressive model
Mapping between temporally consecutive time steps

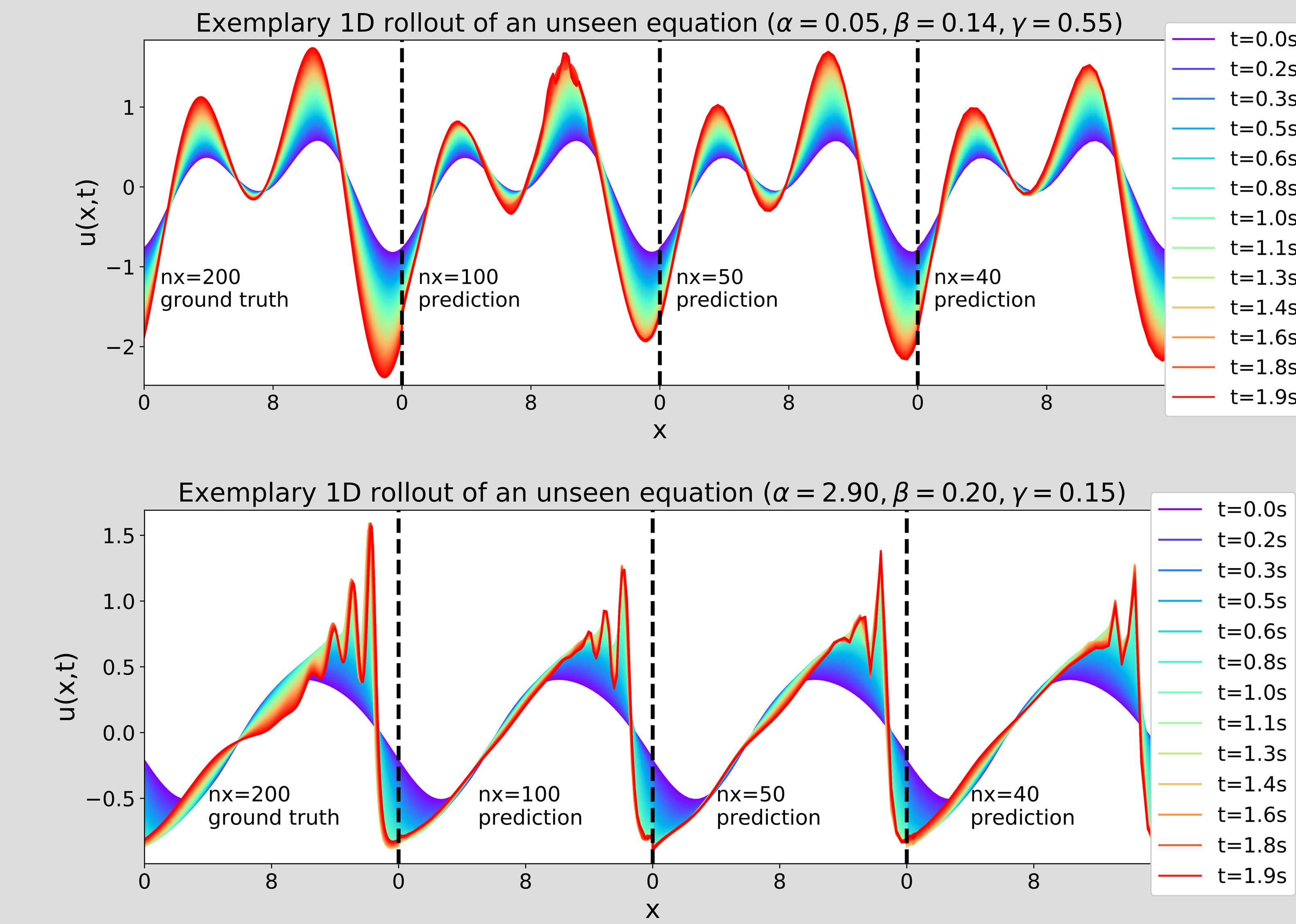
Temporal bundling, pushforward trick

- **Pushforward trick**: mimics distribution shift via adversarial perturbation.
- **Temporal bundling**: synchronous prediction of multiple future timesteps.

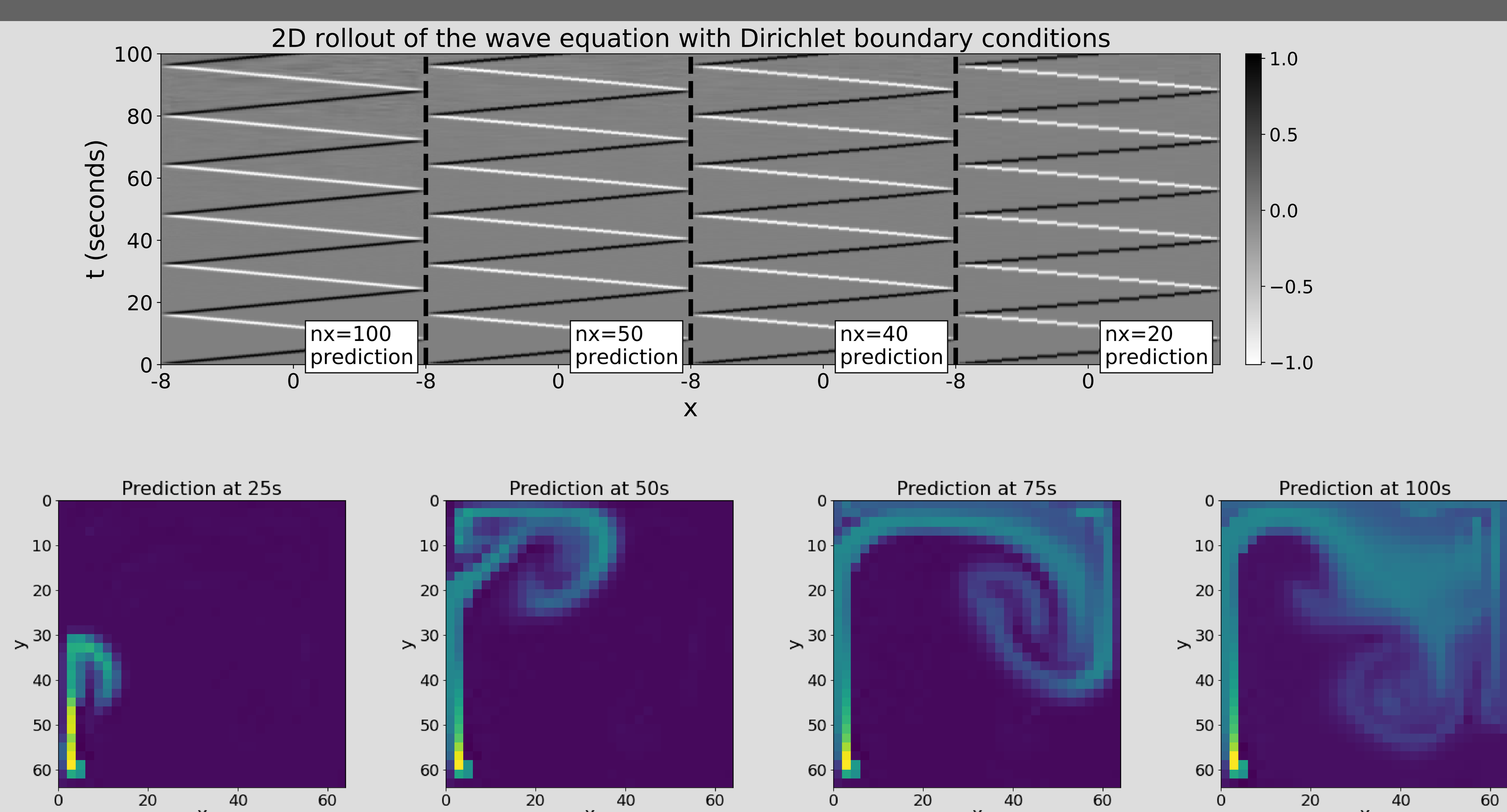


Pushforward training
Gradients flow only through last time step

Generalization across different equations, different resolutions



Generalization across boundary conditions, irregular grids, applicability to higher dimensional problems



References

- Paper: https://arxiv.org/abs/2202.03376
- Code: https://github.com/brandstetter-johannes/MP-Neural-PDE-Solvers