AI for scientific design: Part I

Tailin Wu & Minkai Xu Stanford University 4/9/2023 @ 集智斑图





Outline

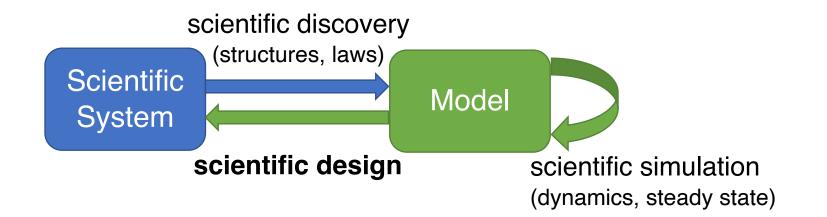
- Overview of AI for scientific design (Tailin Wu)
- Surrogate model + backpropagation for scientific design (Tailin Wu)
 - Inverse design of boundary and system parameters
 - State/parameter estimation: inferring unknown system states/parameters [2]
- Diffusion model for scientific design (Minkai Xu)
 - Overview for equivariant GNNs
 - Overview for diffusion models
 - Diffusion models for molecular design

Outline

• Overview of AI for scientific design (Tailin Wu)

- Surrogate model + backpropagation for scientific design (Tailin Wu)
 - Inverse design of boundary and system parameters
 - State/parameter estimation: inferring unknown system states/parameters [2]
- Diffusion model for scientific design (Minkai Xu)
 - Overview for equivariant GNNs
 - Overview for diffusion models
 - Diffusion models for molecular design

What is scientific design?



Here the **scientific design** includes:

- design the system (parameter/boundary/components) to optimize certain objective Today's talk
- control the system to optimize certain objective
- design experiments for scientific discovery

Next week's talk

Why scientific design?

These three tasks are critical in all **sciences**:

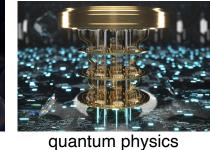
- design the system
- control the system
- design experiments

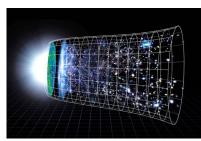
These two tasks are the gist of all **engineering**:

- **design** the system
- control the system

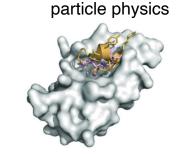
Solving the challenges can also boost AI







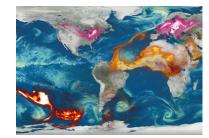
astronomy



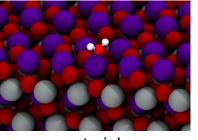
molecular dynamics



robotics



weather forecasting

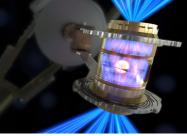




life sciences



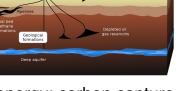




nuclear fusion



policy design: 5 carbon credits



energy: carbon capture

aerospace



What is the most **important insight** from 30 years of deep learning?



Method 1: go from A to B in a straight line (learn a direct mapping from A to B) Hard to learn due to the complexity of A, B and their difference!

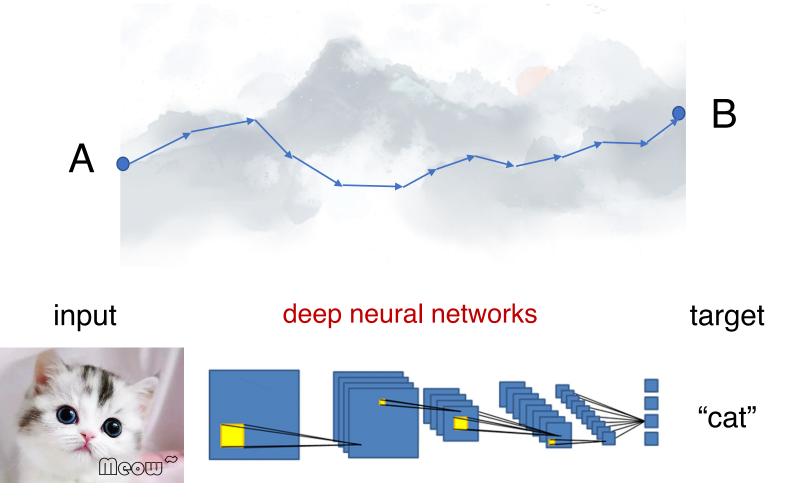
What is the most **important insight** from 30 years of deep learning?



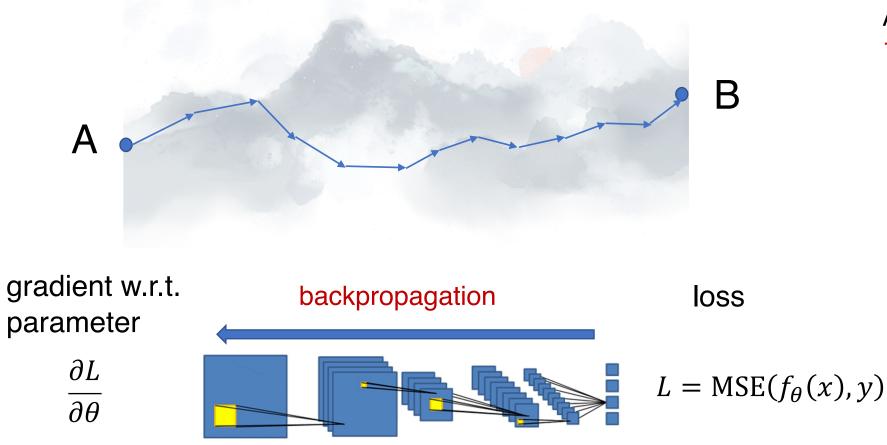
Method 2: go from A to B in small, easier steps (compose step-by-step simple mappings to map A to B)

Much easier!

Insight: to construct a complex mapping from A to B, it is much easier to compose simple mappings

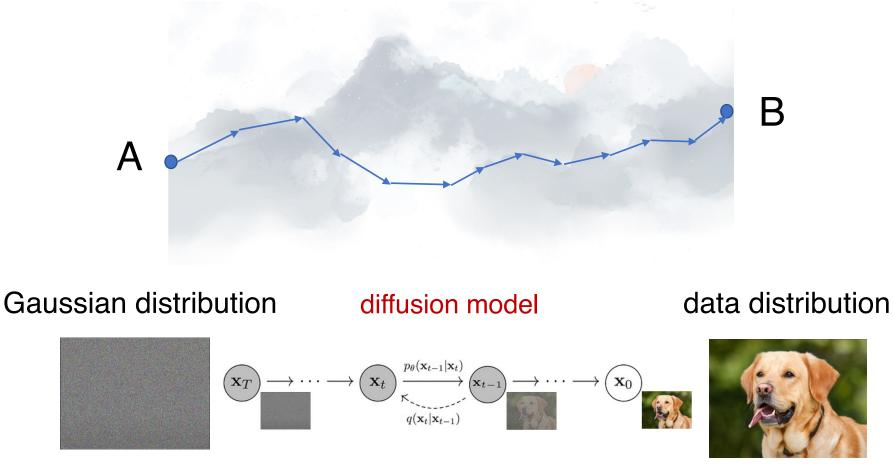


Insight: to construct a complex mapping from A to B, it is much easier to compose simple mappings



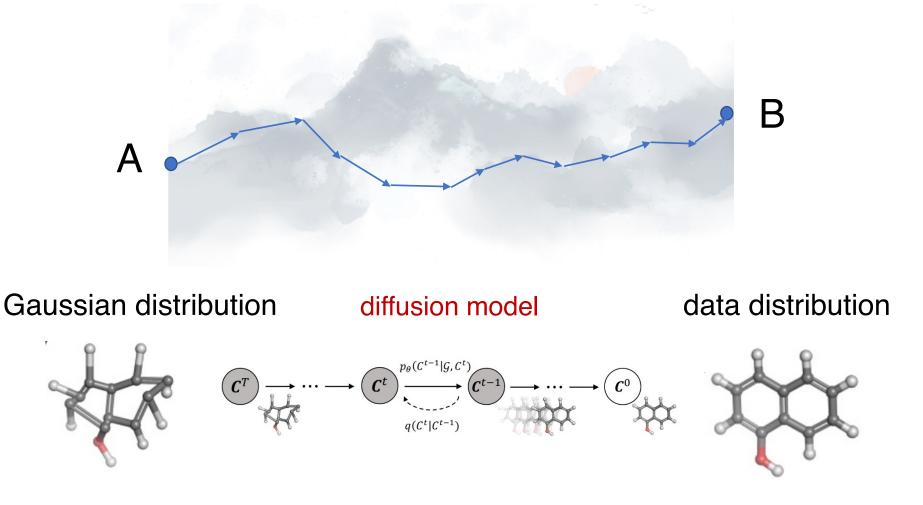
AI for scientific design:1. backpropagation

Insight: to construct a complex mapping from A to B, it is much easier to compose simple mappings



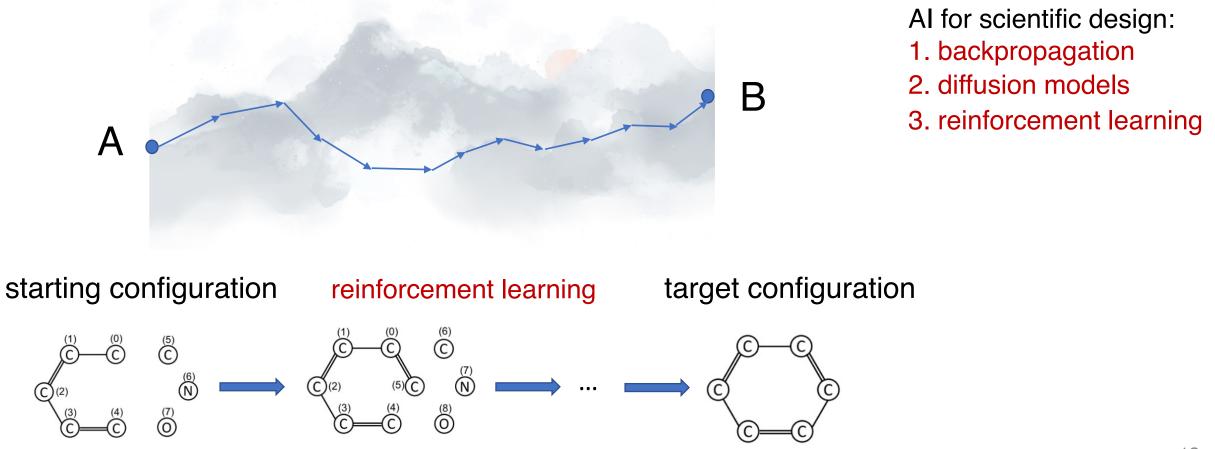
AI for scientific design:1. backpropagation2. diffusion models

Insight: to construct a complex mapping from A to B, it is much easier to compose simple mappings



AI for scientific design:1. backpropagation2. diffusion models

Insight: to construct a complex mapping from A to B, it is much easier to compose simple mappings



- 1. Surrogate model + backpropagation [1][2]
- 2. Diffusion models [3]
- 3. **Reinforcement learning** (e.g., GCPN [4], GFlowNet [5], RL for controlling plasma

fusion [6], RL for designing quantum experiments [7])

- 4. Direct mapping [8]
- 5. Bayesian methods / sample-based methods [9]
- 6. ...

[1] Allen, Kelsey R., et al. "Physical design using differentiable learned simulators." *NeurIPS* 2022

[2] Zhao, Qingqing, David B. Lindell, and Gordon Wetzstein. "Learning to solve pdeconstrained inverse problems with graph networks." ICML 2022

[3] Xu, Minkai, et al. "Geodiff: A geometric diffusion model for molecular conformation generation." *ICLR* 2022

[4] You, Jiaxuan, et al. "Graph convolutional policy network for goal-directed molecular graph generation." Advances in neural information processing systems 31 (2018).
[5] Jain, Moksh, et al. "Biological sequence design with gflownets." ICML 2022

[6] Degrave, Jonas, et al. "Magnetic control of tokamak plasmas through deep reinforcement learning." *Nature* 602.7897 (2022): 414-419.

[7] Nautrup, Hendrik Poulsen, et al. "Optimizing quantum error correction codes with reinforcement learning." Quantum 3 (2019): 215.

[8] Guo, Ruchi, Shuhao Cao, and Long Chen. "Transformer meets boundary value inverse problems." NeurIPS 2022

[9] Rubinstein, R. Y. and Kroese, D. P. "The cross-entropy method: A unified approach to monte carlo simulation, randomized optimization and machine learning". Information Science & Statistics, Springer Verlag, NY, 2004.

- 1. Surrogate model + backpropagation [1][2]
- 2. Diffusion models [3]

Today's talk

- 3. **Reinforcement learning** (e.g., GCPN [4], GFlowNet [5], RL for controlling plasma fusion [6], RL for designing quantum experiments [7])
- 4. Direct mapping [8]
- 5. Bayesian methods / sample-based methods [9]
- 6. ...

[1] Allen, Kelsey R., et al. "Physical design using differentiable learned simulators." *NeurIPS* 2022

[2] Zhao, Qingqing, David B. Lindell, and Gordon Wetzstein. "Learning to solve pdeconstrained inverse problems with graph networks." *ICML* 2022

[3] Xu, Minkai, et al. "Geodiff: A geometric diffusion model for molecular conformation generation." *ICLR* 2022

[4] You, Jiaxuan, et al. "Graph convolutional policy network for goal-directed molecular graph generation." Advances in neural information processing systems 31 (2018).
[5] Jain, Moksh, et al. "Biological sequence design with gflownets." ICML 2022

[6] Degrave, Jonas, et al. "Magnetic control of tokamak plasmas through deep reinforcement learning." *Nature* 602.7897 (2022): 414-419.

[7] Nautrup, Hendrik Poulsen, et al. "Optimizing quantum error correction codes with reinforcement learning." Quantum 3 (2019): 215.

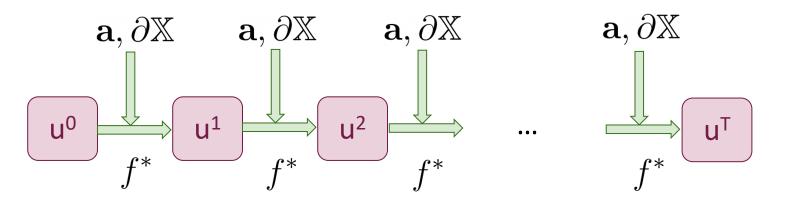
[8] Guo, Ruchi, Shuhao Cao, and Long Chen. "Transformer meets boundary value inverse problems." NeurIPS 2022

[9] Rubinstein, R. Y. and Kroese, D. P. "The cross-entropy method: A unified approach to monte carlo simulation, randomized optimization and machine learning". Information Science & Statistics, Springer Verlag, NY, 2004.

Outline

- Overview of AI for scientific design (Tailin Wu)
- Surrogate model + backpropagation for scientific design (Tailin Wu)
 - Inverse design of boundary and system parameters
 - State/parameter estimation: inferring unknown system states/parameters [2]
- Diffusion model for scientific design (Minkai Xu)
 - Overview for equivariant GNNs
 - Overview for diffusion models
 - Diffusion models for molecular design

Dynamical system: forward simulation



PDE: partial differential equation ODE: ordinary differential equation

u^t: original **state** of the system. Can be an infinite-dimensional function $u_t(x)$ as solution to a PDE, or a graph (e.g., mesh, particles, molecules), or a vector (for ODE)

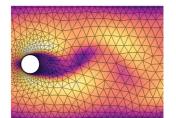
 f^* : Evolution. Can be a PDE evolved by classical solver, or evolution in the real world

a: static parameters of the system that does not change with time (e.g., parameters of PDE, spatially varying diffusion coefficient)

 $\partial \mathbb{X}$: boundary condition of the system



Fluid dynamics, computer graphics



Mesh-based simulation for PDEs



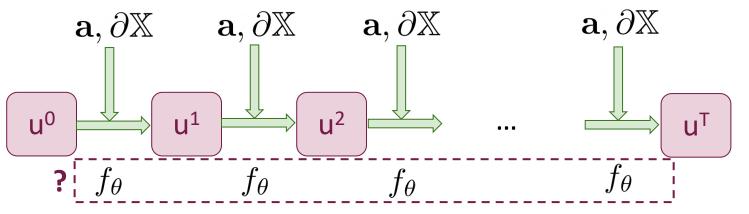
Dislocation in materials



Proteins and small molecules

Neural surrogate models: learning forward simulation

Goal: learn f_{θ} that maps u^t to u^{t+1} :

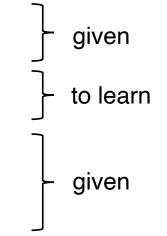


u^t: original **state** of the system. Can be an infinite-dimensional function $u^t(x)$ as solution to a PDE, or a graph, or a vector (for ODE)

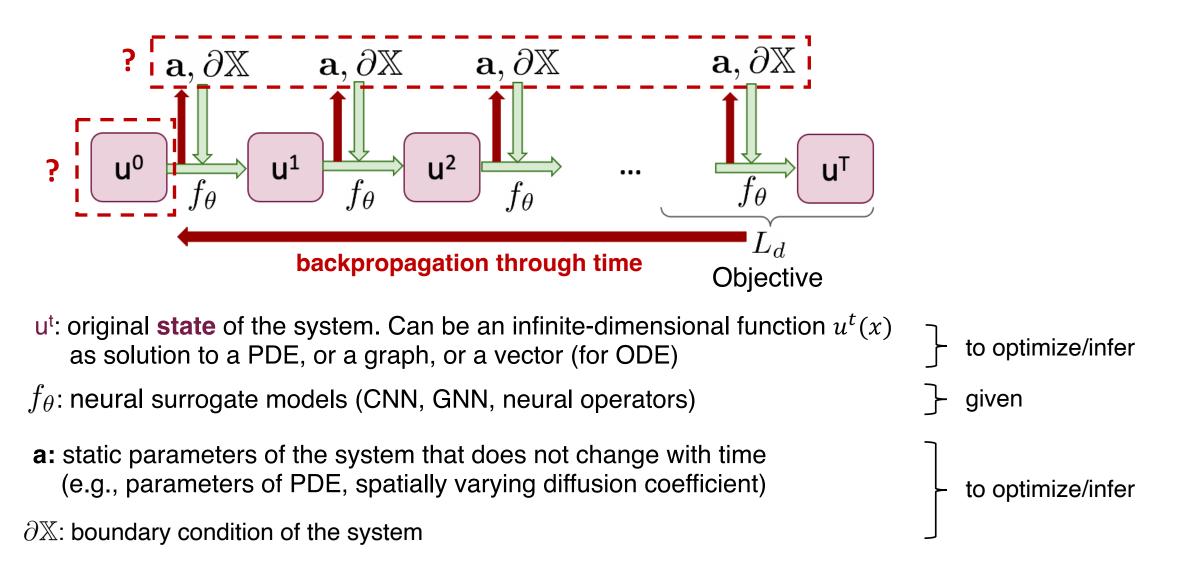
 f_{θ} : neural surrogate models (CNN, GNN, neural operators)

a: static parameters of the system that does not change with time (e.g. parameters of PDE, spatially varying diffusion coefficient)

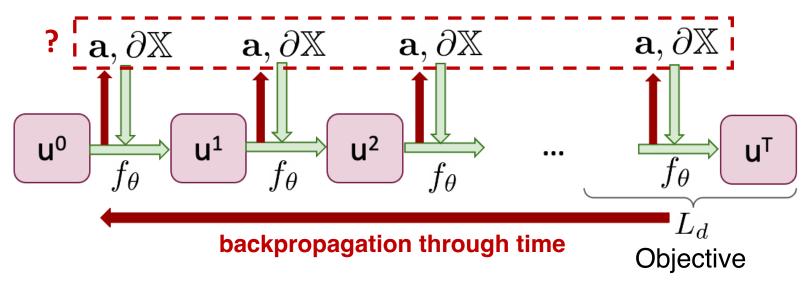
 $\partial \mathbb{X}$: boundary condition of the system



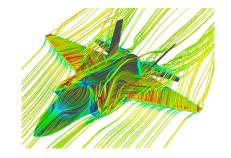
AI for scientific design: inverse optimization/design



AI for scientific design: inverse optimization/design



1. Optimize $L_d[u^T(\partial X, a)]$ w.r.t. boundary ∂X [1] or system parameter *a*: L_d : drag (for plane and under water robots), specific impulse (for rocket engine) ∂X : plane shape, rocket engine shape, robot shape

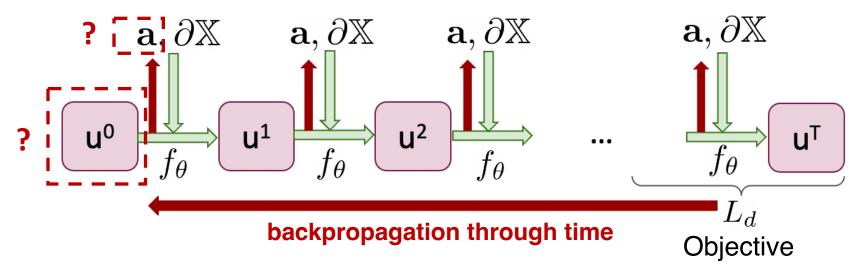




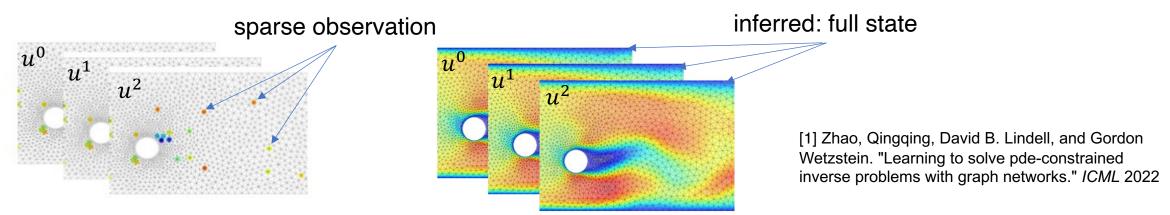


[1] Allen, Kelsey R., et al. "Physical design using differentiable learned simulators." *NeurIPS* 2022

AI for scientific design: inverse optimization/design



2. State/parameter estimation: optimize $MSE[\hat{u}^t(x), u^t(x)]$ w.r.t. initial state u^0 or parameter a [1], given sparse observation of $u^t(x)$ at certain locations $x \in \Omega$



Outline

- Overview of AI for scientific design (Tailin Wu)
- Surrogate model + backpropagation for scientific design (Tailin Wu)
 - Inverse design of boundary and system parameters [1]
 - State/parameter estimation: inferring unknown system states/parameters [2]
- Diffusion model for scientific design (Minkai Xu)
 - Overview for equivariant GNNs
 - Overview for diffusion models
 - Diffusion models for molecular design

[1] Allen, Kelsey R., et al. "Physical design using differentiable learned simulators." *NeurIPS* 2022

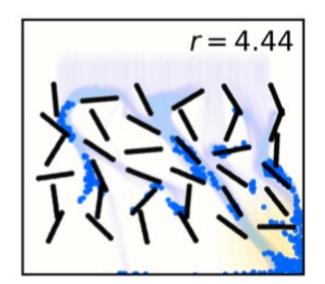
[2] Zhao, Qingqing, David B. Lindell, and Gordon Wetzstein. "Learning to solve pde-constrained inverse problems with graph networks." *ICML* 2022

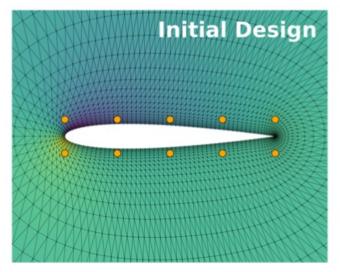
Task: design boundary for physical systems to optimize certain objective

Main contribution:

• Backpropagation through the entire simulation rollout

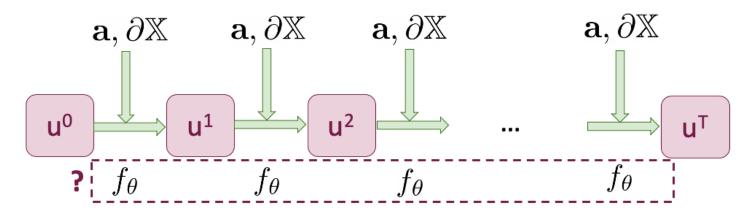




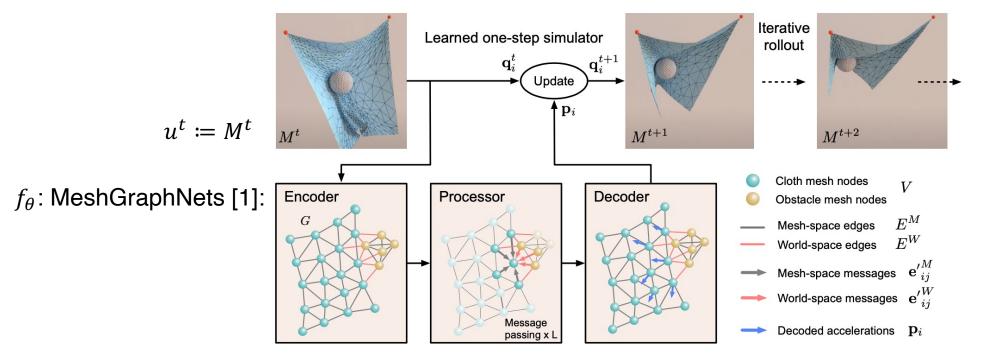


minimize drag for airfoil

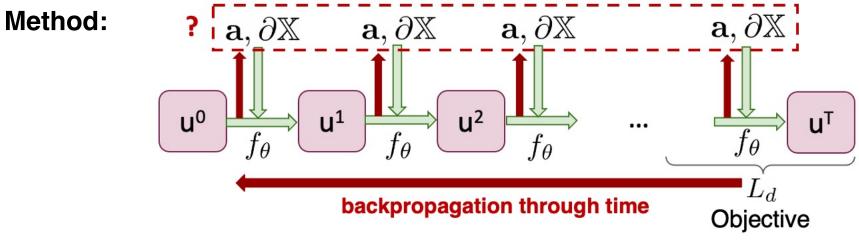
[1] Allen, Kelsey R., et al. "Physical design using differentiable 22 learned simulators." NeurIPS 2022



Training: given tuples of $\{(u^t, a, \partial X, u^{t+1})\}$, learn $f_{\theta}: (u^t, a, \partial X) \rightarrow u^{t+1}$



[1] Pfaff, Tobias, et al. "Learning mesh-based simulation with graph networks." ICLR 2021

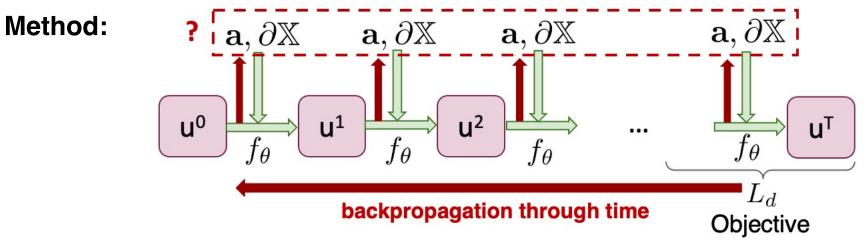


Inference:

Objective for inverse design: maximize $L_d[\partial X]$

 ∂X : boundary

To compute $\frac{\partial L_d[\partial X]}{\partial(\partial X)}$, do backpropagation through time through the entire simulation

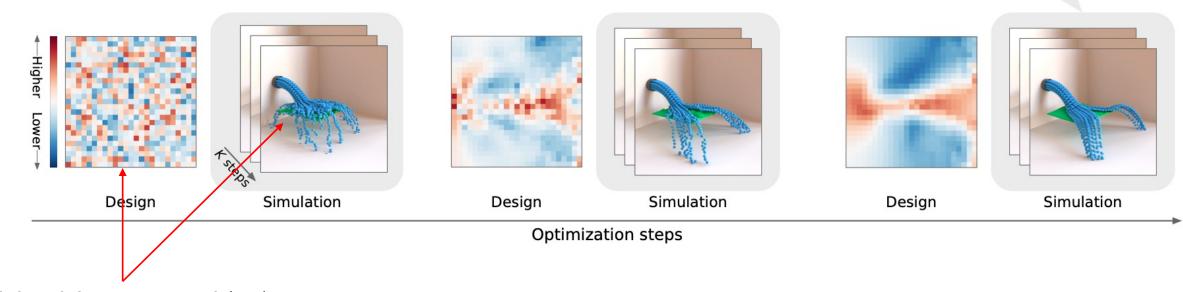


Inference:

Also employs two techniques:

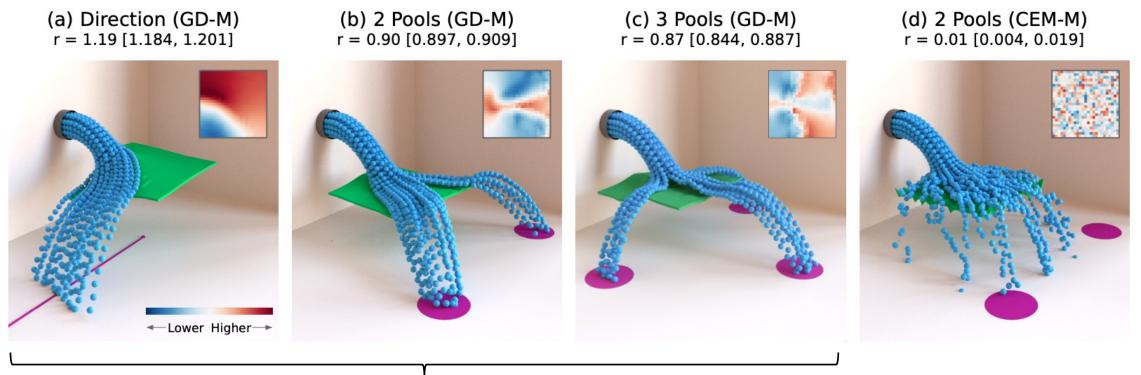
- 1. Ensembles: train multiple forward models each with different data split. At inference, average the gradient
- 2. Gradient checkpointing: recompute the intermediate activations for each step as needed when doing the backpropagation through time

Example optimization process:



height of the green pad (∂X)

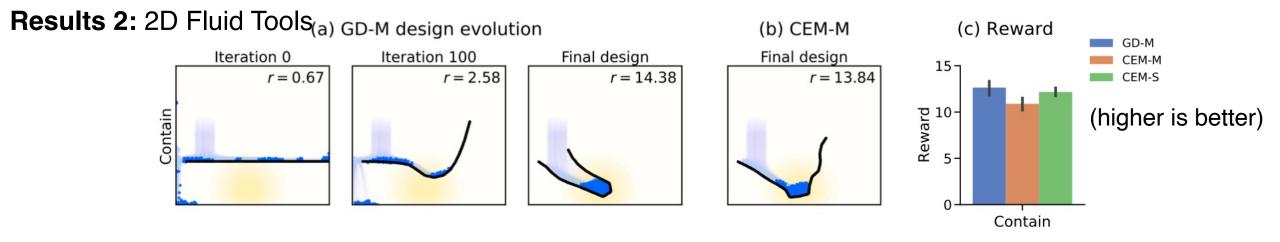
Results 1: 3D Watercourse

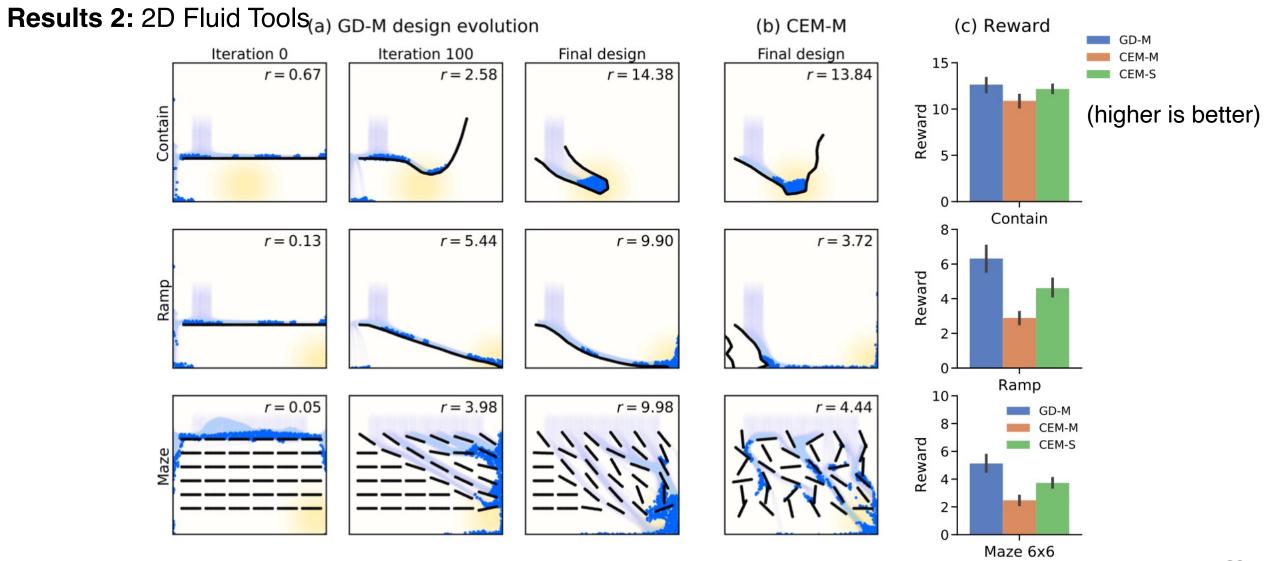


Proposed method (GD-M)

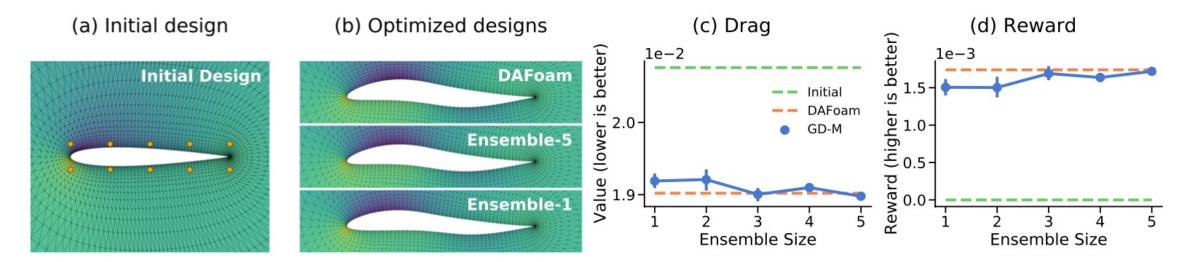
CEM: cross-entropy method [1], a sample-based method

[1] Rubinstein, R. Y. and Kroese, D. P. "The cross-entropy method: A unified approach to monte carlo simulation, randomized optimization and machine learning". Information Science & Statistics, Springer Verlag, NY, 2004.





Results 3: airfoil



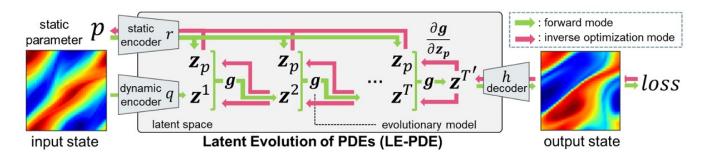
Significance: first work that combines GNN-based surrogate model with backpropagation through time for inverse design

Limitation: computationally expensive, since

- The surrogate model f_{θ} is learning a mapping in the **input space**, which can have high dimension
- Simulation length is long (up to 300 steps)

Improvement: perform the backpropagation through time in latent space

We proposed LE-PDE[1] that learns forward model and performs inverse design in **latent space**, significantly improve speed while maintaining competitive accuracy (compared with FNO, MP-PDE, etc.)



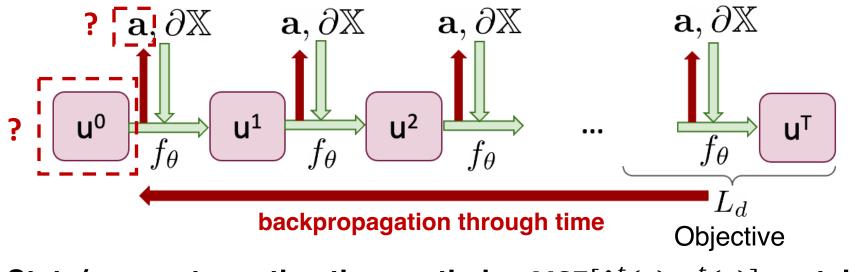
[1] Wu, Tailin, Takashi Maruyama, and Jure Leskovec. "Learning to accelerate partial differential equations via latent global evolution." *NeurIPS* 2022

Outline

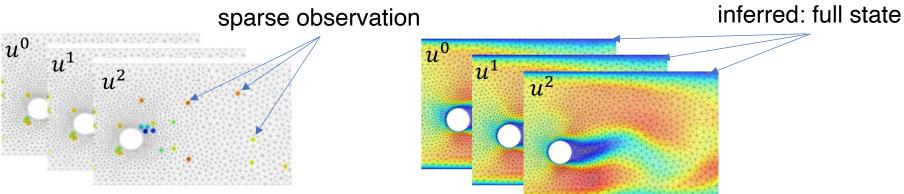
- Overview of AI for scientific design (Tailin Wu)
- Surrogate model + backpropagation for scientific design (Tailin Wu)
 - Inverse design of boundary and system parameters [1]
 - State/parameter estimation: inferring unknown system states/parameters [2]
- Diffusion model for scientific design (Minkai Xu)
 - Overview for equivariant GNNs
 - Overview for diffusion models
 - Diffusion models for molecular design

[1] Allen, Kelsey R., et al. "Physical design using differentiable learned simulators." *NeurIPS* 2022
[2] Zhao, Qingqing, David B. Lindell, and Gordon Wetzstein. "Learning to solve pde-constrained inverse problems with graph networks." *ICML* 2022

Al for scientific design: state/parameter estimation



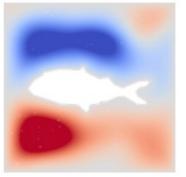
State/parameter estimation: optimize $MSE[\hat{u}^t(x), u^t(x)]$ w.r.t. initial state u^0 or **parameter a**, given sparse observation of $u^t(x)$ at certain locations $x \in \Omega$



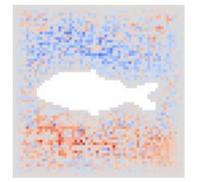
AI for scientific design: state/parameter estimation

Challenge: the u^0 and a have extremely high dimension, can optimize to **adversarial modes**

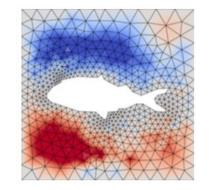
Ground Truth for u^0



Inferred u^0 with U-Net



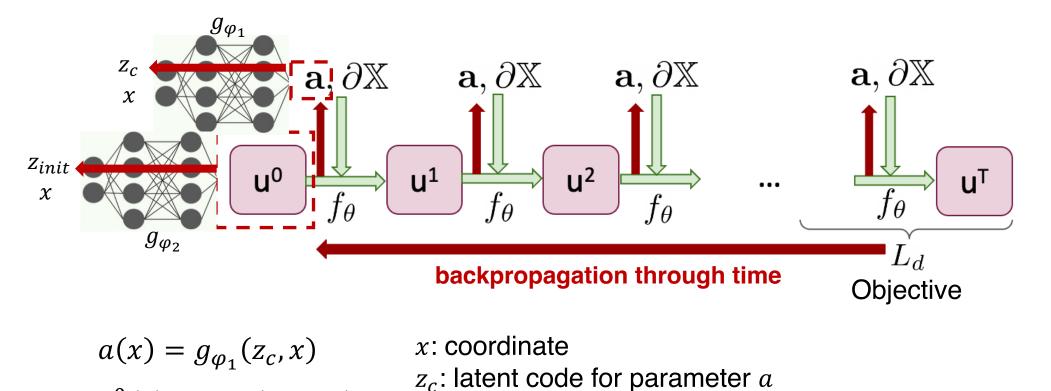
Inferred u^0 with GNN



Case study: state/parameter estimation with priors

Method: learn prior models that map latent code z_c , z_{init} and coordinate x to parameter a and initial state u^0

 z_{init} : latent code for initial state u^0



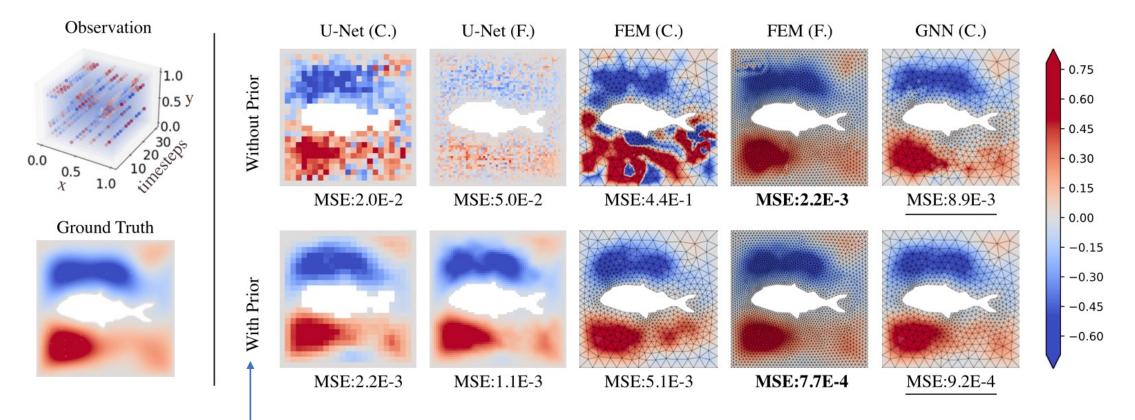
 $u^0(x) = g_{\varphi_2}(z_{init}, x)$

35

[1] Zhao, Qingqing, David B. Lindell, and Gordon

Case study: state/parameter estimation with priors

Result: much better state/parameter estimation



[1] Zhao, Qingqing, David B. Lindell, and Gordon Wetzstein. "Learning to solve pde-constrained inverse problems with graph networks." *ICML* 2022

Proposed method: with prior

Outline

- Overview of AI for scientific design (Tailin Wu)
- Surrogate model + backpropagation for scientific design (Tailin Wu)
 - Inverse design of boundary and system parameters
 - State/parameter estimation: inferring unknown system states/parameters [2]
- Diffusion model for scientific design (Minkai Xu)
 - Overview for equivariant GNNs
 - Overview for diffusion models
 - Diffusion models for molecular design

Outline

- Overview of AI for scientific design (Tailin Wu)
- Surrogate model + backpropagation for scientific design (Tailin Wu)
 - Inverse design of boundary and system parameters
 - State/parameter estimation: inferring unknown system states/parameters [2]

• **Diffusion model** for scientific design (Minkai Xu)

- Overview for equivariant GNNs
- Overview for diffusion models
- Diffusion models for molecular design