Hybrid AI 2.0, a case study: Numerical simulation for PDEs

Marc Schoenauer
Equipe A&O/TAU, INRIA, LISN, CNRS et Université Paris-Saclay

Data Science departement day
Sept. 12., 2022

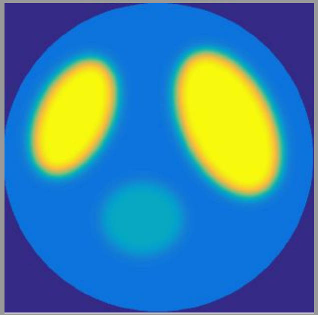
Context

- Alan Turing (1950), *Can machines think?* AI as a tool
- Dartmouth wkp (1956), AI as a goal

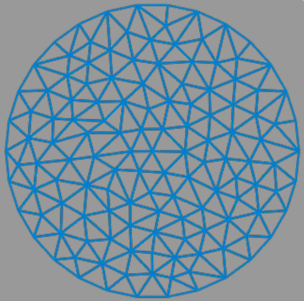
  intelligence as a sequence of logical operations
- … symbolic / subsymbolic “debate”
- **Deep Learning** coming of age (2012+): a universal tool?
  - seemingly hitting a glass ceiling (now)

Deep Hybridization is the only way to move forward

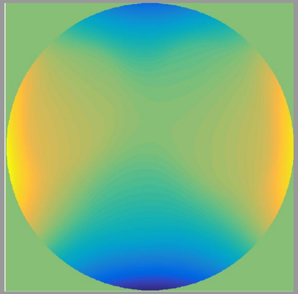
Good Old Numerical Simulation of PDEs



Conductivity σ



Finite Element Method

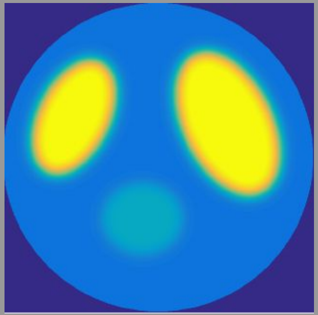


Computed current u

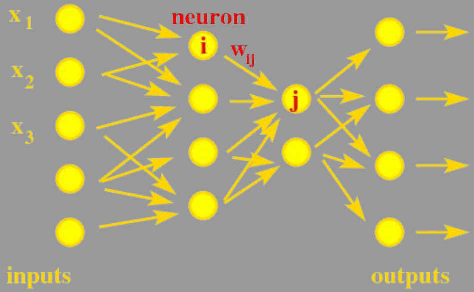
$$
\begin{aligned}
\nabla \cdot \Big(\sigma(x)\nabla u(x)\Big) &= 0, && x \in \Omega \subset \mathbb{R}^2 \\
u(x) &= u_0(x), && x \in \partial\Omega.
\end{aligned}
$$

An image recognition problem?

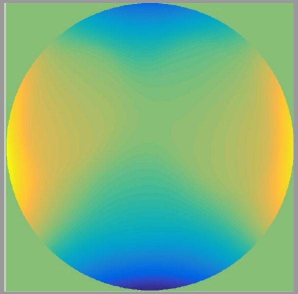
Hybrid AI 1.0



Conductivity σ



Deep Neural Network



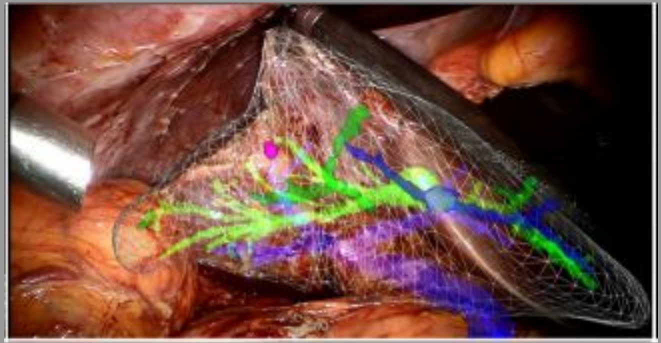
Computed current

$$
\begin{aligned}
&\nabla \cdot \Big(\sigma(x)\nabla u(x)\Big) = 0, && x \in \Omega \subset \mathbb{R}^2 \\
&u(x) = u_0(x), && x \in \partial\Omega.
\end{aligned}
$$

Soft Body Deformation

Stéphane Cotin, Mimesis

Need patient-specific real-time simulation of laparoscopy



- Liver is hyper viscoelastic and anisotropic
- Several complex PDEs for soft tissues
  - anyway an approximation
- Material identification
- Patient-specific geometry
  - Not a big issue, but time consuming
- Boundary conditions are essential
  - but difficult to obtain from images

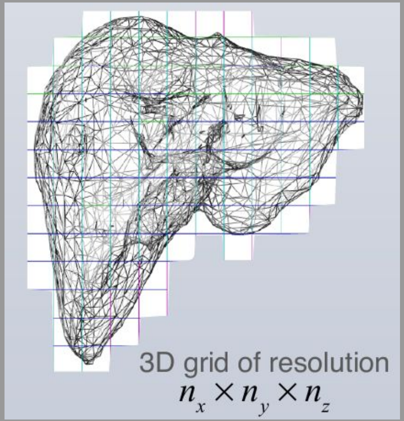
- Need less than 3mm error
- in less than 50ms per image

Mendizabal, Brunet and Cotin - Physics-based Deep Neural Network for Augmented Reality during Liver Surgery, MICCAI 2019.

Soft Body Deformation

Bottleneck: real-time simulation

→ Replace FEM simulation by a Deep Network
Supervised learning (regression) of simulation results



Deep network

input tensor ($3 \times n_x \times n_y \times n_z$) for contact forces

output tensor ($3 \times n_x \times n_y \times n_z$) for displacements

$$\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \left\| h(\mathbf{f}_n) - \mathbf{u}_n \right\|$$

weights of *h*

Training data: N FEM solutions

$$\{(\mathbf{f}_n, \mathbf{u}_n)\}_{n=1}^{N}$$

800 exemples, standard MSE loss

Soft Body Deformation

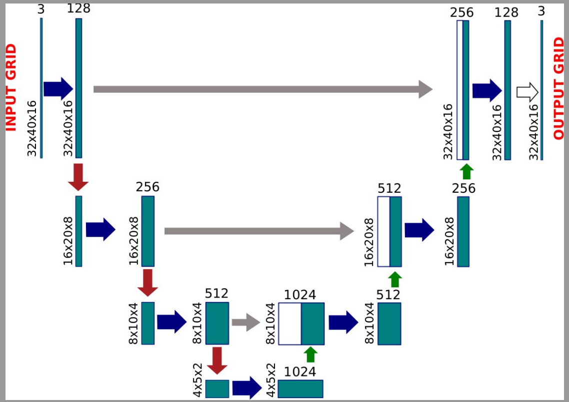
Problem solved?

- Error w.r.t. FEM below 10% → sufficient for surgeons
- 300 times faster than FEM → sufficient for real time surgery
- But single patient → only need generalization w.r.t. external forces
- Nothing there to learn for AI ?

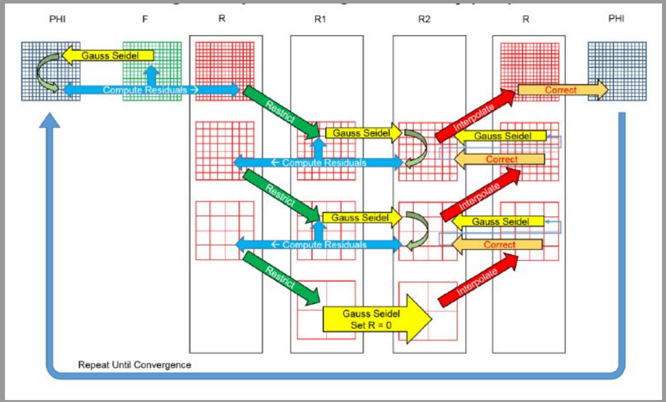
Soft Body Deformation

Problem solved?

- Error w.r.t. FEM below 10% → sufficient for surgeons
- 300 times faster than FEM → sufficient for real time surgery
- But single patient → only need generalization w.r.t. external forces
- Nothing there to learn for AI ?



U-Net architecture
Ronneberger et al., 2015



Multigrid V-cycle architecture
Briggs et al., Multigrid Tutorial, 2000

Toward Hybrid AI 2.0

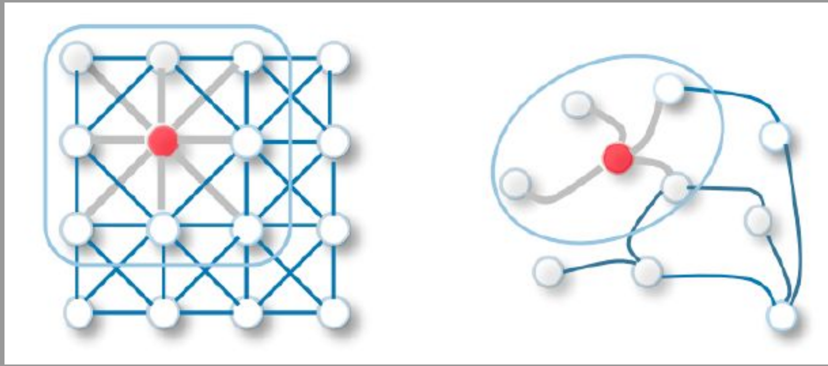
- Performance
  - Irregular geometries → Unstructured meshes → GNNs
  - Few shot learning, OoD generalization → Meta-learning
  - Accuracy → Data-driven inference as pre-processor
- Physical relevance
  - Physics in the loss
  - Residual learning
- Conclusions

Toward Hybrid AI 2.0

- Performance
  - Irregular geometries → Unstructured meshes → GNNs
  - Few shot learning, OoD generalization → Meta-learning
  - Accuracy → Data-driven inference as pre-processor
- Physical relevance
  - Physics in the loss
  - Residual learning
- Conclusions

Convolutional Graph Neural Networks

- The local “convolution” needs to accept any number of inputs, and be **permutation-invariant**
- The global mapping must be **permutation equivariant**



Convolution operation on

Regular grid Graph domain

MoNet* convolution operator

- Given **node features** x_i and **edge features** e_{ij}
- Computes node feature values from **trainable matrix** Θ_k and **kernels** w_k

$$\mathbf{x}'_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \frac{1}{K} \sum_{k=1}^{K} \mathbf{w}_k(\mathbf{e}_{ij}) \odot \mathbf{\Theta}_k \mathbf{x}_j$$

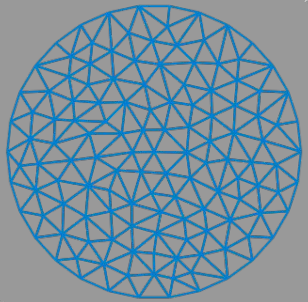
(*) Monti, F., et al. (2016): Geometric deep learning on graphs and manifolds using mixture model CNNs, CVPR.

MoNet for PDEs

PhD Wenzhuo Liu
M. Yagoubi, MS

E.g., solving some nonlinear Poisson equation

$$-\nabla((1-u(x)+u(x)^2)\cdot\nabla u(x)) = f(x) \quad in\ \Omega \text{ with } u(x)|_{\partial\Omega} = 1$$



- A mesh is a graph
- Edge features: $e_{ij} = p_j - p_i$
  where p_i is the coordinate vector of node i
- Node features:
  - value of source term at node i: $f(p_i)$
  - boundary indicator (1 if node i on the boundary, 0 otherwise)

Experimental conditions

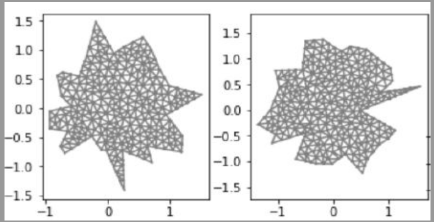
- **42 000** examples, varying both domain Ω and source function f
- FEniCSx used for simulations and mesh generation
- Number of nodes of fine meshes: ~1000
- 12-fold cross-validation,
- Loss = Mean Absolute Error w.r.t. FEM results on finest mesh
- Wilcoxon signed-rank test with 95% confidence

Hyperparameters

- Number of layers, of channels for the NNs — 2-4, 32-64
- Number of nearest neighbors for up- and down-sampling — 6
- Learning rate schedule — step decay
- Batch size, choice of optimizer and its parameters — Adam, default param.
- Initialization method : — Glorot
- Early stopping (with validation set)

Generalisation Results

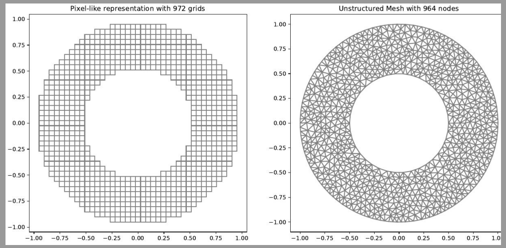
Training set: 30 vertices, ~1000 elements per mesh



		GNN	CNN
Same distribution	Test	2.20 ± 0.16	6.14 ± 0.26
30 vertices, finer meshes	1200-1600	2.53 ± 0.17	6.55 ± 0.32
	1600-2000	3.71 ± 0.25	7.21 ± 0.41
Less vertices, ~1000 elements per mesh	20 vertices	2.39 ± 0.18	5.72 ± 0.39
	10 vertices	3.56 ± 0.31	6.11 ± 0.54
	5 vertices	5.29 ± 0.41	6.40 ± 0.72

Inference Computational Cost

Time(s)	Square	Donut	Polyg.
MGMI	0.912	0.643	7.336
MG-UNet	1.075	0.782	11.312
FEM	173.87	164.35	163.72



- Inference time for solving 5000 sample PDEs
  - By batches of 100 on one GPU GTX 1080Ti for learned MGX models
  - Sequentially by FEniCSx on a Intel(R) Xeon(R) Silver 4108 CPU

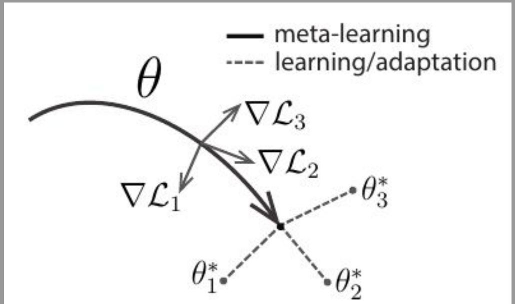
Toward Hybrid AI 2.0

- Performance
  - Irregular geometries → Unstructured meshes → GNNs
  - Few shot learning, OoD generalization → Meta-learning
  - Accuracy → Data-driven inference as pre-processor
- Physical relevance
  - Physics in the loss
  - Residual learning
- Conclusions

Model-Agnostic Meta-Learning - MAML(*)

- Given a set of tasks T_i and corresponding examples
- Learn a partial model f_θ such that
  - few examples and few gradient steps are needed

    to learn θ_i* that solves task i, forall i

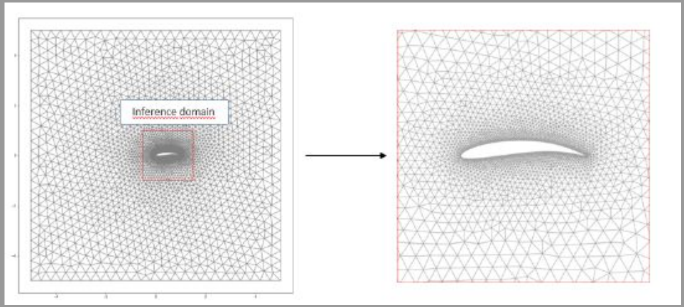


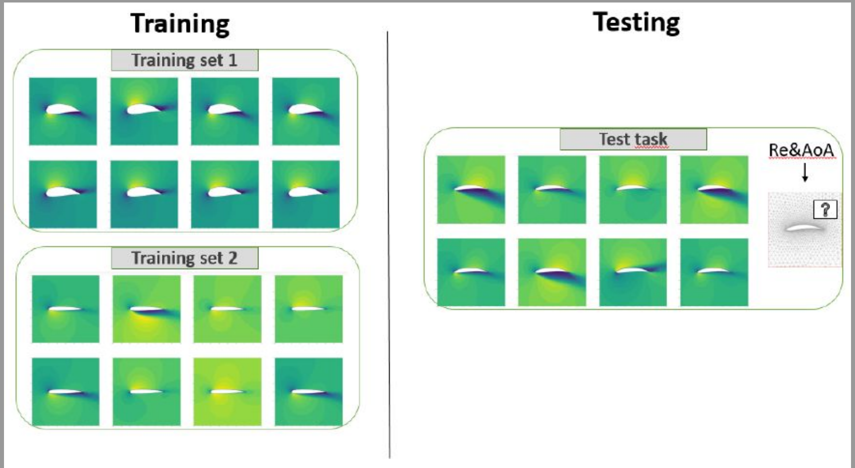
- MAML++(**) relaxes the loss over several steps

(*) Finn, Abbeel, and Levine (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. 34th ICML
(**) Antoniou, Edwards, Storkey (2018). How to train your MAML. arXiv 1810.09502.

MAML++ for RANS

- Reynolds-Averaged Navier-Stokes Equations
- Single-task: U-Net GNNs
  - with limited inference domain
  - Poor generalization w.r.t. airfoil geometry
- Multi-task: One task = one airfoil
  - One example = Reynolds number and Angle of Attack





MAML++ Results

10-fold cross-validation for the whole process

- Test Set 1 : same 72 training airfoils, 10 new examples each
- Interpolation Set 2 : 20 new NACA airfoils, 50 examples each
- OoD* Test set : 20 thinner airfoils, 50 examples each

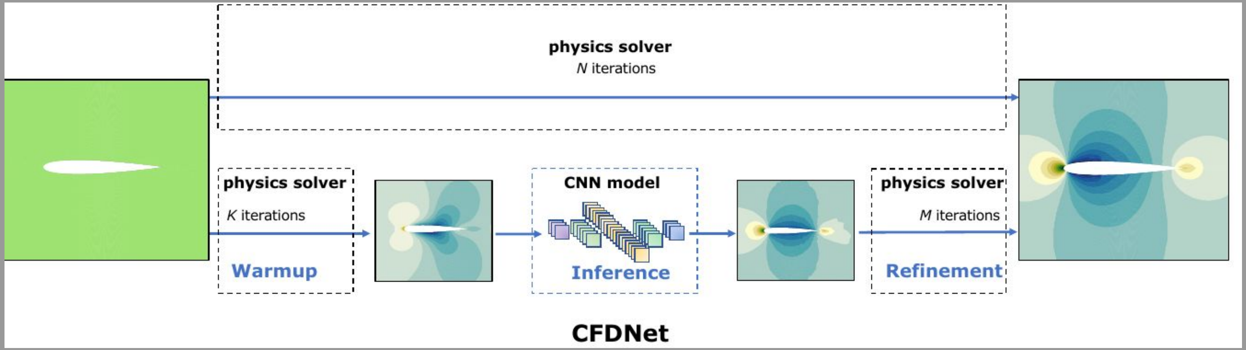
	MAML++	Baseline	Baseline with FineTune
Test set 1	0.0079 ± 0.0008	$\mathbf{0.0071 \pm 0.0003}$	0.0071 ± 0.0003
Interpolation set	$\mathbf{0.0117 \pm 0.0014}$	0.0158 ± 0.0015	0.0156 ± 0.0014
OOD set	$\mathbf{0.0369 \pm 0.0028}$	0.0643 ± 0.0179	0.0601 ± 0.0155

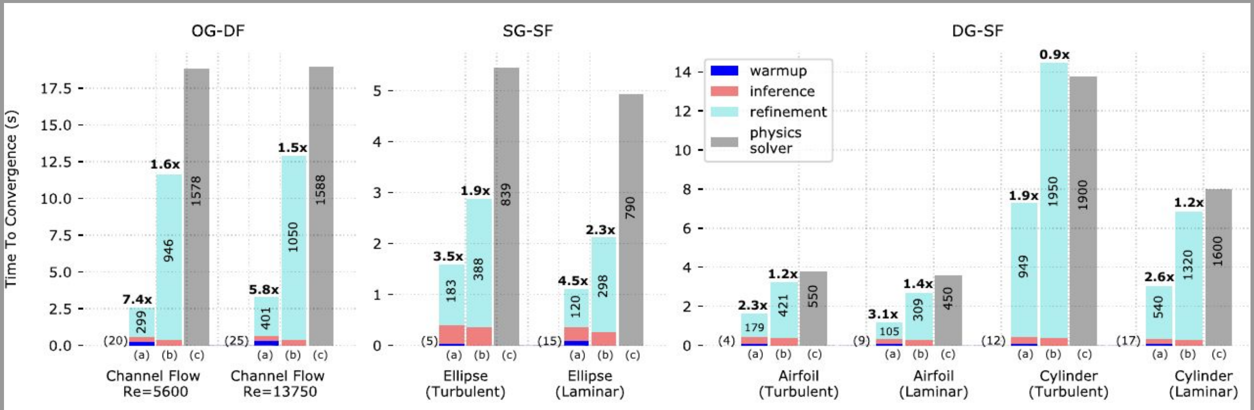
(*) OoD = Out of Distribution

Toward Hybrid AI 2.0

- Performance
  - Irregular geometries → Unstructured meshes → GNNs
  - Few shot learning, OoD generalization → Meta-learning
  - Accuracy → Data-driven inference as pre-processor
- Physical relevance
  - Physics in the loss
  - Residual learning
- Conclusions

Sequential Data-based - FEM





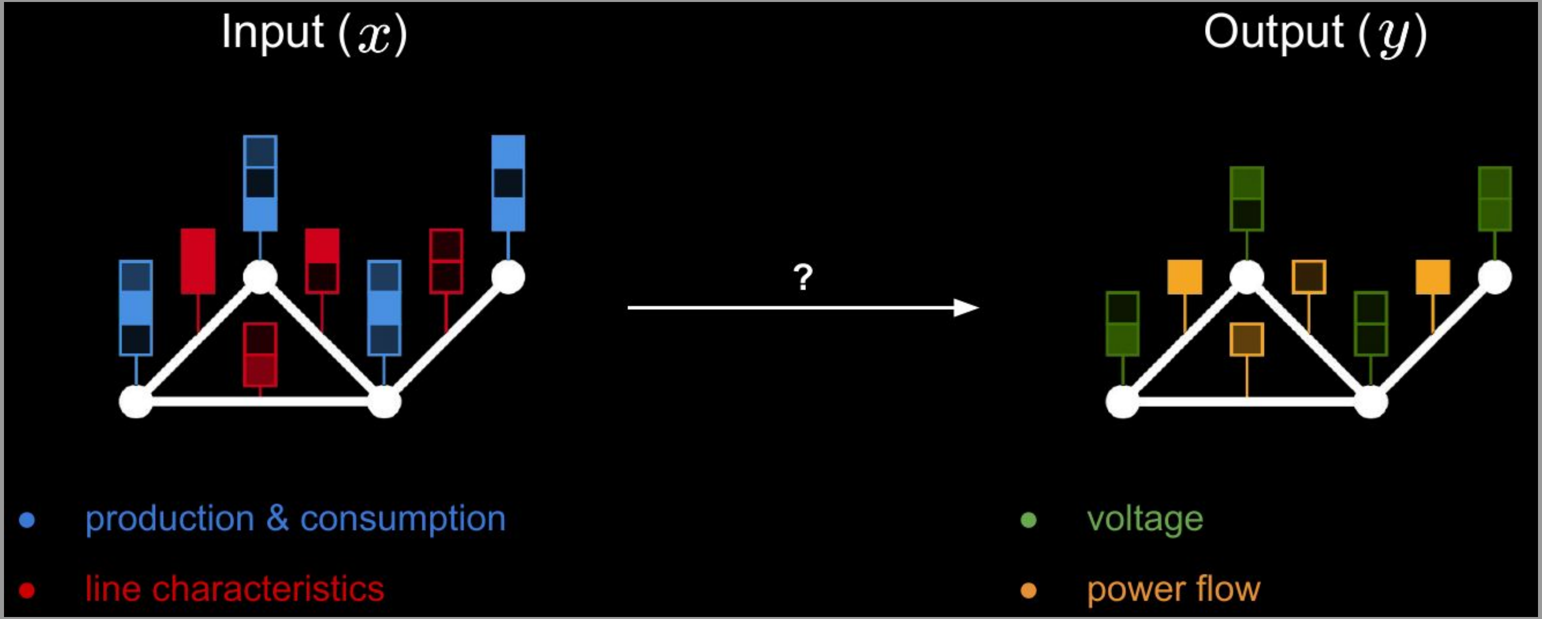
Obiols-Sales, Vishnu, Malaya, et al. (2020). CFDNet: A deep learning-based accelerator for fluid simulations. ICS’20.

Toward Hybrid AI 2.0

- Performance
  - Irregular geometries → Unstructured meshes → GNNs
  - Few shot learning, OoD generalization → Meta-learning
  - Accuracy → Data-driven inference as pre-processor
- Physical relevance
  - Physics in the loss
  - Residual learning
- Conclusions

Power Grid simulation

PhD Balthazar Donon
(I. Guyon, MS)



- Physics: Kirchhoff law in all lines
- Traditional approach: Newton-Raphson, but too slow

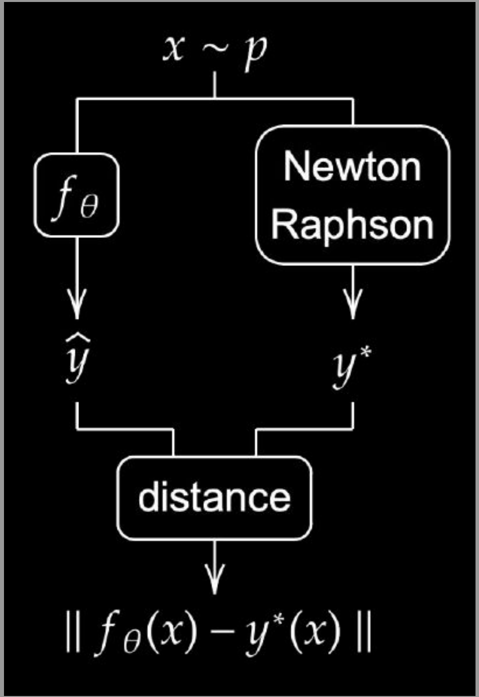
Donon, Liu, Marot, Liu, Guyon, Schoenauer (2020). Deep Statistical Solvers. NeurIPS.

The surrogate approach

- Compute solution y*(x) for many x~*p*
- Learn a surrogate model f_θ

Main issue: generalization to different grid topologies

  - Graph Neural Networks mandatory
    - invariance and equivariance
  - Requires many y*(x)



Physics informed approach

- y*(x) can also be defined as

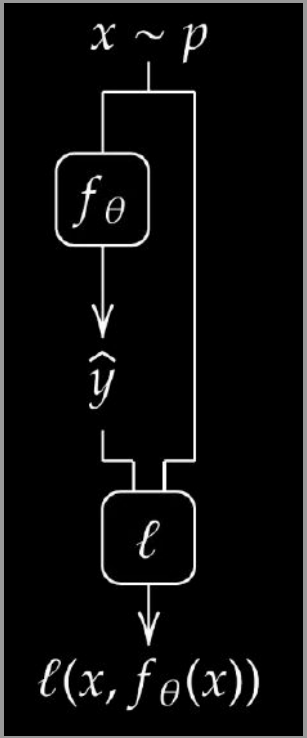
$$y^*(x) = \arg\min_{y \in \mathcal{Y}} \ell(x, y)$$

ℓ : violation of physical laws
(Kirchhoff's laws)

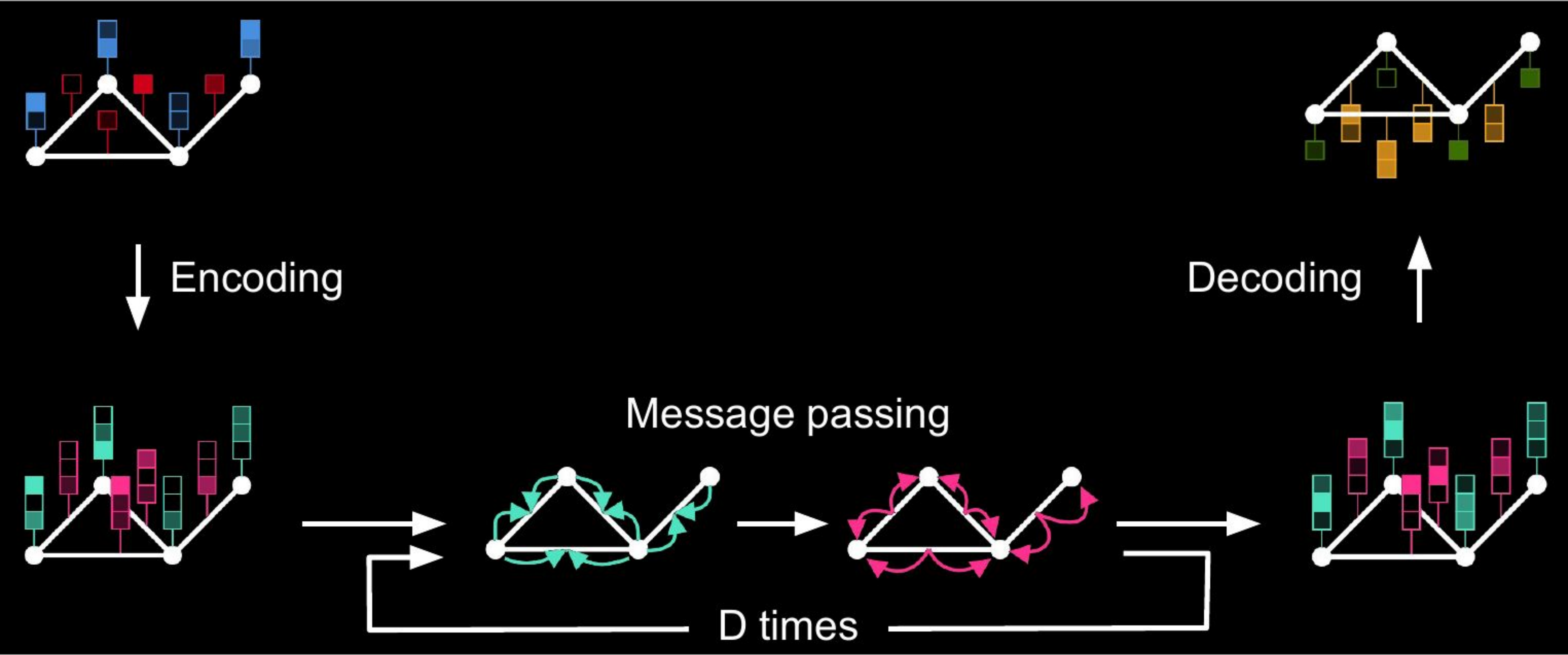
- and the Holy Grail is to find

$$\theta^* = \arg\min_{\theta \in \Theta} \mathbb{E}_{x \sim p} \left[\ell(x, f_\theta(x)) \right]$$

- $\rightarrow$ Message Passing Graph Neural Networks
- aka **Deep Statistical Solver**



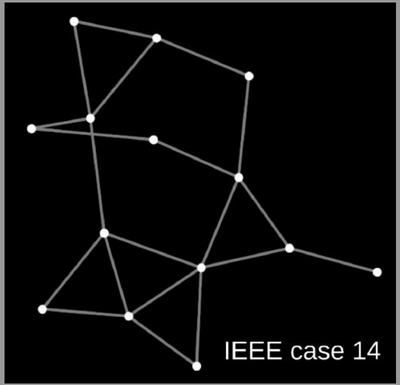
MPGNN in a nutshell

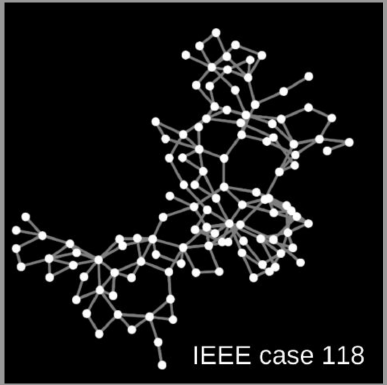


Power Grids Experiments

It works!

- same accuracy than Newton-Raphson
- good generalization to (not too different) topologies
- Two orders or magnitude faster … for batches of topologies





… on small networks

Back to PDEs

PhD Matthieu Nastorg
(A. Bucci, G. Charpiat, Th. Faney, MS)

Let $\mathcal{P}$ be a set of Poisson problems, parametrized by $p \in \Sigma$

$E_p = (\Omega_p, f_p, g_p) \in \mathcal{P}$

$$E_p = \begin{cases} -\Delta u_p = f_p & \in \Omega_p \\ u_p = g_p & \in \partial\Omega_p \end{cases}$$

FEM discretization $\rightarrow$

$E_{h,p} = (\Omega_{h,p}, A_p, B_p) \in \mathcal{P}_h$

$$A_p U_p = B_p$$

$\rightarrow$

ML solver, parametrized by $\theta \in \Theta$

$$U_p = \text{solver}_\theta(E_{h,p})$$

Finite Element discretization

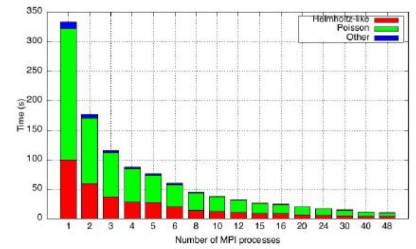
$$A_{ij} = \int_\Omega \nabla\phi_j \cdot \nabla\phi_i \, dx$$

$$B_i = \begin{cases} \int_\Omega f\, \phi_i \, dx & \text{if } i \text{ is interior} \\ g_i & \text{if } i \text{ is boundary} \end{cases}$$

Define a loss function on a pair $(U_p, E_{h,p})$: $\mathcal{L}(U_p, E_{h,p}) = ||A_p U_p - B_p||^2$

Given a distribution $\mathcal{D}$ over Σ, find θ^* such that : $\theta^* = \underset{\theta \in \Theta}{\text{argmin}} \underset{p \sim \mathcal{D}}{\mathbb{E}} [\mathcal{L}(\text{solver}_\theta(E_{h,p}), E_{h,p})]$

e.g., Poisson equation for pressure in the Splitting scheme strategy for incompressible Navier-Stokes equations



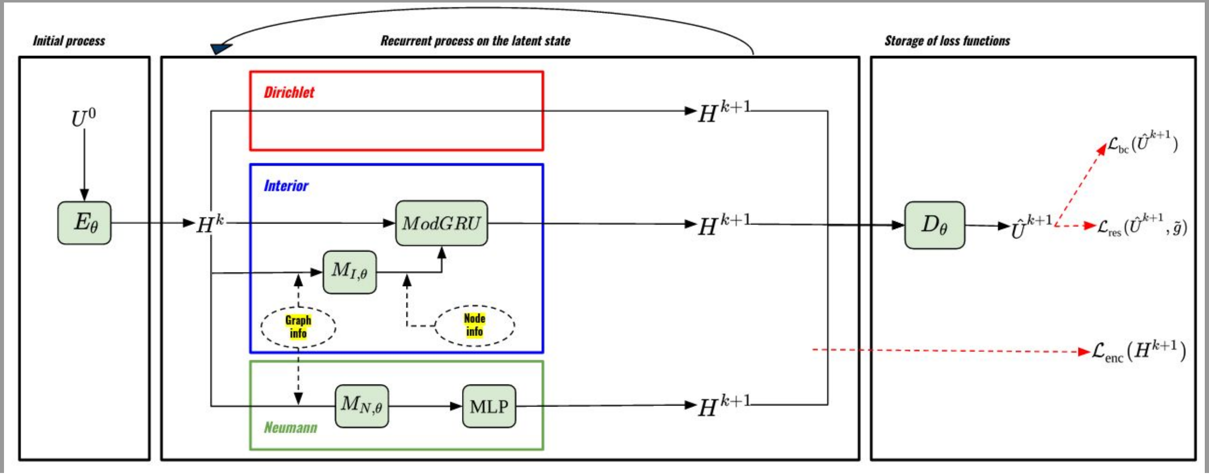
*

(*) Yushan Wang (2015). Solving incompressible Navier-Stokes equations on heterogeneous parallel architectures. PhD thesis, LRI, U. Paris-Sud.

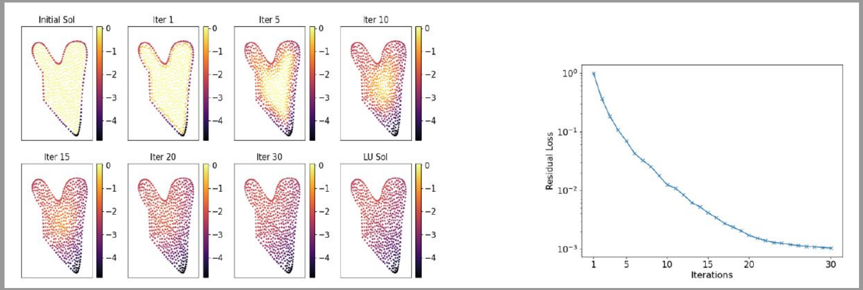
Deep Statistical Solver for discretized PDEs

Problem: solve $A_p U_p = B_p$

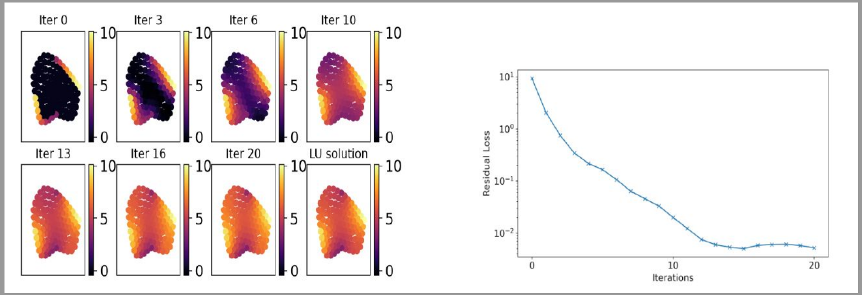
	Original DSS (Donon)	Recurrent MP (Nastorg)
Edge features:	A_{ij}	$p_i - p_j$, $\|p_i - p_j\|$
Node features:	B_i	f_i if i interior,
Loss:	$\sum_{i\in[n]}\left(\sum_{j\in[n]} A_{ij}U_j - B_i\right)^2$	same + $\left\Vert \mathbf{E}_\theta(\mathbf{D}(\mathbf{H}^k)) - \mathbf{H}^k \right\Vert^2$ + $\left\Vert \mathbf{D}_\theta(\mathbf{E}(\hat{\mathbf{U}}^k)) - \mathbf{U}^0 \right\Vert^2_{\partial\Omega}$
Architecture:	37530 weights	4276 weights



First results



Dirichlet boundary condition



Mixed Dirichlet/Neuman boundary cond.

Metrics	Residual loss	RMSE	NRMSE	Time for LU	Time for DSS
Problem $\mathbb{P}_1$	9.4e-4	8.2e-3	3.e-3	2.4	1.8
Problem $\mathbb{P}_2$	5.0e-3	9.0e-2	8.0e-3	2.4	1.8

Open issue: number of iterations …

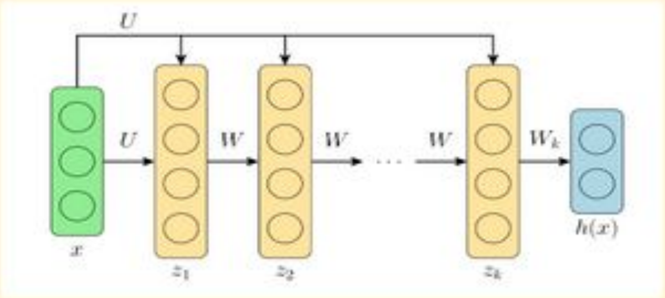
Deep Equilibrium Models(*)

- Directly solve for the fixed point

$$z^\star = \sigma(Wz^\star + Ux + b)$$

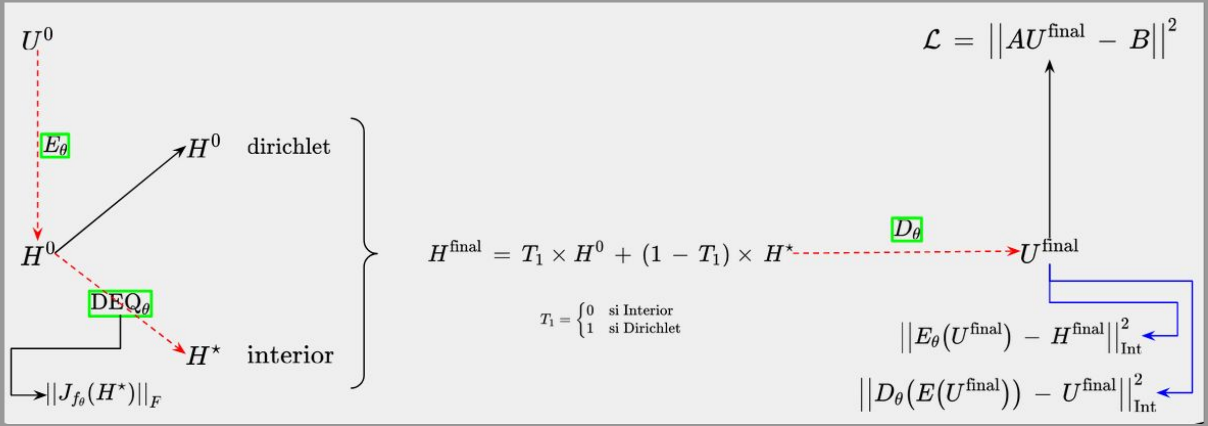
  - using some RootFind routine
  - + approximation of the gradient
  - + regularization of its norm

$$\begin{cases} z_1 &= 0 \\ z_{i+1} &= \sigma(Wz_i + Ux + b), \quad i = 1, \cdots, k-1 \\ h(x) &= W_k z_k + b_k \end{cases}$$



(*) Bai, Kolter, Koltun, NeurIPS 2019

Deep Equilibrium Models



$$\mathrm{DEQ}_\theta = \mathrm{RootFind}(f_\theta(H, G) - H)$$

$$\text{Training Loss} : L = \underbrace{\left|\left|AU^{\text{final}} - B\right|\right|^2}_{\textbf{Residual}} + \underbrace{w \times ||J_{f_\theta}(H^\star)||_F}_{\textbf{Jacobian Frobenius Norm}} + \underbrace{\left|\left|E_\theta(U^{\text{final}}) - U^{\text{final}}\right|\right|^2}_{\textbf{Encoder}} + \underbrace{\left|\left|D_\theta(E(U^{\text{final}})) - U^{\text{final}}\right|\right|^2}_{\textbf{Decoder}}$$

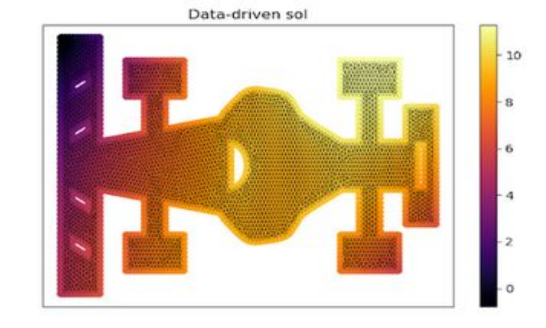
DEQ Results

- ➔ Hyperparameters :
  - Latent dimension : 20
  - Solver : Broyden (quasi-Newton method)
  - Tolerance forward : 1.e-6
  - Threshold forward : 600
  - Tolerance backward : 1.e-8
  - Threshold backward : 600
  - Learning rate : 0.01
  - Gradient clipping : 0.01
  - Scheduler : Reduce LR in plateau with step : 0.5
  - Jacobian spec.rad weight : 1.0

- ➔ Total number of weights : 7241
- ➔ Sanity check: ||D(E(U)) - U||^2 around 1.e-5

Nb nodes	Residual	MSE
250-950	**2.27e-3**	**1.06e-2**

10000 training examples
Avg results on 1000 test samples



3882	**1.20e-3**	**3.82e-2**

result on one OoD example

Toward Hybrid AI 2.0

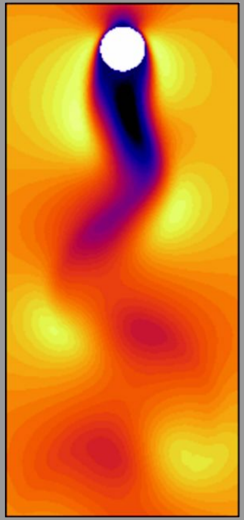
- Performance
  - Irregular geometries → Unstructured meshes → GNNs
  - Few shot learning, OoD generalization → Meta-learning
  - Accuracy → Data-driven inference as pre-processor
- Physical relevance
  - Physics in the loss
  - Residual learning
- Conclusions

Reduced Order Models

- Dynamical systems governed by $$\frac{\partial u(x,t)}{\partial t} = g(u(x,t))$$

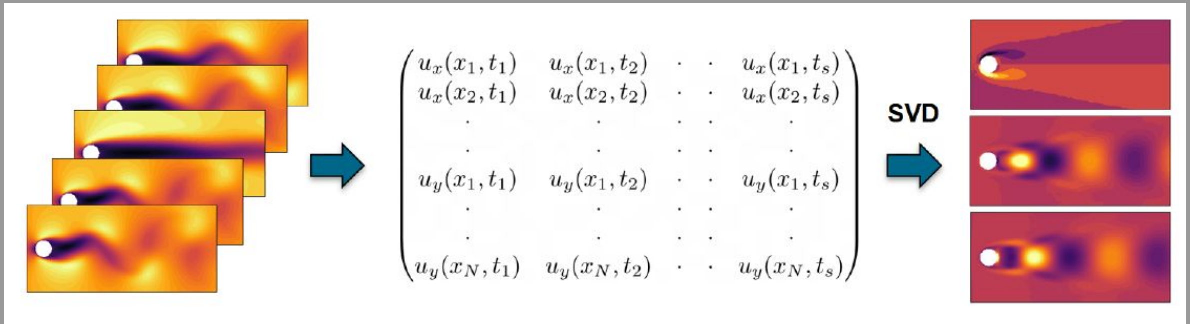
- Issues with numerical simulations
  - small scale structures and fast events → fine discretization
  - stability, numerical dispersion and dissipation

- Does not scale with intrinsic dimension



Limit cycle
Intrinsic dim 2

POD (aka SVD, aka PCA, ...)



- Dimension reduction to a reduced number of modes

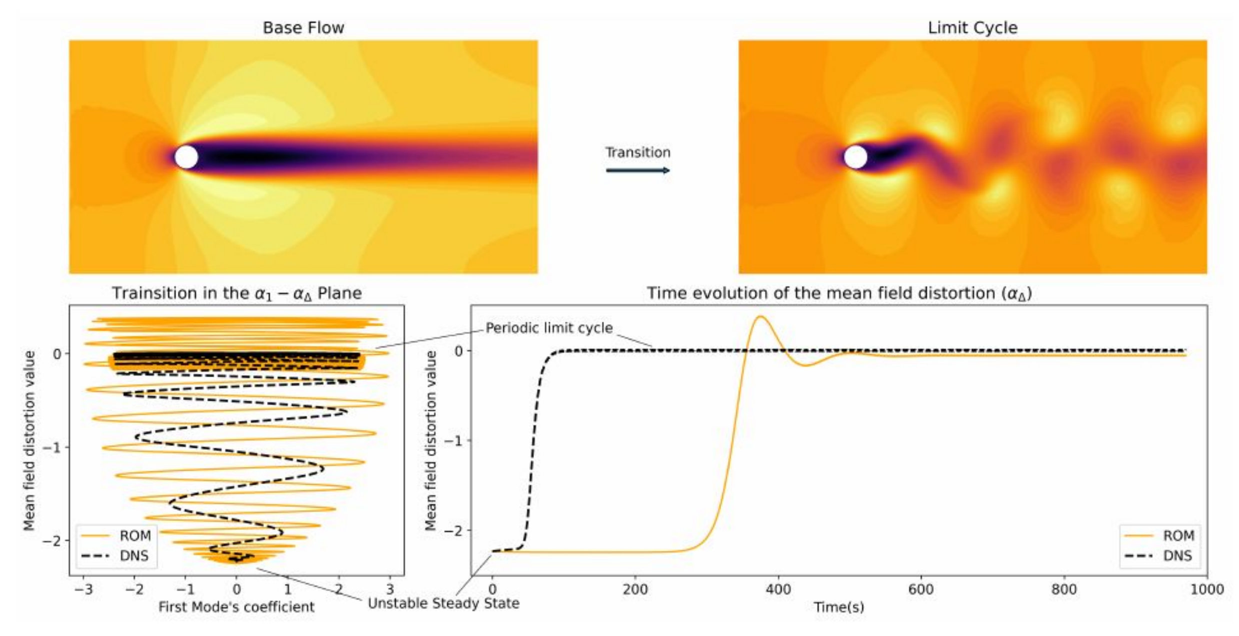
$$\begin{matrix} u(x,t) \approx \bar{u} = V(x)\alpha(t) \\ V \in \mathbb{R}^{n_x \times r}, \alpha \in \mathbb{R}^r \end{matrix} \implies \frac{d\alpha}{dt} = V^T g(u(x,t))$$

- Dynamics computed from simplified solution → error

$$\frac{d\alpha}{dt} \approx V^T g(\bar{u}) \neq V^T g(u(x,t))$$

ROM error

From Noack* equations:



(*) Noack et al. (2003), A hierarchy of low-dimensional models for the transient and post-transient cylinder wake", J. Fluid Mechanics.

Complementary Deep - ROM

PhD Emmanuel Menier

- Goal: learn a correction from data:

$$\frac{d\alpha}{dt} = V^T g(u) = V^T g(\bar{u}) + f$$

- Taken's theorem → all information can be retrieved from past trajectory
- Delay Differential Equation to aggregate information from the past in a time continuous manner:

$$\frac{dx}{dt} = f(t, x, y), \qquad y(t) = \int_{-\infty}^{0} x(t+\tau) e^{\lambda\tau} d\tau, \qquad \lambda \in \mathbb{R}_+$$

  - is solved as an augmented ODE system:

$$\begin{array}{ccc} \frac{dx}{dt} & = & f(t, x, y) \\ \frac{dy}{dt} & & x - \lambda y \end{array}$$

Menier, Bucci, Yagoubi, Mathelin, Schoenauer (2022), CD-ROM: Complementary Deep-Reduced Order Model, arXiv.

Neural correction

- Learn ROM correction from memory y:

$$\begin{aligned}\frac{d\alpha}{dt} &= V^T g(V\alpha) &+& \quad \mathcal{N}\mathcal{N}(y)\\ \frac{dy}{dt} &= \alpha &-& \quad \Lambda y\end{aligned}$$

where Λ is a matrix to account for the different time scales

- Augment memory dimension (information bottleneck)

$$\begin{aligned}\frac{d\alpha}{dt} &= g(V\alpha) &+& \quad \mathcal{N}\mathcal{N}(Y)\\ \frac{dY}{dt} &= Enc(\alpha) &-& \quad \Lambda Y\end{aligned}$$

where Enc is some learnable encoding

Training CD-ROM

- **Training data** from observed (computed with DNS) trajectory

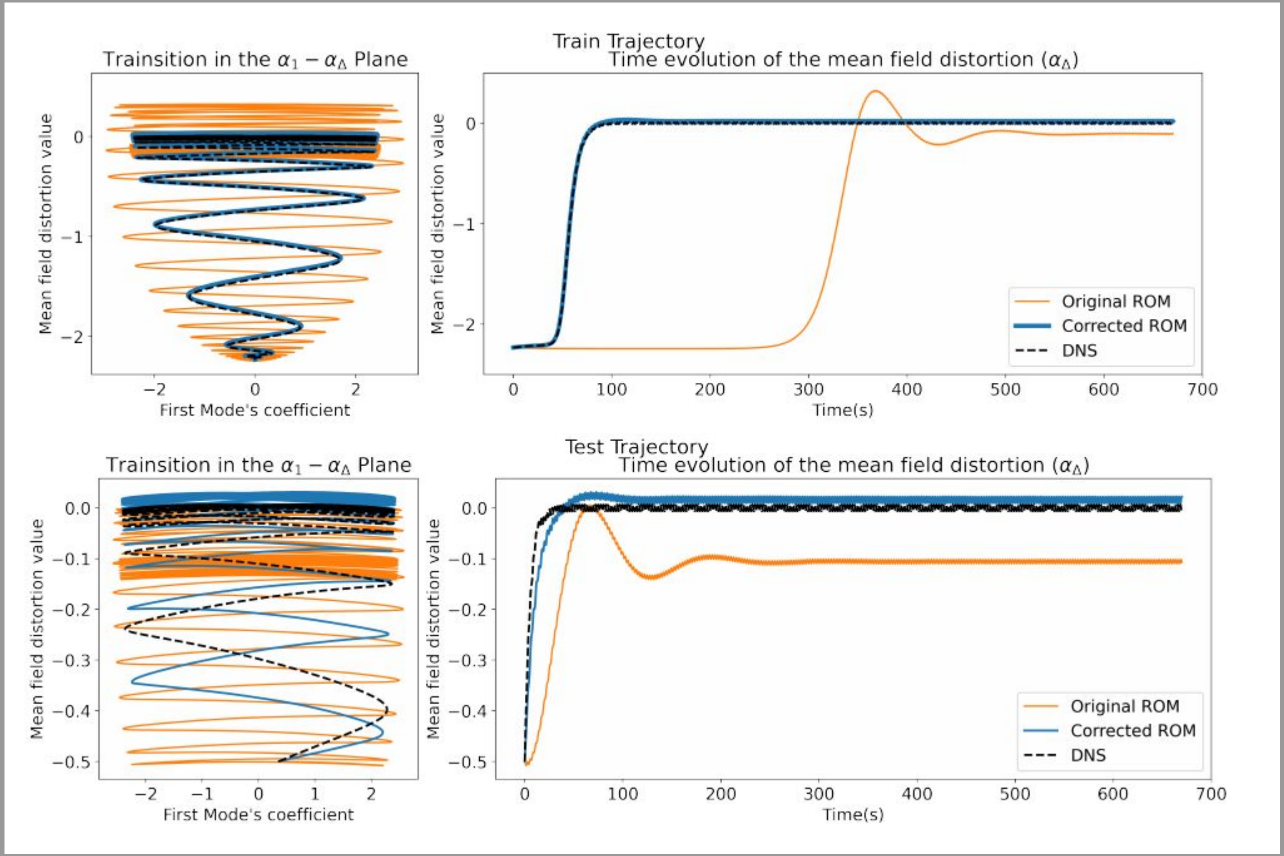
$$\hat{\alpha}_{t_i} = V^T u_{DNS,t_i}$$

- with **loss** $\mathcal{L}(\alpha) = \sum_{t_0}^{t_n} \|\alpha_{t_i} - \hat{\alpha}_{t_i}\|_2$

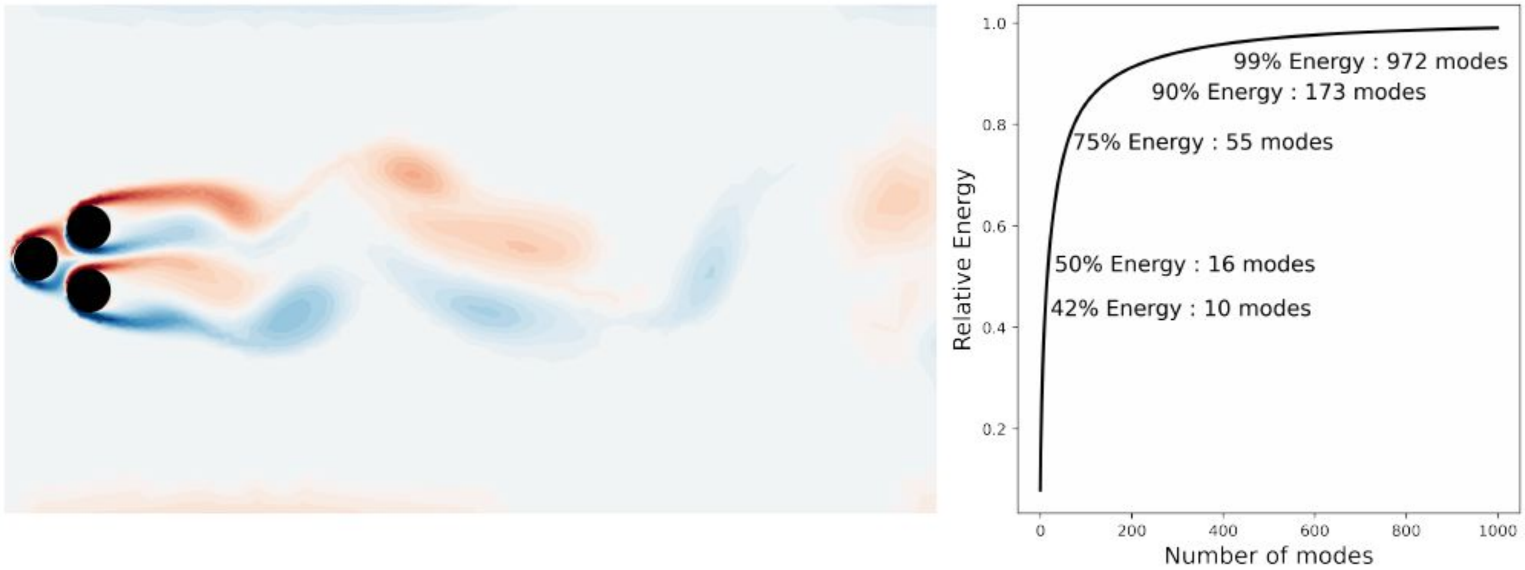
  (+ regularization terms)

- Residual and Encoder are simple multilayer perceptrons (~5 layers) with ReLU activation fn.
- Optimizer is Adam with standard parameters
- Λ is initialized randomly
- Memory is initialized from the past of the true trajectory

Results: the cylinder

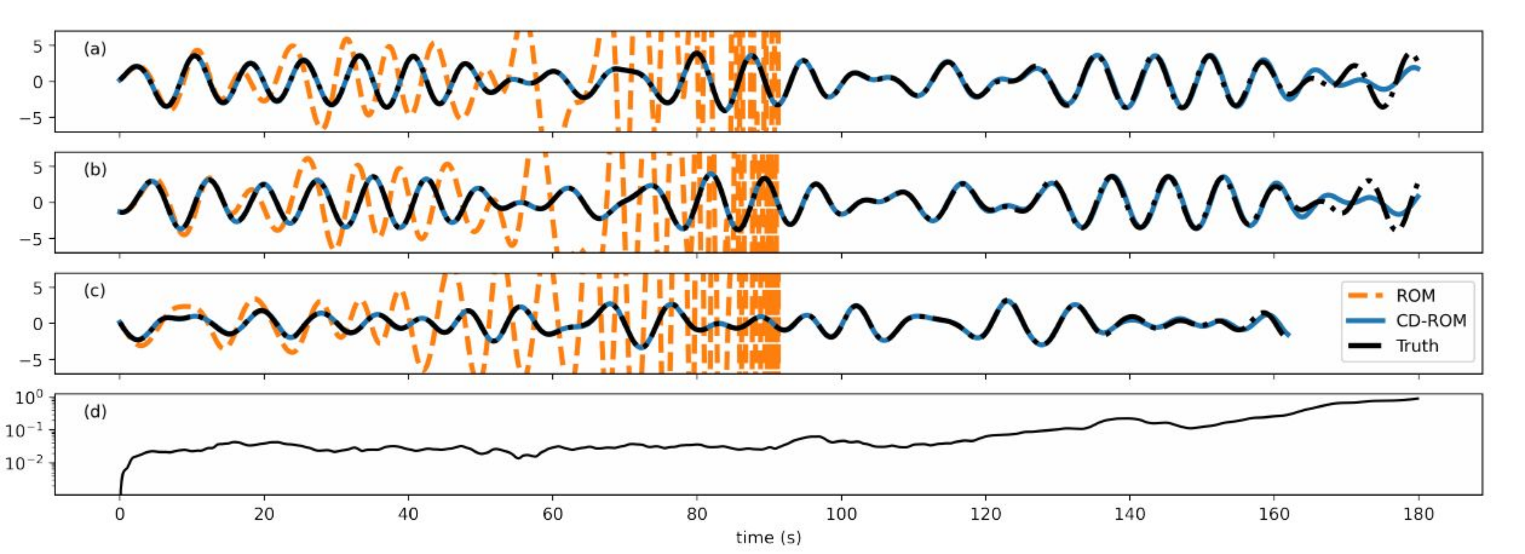


The Fluidic Pinball



Much less energy in the first modes

Results on the Pinball



The 10-mode ROM: First 3 modes, and relative distance between true and simulated trajectory

Data-driven / model-based PDE numerical simulations

- DL (e.g., GNNs) as a **powerful surrogate model**, almost “off-the-shelf”
  - Requires huge number of examples
  - Lack of accuracy (and certification thereon)
  - + delicate tuning of hyperparameters → AutoML
- **Meta-learning** improves the situation, but
  - OoD DL results still not accurate enough → warm start of FEM
- Inserting **mechanistic knowledge**
  - In the loss
  - Based on standards approximations, e.g. ROM

Requires expert knowledge **in both fields**

Toward Hybrid AI 2.0

- Performance
  - Irregular geometries → Unstructured meshes → GNNs
  - Few shot learning, OoD generalization → Meta-learning
  - Accuracy → Data-driven inference as pre-processor
- Physical relevance
  - Physics in the loss
  - Residual learning
- Conclusions

Context

- Alan Turing (1950), *Can machines think?* AI as a tool
- Dartmouth wkp (1956), AI as a goal

  intelligence as a sequence of logical operations
- … symbolic / subsymbolic "debate"
- **Deep Learning** coming of age (2012+): a universal tool?
  - seemingly hitting a glass ceiling (now)

Deep Hybridization is the only way to move forward

Hybrid AI 2.0

Deeply Hybridize Deep Learning with

- Symbolic AI in the quest for **trustworthiness**
- Formal proofs and certification
- Control theory
- PDE theory
  - numerical simulation (this talk)
  - theoretical insights (ODE-Net, …)
- Statistical physics
- Biology (genetics, omics, …)
- you name it

All talks in this workshop :-)

Hybrid AI 2.0

AI as a tool

What has changed

- GAFAMs, BATXs, and other tech giants invest in basic research
  - more than public research can afford
  - in both material and human resources (brain drain)
  - “freedom or research” not any more an exclusive argument
- Irrational exuberance (>10 000 submissions at NeurIPS)

But

- Modern AI is, or could/will be, ubiquitous
- Only Public Research has the required diversity

The future of Public Research in AI lies in hybridizations

Hybrid AI 2.0

Thank you