AI and Physical Sciences @AMU
Nov. - 12 – 2024
https://ai-physicalsciences.lis-lab.fr/

Physics-Aware Deep Learning for Modeling Dynamical Systems

Patrick Gallinari
Sorbonne University & CRITEO AI Lab – Paris
patrick.gallinari@sorbonne-universite.fr, p.gallinari@criteo.com
Summary of PG AI4Science activity available at https://pages.isir.upmc.fr/gallinari/chaire-ia/

Outline

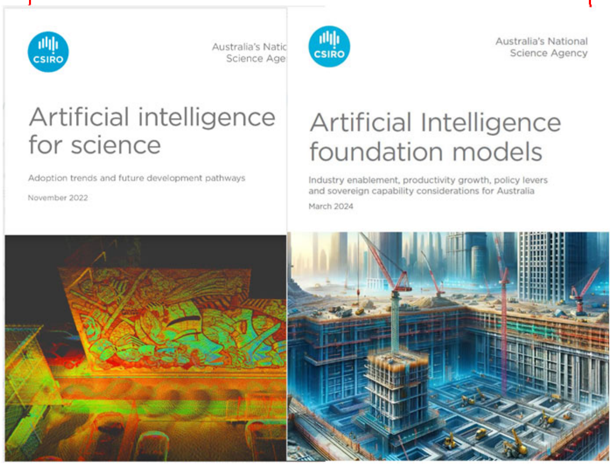
- AI4Science
  - Domains and opportunities
- Modeling dynamical systems: challenges for the adoption of AI
  - $C1$: Integration of physical and deep learning models: hybrid modeling
  - $C2$: Generalization: data-driven approaches beyond training data distribution
  - $C3$: Neural operators: mesh free approaches for simulation

AI4Science
AI as a new scientific paradigm

AI for Science

- Since 2010 AI successes mainly concern the virtual world (semantics, game)
- AI4Science emerged in the 2020
- AI for science as a new scientific paradigm

Australia NSA 2022 & 2024



US-DOE 2020 & 2022

EU 2024

AI4Science: domains and opportunities

2020: emerging topic

Examines the potential of AI for application domains

- Materials, environment, life sciences, high energy physics, smart energy infrastructures, etc

Challenges

- Data
  - AI Assisted data acquisition, hypothesis testing, compression, platforms, …
- Experiments
  - self driving laboratories (biology), coupling AI and robots
- Models
  - e.g. digital twins
- AI challenges
  - data complexity, physical plausibility, integrating domain knowledge, etc

2022: Evolution large scale

AI reflects an inflection point.

- Perspective change: from AI as a tool towards AI as support for scientific exploration

Highlights 6 AI paradigms including

- Surrogate models
  - Climate, cosmology, high energy physics, ...
- Foundation models
- Inference and inverse design
  - Material, chemistry, biology, molecular discovery, …
- …

AI4Science – example: weather forecasting

2022-2023 – Foundation Models for weather prediction (ERA5 dataset 40 years hourly reanalysis data)

- GraphCast – Google & DeepMind 2022
  https://arxiv.org/abs/2212.12794
  Blog&Demo: online demo of weather prediction
- ClimaX – Msoft & UCLA 2023
  https://arxiv.org/abs/2301.10343
- Pangu-Weather – Huawei 2023
  http://arxiv.org/abs/2211.02556
- FourCastNet – NVIDIA&Lawrence Berkeley lab.&al.
  http://arxiv.org/abs/2202.11214
- Neural General Circulation Model – Google 2023
  https://arxiv.org/abs/2311.07222
- Aurora - Microsoft 2024
  https://arxiv.org/abs/2405.13063
- AIFS, Artificial Intelligence Forecasting System - ECMWF 2024
  - https://arxiv.org/abs/2406.01465

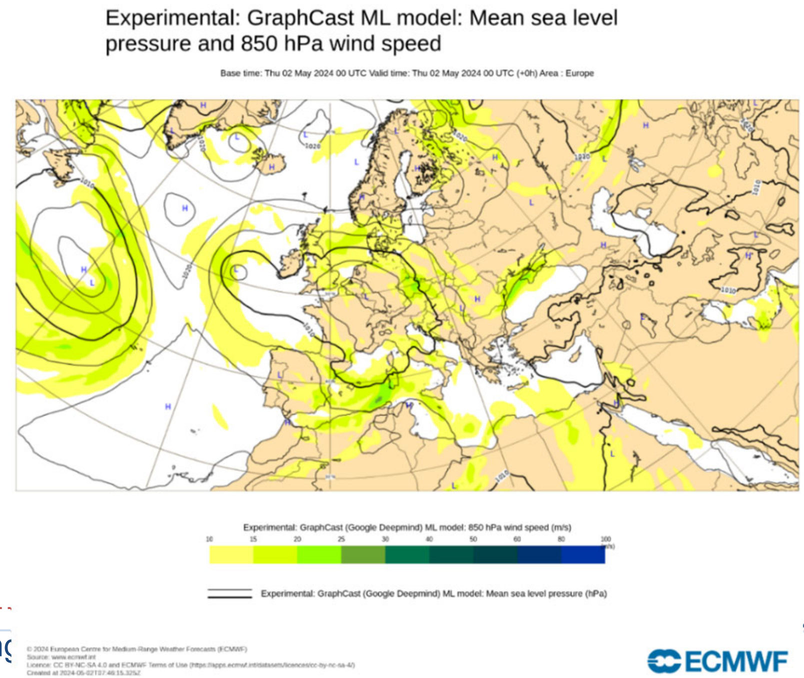
C4: Foundation models for weather forecasting: GraphCast (Lam et al. 2023 - Google)

- Data-driven approach to weather forecast
- Learn from historical data
  - Training: 39 years (1979-2017) of historical data from ECMWF ERA5 reanalysis archive – petabytes of data
    - ECMWF: European Center for Medium-Range Weather Forecast
  - Test: 2018 onward
  - Time step: 6 hours
  - State variables
    - 5 surface variables (temperature, wind speed, etc)
    - 6 athmospheric variables (temp., wind, etc) at 37 pressure levels
    - 0.25° latitude/ longitude grid, 28x28 kilometer resolution, $1M$ points
- Objective
  - Given state variables at t and $t - 6$ hours, predict next state $(t + 6)$
  - Prediction horizon: 10 days (medium range), auto-regressive model

AI4Science: example: weather forecasting and climate
Foundation models for weather forecasting: GraphCast (Lam et al. 2023 - Google)

- ECMWF is running a series of data-driven forecasts as part of its experimental suite.
  - Quote: “These ML-based weather forecasts first approached the skill of the IFS (used as the benchmark for high-quality forecasting), then matched IFS skill, and then claimed to surpass our scores. What’s more, making a forecast with these models requires only a single GPU, takes less than a minute, and consumes a tiny fraction of the energy required for an IFS forecast.”

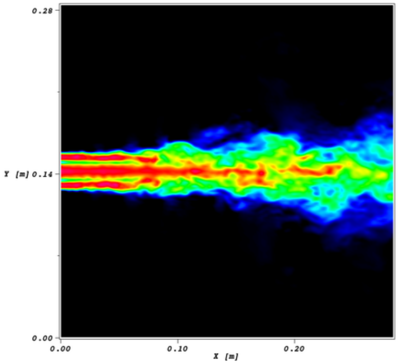
[See model’s forecasts on ECMWF website]()



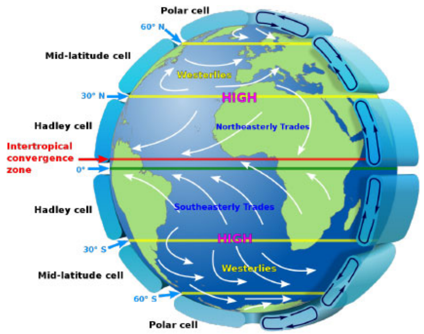
These models are free and can be downloaded

- Applications domains - examples

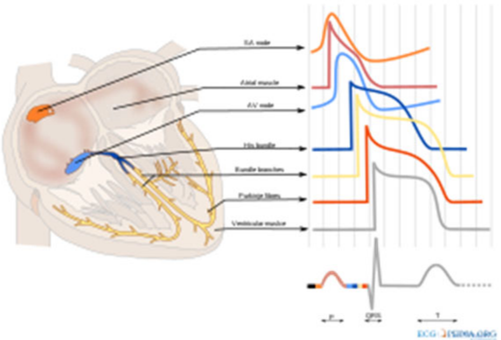
Computational Fluid Dynamics



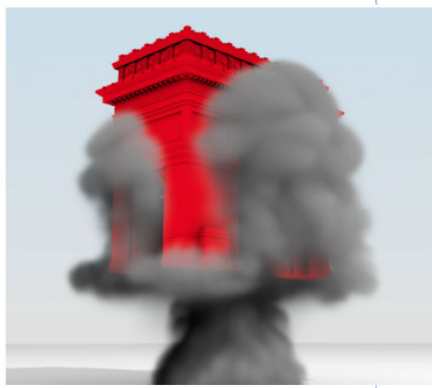
Earth System Science - Climate



Biology



Graphical design



Tompson et al. 2017

C1: Incorporating physical knowledge in statistical dynamics models – hybrid systems

C1 -Incorporating physical knowledge
Why hybrid systems - motivation

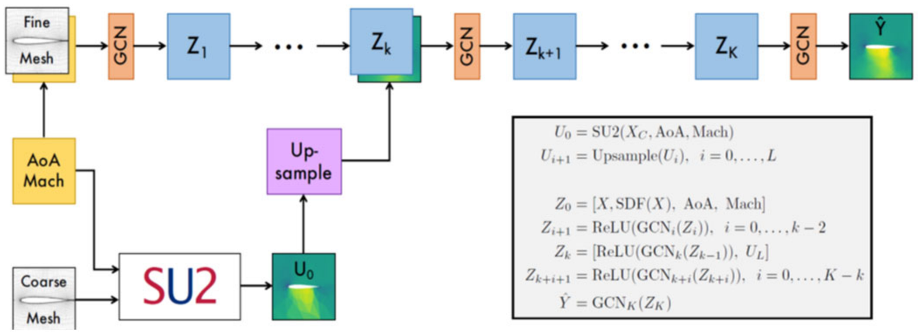
Accelerate computations

- Direct numerical simulation is usually intractable
- Approximate the high-precision simulation at a lower computational cost

Complement physical models

- Part of the physics is unknown or not considered in the model
- Learn the missing information from data

Low to high resolution (Belbute-Peres et al. 2020)



$$
\begin{aligned}
U_0 &= \mathrm{SU2}(X_C, \mathrm{AoA}, \mathrm{Mach}) \\
U_{i+1} &= \mathrm{Upsample}(U_i), \quad i = 0, \ldots, L \\
Z_0 &= [X, \mathrm{SDF}(X), \mathrm{AoA}, \mathrm{Mach}] \\
Z_{i+1} &= \mathrm{ReLU}(\mathrm{GCN}_i(Z_i)), \quad i = 0, \ldots, k-2 \\
Z_k &= [\mathrm{ReLU}(\mathrm{GCN}_k(Z_{k-1})), U_L] \\
Z_{k+i+1} &= \mathrm{ReLU}(\mathrm{GCN}_{k+i}(Z_{k+i})), \quad i = 0, \ldots, K-k \\
\hat{Y} &= \mathrm{GCN}_K(Z_K)
\end{aligned}
$$

Neural General Circulation Model – weather prediction (Kochkov et al. 2023)

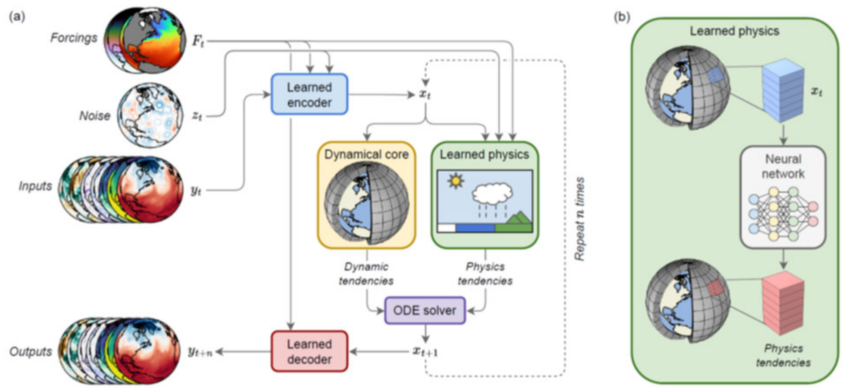


Fig. 1 Structure of the NeuralGCM model. (a) Overall model structure, showing how forcings F_t, noise z_t (for stochastic models), and inputs y_t are encoded into the model state x_t. Model state is fed into the dynamical core, and alongside forcings and noise into the learned physics module. This produces tendencies (rates of change) used by an implicit-explicit ODE solver to advance the state in time. The new model state x_{t+1} can then be fed back into another time step, or decoded into model predictions. (b) Inset of the learned physics module, which feeds data for individual columns of the atmosphere into a neural network used to produce physics tendencies in that vertical column.

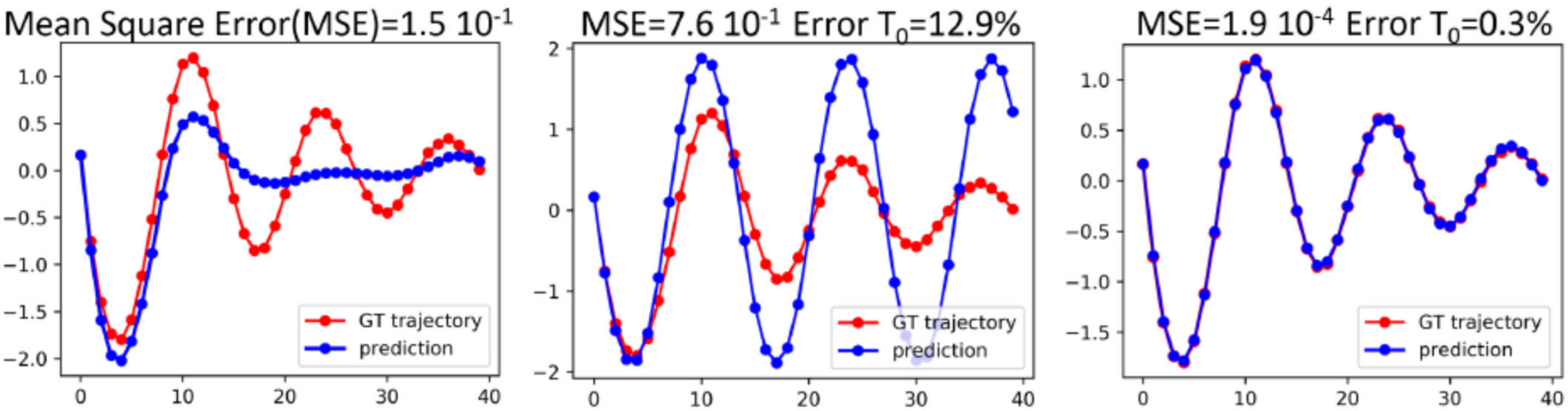
C1 - Incorporating physical knowledge

APHYNITY: Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting (Yin et al. 2021, Donà et al. 2022)

- Context
  - Assumptions: incomplete background knowledge is available
    - e.g. PDE that only explains partially the phenomenon
  - Complement the physical model with a statistical component
    - Learn the missing information from data
  - Provide a principled framework to make model-based and data-based frameworks cooperate
  - Objective
    - Identify the physical parameters (inverse problem)
    - The NN component should learn to describe the information that cannot be captured by the physics (direct problem)

APHYNITY: Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting (Yin et al. 2021, Donà et al. 2022)

- Illustration: damped pendulum



(a) Data-driven Neural ODE (b) Simple physical model (c) Our APHYNITY framework

Figure 1: Predicted dynamics for the damped pendulum vs. ground truth (GT) trajectories $\mathrm{d}^2\theta/\mathrm{d}t^2 + \omega_0^2 \sin\theta + \alpha \mathrm{d}\theta/\mathrm{d}t = 0$. We show that in (a) the data-driven approach (Chen et al., 2018) fails to properly learn the dynamics due to the lack of training data, while in (b) an ideal pendulum cannot take friction into account. The proposed APHYNITY shown in (c) augments the over-simplified physical model in (b) with a data-driven component. APHYNITY improves both forecasting (MSE) and parameter identification (Error T_0) compared to (b).

C1 - Incorporating physical knowledge
APHYNITY: Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting (Yin et al. 2021, Donà et al. 2022)

- We consider
  - General dynamics of the form $\frac{dX_t}{dt} = F(X_t)$
- Assumption
  - Evolution function F, will be modeled by a combination of:
    - A physical – incomplete model $F_p \in \mathcal{F}_p$
    - An agnostic model (a neural network) $F_a \in \mathcal{F}_a$
- Example:
  - Additive decompositions
  - $\boxed{\frac{dX_t}{dt} = F(X_t) = F_p(X_t) + F_a(X_t) \text{, with } F_p \in \mathcal{F}_p \text{ and } F_a \in \mathcal{F}_a}$
- Ill-posed problem
  - The decomposition $F_p(X_t) + F_a(X_t)$ is usually not unique
    - Turned into a well posed optimization problem

C1 - Incorporating physical knowledge
APHYNITY: Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting (Yin et al. 2021, Donà et al. 2022)

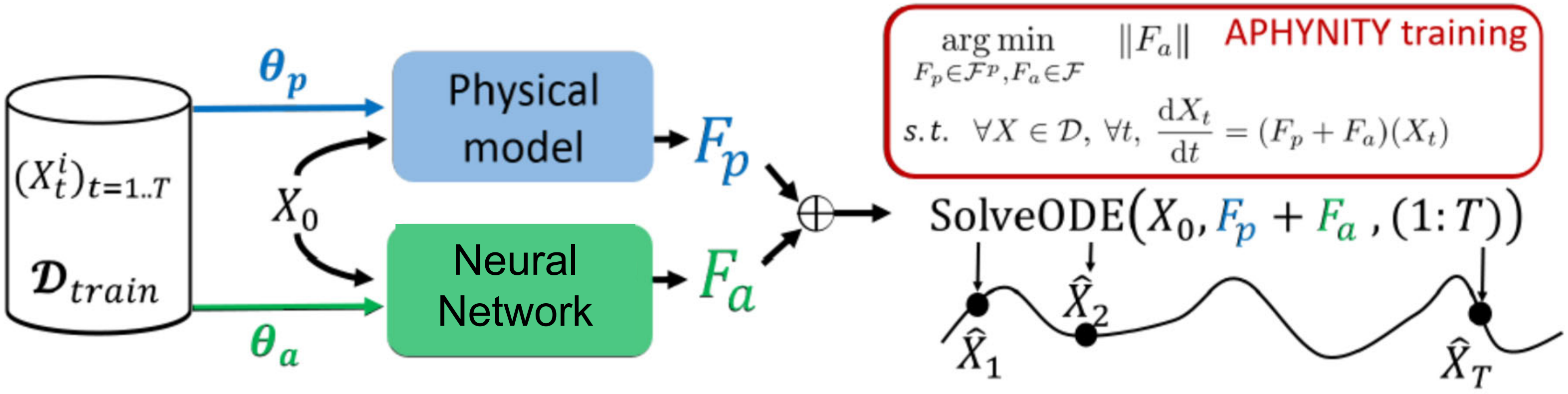
- Intuition
  - F_p should explain as much of the dynamics as possible
    - Learn F_a and F_p so that F_a explains only the residual unexplained by F_p

- Formalization: training objective
  - Given a normed vector space $(\mathcal{F}, \|.\|)$
    - $Min_{F_p \in \mathcal{F}_p, F_a \in \mathcal{F}_a} \|F_a\|$, s.t. $\forall X \in D, \frac{dX_t}{dt} = F_p(X_t) + F_a(X_t)$

- Theoretical insights
  - If $\mathcal{F}_p$ is a proximinal set, there exists a minimizing decomposition.
  - If $\mathcal{F}_p$ is a Chebyshev set, the optimization problem admits a unique minimizer, hence identifiability is guaranteed.

APHYNITY: Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting (Yin et al. 2021, Donà et al. 2022)

- The evolution function solution, combines a differentiable numerical solver with a NN residual component
- The parameters of the solver and of the neural network are learned from data
- Solving amounts at integrating this evolution function in time



Differentiable solver

Cardiac electrophysiology (Kashtanova, Sermesant et al. 2022 - Epione team Inria Sophia)

- Objective
  - Modeling the dynamics of cardiac electrical activity
    - Normal and pathological conditions
  - Variable of interest: Action Potential (mVolts) wave propagation

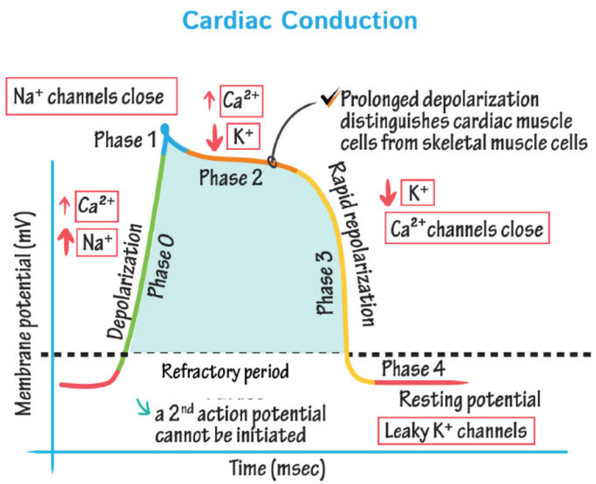


Fig. drawittoknowit.com

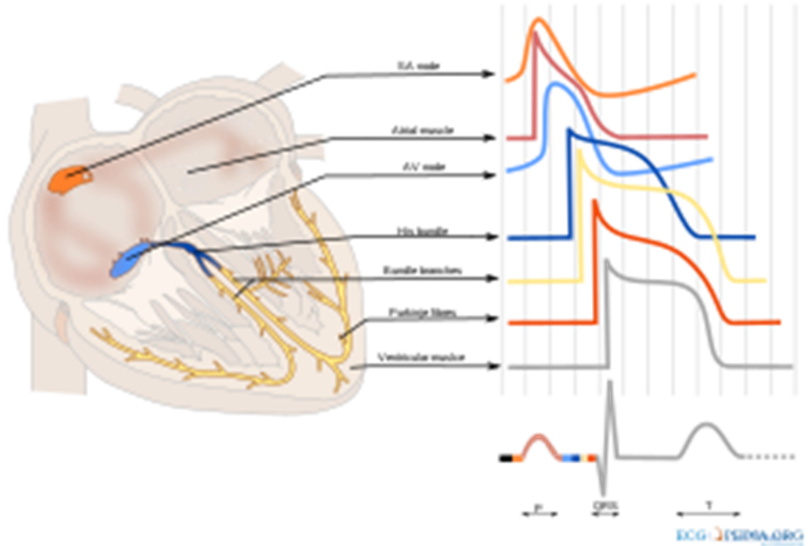


Fig. Wikipedia

C1 - Incorporating physical knowledge
Cardiac electrophysiology (Kashtanova et al. 2022)

- Objective
  - Modeling the dynamics of cardiac electrical activity
- Setting 1: In Silico Data
  - Complex high fidelity model – considered as Ground Truth
    - e.g. Ten Tusscher-Panvilov 2004
    - # hidden variables and parameters, computationally expensive (43 variables)
  - Surrogate low fidelity model – Incomplete phenomenological model
    - e.g. Mitchell Schaeffer 2003 – 6 parameters model
      - In the experiments, 3 parameters to be estimated + 3 fixed
      - Rapid prototyping, less precise - Reaction-diffusion model
  - Objective
    - Learn to simulate high fidelity data using a combination of low fidelity model (Mitchell Schaeffer) and residual neural network – APHYNITY framework

C1 - Incorporating physical knowledge
Cardiac electrophysiology (Kashtanova et al. 2022)

- In silico data - Example: polarization phase
  - Slab of 2D cardiac tissue of 24x24 elements

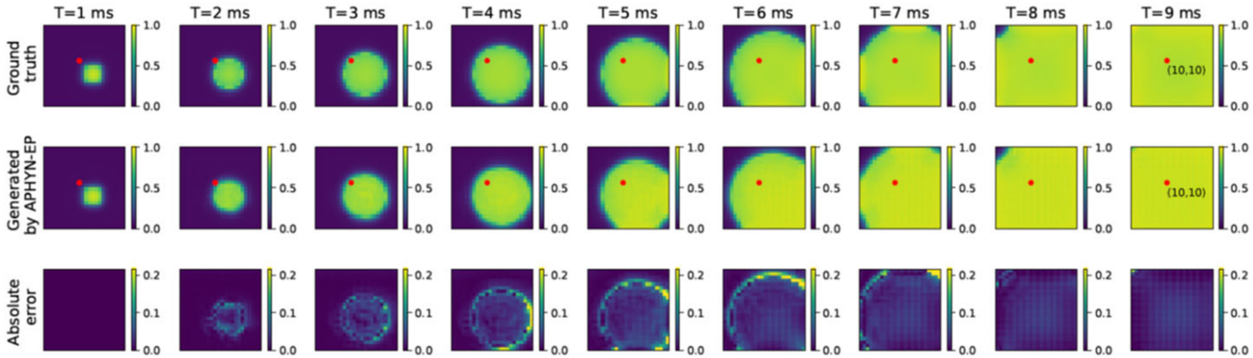


Figure 2: APHYN-EP predicted dynamics for the transmembrane potential diffusion. The figure shows a 9 ms of forecast).

- Ex-vivo data
  - Optical data from swine hearts

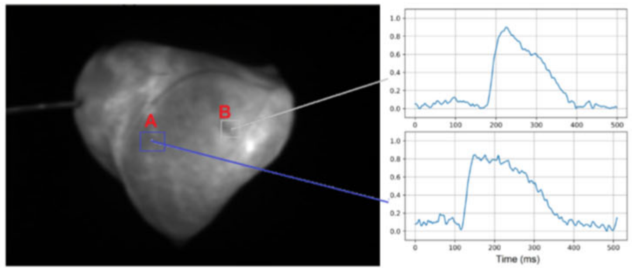


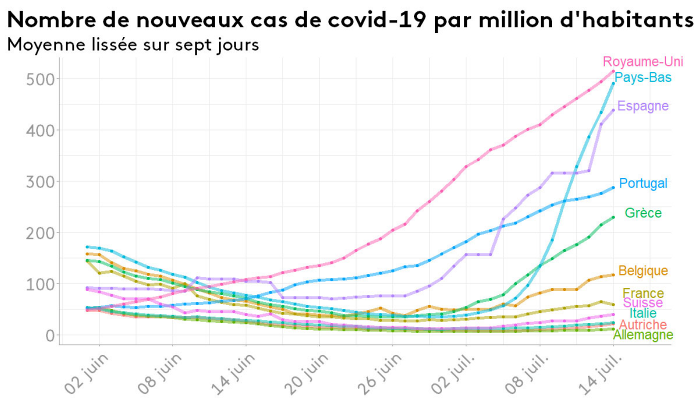
Fig. 2: Example of optical mapping data (tracings of denoised action potential waves) recorded ex vivo in a porcine heart. ROI B represents an ischaemic region characterized by a shorten action potential duration (APD) compared to the normal APD recorded in ROI A.

C2: Tackling the generalization problem

C2: Tackling the generalization problem for dynamical systems
Motivating examples

One underlying process – Multiple environments

Modelling epidemics in different countries



Modeling heart electrical diffusion from different patients

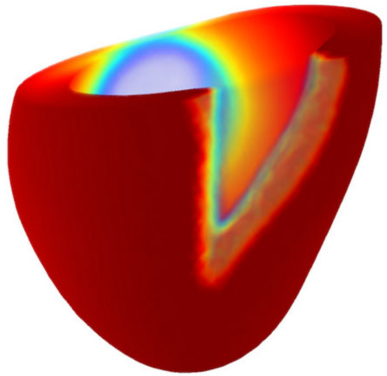
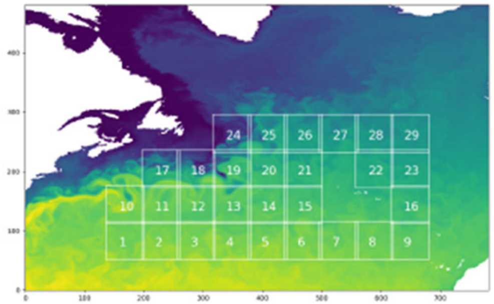


Fig. Fresca 2020



Sub regions extracted for the dataset. Test regions are regions 17 to 20.

Predictions of sea surface temperature from satellite data – different areas

C2: Tackling the generalization problem for dynamical systems Domain Generalization

- Problem setting
  - Assumption: there exists a set of environments $E = \{e^i\}$, each governed by a differential equation $\frac{dx_t^e}{dt} = f_e(x_t^e)$
    - Sharing commonalities e.g. general form of the dynamics (shared parameters θ_c)
    - With specificities, e.g. coefficients of the PDE, initial & boundary conditions, forcings, spatio-temporal domains, etc (Specific parameters θ_e)
- Challenge
  - How to leverage this setting in order to generalize to unseen situations and new environments?

C2: Tackling the generalization problem for dynamical systems
Domain Generalization

- Usual practice in ML (Empirical Risk Minimization)
  - Training dataset: sample environment distribution and for each environmen sample the trajectory distribution
  - Expect this will generalize to new environments
  - This assumes:
    - i.i.d. distribution, dataset large enough to cover the data distribution and represent the diversity of situations
  - Not realistic
- Claim
  - The models should leverage adaptive conditioning to the environment

Domain Generalization (Kassai et al. 2024)

ERM baselines vs environment adaptive conditioning

Gray-Scott

Burgers

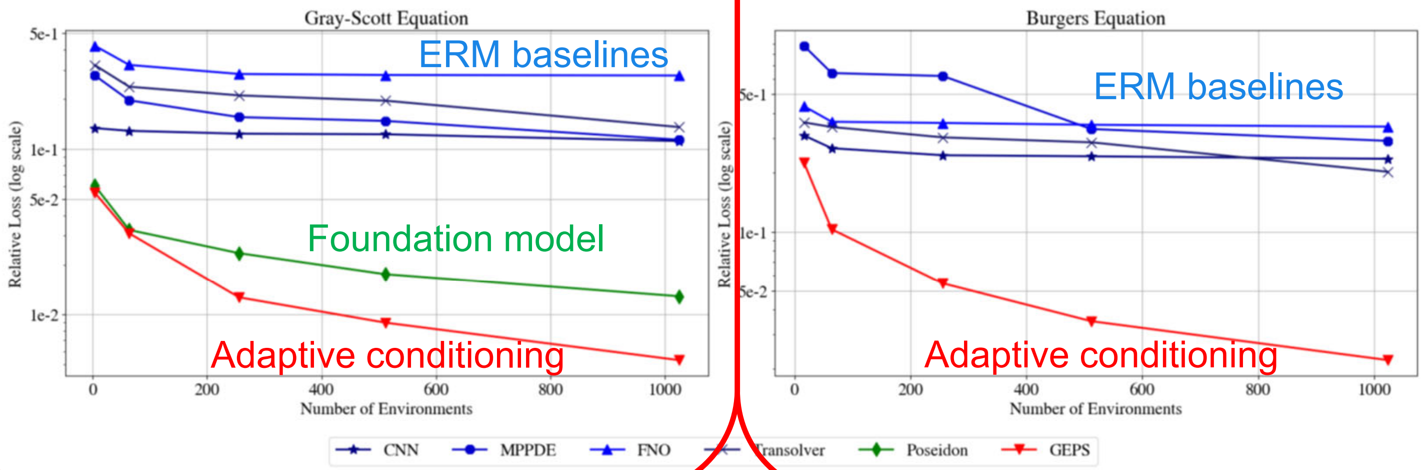
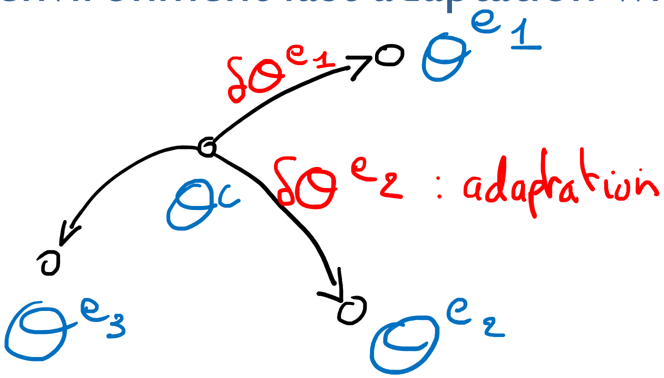


Figure 2: Comparison of ERM approaches (shades of blue) and Poseidon foundation model (green) with our framework GEPS (red) when increasing the number of training environments.

C2: Tackling the generalization problem for dynamical systems
CODA framework (Kirchmeyer et al. 2022, Kassai et al. 2024

- How to – Intuition: Meta-learning for fast adaptation to new environments
  - Training
    - Learn on a sample of the domains' distribution (i.e. different environments)
      - $\theta^e = \theta^c + \delta\theta^e$
      - θ^c shared parameters across environments, $\delta\theta^e$ environment specific parameters
    - So that it could adapt fast and with a few shots to a new environment
  - Inference: for a new environment fast adaptation with a few samples



C2:Tackling the generalization problem for dynamical systems
CODA framework (Kirchmeyer et al. 2022)

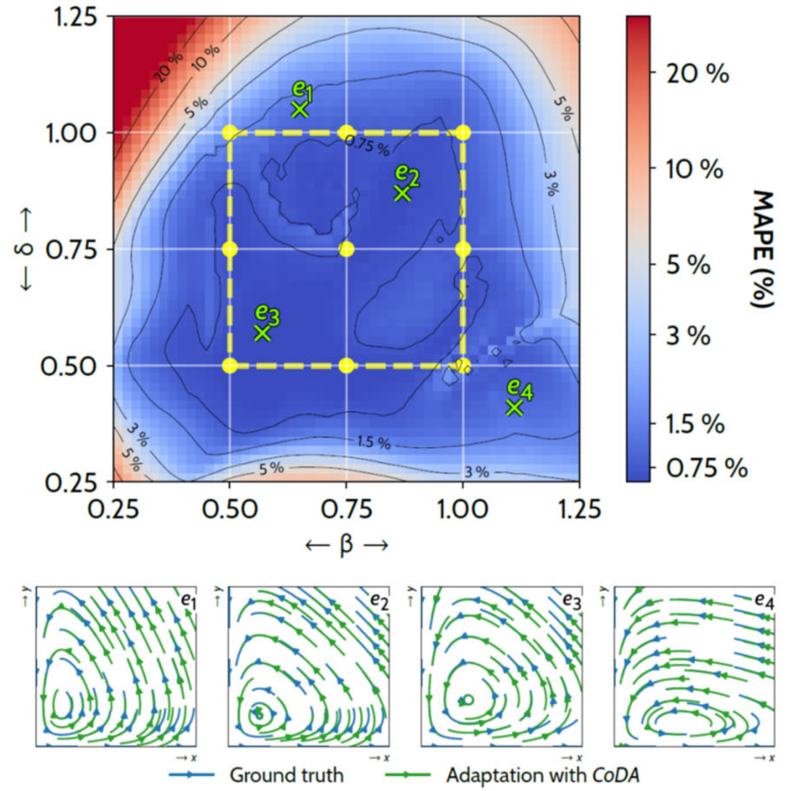


Figure 2. Adaptation results with CoDA-ℓ_1 on LV. Parameters (β, δ) are sampled in $[0.25, 1.25]^2$ on a 51×51 uniform grid, leading to 2601 adaptation environments $\mathcal{E}_{\text{ad}}$. ● are training environments $\mathcal{E}_{\text{tr}}$. We report MAPE (↓) across $\mathcal{E}_{\text{ad}}$ (Top). On the bottom, we choose four of them (×, e_1–e_4), to show the ground-truth (blue) and predicted (green) phase space portraits. x, y are respectively the quantity of prey and predator in the system in Eq. (15).

Lotka-Volterra (LV, Lotka, 1925) The system describes the interaction between a prey-predator pair in an ecosystem, formalized into the following ODE:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \alpha x - \beta xy$$
$$\frac{\mathrm{d}y}{\mathrm{d}t} = \delta xy - \gamma y \qquad (15)$$

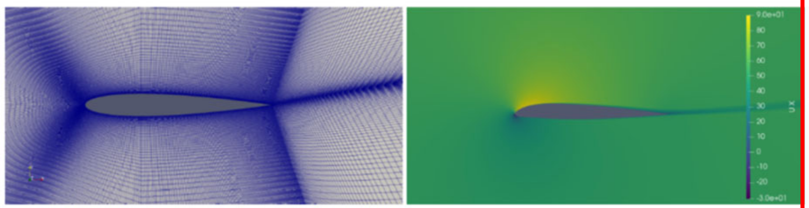
where x, y are respectively the quantity of the prey and the predator, $\alpha, \beta, \delta, \gamma$ define how two species interact.

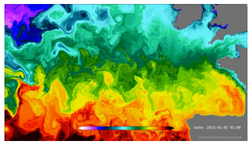
- Four parameters, two fixed (α, γ) and two (β, δ) change accross environments
- Training on 9 environments (yellow)
- Top: Evaluation on 2600 new environments
- Bottom: phase portraits for 4 new environments $\boldsymbol{e_1}$ **to** $\boldsymbol{e_4}$
  - **Blue trajectories: ground truth**
  - **Green trajectories: predicted**

C3: Neural operators: beyond mesh based approaches for simulation

C3: Neural operators

Classical numerical solvers operate on grids or meshes (finite differences, finite elements, finitie volumes)





Neural solvers operate on tensors (grids) or on graphs (irregular meshes)

Neural operators is a recent topic aiming at learning maps between function spaces instead of vector spaces

- e.g. images are considered as continuous functions

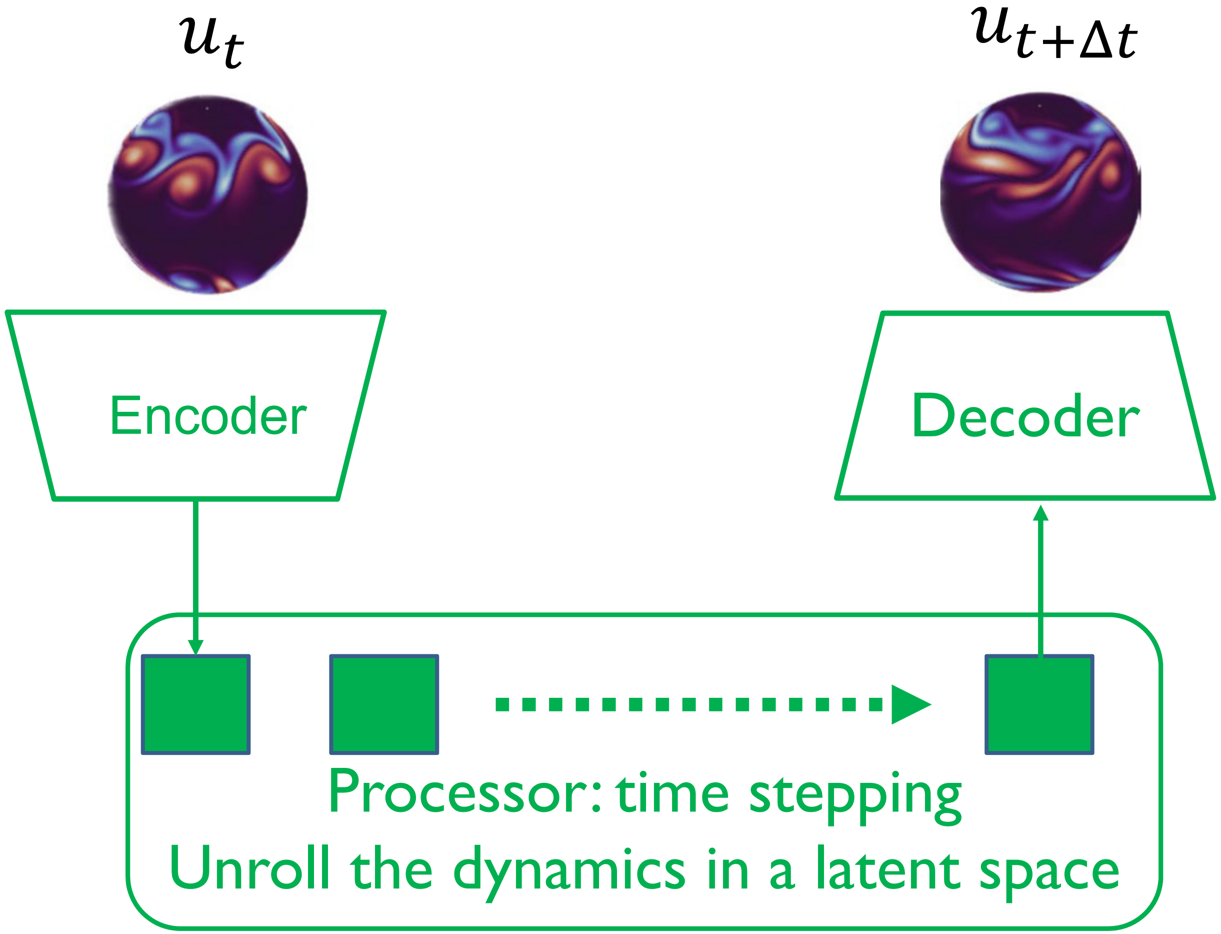
Key ideas

- Functions and operators are mesh/ resolution invariant
- They can be applied for different geometries, for multiple resolutions

Learning operator methods are data driven

C3: Neural operator
Encode – Process – Decode framework

Encode-Process-Decode has become the standard framework for many spatio-temporal forecasting problems

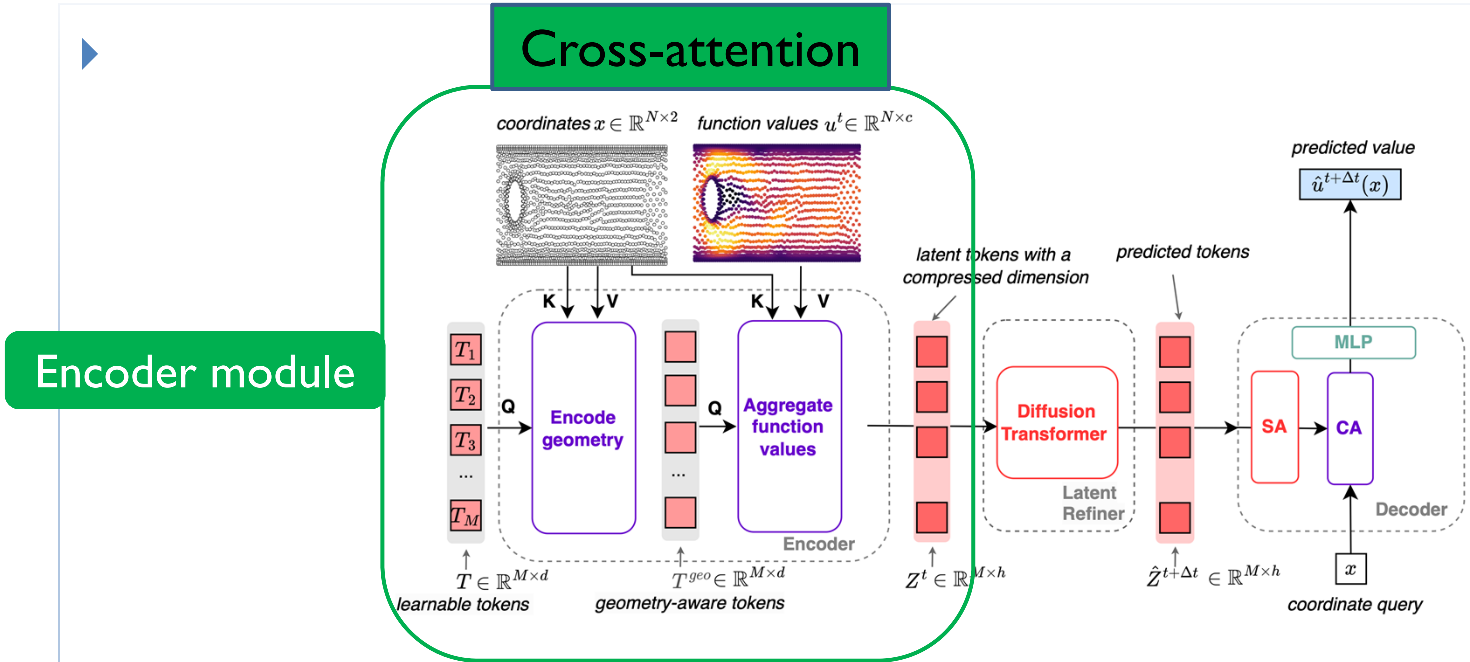


C3: Neural operators
AROMA: Attentive Reduced Order Model with Attention (Serrano et al. 2024)

- Principled Framework:
  - Properties
    - Handle diverse geometries: inputs and outputs may consist in point sets, grids, meshes
    - Can be queried at any spatial position
  - Demonstrates how modern NN components allow building versatile PDE solvers
    - Encode/ Process/ Decode framework
      - ☐ Encoding: cross-attention maps variable-size inputs to a fixed-size compact latent token space encoding local spatial information
      - ☐ Processing: a diffusion transformer architecture to model dynamics and exploit spatial relations locally and globally via self-attention + model uncertainty
      - ☐ Decoding: uses a conditional neural field + cross-attention to query forecast values at any spatial point within the equation's domain

AROMA: Attentive Reduced Order Model with Attention (Serrano et al. 2024)

General framework



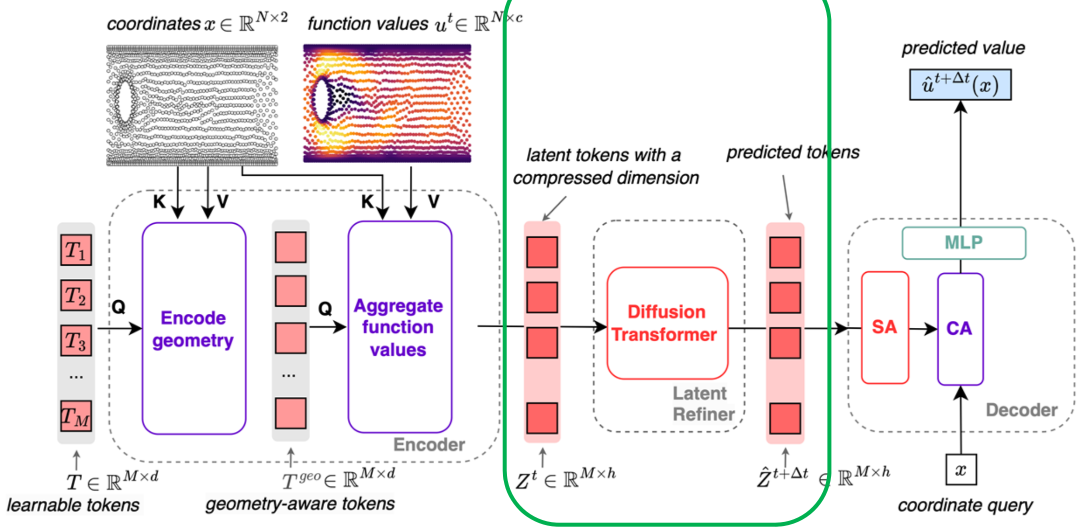
Cross-attention encoder: $u^t \to Z^t$

- Encodes variable size discretized input $u()$ into a fixed size & small dimensional sequence of latent embedding tokens Z
- Z encodes local spatial information on problem geometry + variable values

C3: Neural operators
AROMA: Attentive Reduced Order Model with Attention (Serrano et al. 2024)
General framework

Self-attention

Processor



Time stepping transformer: $Z^t \to Z^{t+\Delta t}$

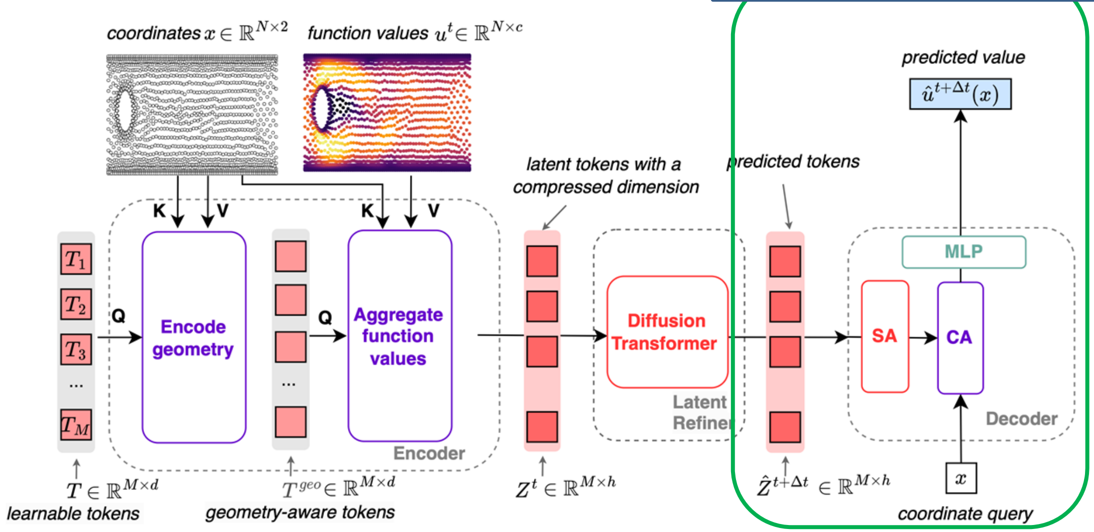
- Learns the dynamics in the small dimensional latent space
- Self attention models relations between spatial latent tokens
- Inference: dynamics is enrolled in the latent space starting from an initial condition– low complexity
- Diffusion: introduces a stochastic component

AROMA: Attentive Reduced Order Model with Attention (Serrano et al. 2024)

General framework

Cross-attention

Decoder module



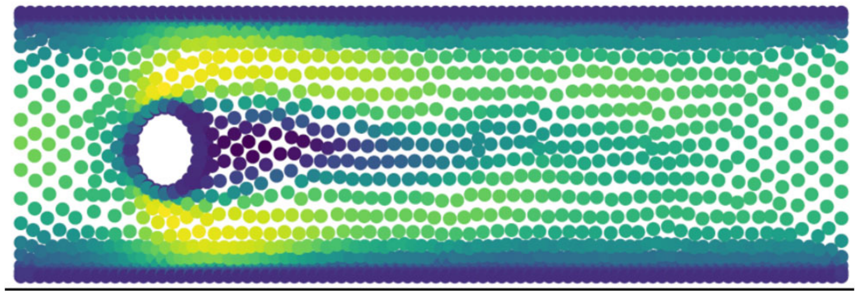
Cross-attention neural fields decoder: $Z^{t+\Delta t} \rightarrow u^{t+\Delta t}$

- Maps the latent representation $Z^{t+\Delta t}$ to the original physical space
- Can be queried at any position x of the physical space

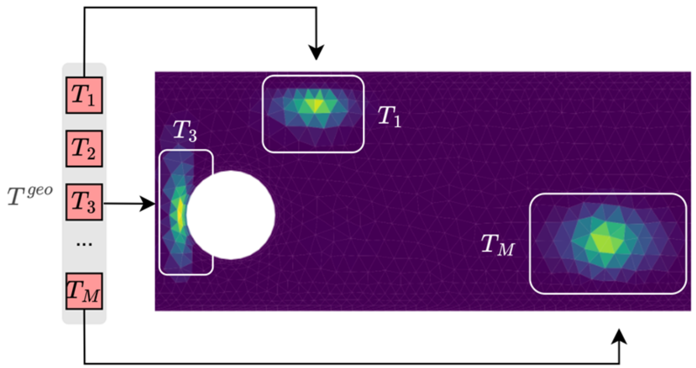
C3: Neural operators
AROMA: Attentive Reduced Order Model with Attention
Cross-attention encoder captures spatial attention

Example: Navier Stokes – cylinder flow
Cross attention illustration

Cylinder flow ground truth

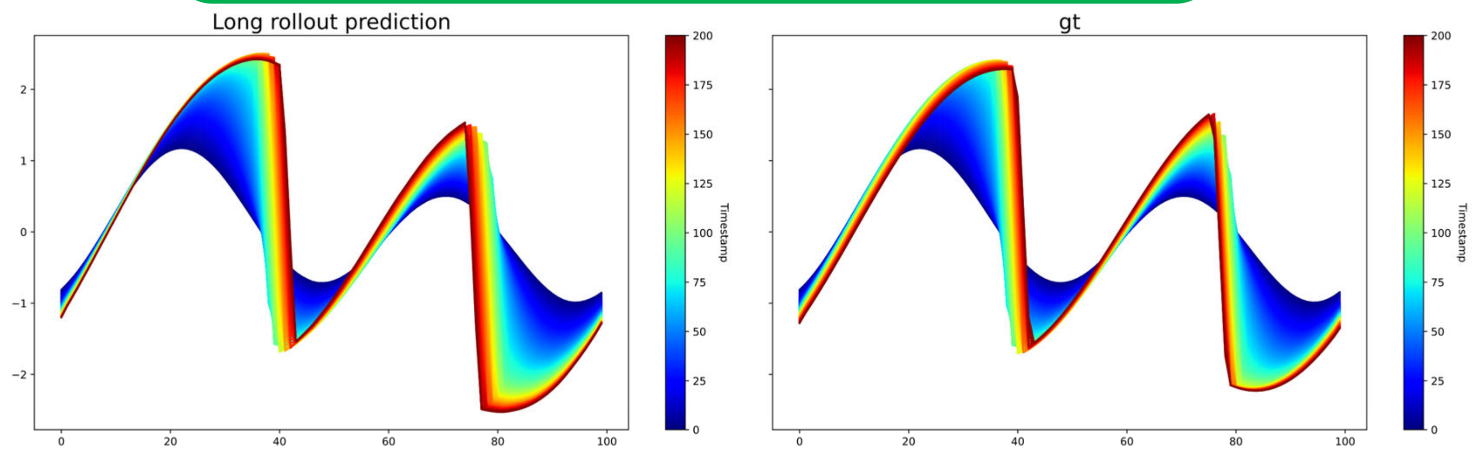


Tokens capture and encode local spatial information – cross attention between T^{geo} tokens and "x"



AROMA: Attentive Reduced Order Model with Attention
Stability on long rollouts

Burgers equation
Trained to predict next step on 50 time steps trajectories
Unrolled for 200 steps



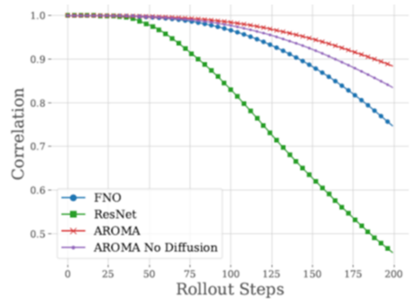


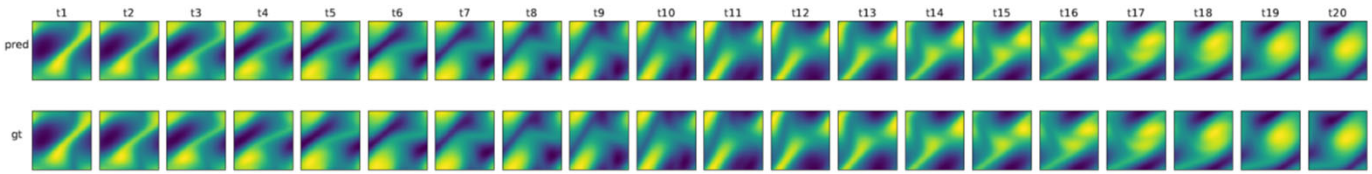
Figure 3: Correlation over time for long roll-outs with different methods on *Burgers*

AROMA: Attentive Reduced Order Model with Attention
Stability on long rollouts

Navier Stokes
Trained to predict next step on 20 time steps trajectories
Unrolled at test for 40 steps

Ground truth

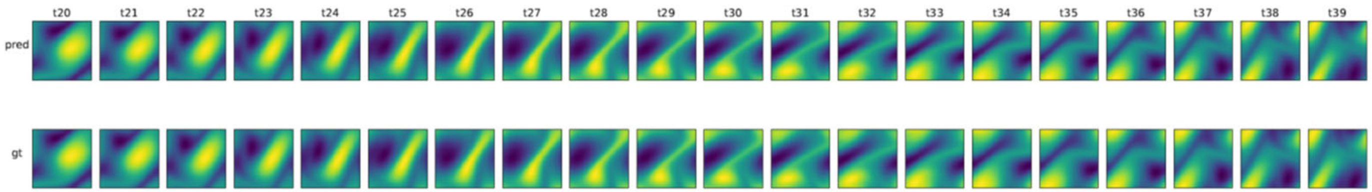
Prediction



(a) *In-t*

Ground truth

Prediction



(b) *Out-t*

Figure 12: Test example rollout trajectories with AROMA on *Navier-Stokes*. **Top:** predicted trajectory on *In-t*. **Bottom:** trajectory on *Out-t*. First row in each subfigure shows the prediction, the second row shows the ground truth.

Conclusion

- AI4Science
  - Still an open field, with several challenges
  - Already significant demonstrations in fields like weather forecasting, biology, materials, molecular design, …
  - Crucial role of curated data collections
  - Quest for foundation models

- Key issue
  - Crucial role of pluridisciplinary teams and efforts

References used in the presentation

General references (AI4Science)

- AI index report 2024, Stanford, https://aiindex.stanford.edu/report/
- Artificial Intelligence foundation models report, CSIRO, Australia, https://www.csiro.au/en/research/technology-space/ai/AI-foundation-models-report
- DOE report 2020, https://www.anl.gov/ai/reference/ai-for-science-report-2020
- DOE report 2023, https://www.anl.gov/ai/reference/AI-for-Science-Energy-and-Security-Report-2023
- EU reports on AI4Science 2024, https://scientificadvice.eu/advice/artificial-intelligence-in-science/

References used in the presentation

Scientific references

- Belbute-Peres, F. de A., Economon, T. D., & Kolter, J. Z. (2020). Combining Differentiable PDE Solvers and Graph Neural Networks for Fluid Flow Prediction. ICML, 2402–2411. http://arxiv.org/abs/2007.04439
- Bommasani, R., Hudson, D.A., and Adeli, E., et al., 2021. On the opportunities and risks of foundation models. *arXiv*. https://doi.org/10.48550/arXiv.2108.07258
- Bonnet, F., Mazari, J. A., Cinnella, P., & Gallinari, P. (2022). AIRFRANS : High Fidelity Computational Fluid Dynamics Dataset for Approximating Reynolds-Averaged Navier – Stokes Solutions. *Neurips 2022*.
- Dona, J., Dechelle, M., Gallinari, P., & Levy, M. (2022). Constrained Physical-Statistical Models for Dynamical System Identification and Prediction. *ICLR*.
- Fathony, R., Sahu, A. K., Willmott, D., & Kolter, J. Z. (2021). Multiplicative Filter Networks. *ICLR*, 1–10.
- Kashtanova, V., Ayed, I., Arrieula, A., Potse, M., Gallinari, P., & Sermesant, M. (2022). Deep Learning for Model Correction in Cardiac Electrophysiological Imaging. *MIDL*, 1–11.
- Kassai Koupai, A., Benet, J. M., Yin, Y., Vittaut, J.-N., & Gallinari, P. (2024). GEPS: Boosting Generalization in Parametric PDE Neural Solvers through Adaptive Conditioning. NeurIPS. https://geps-project.github.io/
- Kirchmeyer, M., Yin, Y., Donà, J., Baskiotis, N., Rakotomamonjy, A., & Gallinari, P. (2022). Generalizing to New Physical Systems via Context-Informed Dynamics Model. *ICML*. http://arxiv.org/abs/2202.01889
- Kochkov D, Yuval J, Langmore I, et al. Neural General Circulation Models. In: *ArXiv:2311.07222v2*. ; 2024.
- Lam R, Sanchez-Gonzalez A, Willson M, et al. Learning skillful medium-range global weather forecasting. *Science (80-)*. 2023;382(6677):1416-1422. doi:10.1126/science.adi2336
- Park, J.J, Florence, P, , Straub, J, Newcombe, R, and Lovegrove,, S, "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
- Serrano, L., Boudec, L. Le, Koupaï, A. K., Wang, T. X., Yin, Y., Vittaut, J.-N., & Gallinari, P. (2023). Operator Learning with Neural Fields: Tackling PDEs on General Geometries. *NeurIPS*
- Serrano, L., Wang, T., le Naour, E., Vittaut, J.-N., & Gallinari, P. (2024). AROMA : Preserving Spatial Structure for Latent PDE Modeling with Local Neural Fields. *NeurIPS*.
- Sitzmann, V., Martel, J. N. P., Bergman, A. W., Lindell, D. B., Wetzstein, G., & University, S. (2020). Implicit Neural Representations with Periodic Activation Functions. *Neurips*.
- Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., & Ng, R. (2020). Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. *Neurips*.
- Yagoubi, M., Danan, D., Leyli-abadi, M., Brunet, J., Gmati, M., Systemx, I. R. T., Schoenauer, M., & Gallinari, P. (2024). ML4CFD Competition : Harnessing Machine Learning for Computational Fluid Dynamics in Airfoil Design. *NeurIPS Competition Track*
- Yin, Y., Kirchmeyer, M., Franceschi, J.-Y., Rakotomamonjy, A., & Gallinari, P. (2023). Continuous PDE Dynamics Forecasting with Implicit Neural Representations. *ICLR*, 1–19. http://arxiv.org/abs/2209.14855
- Yin, Y., Le Guen, V., Dona, J., de Bezenac, E., Ayed, I., Thome, N., & Gallinari, P. (2021). Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting. *ICLR*.