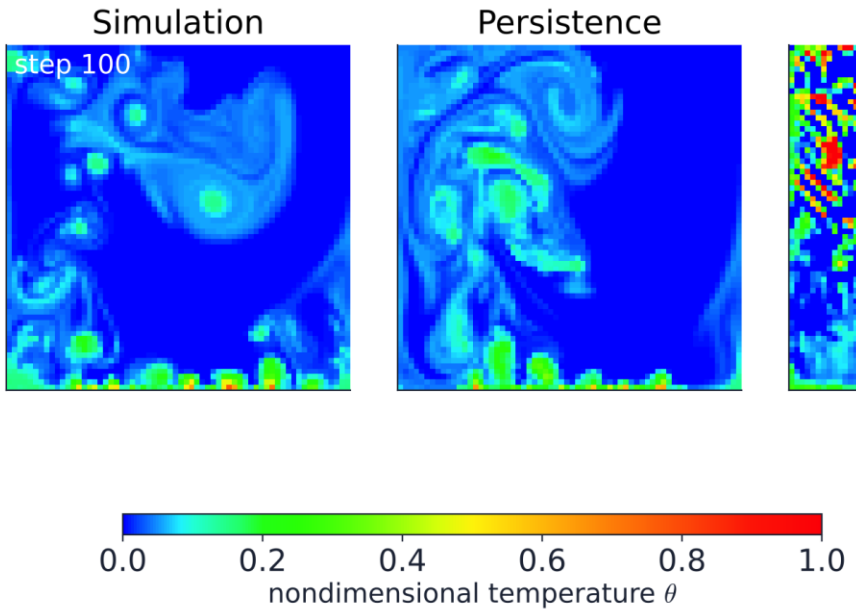


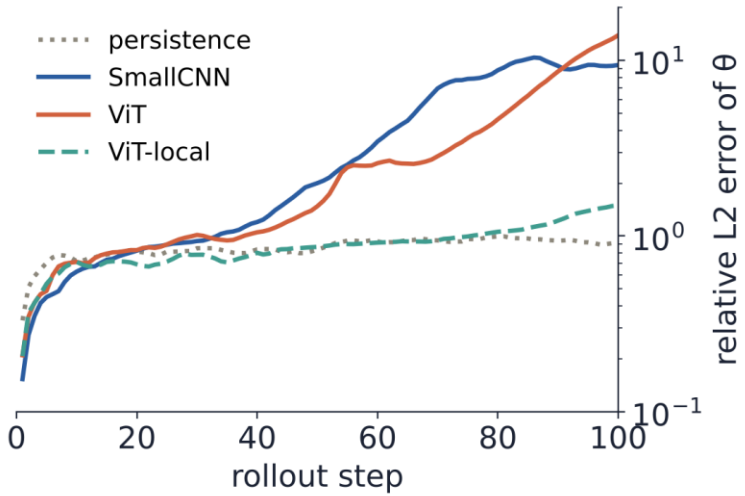
ML FOR SCIENCE · LECTURE 10

Evaluating SciML

Aparna Chandramowlishwaran · EECS 195/221 · Fall 2026

Which Lecture 9 model is better?

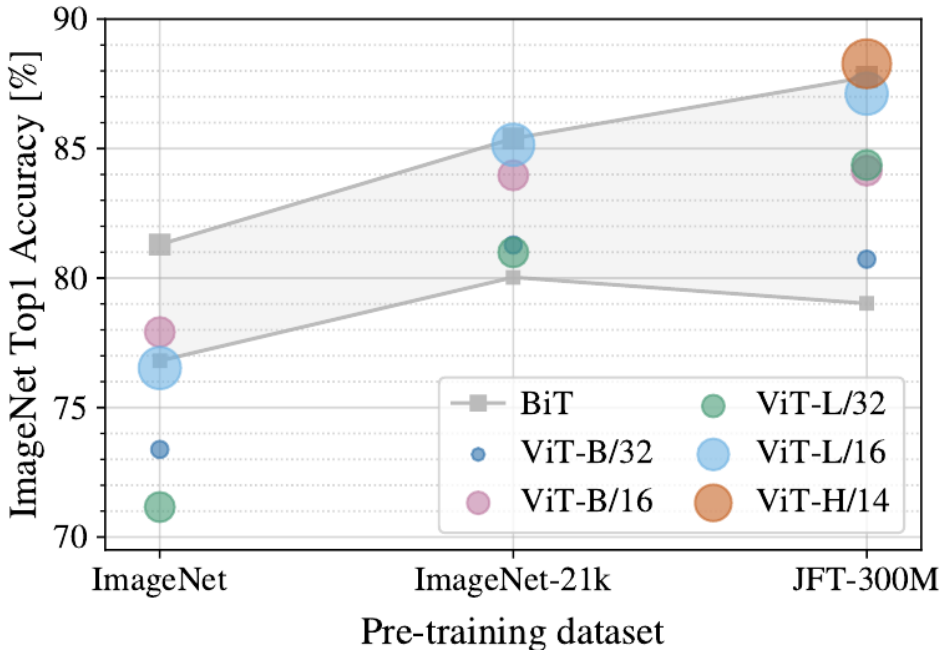


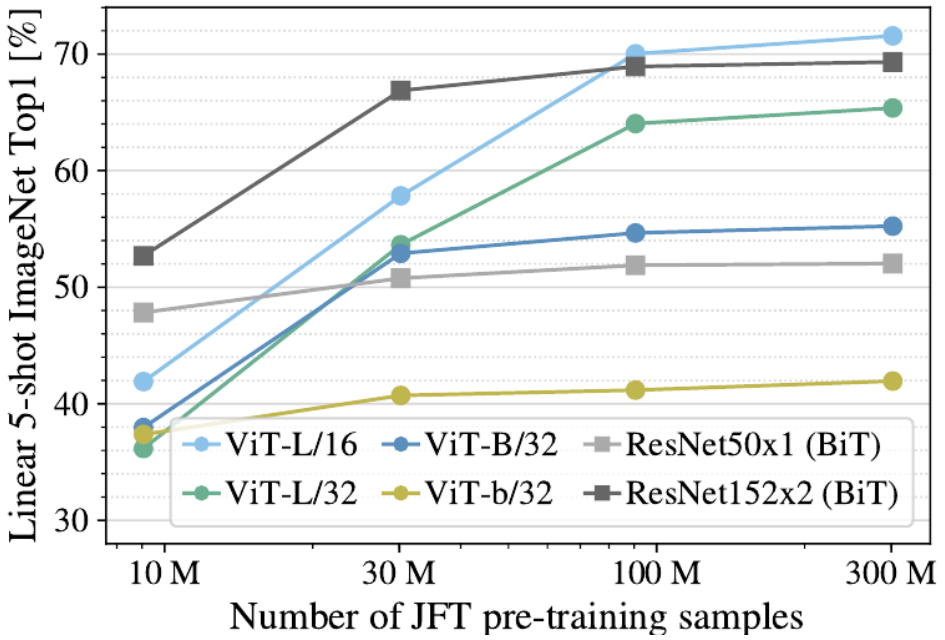


- **One step**: SmallCNN has the lowest error; all three models beat **persistence** (copy the last frame)
- **100-step rollout**: SmallCNN and ViT grow past persistence and leave the physical range; ViT-local stays much closer
- Correlation, mean temperature and spectra give other orderings. Each metric answers a different question

The ranking depends on what you measure

Transformers and CNNs at different data sizes





- A CNN assumes **locality** and **translation equivariance**. Global attention must learn them from data
- Left: pretrained on 1.3M images, ResNets (gray band) beat every ViT. On 300M images, large ViTs are better.
- Right: same trend on subsets of 10M to 300M images
- Scientific datasets are small: BubbleML 2.0 has about 160 simulations. Pretraining on many PDEs as in Poseidon is an active research direction

Built-in structure matters most when data are scarce

Figure: Dosovitskiy et al., ICLR 2021 (BiT: ResNets of Kolesnikov et al. 2020).
BubbleML 2.0: Hassan et al., NeurIPS 2025. Herde et al., Poseidon, NeurIPS 2024

Questions for evaluating a SciML model

- **Accuracy**: how close is the prediction to the reference solution, at one step and over a rollout?
- **Baseline**: is the model better than a simple method (persistence, a cheaper numerical solver)?
- **Physics**: does the prediction conserve quantities, the constraints and the statistics of the flow?
- **Generalization**: is it as accurate on new trajectories, parameters, resolutions and longer times?
- **Cost**: is it faster than a numerical solver at the same accuracy, including the cost of data generation and training?
- **Rigor**: is the result the same across seeds and test cases, with the test set used only once?

Choose the questions, baselines and metrics before training

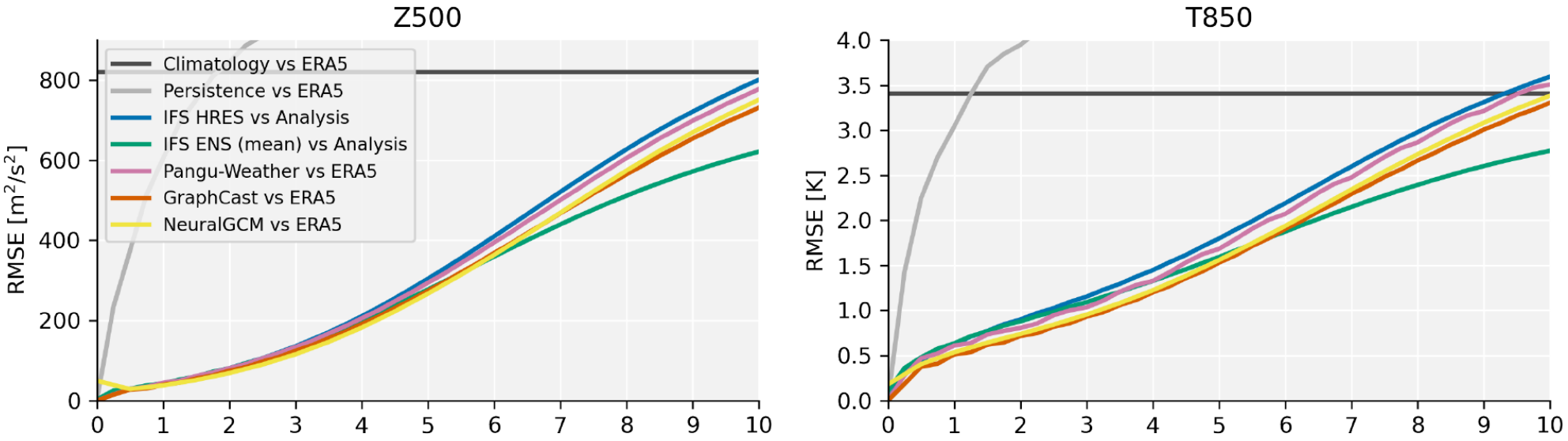
Pointwise error metrics

- Compare the prediction $\hat{u}$ with the reference u at every grid point, then average
  - MSE = mean $(\hat{u} - u)^2$, RMSE = $\sqrt{\text{MSE}}$: in the units of the field
- Relative L2 error $\|\hat{u} - u\|_2 / \|u\|_2$: no units, so fields and datasets can be compared
  - **Divides by the size of the field, including its mean.** For T = 300 K ± 1 K, predicting 300 K everywhere gives a relative error of 0.3%
- Variance-scaled RMSE: VRMSE = $\sqrt{(\text{ mean } (\hat{u} - u)^2 / \text{var } u)}$
  - Divides by the variation of the field around its mean. Predicting the mean scores 1 (The Well)
- Report each field separately (θ, u, v, p). An average over fields hides the field that fails
- Report the error at one step and as a function of rollout step

Normalize by the variation you want the model to predict

VRMSE: Ohana et al., The Well, NeurIPS 2024. PDEBench uses the relative L2 error (nRMSE): Takamoto et al., NeurIPS 2022

Persistence and climatology baselines



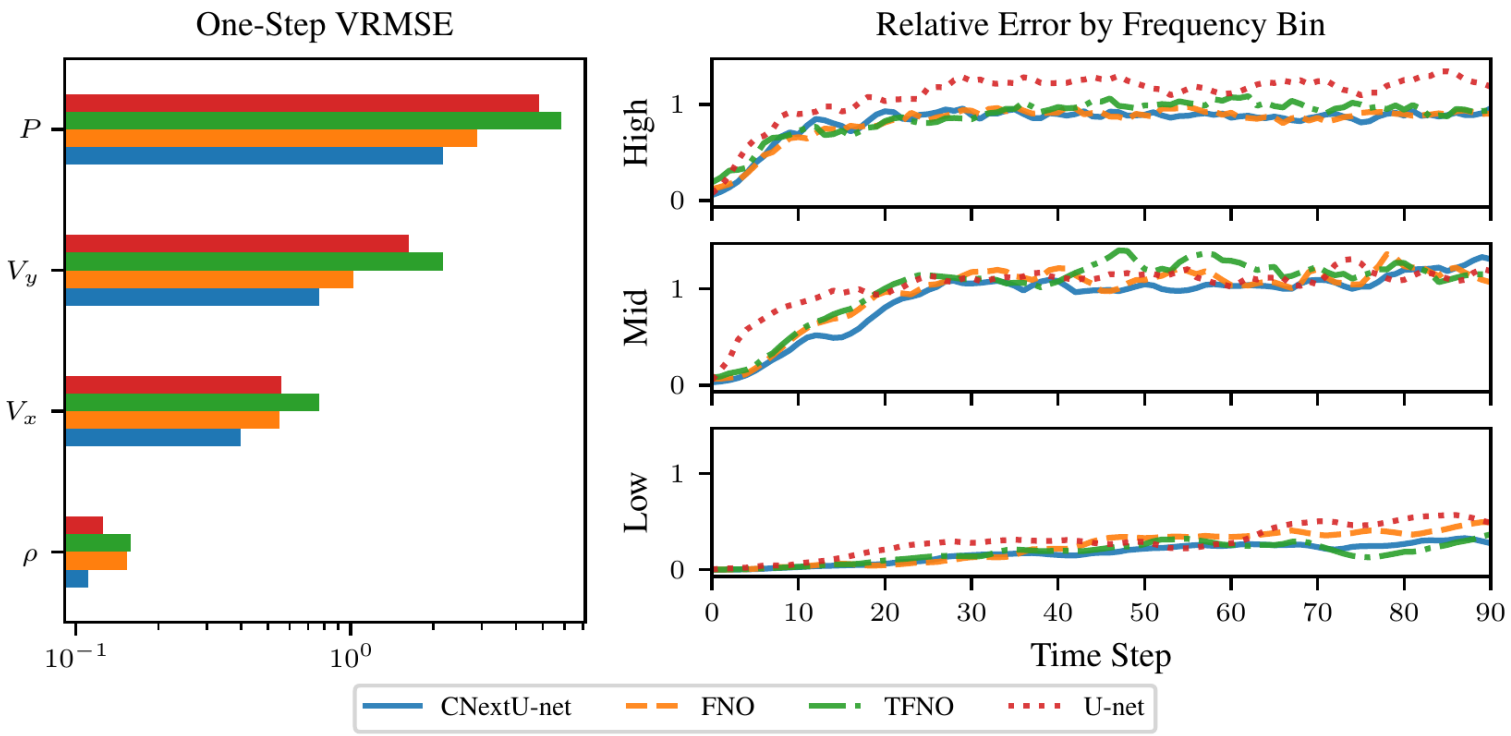
Global RMSE of geopotential at 500 hPa (Z500) and temperature at 850 hPa (T850) over 10 days.

- **Persistence**: predict no change, $\hat{u}_{n+1} = u_n$
- **Climatology**: predict the long-time average state for that place and date
- Persistence is accurate at short lead times; climatology is the reference at long ones
- A model is useful only while its error is below both

Always report the error of the simplest predictor as a baseline

Figure: Rasp et al., WeatherBench 2, J. Adv. Model. Earth Syst. 2024; RMSE against ERA5 or the IFS analysis, year 2020). Lead time in days

Error by scale

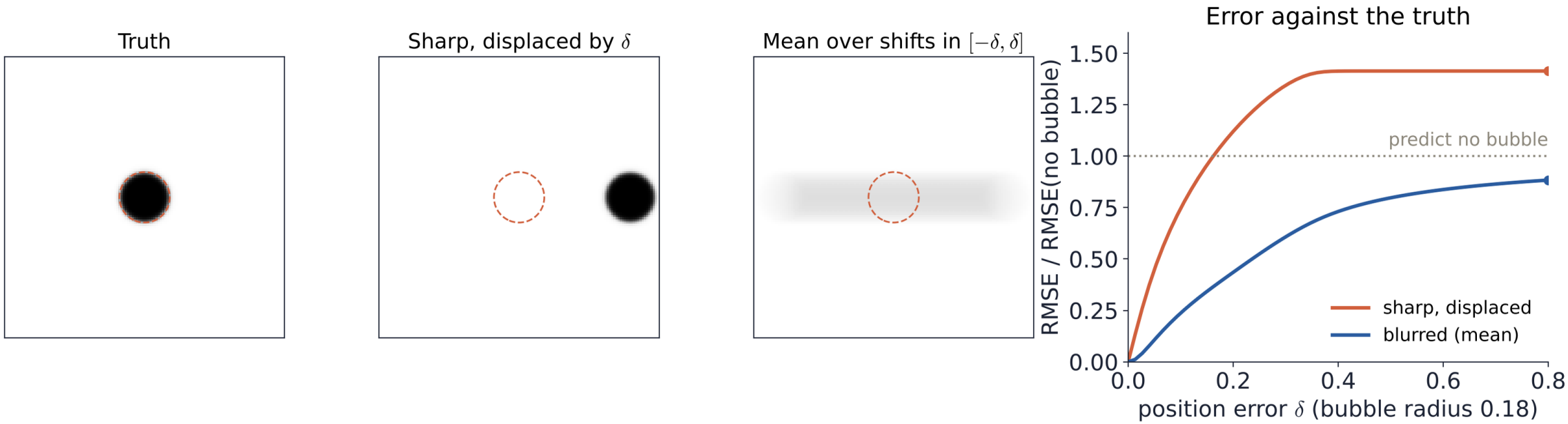


- Split the error by wavenumber k (Fourier modes). Low k: large scales. High k: small scales
- PDEBench: RMSE in low, mid and high frequency bands (fRMSE), at the boundary (bRMSE), and in the conserved quantity (cRMSE)
- The Well (left): error in each band divided by the energy in that band. A value near 1 means that band is lost
- The high and mid bands reach 1 within about 20 steps; the low band stays well below

Small scales are lost first

Figure: Ohana et al., The Well, NeurIPS 2024, Left: one-step VRMSE for pressure P, velocity V_x, V_y, density ρ. Right: rollout error per band
Metrics: Takamoto et al., PDEBench, NeurIPS 2022

Double penalty of pointwise error

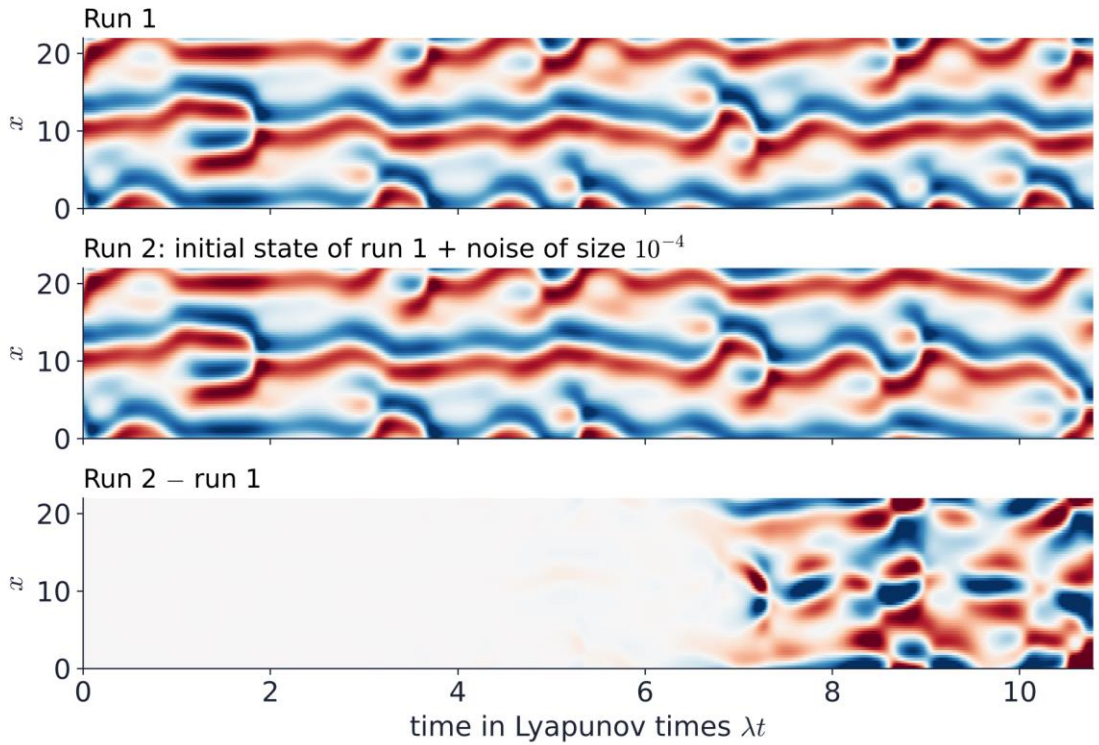


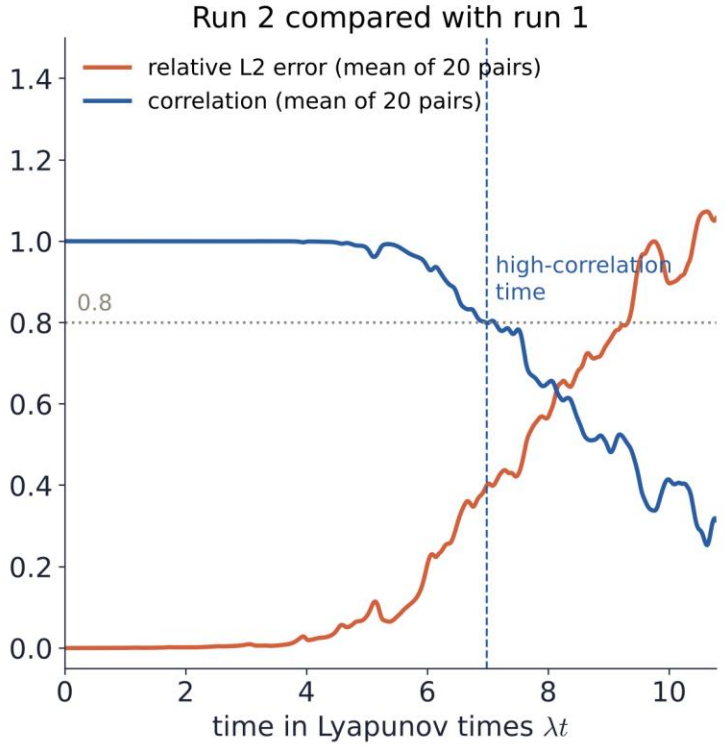
- A sharp bubble at the wrong place is counted twice: missing where it should be and extra where it is
- Once δ exceeds the bubble size, its RMSE is $\sqrt{2}$ times that of predicting **no bubble at all**. The blurred mean scores lower at every δ
- MSE training learns the mean over the outcomes the model cannot distinguish, so it predicts a **blurred** bubble when the position is uncertain

Pointwise metrics reward blur. Check the spectrum or the structures as well

Double penalty: Gilleland et al., Weather and Forecasting 2009

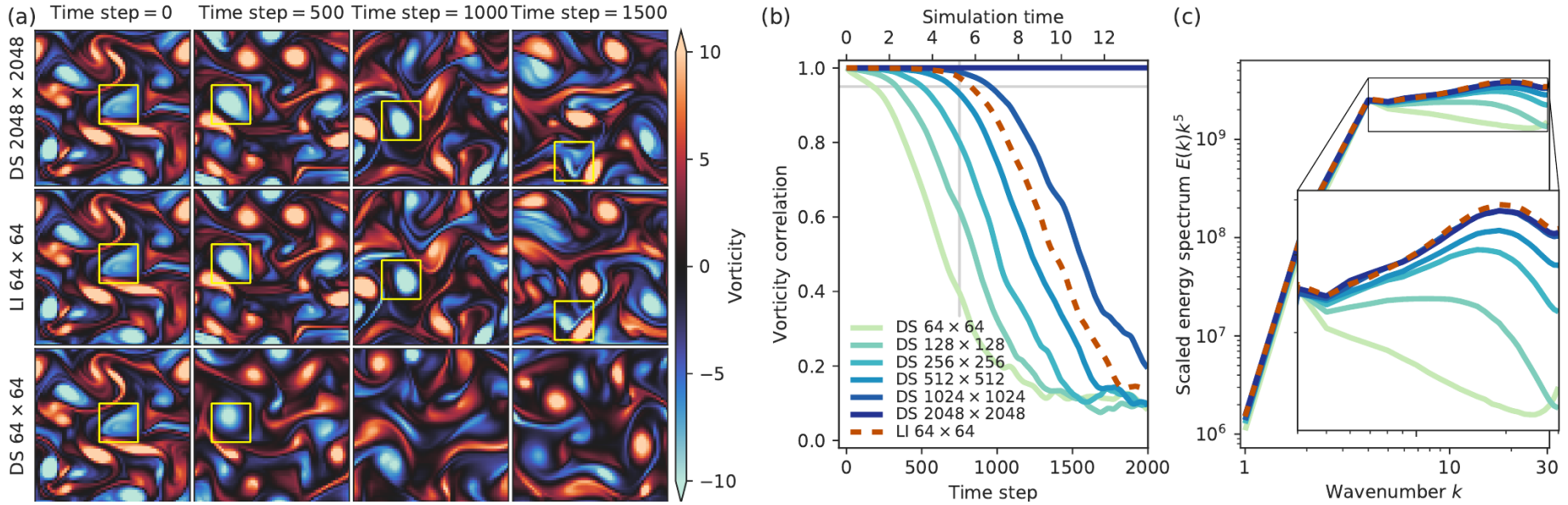
Pointwise error in chaotic systems





- Kuramoto-Sivashinsky (KS) equation $u_t + u\,u_x + u_{xx} + u_{xxxx} = 0$: a 1D model of chaotic flame fronts and thin films
- Two runs of the same solver, starting 10^{-4} apart. Their difference grow as $e^{\lambda t}$; in one Lyapunov time $1/\lambda$, it grows by a factor of e
- After about 7 Lyapunov times even the solver's second run fails a pointwise test: **pointwise error measures the chaos, not the model**

Correlation and high-correlation time

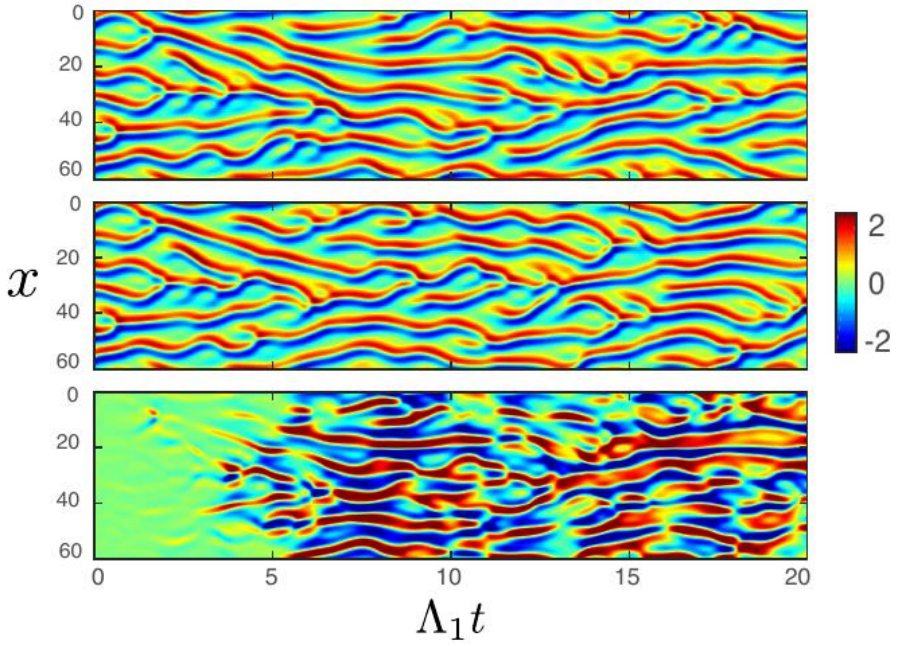


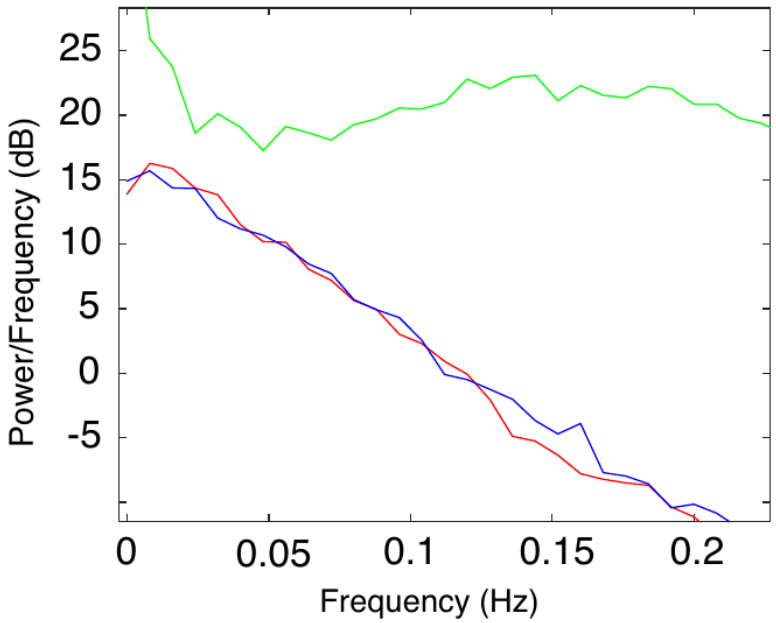
- **Correlation** ρ(û, u): subtract the mean of each field, then take the normalized inner product. 1 for a perfect match, near 0 for an unrelated field
- **High-correlation time**: the time at which the correlation, averaged over test trajectories, drops below a threshold (0.8, 0.9 or 0.95)
- (b): each coarser solver decorrelates earlier. Learned interpolation on 64 x 64 (dashed) stays correlated about as long as a solver on a grid 8-16 times finer

For chaotic systems, report how long the prediction stays correlated

Figure: Kochkov et al., Machine learning–accelerated computational fluid dynamics, PNAS 2021, 2D Kolmogorov flow, Re = 1000

Statistics of a chaotic flow



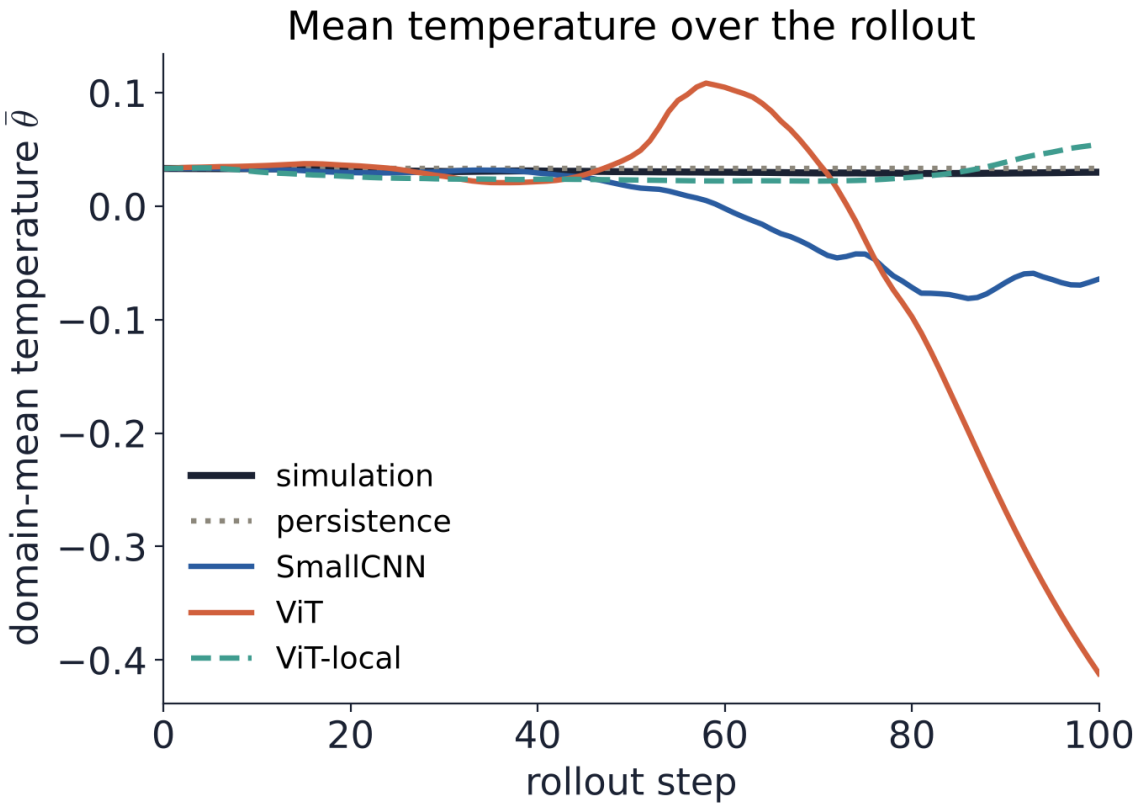


- Left: KS truth (top), a reservoir network's prediction (middle) and their difference (bottom). The prediction follows the truth for about 5 Lyapunov times
- Right: power spectrum of the truth (blue) and of the long prediction (red) agree. A badly tuned network (green) has the wrong spectrum
- After decorrelation, compare **statistics**: mean, variance, PDF of u, energy spectrum, autocorrelation in time

Short times: compare states. Long times: compare statistics

Figures: Pathak, Hunt, Girvan, Lu & Ott, Model-free prediction of large spatiotemporally chaotic systems from data, PRL 2018

Quantities of interest



- A **quantity of interest** is what the scientist or engineer uses the simulation for
  - Boiling: heat flux at the heater, bubble departure frequency, vapor fraction
  - Weather: storm track and intensity, precipitation
  - Turbulence: drag, energy spectrum
- Left: domain-mean temperature, a measure of the thermal energy in the domain. The simulation keeps it nearly constant
- SmallCNN and ViT drift **below θ = 0**, colder than the coldest liquid in the system

Evaluate the quantity the simulation is meant to predict

Conservation and constraints

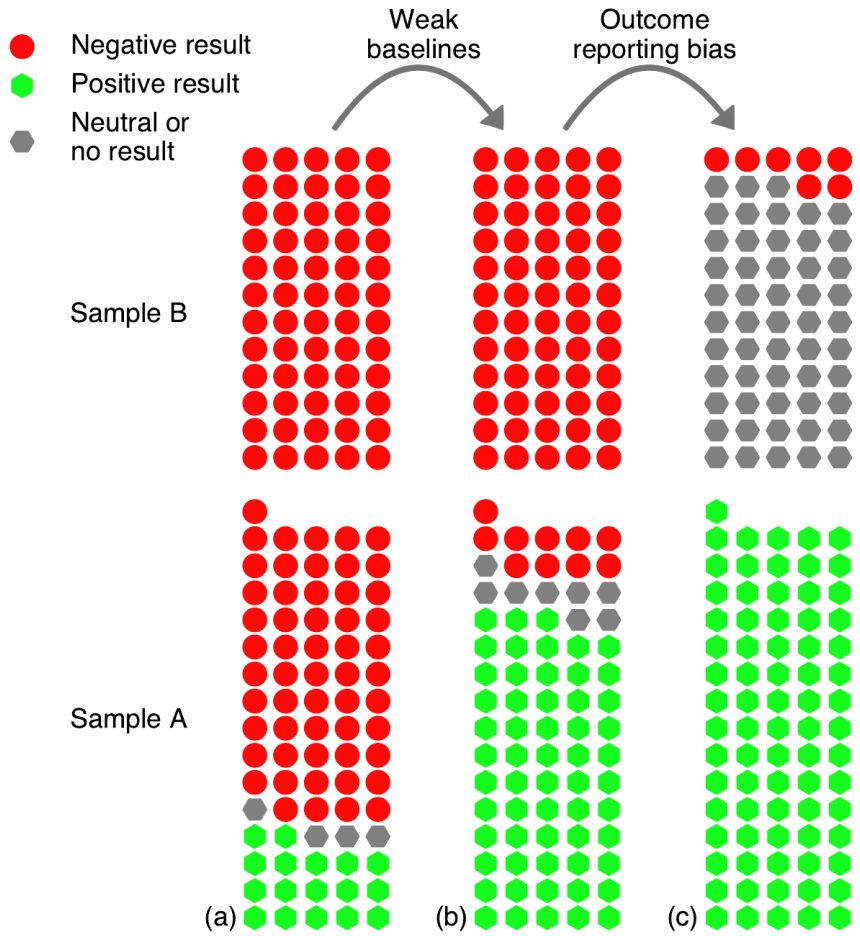
Property	What to compute	Example
Conserved quantity	Total over the domain, at every rollout step, against the reference	Heat equation with insulated walls (Lecture 1): $\int u\,dx$ is constant
Flux balance	Change of the total = flux in − flux out	Boiling: heat from the heater, to the bulk liquid and the outflow
Constraint	Size of the violation	Incompressible flow: $\nabla \cdot u = 0$
Bounds	Fraction of points outside the bounds	Density ≥ 0, volume fraction in $[0, 1]$
Boundary conditions	Error on the boundary only (bRMSE)	Wall temperature, no-slip velocity

- Small violations at each step add up over a rollout. Measure them at the end of long rollouts

A small error can still violate a conservation law

bRMSE, cRMSE: Takamoto et al., PDEBench, NeurIPS 2022. u: velocity in the divergence row, temperature in the heat-equation row

Weak baselines in the literature

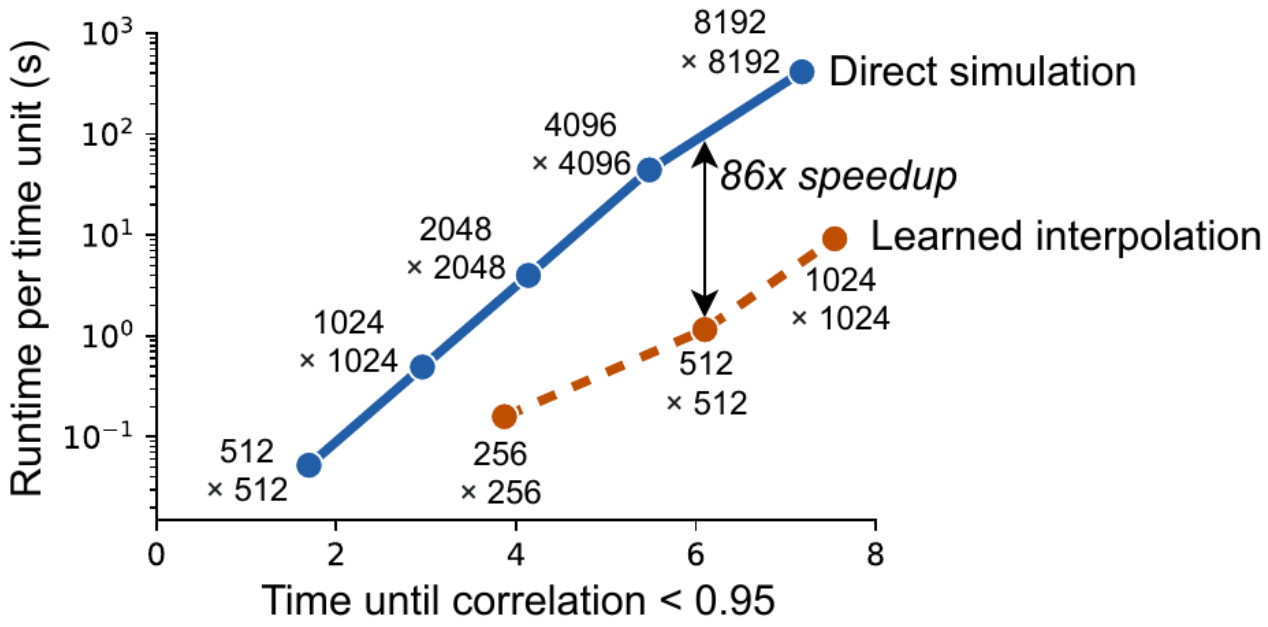


- Systematic review of ML solvers for fluid-related PDEs (McGreivy & Hakim 2024)
- Of 76 articles that report outperforming a standard numerical method, **79% (60/76)** compared against a weak baseline
  - Rule 1 broken: the ML solver was compared with the accurate solver that generated the training data, not with a cheaper solver of equal accuracy
  - Rule 2 broken: the numerical method was not an efficient one for that PDE
- **Reporting bias**: 80% of these articles report their speed against a solver in the abstract. Only 8% of a sample of PINN papers do, which are slower than solvers
- Figure: (a) results with strong baselines, (b) as published with weak baselines, (c) after reporting bias

The more impressive the reported speedup, the more likely the baseline was weak

Figure: McGreivy & Hakim, Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations, Nature Machine Intelligence 2024
Sample A: 76 articles. Sample B: 60 PINN articles

Comparing at equal accuracy or equal runtime



- Kochkov et al. follow rule 1; McGreivy & Hakim note that a faster numerical method exists for this flow (rule 2)

- A numerical solver trades accuracy for runtime: plot both methods as curves of runtime against accuracy
- Measure the speedup at **equal accuracy** (vertical arrow) or the accuracy at equal runtime
- Use an efficient solver and the same hardware class for both
- Count data generation and training if the model is used for only a few runs

A speedup is meaningful only at equal accuracy

Figure: Kochkov et al., PNAS 2021, 2D Kolmogorov flow, one TPU core. Rules: McGreivy & Hakim, Nature Machine Intelligence 2024

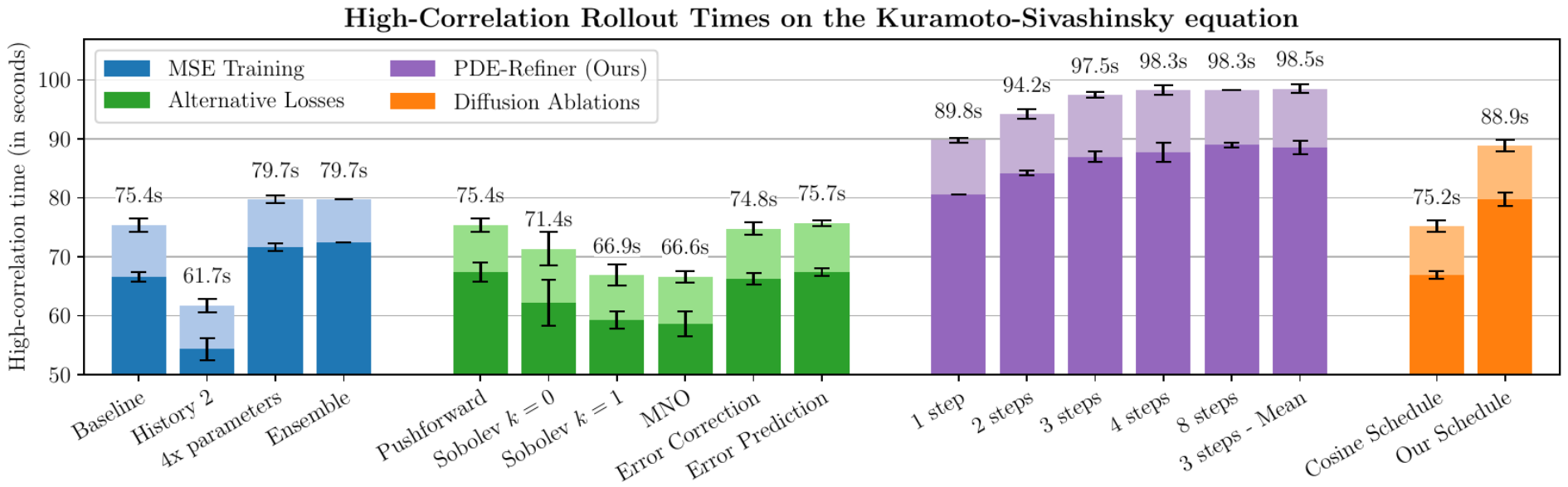
Generalization tests

Hold out	Tests	BubbleML example
Trajectories	New initial conditions, same physics	Other runs at Twall = 106 ° C
Parameters	Interpolation or extrapolation in a physical parameter	Train 95-105 ° C, test 106 ° C (inside) or 110 ° C (outside)
Time	Later times and rollouts longer than in training	Train on frames up to 600, test on later frames, roll out 100+ steps
Resolution	A finer or coarser grid than in training	Train at 64 x 64, test at 128 x 128
System	New geometry, fluid or boundary condition	Flow boiling or a different fluid

- Split by what should generalize, never by random frames of one trajectory
- State which kind of test each number comes from. Out-of-distribution error is usually much larger

Report results on in-distribution and out-of-distribution tests separately

Seeds, test cases and error bars



- Training is random: initialization, batch order, noise. Train with **3-5 seeds** and report the mean and standard deviation
- Average over many test trajectories and report the spread as well as the mean
- Tune hyperparameters on a validation set. Use the test set once, at the end
- Give every baseline the same tuning effort and compute budget (The Well: 12 h on one H100 per model)

A difference smaller than the spread over seeds is not a result

Figure: Lippe, Veeling, Perdikaris, Turner & Brandstetter, PDE-Refiner, NeurIPS 2023
Dark/light: time until mean correlation drops below 0.9/0.8. Error bars: standard deviation over 5 seeds. y-axis starts at 50 s

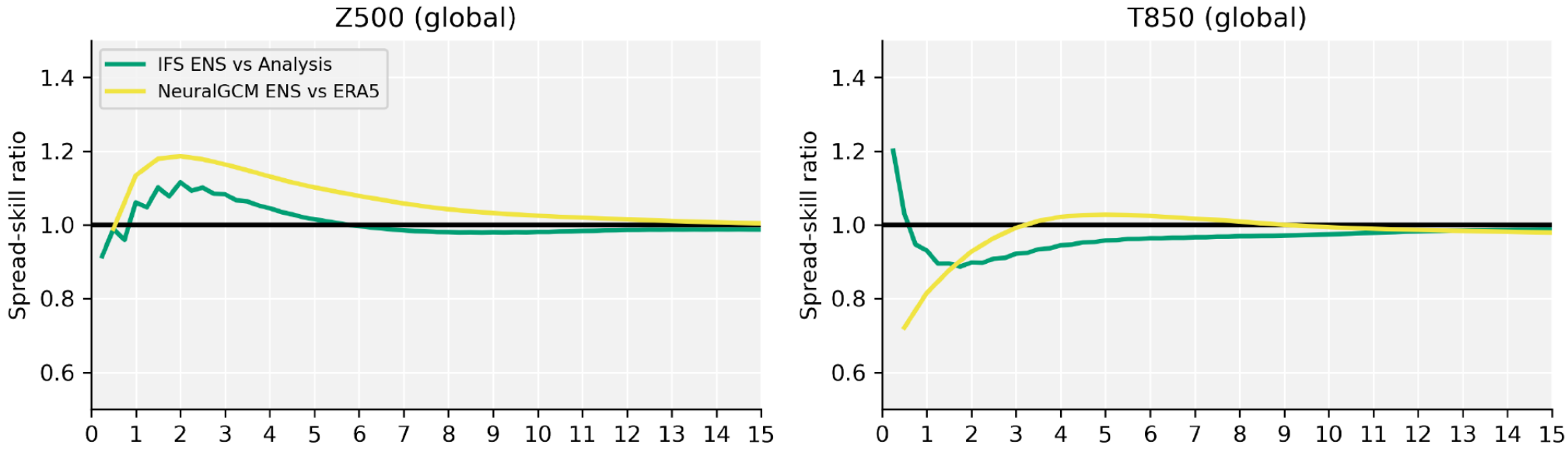
Benchmarks for SciML

Benchmark	Data	Metrics and baselines
PDEBench (2022)	1D-3D PDEs: advection, Burgers, diffusion-reaction, Darcy, compressible and incompressible Navier–Stokes	RMSE, nRMSE, fRMSE, bRMSE, cRMSE; U-Net, FNO, PINN
PDEArena (2023)	Navier-Stokes, shallow water, Maxwell	Rollout error; U-Net and FNO variants; common training code
The Well (2024)	16 datasets, 15 TB: fluids, plasma, astrophysics, biology	VRMSE by field, by time window and by frequency band
WeatherBench 2 (2024)	ERA5 reanalysis, global weather	RMSE, ACC, spectra, CRPS; persistence, climatology, IFS
BubbleML (2023)	Pool and flow boiling	Field errors and physical metrics such as heat flux

- Benchmarks fix the data, the splits and the metrics, so results are comparable across papers
- **You still choose** the baselines, the compute budget and the physical checks

Takamoto et al., NeurIPS 2022. Gupta & Brandstetter, TMLR 2023. Ohana et al., NeurIPS 2024. Rasp et al., JAMES 2024. Hassan et al., NeurIPS 2025

Evaluating probabilistic forecasts



- An ensemble or a generative model predicts a distribution. Score the distribution, not only its mean
- **CRPS** = $E|X - y| - \frac{1}{2} E|X - X'|$, for samples X, X' of the forecast and observation y. It equals the absolute error for a single deterministic forecast
- **Spread-skill ratio**: ensemble spread / RMSE of the ensemble mean. Near 1: the ensemble spread matches its actual error

Future lecture: generative models and uncertainty quantification

Figure: Rasp et al., WeatherBench 2, JAMES 2024. CRPS: Gneiting & Raftery, JASA 2007

Evaluation checklist for your project

- Split by trajectory, parameter or time. Label in- and out-of-distribution results
- Baselines: persistence, a published model with public code, and a numerical solver at equal accuracy/runtime if you claim speed
- Metrics: VRMSE or relative L2 for each field, one step and over the rollout
  - Chaotic systems: high-correlation time, then statistics (spectrum, PDF)
  - Physics: conserved quantities, constraints, one quantity of interest
- 3 or more seeds; mean and standard deviation; test set used once
- Same tuning budget for every model; report training and inference cost
- Show failure cases as well as successes

Mid-project check-in: baselines reproduced and the metrics computed

Summary

- Report the error of each field with a normalization that matches the question (relative L2, VRMSE), at one step and over the rollout
- Compare with persistence; split the error by scale
- Pointwise metrics reward blur (double penalty). Check spectra and structures
- Chaotic systems: high-correlation time for the state, statistics after decorrelation
- Check conserved quantities, constraints and the quantities the scientist will use
- Speedups: equal accuracy or equal runtime, against an efficient solver
- Seeds, test cases, error bars; the test set is used once

After class

- McGreivy & Hakim, Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related PDEs, Nature Machine Intelligence 2024
- Ohana et al., The Well, NeurIPS 2024: Section 4 and Appendix E (metrics)
- Rasp et al., WeatherBench 2, JAMES 2024: Section 4 (evaluation)
- Kochkov et al., Machine learning-accelerated computational fluid dynamics, PNAS 2021

Next, Lecture 11 (Thu, Oct 29): Project abstract presentations

NEXT: HANDS-ON

From concepts to code

→ Open the Lecture 10 notebook

1. Evaluate the Lecture 9 models and persistence: relative L2 and VRMSE per field, correlation and high-correlation time, error by frequency band
2. Double penalty: score a sharp displaced bubble and a blurred one
3. Quantity of interest: domain-mean θ over the rollout