

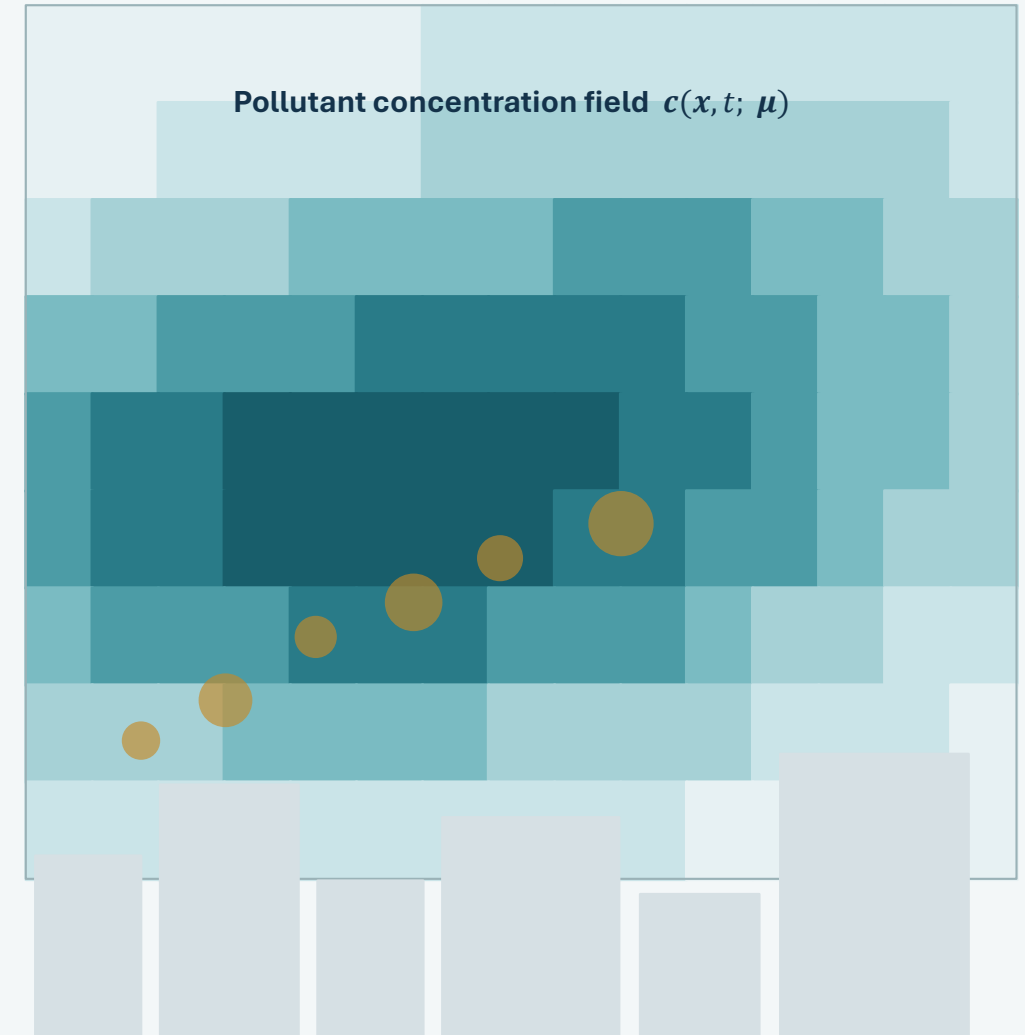
Physics-Informed Machine Learning for Air-Quality Digital Twins

Parametric PINNs and Physics-Informed Neural Operators (PINO)

A 15-minute technical briefing for academic and engineering audiences

Central proposition

For a digital twin, the computational target is a reusable map from changing atmospheric inputs and parameters to concentration fields — not a single PDE solution.



What is an Air Quality Digital Twin?

A dynamic digital representation of the air-quality system, continuously updated from measurements and used for prediction, diagnosis, and scenario analysis.

Real data.
Smarter insights.
Cleaner air.
Healthier communities.



Physical asset / real-world air-quality system

The real environment
we monitor and manage



Continuous data exchange

Communication
and synchronisation

Real-world data
(physical → digital)

- Sensor streams
- Meteorology (wind, T, humidity)
- Emissions (traffic, industry, etc.)
- Boundary conditions
- Satellite/remote sensing
- Operational data (e.g. traffic)

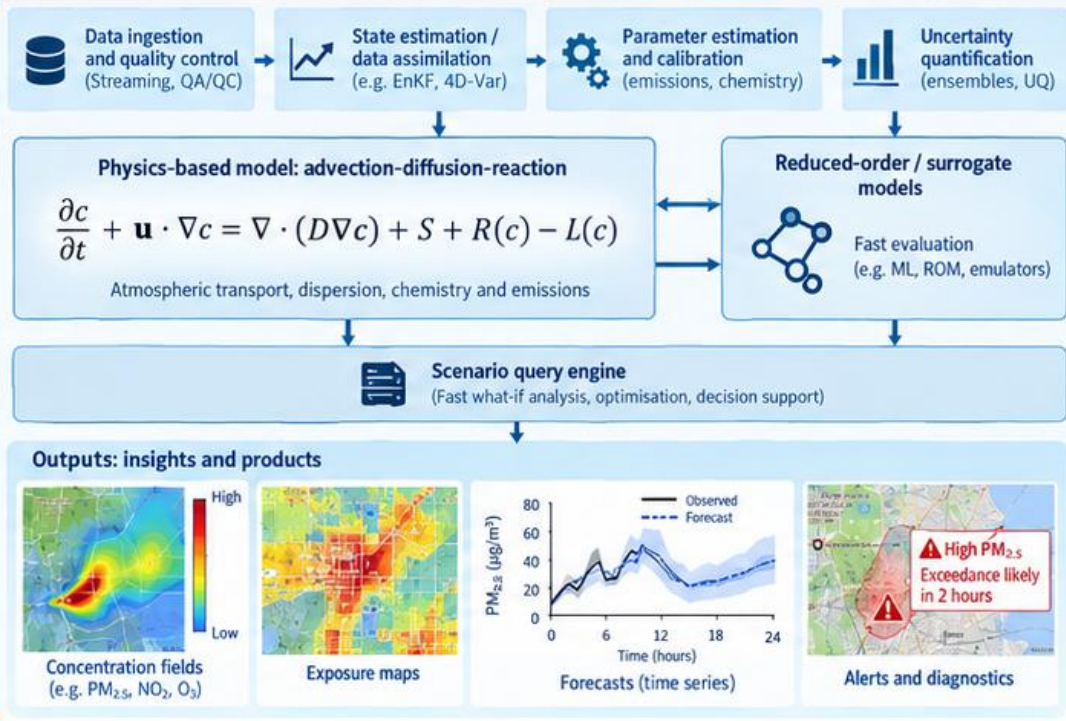
Information for action
(digital → physical)

- Predictions and forecasts
- Diagnostics and source attribution
- Control recommendations
- What-if scenario results
- Alerts and early warnings

Closed loop for
a cleaner, healthier tomorrow

Digital asset / air-quality digital twin

A continuously updated
computational representation



Computational requirements

Enabling near real-time, continuously updated digital twins

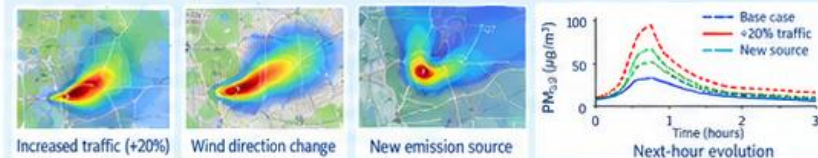


Near real-time scenario queries

Rapidly explore "what if?" questions and quantify impacts

- ✓ What if traffic increases by 20%?
- ✓ What if wind direction changes?
- ✓ What is the effect of a new emission source?
- ✓ How will concentrations evolve over the next hour?
- ✓ Which mitigation strategy is most effective?

Fast results (seconds – minutes)



Why it matters

Turning data and models into cleaner
air and healthier communities



Rapid forecasting

Anticipate air-quality issues



Design and policy evaluation

Test interventions before implementation



Source attribution and diagnostics

Identify and quantify sources



Exposure/risk assessment

Inform public health decisions

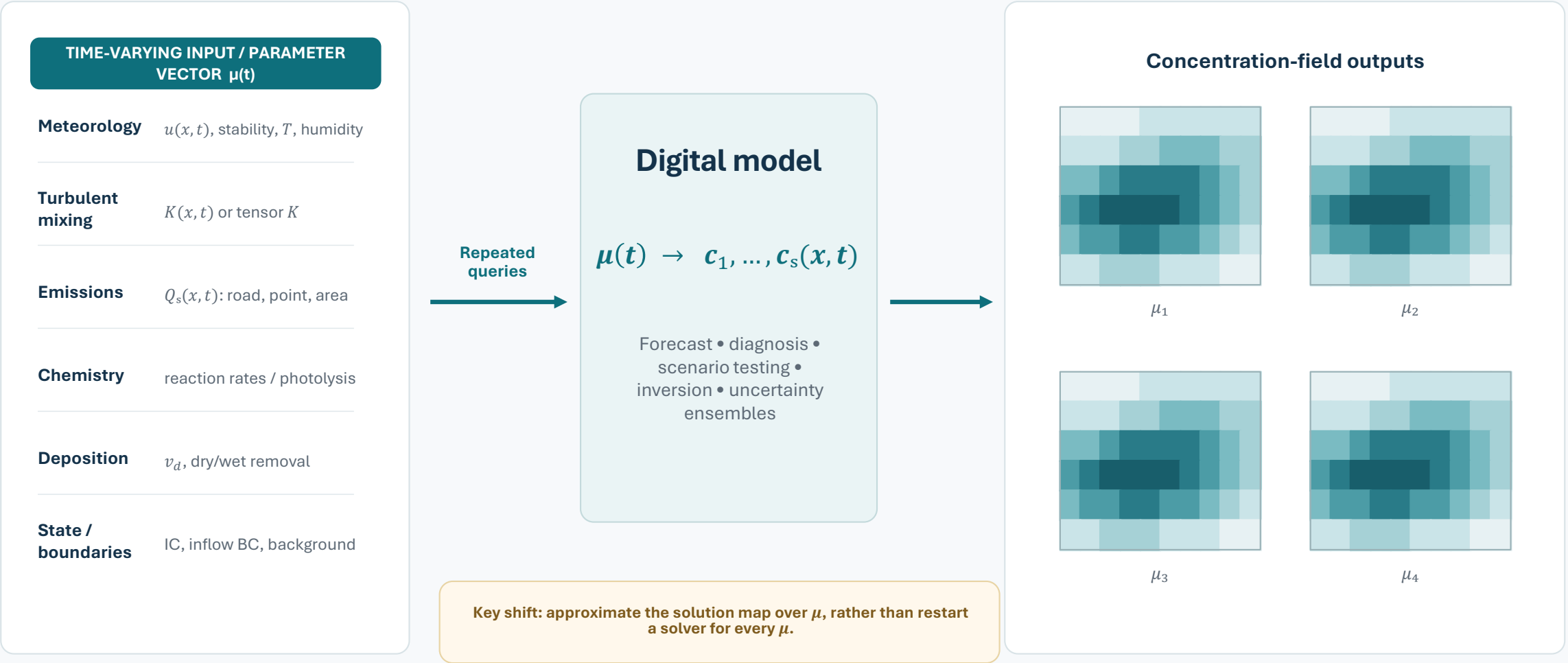


Decision support for operations

Enable proactive, data-driven action

Digital-twin requirement: solve a family of transport PDEs

The atmosphere, emissions and boundary conditions change continuously — the twin must respond repeatedly and quickly.



Pollutant transport PDE: the physics to be embedded

A representative species equation combines advection, turbulent diffusion, sources, chemistry and removal.

$$\partial c_s / \partial t + \nabla \cdot (u c_s) = \nabla \cdot (K \nabla c_s) + Q_s + R_s(c, T, I, \dots) - \lambda_{dep, s} c_s$$

Storage

$$\partial c_s / \partial t$$

transient evolution

Advection

$$\nabla \cdot (u c_s)$$

wind-driven transport

Diffusion

$$\nabla \cdot (K \nabla c_s)$$

turbulent mixing

Sources

$$Q_s$$

traffic / point / area

Chemistry

$$R_s(\cdot)$$

coupled reactions

Removal

$$-\lambda_{dep, s} c_s$$

deposition / scavenging

Parametric representation

$$\mu = \{ u, K, Q, k_{chem}, \lambda_{dep}, IC, BC, \dots \}$$

The digital twin needs $c(x, t; \mu)$ over the admissible operating distribution of μ .

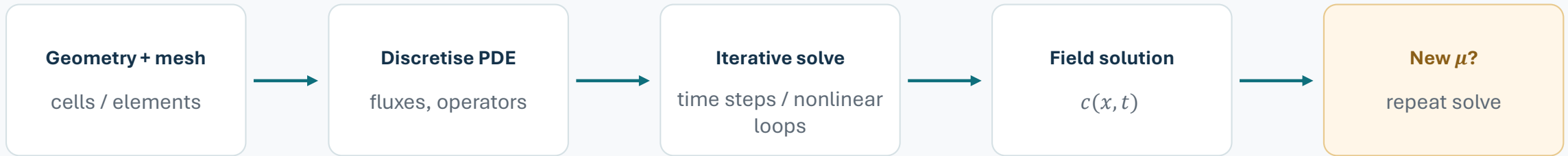
Boundary / initial constraints remain essential

$c(x, 0) = c_0(x)$ • inflow Dirichlet • outflow/flux Neumann • ground deposition Robin-type conditions

Representative formulation; specific chemistry/deposition operators depend on the atmospheric model and species set.

Conventional numerical solvers are the reference — but costly to repeat

Finite-volume / finite-element / finite-difference solvers remain essential for fidelity, validation and high-risk regimes.



Why we keep the numerical solver

- Well-established discretisation, convergence and conservation machinery.
- Handles complex geometry, boundary conditions and coupled physics directly.
- Provides high-fidelity labels, benchmark cases and a deployment fallback.

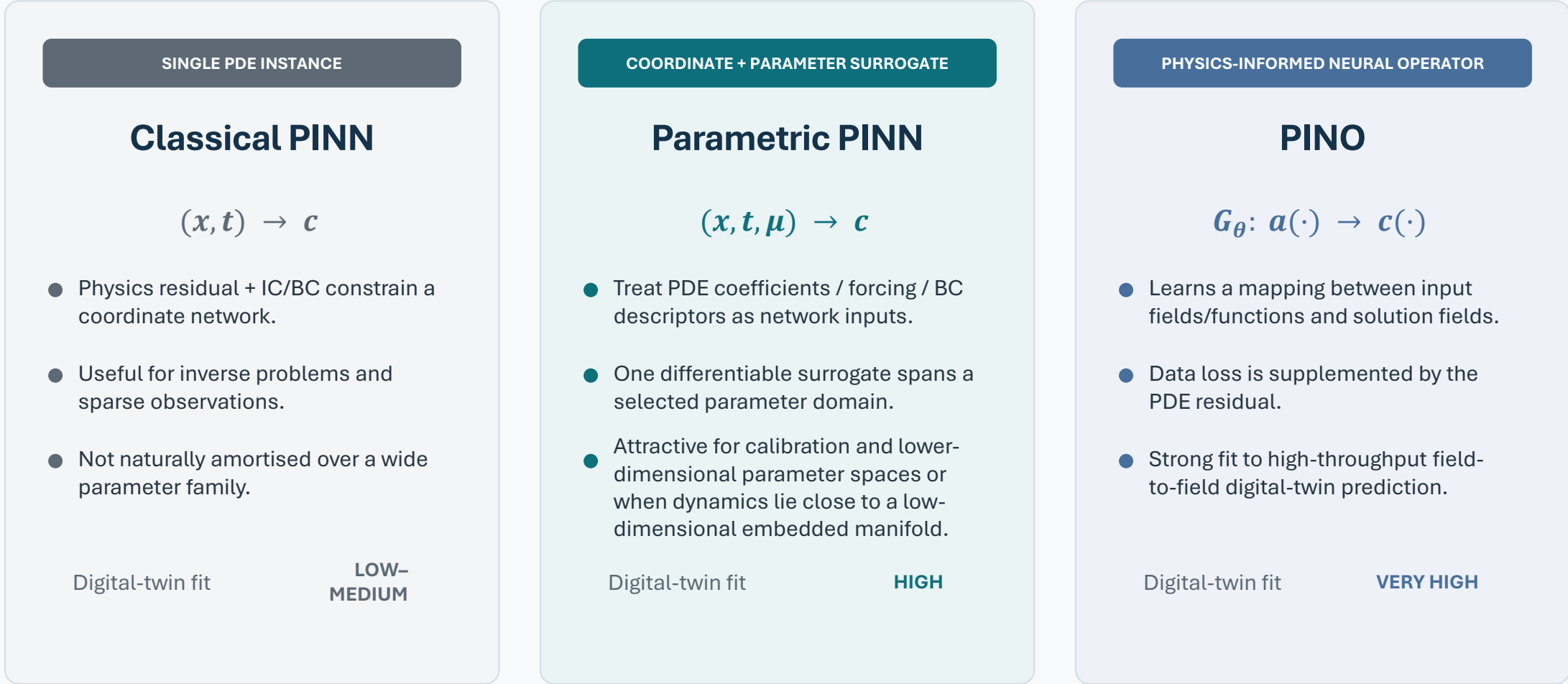
Digital-twin bottleneck

- Cost is paid again for every new wind / emissions / parameter scenario.
- Data assimilation, optimisation and uncertainty propagation multiply the number of required solves.
- Wall-clock latency can become incompatible with rapid online diagnostics or large ensembles.

PIML changes where the computation is paid; more offline training, much less repeated online solve cost.

Physics-informed ML landscape for the air-quality twin

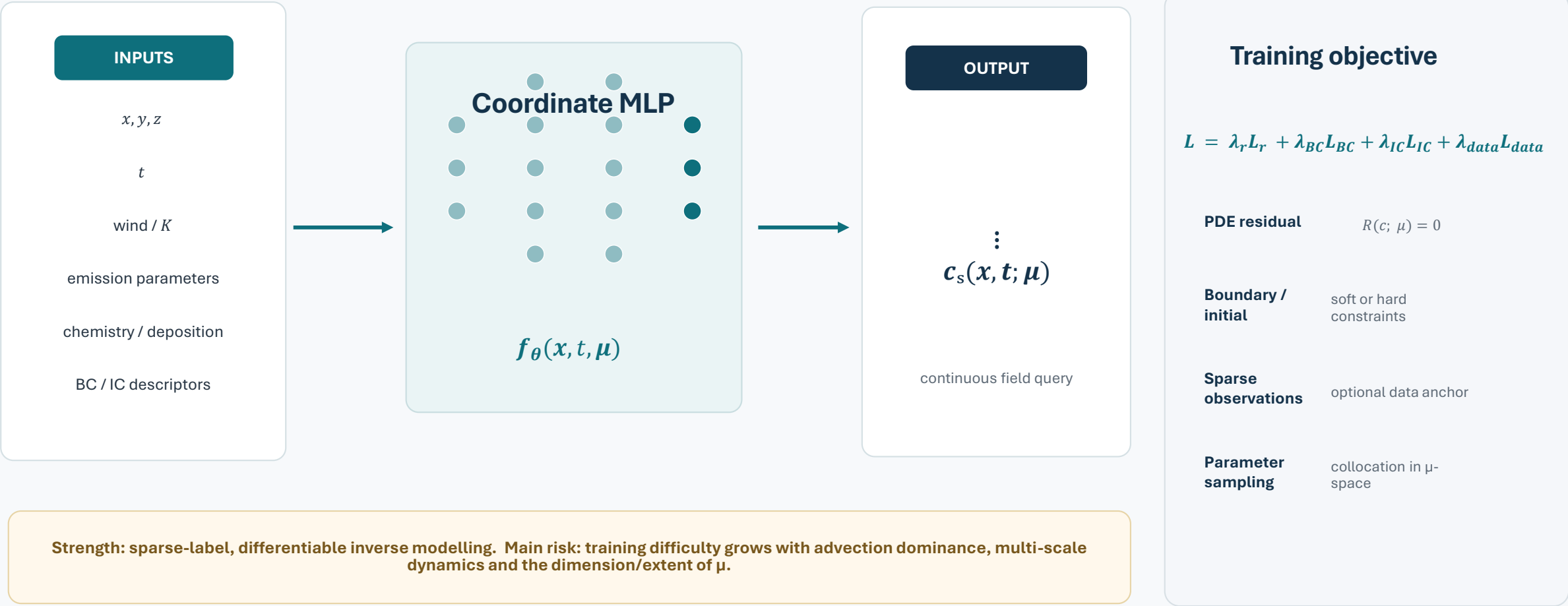
The key distinction is whether the network approximates one solution, a parametric solution field, or the full solution operator.



PINN: Raissi, Perdikaris & Karniadakis, J. Comput. Phys. 378 (2019). PINO: Li et al., ICLR 2022 / arXiv:2111.03794.

Conventional Parametric PINN: one differentiable surrogate over the selected parameter domain

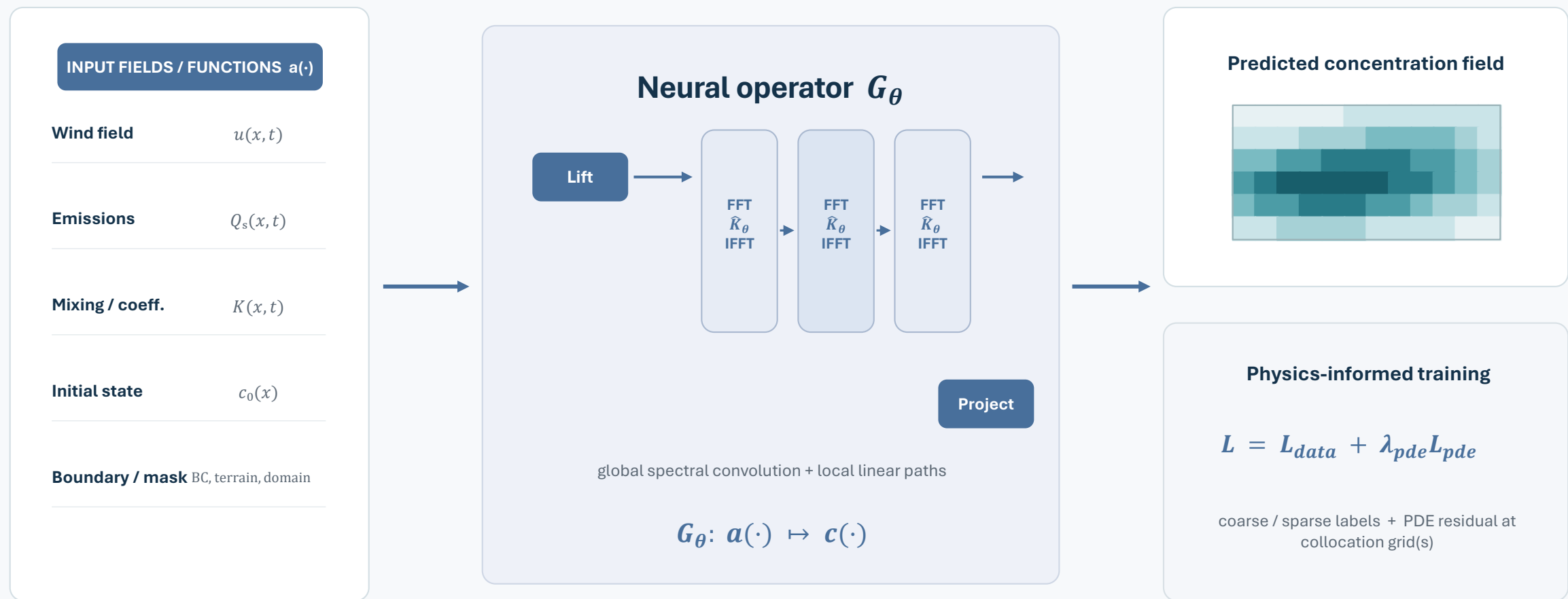
Physics residuals are evaluated by automatic differentiation at collocation points in space, time and parameter space.



Parametric PINN is a natural extension of the PINN formulation by including PDE parameters / forcing descriptors among network inputs.

PINO: learn the solution operator, not a single solution

A Physics-Informed Neural Operator combines operator learning with a PDE residual, often using an FNO backbone.



The lifting layer maps the relatively low-dimensional physical input at each spatial point into a higher-dimensional feature representation that the neural operator can process effectively. For example, the lifting layer might map 4 inputs into 64 latent features:

Li et al., "Physics-Informed Neural Operator for Learning Partial Differential Equations", ICLR 2022 / arXiv:2111.03794; FNO backbone: Li et al., ICLR 2021.

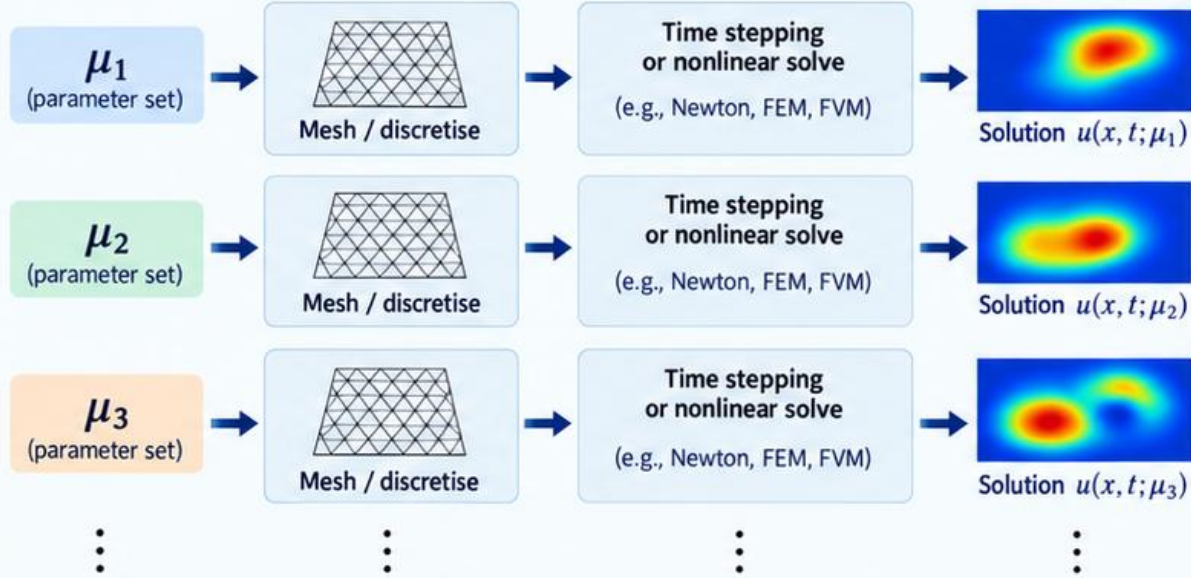
Advantages of PIML PDE Solvers in Many-Query Scenarios

Illustrated using parametric PINN and PINO

Best suited when the same PDE family is queried repeatedly; training cost is paid upfront.

Traditional numerical PDE solver

For each query: solve $\mathcal{N}(u; \mu) = 0$



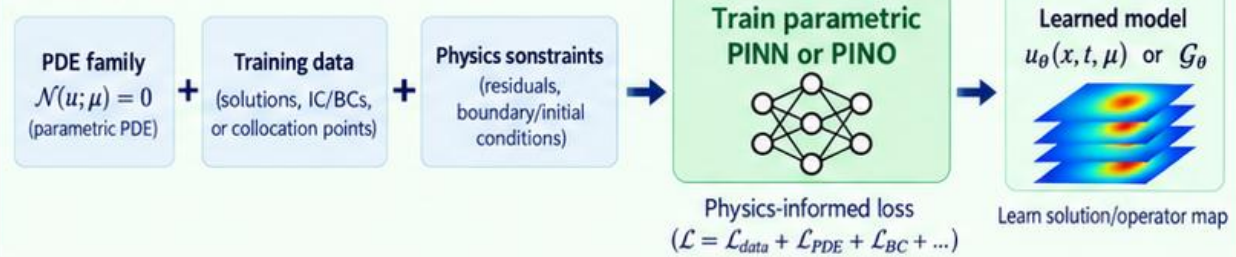
Consequence

- High cumulative cost
- Slow parameter sweeps
- Difficult real-time use

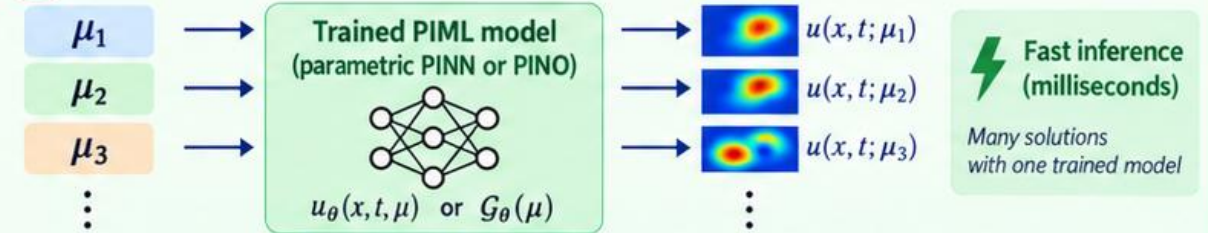
PIML surrogate solver (parametric PINN / PINO)

1 Offline training (one-time cost)

Learn $u_\theta(x, t, \mu)$ or $\mathcal{G}_\theta : \mu \mapsto u(x, t)$



2 Online inference (fast repeated queries)



Consequence

- Amortised cost over many queries
- Near real-time prediction
- Rapid parameter studies

Why this matters in many-query settings



Design optimisation

Explore design spaces quickly



Uncertainty quantification

Evaluate many samples (e.g., Monte Carlo)



Inverse problems / calibration

Repeated solves during optimisation



Model predictive control / digital twins

Fast evaluations for real-time decisions



Large parameter sweeps

Map system behaviour over wide parameter ranges

Key advantages after training

1

Fast evaluation for new parameters

2

No full re-solve for each query

3

Differentiable surrogate for sensitivities

4

Suitable for repeated online use

5

PINO often fastest at inference; parametric PINN flexible when data are limited

Amortised Cost in an Air-Quality Digital Twin

1. Upfront training

Use physics (PDE/CFD) data to train a surrogate model.



2. Trained surrogate model

A physics-informed neural network that emulates the PDE solver.

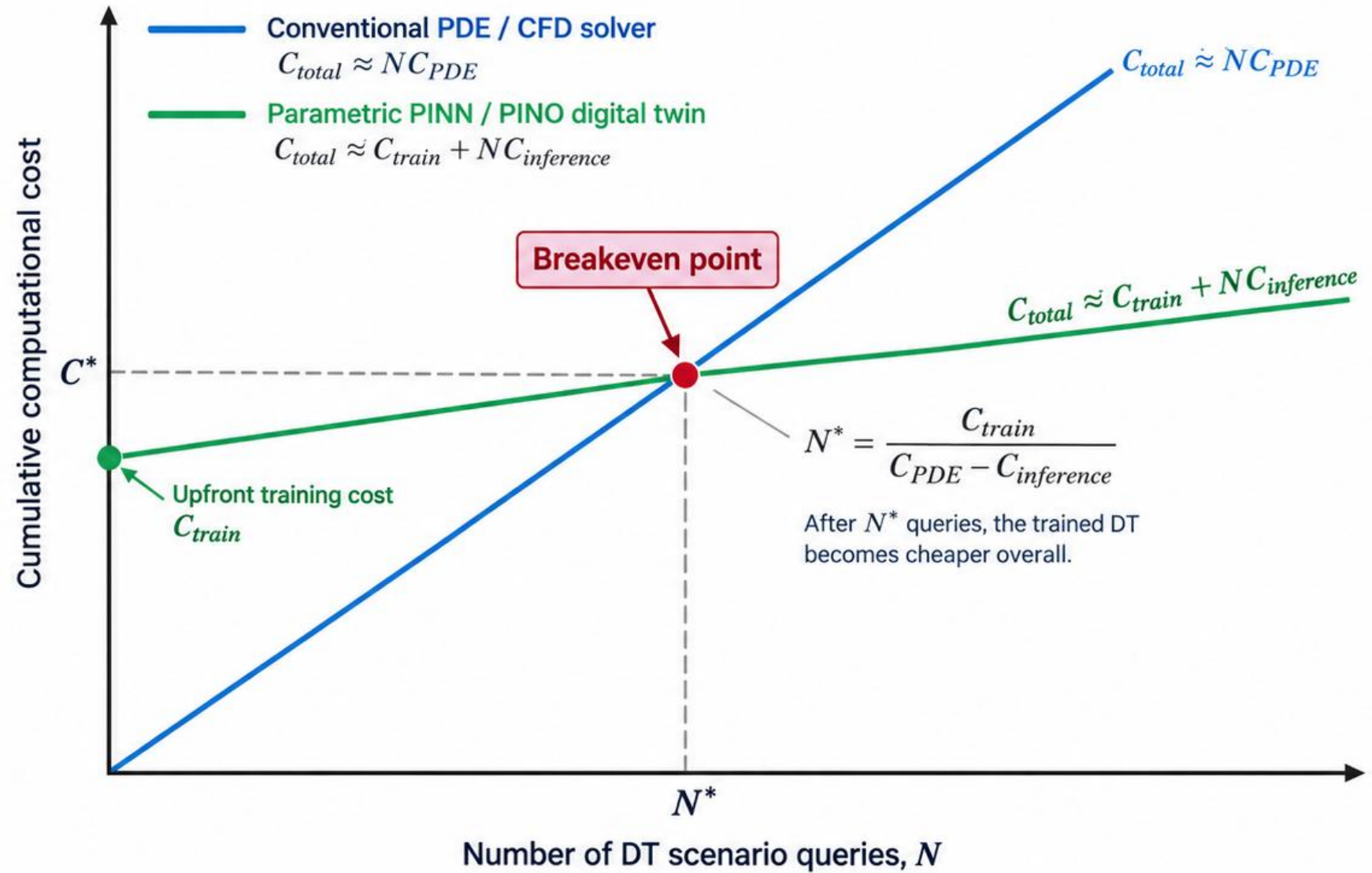


3. Fast many-query inference

Rapid evaluation for different scenarios (e.g. emissions, wind, pollutant fields).



- **Amortised** = pay a large cost once, then reuse the model many times.
- **Ideal for many-query air-quality tasks:** scenario testing, control options, uncertainty sampling.
- **Near real-time answers** replace repeated expensive PDE solves.



Key idea: expensive training is spread across many future digital-twin queries, reducing average cost per prediction.

Chemical stiffness: why it matters to the digital twin

Fast reaction modes can evolve much more rapidly than transport, deposition and the measurement-update cycle.

Where stiffness enters

$$\frac{\partial \mathbf{c}_s}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{c}_s) = \nabla \cdot (\mathbf{K} \nabla \mathbf{c}_s) + \mathbf{Q}_s + \mathbf{R}_s(\mathbf{c}, T, I, \dots) - \lambda_{dep, s} \mathbf{c}_s$$

stiff chemistry: $\mathbf{R}_s(\mathbf{c}, T, I, \dots)$

Reaction Jacobian $\mathbf{J}_R = \partial \mathbf{R} / \partial \mathbf{c}$

$$|\lambda_{fast}| \gg |\lambda_{slow}|$$

$$\tau_i \approx 1 / |\lambda_i|$$

Characteristic timescales



1 Numerical latency $\uparrow$

- Explicit steps become restrictive
- Implicit/BDF/Rosenbrock solves add cost
- Repeated scenarios multiply that cost

2 Estimation becomes harder

- Fast modes may occur between sensor updates
- Sensitivity matrices can be ill-conditioned
- Calibration becomes more fragile

3 UQ cost multiplies

- Each ensemble member inherits the stiff solve
- Monte-Carlo / optimisation become expensive
- Online uncertainty budgets tighten

4 PIML training is harder

- Physics residual terms become badly scaled
- Fast transients demand richer time sampling
- Vanilla PINN/PINO are not stiffness-proof

Digital-twin constraint:

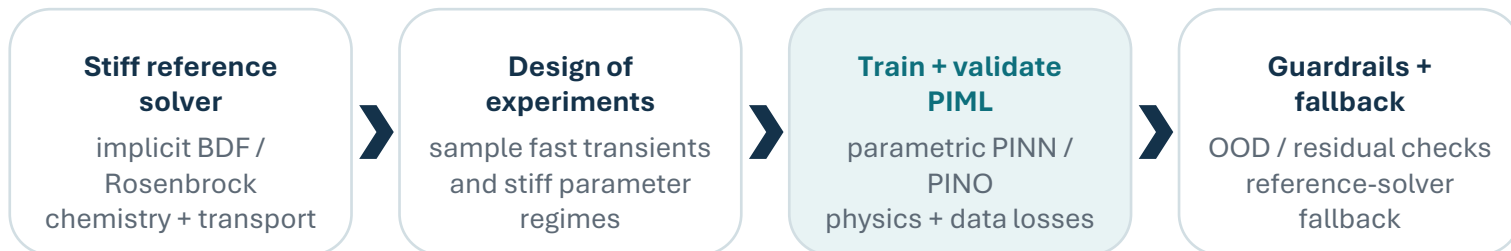
$$T_{compute} \gg T_{physical}$$

— stiffness pushes compute time upward when the digital twin needs repeated, near real-time scenario queries.

Stiffness-aware digital-twin architecture

Keep validated stiff physics as the offline authority; deploy a validated surrogate for the online many-query workload.

OFFLINE / PERIODIC HIGH-FIDELITY FOUNDATION



ONLINE / NEAR REAL-TIME DIGITAL-TWIN LOOP



← new observations / residuals refresh the state, parameters and surrogate when required

Stiffness-aware PIML design

- Dimensionless residual terms
- Oversample rapid chemical transients
- Use adaptive / causal loss weighting where needed
- Validate latency and accuracy across stiffness regimes
- Retain OOD detection and numerical-solver fallback

Research option: slow-fast latent chemistry

$$\mathbf{c} \rightarrow \mathbf{E}_c \rightarrow [\mathbf{z}_s, \mathbf{z}_f]$$

$$\varepsilon \frac{d\mathbf{z}_f}{dt} - f(\mathbf{z}_s, \mathbf{z}_f) \approx 0$$

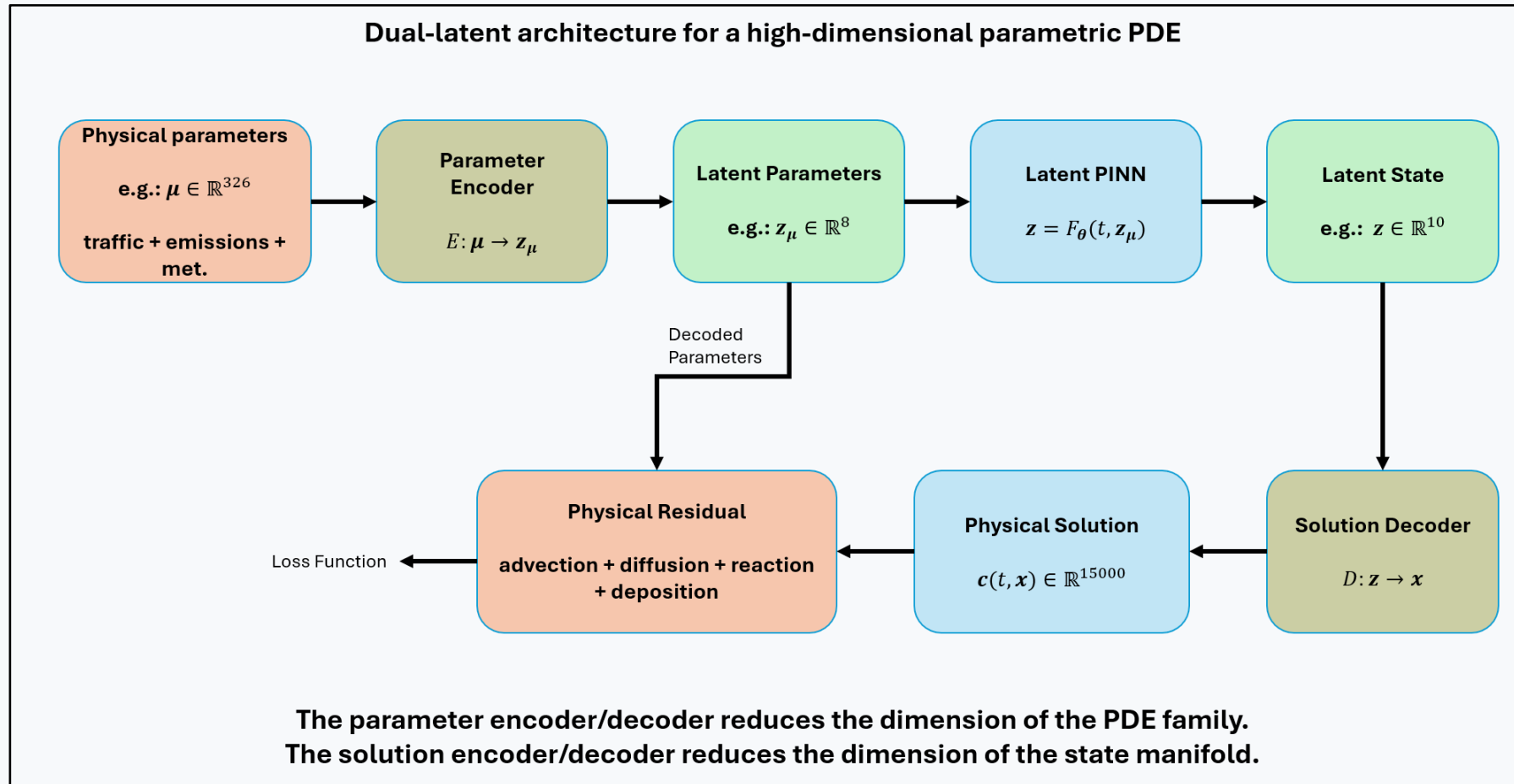
Potentially reduces the effective online dimensionality of stiff chemistry.

Key implication:

Stiffness strengthens the economic case for amortised surrogates — but raises the training, validation and robustness burden.

Dual-latent architecture

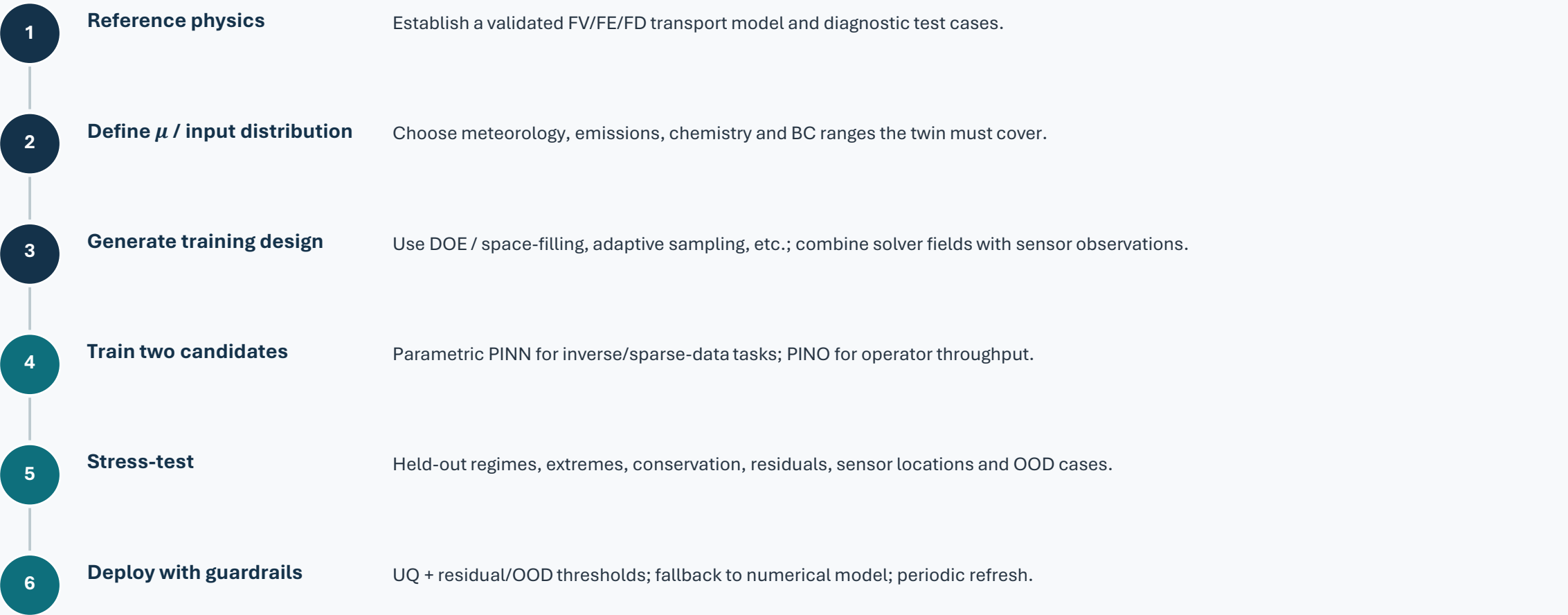
Parameter compression + nonlinear solution-manifold reduction



Two encoders address two different manifolds; the physical PDE residual couples them.

Recommended development strategy for an air-quality digital twin

Treat the numerical model, parametric PINN and PINO as complementary assets rather than mutually exclusive alternatives.



Likely operational split: PINO for repeated forward prediction + parametric PINN / differentiable solver for parameter inference and calibration.

Open-source implementation stack (August 2026)

A practical stack exists today for PINNs, neural operators, numerical reference models and GPU deployment.

PIML FRAMEWORK

NVIDIA PhysicsNeMo / physicsnemo.sym • PyTorch

MODEL ARCHITECTURES

Fully connected PINNs • FNO / neural operators • PINO-style physics-informed operator training

NUMERICAL REFERENCE / DATA GENERATION

OpenFOAM • FEniCSx / PETSc • established atmospheric / CFD solvers

DATA + DEPLOYMENT

xarray / Zarr / HDF5 • GPU inference • monitoring, UQ and OOD guardrails

PhysicsNeMo documentation currently exposes PINN workflows and FNO/neural-operator architectures; PhysicsNeMo-Sym documentation includes PINO examples.

NVIDIA PhysicsNeMo Framework documentation, v26.05 / latest (accessed Aug 2026); PhysicsNeMo-Sym PINO documentation v26.03.

Which PIML solver is most suitable for a stiff pollutant transport PDE?

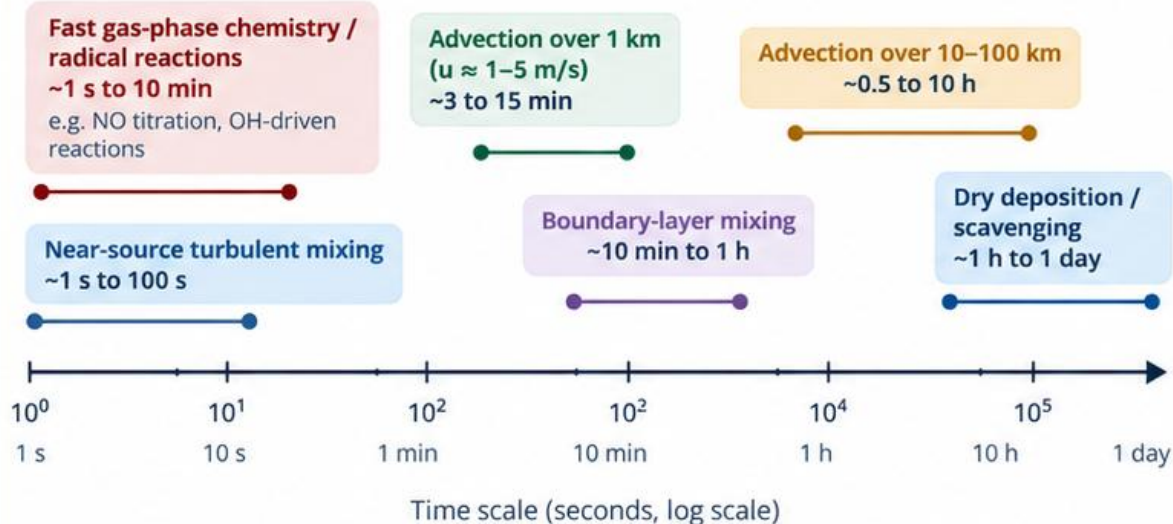
Conventional PINN, latent-variable parametric PINN, and PINO (FNO) compared for chemistry-transport digital twins.

$$\partial c_s / \partial t + \nabla \cdot (u c_s) = \nabla \cdot (K \nabla c_s) + Q_s + R_s(c, T, I, \dots) - \lambda_{\text{dep}, s} c_s$$

Stiffness arises because the characteristic time scales differ by several orders of magnitude.

Physical time scales in pollutant transport

Key processes span seconds to days, leading to a stiff, multi-scale system.



**Scale separation: seconds to day-scale removal ($\approx 10^4$ – 10^5)
 $\Rightarrow$ stiff chemistry-transport system.**

Suitability of candidate solvers

Different PIML solvers offer different trade-offs for stiff chemistry-transport problems.

Conventional PINN

Limited for stiff many-query AQ PDEs

- Physics-informed and data-efficient for small problems
- Good for inverse/parameter-estimation tasks
- Training can become unstable when fast chemistry and slower transport coexist
- Usually one network per case; poor many-query scalability.

Parametric PINN + latent variable auto-encoders

Good compromise

- Learns a reduced latent manifold for parameters/solutions
- Better suited to repeated scenario queries than a conventional PINN
- Can reduce dimensionality and improve amortisation
- Use fast and slow latent variable representation to handle stiff equations

PINO based on FNO

Best for rapid digital-twin querying

- Learns the solution operator over a family of conditions
- Very fast inference after training
- Well suited to many-query forecasting and scenario analysis
- Needs more high-quality training snapshots and may require wavelet operators to handle stiff dynamics



Overall ranking for a stiff pollutant air-transport digital twin: PINO(FNO) for fastest many-query deployment or latent parametric PINN when data are scarcer and the solution manifold is compressible; conventional PINN mainly for smaller or single-scenario inverse problems.

Take-home message

For digital twins, the decisive advantage is not “ML versus physics” — it is amortised physics with explicit validation and guardrails.

1

Use the numerical solver as the physics authority

It remains the source of high-fidelity benchmarks, training data, stress tests and fallback behaviour.

2

Use parametric dual-latent variable PINNs when stiffness, differentiation and sparse-data inversion matter

They provide a continuous $c(x,t; \mu)$ surrogate and direct gradients with respect to unknown parameters.

3

Use PINO where the twin needs repeated full-field predictions and stiffness is relatively low

Operator learning is naturally aligned with families of PDEs and high-throughput scenario / ensemble evaluation.

Selected references and software sources

- Raissi, M., Perdikaris, P. & Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear PDEs. *Journal of Computational Physics*, 378, 686–707. doi:10.1016/j.jcp.2018.10.045.
- Li, Z. et al. (2021). Fourier Neural Operator for Parametric Partial Differential Equations. ICLR 2021; arXiv:2010.08895.
- Li, Z. et al. (2022). Physics-Informed Neural Operator for Learning Partial Differential Equations. ICLR 2022; arXiv:2111.03794.
- NVIDIA PhysicsNeMo Framework documentation (latest / v26.05): PINN workflows, physicsnemo.sym, neural operators and FNO architectures.
- NVIDIA PhysicsNeMo-Sym documentation v26.03: Physics-Informed Neural Operator theory and Darcy PINO tutorial.
- Karniadakis, G.E. et al. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3, 422–440. doi:10.1038/s42254-021-00314-5.

Software note: package APIs and examples evolve; verify against the deployed PhysicsNeMo release when implementation starts.