TURING: Towards Robust Foundation Models and Platform Services for Complex Physical Systems

Ioannis Kontopoulos Miloš Savić Dusan Jakovetic Jelena Topić Božič
Maria Pateraki Konstantinos Tserpes

National Technical University of Athens, Greece
Institute of Communication and Computer Systems, Greece
Faculty of Sciences, University of Novi Sad, Serbia
Rudolfovo, Science and Technology Centre, Slovenia

GECON 2026
Special Session on European Research Projects

What Are Physical Models?

Physical models are formal representations of real-world physical systems.

They usually combine:

- governing equations;
- conservation laws;
- material properties;
- boundary and initial conditions;
- numerical solvers;
- domain-specific assumptions and approximations.

Core idea

A physical model encodes knowledge about how the system is expected to behave.

Examples of Physical Models

Domain	Example physical model
High-energy physics	Particle-detector simulations modelling how particles interact with detector materials and geometries
Nuclear engineering	Thermal-hydraulic and reactor-core simulations modelling heat transfer, fluid flow, neutron transport, and safety-relevant parameters
Criticality safety	Neutronics models estimating quantities such as multiplication factors, sensitivities, and safety margins
Severe-accident analysis	Debris-bed cooling and corium coolability models describing complex thermal and hydraulic accident processes
Meteorology	Numerical weather prediction models solving atmospheric dynamics over large spatial and temporal domains

Why Physical Models Are Important

Physical models are important because they support:

- scientific understanding of complex phenomena;
- engineering design and optimisation;
- safety analysis and risk assessment;
- prediction of rare or expensive-to-observe events;
- policy, planning, and operational decision support.

Why they are trusted

They are grounded in physical principles and can often be inspected, validated, and interpreted using domain knowledge.

But

Their accuracy and physical grounding often come at high computational cost.

Physical Models Limitations

High-fidelity simulation becomes expensive when systems involve:

- multi-scale phenomena;
- multi-physics interactions;
- nonlinear dynamics;
- large three-dimensional domains;
- long time horizons;
- complex geometries.

Consequence

Every new configuration, scenario, or operating condition may require a new expensive simulation run.

Limitation 1: Multi-scale Phenomena

Multi-scale means that relevant behaviour occurs at very different spatial or temporal scales.

- In high-energy physics, microscopic particle interactions must be related to detector-level signals.
- In weather prediction, small-scale atmospheric processes influence regional or larger-scale forecasts.
- In nuclear systems, local thermal or neutronic effects can influence system-level safety indicators.

Why this is hard

The simulator may need to resolve fine-scale effects while also modelling large-scale system behaviour.

Limitation 2: Multi-physics Phenomena

Multi-physics means that several interacting physical processes must be modelled together.

Examples include:

- heat transfer coupled with fluid flow in nuclear thermal-hydraulics;
- neutron transport coupled with reactor-core behaviour;
- atmospheric dynamics coupled with land-surface processes;
- geometry, material response, and particle interactions in detector simulation.

Why this is hard

Errors or approximations in one physical process may propagate to the others.

Limitation 3: Nonlinear Dynamics

Nonlinear dynamics means that small changes in input conditions can produce disproportionately large changes in outputs.

Examples include:

- instability or transition phenomena in thermal-hydraulic systems;
- long-horizon error growth in weather prediction;
- accident scenarios where regime changes alter system behaviour;
- detector responses that depend nonlinearly on geometry, energy, and material properties.

Why this is hard

The model must remain reliable near regime changes and rare events.

Limitation 4: Large 3D Domains

Many physical models operate over large three-dimensional spaces.

Examples include:

- detector geometries in high-energy physics;
- reactor-core spatial layouts;
- atmospheric grids in regional weather forecasting;
- spatially distributed thermal and hydraulic fields.

Why this is hard

Large 3D domains generate high-dimensional inputs and outputs, leading to large memory, storage, and compute requirements.

Limitation 5: Long Time Horizons

Some systems must be simulated over many time steps.

Examples include:

- multi-step weather forecasts;
- thermal-hydraulic transients;
- accident progression scenarios;
- time-series behaviour in cooling or structural response.

Why this is hard

Small numerical or modelling errors can accumulate, making long-horizon stability and uncertainty critical.

Limitation 6: Complex Geometries

Complex geometries make simulation and generalisation harder.

Examples include:

- changing detector layouts;
- reactor-core configurations;
- irregular debris-bed structures;
- spatially heterogeneous weather domains;
- structural or mechanical systems with nontrivial boundaries.

Why this is hard

A model trained on one geometry may fail when the physical layout changes.

Suggestion: Use AI to Assist Physical Modelling

Research hypothesis

AI models can complement, accelerate, or partially replace physical simulation workflows, provided that they preserve the trustworthiness properties required by scientific and engineering applications.

AI models can learn from:

- simulation outputs;
- observational data;
- benchmark datasets;
- previously trained models;
- domain-specific physical constraints.

Ways to Use AI in Physical Modelling

AI can be introduced at different levels of the modelling workflow.

- **Full surrogate**: replace an expensive simulator for selected inputs and outputs.
- **Partial surrogate**: replace only the most expensive component of a simulation workflow.
- **Hybrid model**: combine equations and learned components.
- **Physics-informed model**: include governing equations or constraints in training or architecture.
- **Reduced-order or latent model**: learn compact representations of high-dimensional physical states.
- **Assistant model**: support scenario exploration, anomaly detection, or model selection.

Challenges in Implementing the Suggestion

AI models may be fast and flexible, but they can fail when used outside their training comfort zone.

They may fail under:

- changed geometries;
- unseen operating conditions;
- rare events;
- noisy observations;
- extrapolative regimes.

They may also produce:

- non-physical outputs;
- point predictions without calibrated uncertainty;
- explanations that are not meaningful to domain experts.

Quick Examples of AI Failure Modes

Failure mode	**Example**
Changed geometry	A detector surrogate trained on one calorimeter layout fails on a modified detector geometry.
Unseen operating condition	A reactor surrogate performs well in nominal regimes but fails under a transient condition.
Rare event	A weather model underestimates an extreme local event because similar cases were scarce in training.
Noisy observations	Measurement noise causes unstable predictions or wrong uncertainty estimates.
Extrapolative regime	A model is queried outside the parameter range used during training.
Non-physical output	A prediction violates conservation laws, boundary conditions, or known invariants.

The Core Trade-off

Physical simulators

- expensive;
- interpretable;
- domain grounded;
- based on explicit assumptions;
- usually easier to validate physically.

Data-driven models

- fast;
- flexible;
- reusable across tasks;
- potentially opaque;
- possibly unstable or invalid outside the training distribution.

Main challenge

Make it trustworthy enough for scientific use.

Six Recurring Trustworthiness Requirements

Trustworthy AI-assisted physical models must address:

1. **Limited transfer across configurations**
2. **Physical inconsistency**
3. **Noisy and low-data adaptation**
4. **Weak uncertainty awareness**
5. **Limited interpretability**
6. **High operational cost**

Scientific Foundation Models

A scientific foundation model is a reusable AI model trained to capture structure across scientific data, simulations, or physical tasks.

It differs from a narrow surrogate because it is not designed only for one fixed configuration.

Foundation-model idea

Train once on broad or heterogeneous data, then adapt to multiple downstream tasks.

TURING Models

TURING models are scientific foundation and task-specific models designed for complex physical systems.

Their distinguishing characteristics are that they combine:

- physical consistency;
- transfer across configurations and regimes;
- low-shot and parameter-efficient adaptation;
- robustness against noise, perturbations, and adversarial effects;
- uncertainty quantification;
- explainability for domain experts;
- efficient inference and deployment awareness.

TURING Models: Trustworthiness Challenges

Challenge	TURING model-level response
Limited transfer	Foundation-model pre-training, meta-learning, low-shot adaptation
Physical inconsistency	Physics-informed learning, latent dynamics, physical constraints, output correction
Noisy and low-data adaptation	Robust fine-tuning, parameter-efficient adaptation, adversarially robust training
Weak uncertainty awareness	Epistemic and aleatoric uncertainty estimation
Limited interpretability	Explainability and diagnostics integrated into inference
High operational cost	Optimisation, early exits, compression, compilation, efficient inference

The TURING Framework

The **TURING framework** complements the models with the services needed to make them operational.

It provides:

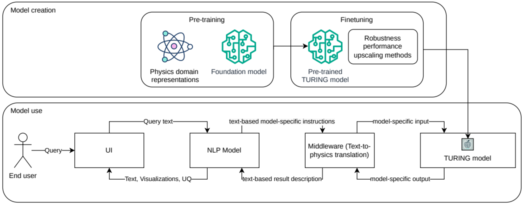
- data access;
- model storage;
- experiment tracking;
- training and inference pipelines;
- orchestration;
- resource management;
- user interaction.

Key point

TURING models are the scientific AI artefacts; the TURING framework is the environment that makes them trainable, adaptable, deployable, and usable.

Conceptual flow

1. White-box simulations provide trusted data and physical constraints.
2. TURING models learn reusable and task-specific capabilities.
3. The TURING framework supports training, adaptation, deployment, and querying.
4. Users receive predictions enriched with uncertainty and explanations.



Implementation Approach: White-Box Analysis

White-box analysis is the bridge between physical modelling and AI model design.

It identifies:

- where AI can be useful;
- what physical properties must be preserved;
- how model outputs should be evaluated;
- which variables, inputs, and outputs matter;
- which solver assumptions and boundary conditions constrain the problem;
- which parts of a workflow can be approximated by data-driven methods.

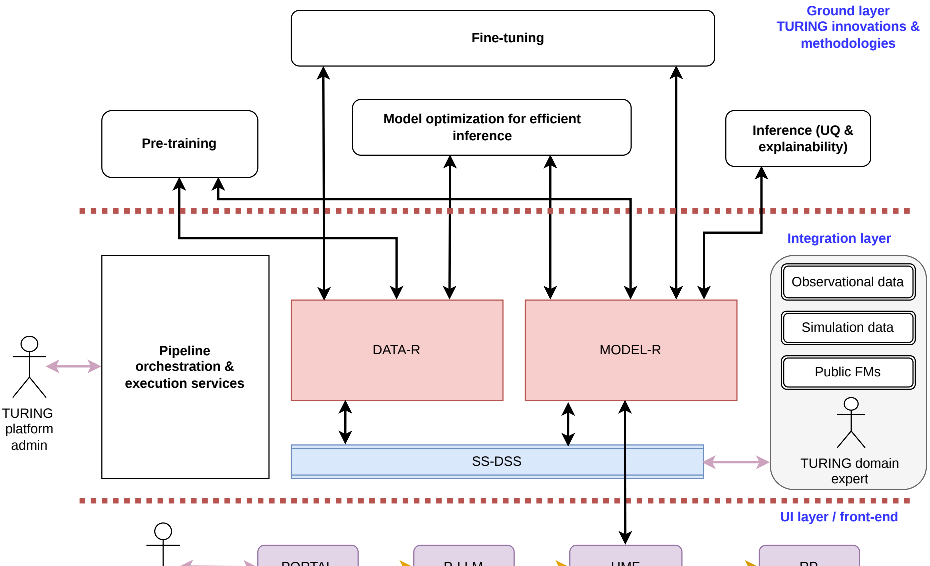
Purpose

AI models are designed around the structure, constraints, and limitations of the original physical models.

TURING Response to Trustworthiness Challenges

Trustworthiness challenges	TURING response
Limited transfer across configurations	Foundation models, meta-learning, low-shot adaptation
Physical inconsistency	Physics-informed learning, latent dynamics, output correction
Noisy and low-data adaptation	Robust fine-tuning, parameter-efficient adaptation, adversarial robustness
Weak uncertainty awareness	Epistemic and aleatoric uncertainty quantification
Limited interpretability	Explainability integrated into inference pipelines
High operational cost	Efficient inference, early exit, compression, compilation, orchestration

Design: TURING Platform Architecture



Architecture: Three Layers

Ground layer

Scientific AI methods and TURING innovations: pre-training, fine-tuning, model optimisation, and inference-time uncertainty and explainability.

Integration layer

Data, model, storage, orchestration, execution, and tracking services connecting the scientific components.

UI layer / front-end

User-facing access to operational TURING models through a portal, language-model interface, model forwarding, and response presentation.

Ground Layer: Scientific AI Capabilities

The ground layer contains four functional groups.

- **Pre-training**: produces TURING foundation models from observational data, simulation data, public benchmarks, or public foundation models.
- **Fine-tuning**: adapts pre-trained models to downstream tasks, new configurations, or specific operating regimes.
- **Model optimisation**: improves inference efficiency through early-exit mechanisms, compression, or hardware-specific compilation.
- **Inference with UQ and explainability**: provides uncertainty estimation, output stabilisation, quality assessment, and explanation.

Connection to the gaps

These groups map directly to the transfer, physical-consistency, uncertainty, explainability, and efficiency gaps.

Evaluation Through Use-Case Scenarios

The TURING framework and models are evaluated through proof-of-concept scenarios.

These scenarios define:

- the physical modelling baseline;
- the data and simulation workflow;
- the trustworthiness requirements;
- the relevant uncertainty and explainability needs;
- the computational and deployment constraints.

Evaluation principle

Each use case tests whether AI-assisted modelling can be fast, physically grounded, uncertainty-aware, interpretable, and operationally useful.

Main Takeaways

1. Physical models are scientifically trusted, but often computationally expensive.
2. AI models can accelerate or complement physical simulation workflows.
3. This introduces trustworthiness challenges: transfer, physical consistency, low-data adaptation, uncertainty, interpretability, and cost.
4. TURING models specialise scientific foundation models for complex physical systems.
5. The TURING framework and platform make these models operational through repositories, orchestration, execution services, and user interaction.
6. Evaluation is guided by realistic use cases in high-energy physics, nuclear engineering, severe-accident analysis, criticality safety, and meteorology.

Thank You!

Questions?

- Email: info@lists.turing-project.eu
- Website: https://turing-project.eu

This work has been supported by the TURING project, funded by the European Union under the Horizon Europe programme, Grant Agreement No. 101215032. Views and opinions expressed are however those of the authors only and

do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.