From Simulation to Prediction: Data-Driven Modeling of Fluid Dynamics Using Artificial Intelligence

Debabrata Pruseth
AI Architect & Applied AI Researcher
Singapore

Author Note

This article is a research-style companion version of the author's blog post "From Simulation to Prediction: Learning Fluid Dynamics with AI" and the associated GitHub project "Physics-simulation-of-Fluid-dynamics-using-WALRUS"

Abstract

Computational Fluid Dynamics has traditionally relied on numerical methods to approximate the governing equations of fluid motion. These approaches remain foundational in engineering, atmospheric science, aerospace research, energy systems, materials science, and many other domains where the evolution of physical systems must be understood over time. However, traditional simulation approaches can become computationally demanding when physical systems involve high spatial resolution, long time horizons, turbulence, nonlinear interactions, or repeated scenario testing. Recent progress in scientific machine learning has introduced an alternative paradigm: instead of solving the governing equations explicitly for every new scenario, machine learning models can be trained to learn spatiotemporal patterns from physical data and generate predictions of future system states.

This paper presents a research-style study of a data-driven fluid prediction experiment using WALRUS, a cross-domain physics foundation model developed for continuum dynamics. The author implemented a workflow that uses WALRUS to predict the temporal evolution of Rayleigh–Bénard convection, a canonical heated-fluid system in which a fluid layer is heated from below and cooled from above. The implementation uses a structured physical field as input rather than a conventional image or video. The model receives a current state of the system and generates future states through autoregressive rollout, where each predicted state is recursively fed back into the model to produce subsequent predictions.

The study is grounded in the author's blog article and GitHub notebook implementation. The blog frames the experiment as a shift from equation-first simulation toward data-driven prediction, while the GitHub notebook documents a practical implementation using Google Colab, PyTorch, WALRUS model checkpoints, and a Rayleigh–Bénard dataset. The paper discusses the methodology, model pipeline, qualitative observations, limitations, and future research directions. It also positions the experiment within the broader scientific machine learning landscape, including physics-informed neural networks, Fourier Neural Operators, machine-learning-accelerated CFD, and foundation models for physical systems.

The findings suggest that AI-based models can provide a useful complementary approach to traditional fluid simulation by learning approximate physical dynamics from data. However, the study also highlights important limitations, including autoregressive error accumulation, reduced interpretability, lack of explicit conservation-law enforcement, and the need for quantitative validation against numerical solvers.

---

1. Introduction

Fluid dynamics is one of the most important and mathematically challenging areas of physical simulation. It studies how fluids move, interact, mix, transfer heat, and respond to forces. These processes are central to many real-world systems: airflow over aircraft wings, ocean currents, blood flow, industrial mixing, climate modeling, weather prediction, combustion, cooling systems, and astrophysical phenomena. Despite its broad applicability, fluid dynamics remains difficult to simulate because fluid motion is often nonlinear, multiscale, and sensitive to small changes in initial conditions.

Traditional Computational Fluid Dynamics uses numerical methods to approximate the governing equations of fluid motion. These equations describe how quantities such as velocity, pressure, density, viscosity, and temperature evolve through space and time. In many cases, the governing equations are based on conservation principles: conservation of mass, conservation of momentum, and conservation of energy. The Navier–Stokes equations are the most widely recognized mathematical formulation for viscous fluid flow. In simplified conceptual form, they describe how a fluid's velocity changes under the influence of pressure gradients, viscous forces, external forces, and nonlinear self-advection.

Although traditional CFD has produced highly reliable simulation tools, it can be computationally expensive. High-fidelity numerical solvers often require fine spatial grids and small time steps to capture important physical features. Turbulent or chaotic systems are especially demanding because small-scale structures can influence large-scale behavior. Fluid simulation is essential for domains such as weather, climate, aerodynamics, and plasma physics, but solving fluid equations at scale remains difficult because of the cost of resolving small spatiotemporal features. Their work demonstrates

how machine learning can accelerate CFD while retaining stability and generalization in some settings.

The author's project investigates this emerging direction through a practical experiment. Instead of explicitly solving the fluid equations for each time step, the project uses WALRUS, a physics foundation model, to predict how a heated fluid evolves from an initial state. The author's blog describes the central question of the project: given the current state of a system, can an AI model predict its future? The blog further frames the experiment as a movement from "formulate equations → solve numerically → obtain results" toward "observe data → learn patterns → generate predictions."

This paper extends that blog into a formal research-style account. It presents the background, related literature, methodology, experimental pipeline, observations, limitations, and future directions of the WALRUS-based fluid prediction experiment.

---

2. Research Motivation

The motivation for this study comes from a broader shift in scientific computing. Traditional numerical simulation remains indispensable, but machine learning has created new opportunities for surrogate modeling, acceleration, approximation, and discovery. In surrogate modeling, a learned model approximates the behavior of a more expensive simulator. Once trained, the surrogate may generate predictions faster than a full numerical solver, making it useful for rapid exploration, design optimization, uncertainty analysis, or interactive visualization.

Scientific machine learning is especially relevant for systems governed by partial differential equations. Instead of learning only scalar labels or simple outputs, scientific ML models often learn structured mappings between physical fields. For example, a model may learn how a temperature field, velocity field, pressure field, or density field evolves over time. These problems are more complex than ordinary supervised learning because the output is not a single class or value; it is often a full spatial field or a time-evolving sequence of fields.

The author's experiment is motivated by three related ideas.

First, physical systems often contain learnable spatiotemporal structure. A heated fluid may appear chaotic visually, but its motion is not random. It is shaped by physical laws, boundary conditions, temperature gradients, buoyancy effects, and local interactions. A sufficiently capable model may learn statistical regularities in this evolution.

Second, foundation models are expanding beyond language and vision. WALRUS is described as a transformer-based foundation model for fluid-like continuum dynamics. The WALRUS paper reports that the model was pretrained on nineteen diverse scenarios spanning astrophysics, geoscience, rheology, plasma physics, acoustics, and classical

fluids. It also states that WALRUS was designed to address challenges such as data heterogeneity, unstable long-term dynamics, varying resolutions, and different dimensionalities.

Third, practical experimentation is important for understanding how such models behave outside research papers. The author's GitHub notebook demonstrates an applied workflow: installing WALRUS, downloading the checkpoint and configuration, loading the Rayleigh–Bénard dataset, and generating a future sequence through autoregressive prediction.

---

3. Research Questions

This study is guided by the following research questions:

RQ1: Can a pretrained physics foundation model generate meaningful future-state predictions from an initial fluid state?

This question examines whether WALRUS can produce coherent predicted states for Rayleigh–Bénard convection using a structured input field.

RQ2: How effectively can autoregressive prediction preserve spatiotemporal structures during rollout?

This question focuses on the model's ability to maintain visual and physical structure over multiple predicted time steps.

RQ3: What limitations emerge when equation-based simulation is replaced or approximated by learned predictive modeling?

This question addresses known risks such as error accumulation, drift, loss of fine details, lack of explicit conservation guarantees, and limited interpretability.

---

4. Study Contributions

This study contributes the following:

1. It presents an applied implementation of WALRUS for fluid dynamics prediction using Rayleigh–Bénard convection.
2. It translates a practical blog and GitHub implementation into a formal research-style methodology.
3. It explains the autoregressive prediction pipeline used to generate future fluid states.

4. It positions WALRUS within the broader landscape of scientific machine learning, including Physics-Informed Neural Networks, Fourier Neural Operators, and machine-learning-accelerated CFD.
5. It identifies limitations and future research directions required to move from visual demonstration toward rigorous scientific validation.

---

5. Background: Computational Fluid Dynamics

Computational Fluid Dynamics is the field concerned with simulating fluid motion using computational methods. The core idea is to represent a physical system mathematically, discretize the domain into a computational grid or mesh, and solve approximate equations over time.

In classical CFD, physical systems are often modeled using the Navier–Stokes equations.

This equation captures several important physical mechanisms. The time derivative describes how velocity changes over time. The nonlinear advection term describes how the fluid transports its own momentum. The pressure gradient drives motion from high-pressure to low-pressure regions. The viscous term models internal friction and diffusion of momentum. External forces may include gravity, electromagnetic forces, or other domain-specific forces.

The challenge is that these equations are generally not solved analytically for complex real-world systems. Instead, numerical solvers approximate them using methods such as:

- Finite Difference Method,
- Finite Volume Method,
- Finite Element Method,
- Spectral Methods,
- Lattice Boltzmann Methods.

Each method discretizes the continuous physical system in a different way. For example, finite difference methods approximate derivatives using differences between neighboring grid points. Finite volume methods conserve fluxes across control volumes and are widely used in engineering CFD. Finite element methods represent the domain using elements and basis functions, making them useful for complex geometries.

The computational burden arises because accurate simulation often requires small grid spacing, small time steps, and repeated solution of large systems of equations. When physical systems involve turbulence, boundary layers, shocks, or multiscale interactions, the cost increases further. This is why AI-based surrogate modeling has become attractive: if a model can learn approximate dynamics, it may generate predictions faster than a conventional solver after training.

---

6. Related Work

6.1 Physics-Informed Neural Networks

Physics-Informed Neural Networks are neural networks trained with physical constraints embedded into the loss function. Instead of relying only on data, PINNs penalize violations of known differential equations, boundary conditions, or initial conditions. This allows the model to combine data-driven learning with physical knowledge.

The advantage of PINNs is that they can encode governing equations directly into training. This can improve physical consistency and reduce the amount of labeled data required. However, PINNs can be difficult to train for stiff, turbulent, high-dimensional, or long-horizon systems. They may also struggle when the governing equations are incomplete, uncertain, or computationally expensive to evaluate.

For this study, PINNs are relevant because they represent one major pathway for bringing physics into machine learning. The author's WALRUS experiment takes a different approach: instead of explicitly encoding the governing equations into the loss during the author's implementation, it uses a pretrained foundation model to generate future states from field inputs.

6.2 Fourier Neural Operators

Fourier Neural Operators are neural operator models designed to learn mappings between function spaces. Unlike ordinary neural networks, which often map finite-dimensional inputs to finite-dimensional outputs, neural operators learn solution operators for families of partial differential equations. Li et al. describe Fourier Neural Operators as models that directly learn mappings from functional parametric dependence to PDE solutions. Their paper reports experiments on Burgers' equation, Darcy flow, and Navier–Stokes equation, and states that the Fourier Neural Operator can model turbulent flows with zero-shot super-resolution and faster inference than traditional PDE solvers.

FNOs are important in this context because they show how machine learning can approximate PDE behavior at the level of operators, not only at the level of individual states. They are commonly discussed in scientific machine learning because they offer a structured way to learn solution maps. WALRUS can be viewed as part of the broader evolution from task-specific neural PDE solvers toward larger pretrained models for physical dynamics.

6.3 Machine-Learning-Accelerated CFD

Kochkov et al. developed an approach in which deep learning improves approximations inside CFD workflows for two-dimensional turbulent flows. Their work reports that learned components can achieve accuracy comparable to baseline solvers with much finer resolution, producing significant speedups while maintaining stability during long simulations.

This is different from a purely black-box simulation replacement. Instead, it shows a hybrid approach where machine learning is inserted into scientific computing workflows to improve efficiency while retaining some structure from traditional numerical simulation. This is an important distinction. In real scientific and engineering applications, AI is often most useful not as a complete replacement for physics, but as a way to accelerate, approximate, or augment physics-based computation.

6.4 Foundation Models for Physical Systems

Foundation models have had significant impact in language and vision, but physical simulation introduces different challenges. Physical data can vary across spatial dimensions, temporal scales, resolutions, variables, and governing regimes. A model trained only on one system may not generalize well to another.

WALRUS was introduced as a cross-domain foundation model for continuum dynamics. The WALRUS paper states that the model is transformer-based and primarily developed for fluid-like continuum systems. It was pretrained on nineteen diverse scenarios across multiple physical domains and incorporates techniques such as harmonic-analysis-based stabilization, load-balanced distributed training, and compute-adaptive tokenization.

This makes WALRUS especially relevant for the author's experiment. Rather than training a fluid model from scratch, the implementation uses a pretrained model intended to work across physical domains. The study therefore examines the practical use of a physics foundation model in a specific fluid-dynamics setting.

---

7. Physical System: Rayleigh–Bénard Convection

The experiment uses Rayleigh–Bénard convection, a classical fluid-dynamics system in which a fluid layer is heated from below and cooled from above. This creates a temperature gradient. Warm fluid near the bottom becomes less dense and rises, while cooler fluid near the top becomes denser and sinks. As this process continues, organized convection patterns emerge.

The author's blog describes Rayleigh–Bénard convection as a simple stage on which complex behavior can unfold. It notes that a fluid layer heated from below and cooled from above develops instability, causing warm regions to rise and cooler ones to sink. Over time, the system organizes into dynamic convection patterns.

Rayleigh–Bénard convection is useful for this study because it has several desirable properties:

- It is physically meaningful and well studied.
- It produces rich spatiotemporal structure.
- It is visually interpretable.

- It involves nonlinear evolution.
- It provides a canonical example of emergent complexity from simple boundary conditions.

A key dimensionless quantity in this system is the Rayleigh number:

Ra = \frac{g \beta (T_h - T_c)L^3}{\nu \alpha}

where:

- g is gravitational acceleration,
- \beta is the thermal expansion coefficient,
- T_h - T_c is the temperature difference between hot and cold boundaries,
- L is the characteristic length scale,
- \nu is kinematic viscosity,
- \alpha is thermal diffusivity.

The Rayleigh number indicates the relative strength of buoyancy-driven convection compared with dissipative effects. At low Rayleigh numbers, heat transfer may remain dominated by conduction. At higher Rayleigh numbers, convection becomes stronger and more complex. The GitHub notebook output indicates use of a Rayleigh–Bénard dataset with Rayleigh 1e10, suggesting a highly dynamic convection setting in the selected dataset.

---

8. Experimental Methodology

8.1 Overall Research Design

The study follows an implementation-based research design. The author does not train a new model from scratch. Instead, the study uses an existing pretrained WALRUS model and applies it to a Rayleigh–Bénard convection prediction task. The focus is on demonstrating the pipeline, interpreting the output, and identifying implications for AI-based physical prediction.

The methodology consists of the following stages:

1. Environment setup.
2. WALRUS installation.
3. Model checkpoint and configuration download.
4. Dataset loading.
5. Initial physical state selection.
6. Autoregressive rollout.
7. Visualization of predicted fluid evolution.
8. Qualitative analysis of output behavior.
9. Identification of limitations and future evaluation needs.

8.2 Experimental Environment

The GitHub notebook documents a practical Google Colab environment. It reports Python 3.12, Linux runtime, PyTorch 2.10 with CUDA, and a Tesla T4 GPU. The notebook also includes commands to clone the PolymathicAI WALRUS repository, install dependencies, and download the WALRUS checkpoint and configuration from Hugging Face.

8.3 Model Input

The model does not receive an ordinary video. The blog clearly states that the model receives a field: numerical values at each point in a spatial grid, representing the state of the system. These values may describe temperature, motion, or other physical quantities.

This is important because it distinguishes scientific ML from standard image generation. A fluid field is not merely a picture. It is a structured representation of physical quantities. Each grid point has numerical meaning, and spatial relationships between grid points correspond to physical relationships in the system.

8.4 Autoregressive Prediction

The core prediction method is autoregressive rollout. In this approach, the model receives the current state and predicts the next state. The predicted next state is then used as the input for the following prediction. This process is repeated over multiple steps, generating a sequence of future states.

The author's blog describes this process directly: the model takes the initial state, produces a prediction for the next moment, and feeds that prediction back into the model to generate the next step. This allows the model to construct a trajectory into the future.

This design is powerful because it allows a model to generate long sequences from a single starting point. However, it also creates a known vulnerability: any error in one predicted step becomes part of the input for the next step. Over long horizons, errors may accumulate.

---

9. WALRUS Prediction Pipeline

The implementation pipeline can be described as follows.

Step 1: Environment Verification

The notebook begins by checking the Python environment, platform, PyTorch version, CUDA availability, and GPU type. This ensures that the runtime is suitable for running

model inference. The notebook output confirms that CUDA is available and identifies the GPU as Tesla T4.

Step 2: WALRUS Installation

The notebook clones the WALRUS repository from PolymathicAI and installs it in editable mode. It also installs required libraries such as Matplotlib, ImageIO, Einops, Hydra, OmegaConf, and Hugging Face Hub.

Step 3: Model Checkpoint and Configuration

The notebook creates checkpoint and configuration directories, then downloads the WALRUS model checkpoint and extended configuration file from Hugging Face. This indicates that the implementation uses released model artifacts rather than training from scratch.

Step 4: Dataset Selection

The implementation uses a Rayleigh–Bénard convection dataset. The raw notebook metadata shows a downloaded dataset path containing `rayleigh_benard_Rayleigh_1e10`, indicating use of a high-Rayleigh-number convection scenario.

Step 5: Initial State Selection

The model is given an initial physical state from the dataset. This initial state represents the current condition of the fluid system at a specific time.

Step 6: Future-State Prediction

The WALRUS model predicts future states. Each predicted state is recursively passed back into the model as input for subsequent predictions.

Step 7: Visualization

The predicted sequence is converted into a visual representation. The blog describes the output as a moving image in which patterns evolve, boundaries shift, and structures stretch and dissipate. It also explains that darker and brighter regions correspond to lower and higher field values depending on the selected variable.

---

10. Results and Observations

10.1 Qualitative Output

The output of the experiment is a learned simulation. It is not a direct numerical solution produced step-by-step by a conventional solver. It is a predicted future trajectory generated by a learned model. The blog describes the output as a moving image where each frame represents the model's prediction of the system at a future time step.

The qualitative observations are:

- The model produces coherent spatial structures.
- Fluid-like patterns evolve over time.
- Boundaries and internal structures shift across frames.
- The output appears consistent with the idea of dynamic convection.
- The sequence provides an intuitive visualization of learned physical evolution.

10.2 Interpretation of the Learned Dynamics

The author's blog argues that the model is not explicitly encoding physical laws in symbolic form. Instead, it learns relationships: how local variations influence neighboring regions, how patterns propagate through space, and how structures evolve over time. In machine learning terms, the model captures spatiotemporal dependencies. In physical terms, it approximates system dynamics.

This interpretation is important. A model such as WALRUS does not necessarily "understand" fluid dynamics in the same way a physicist understands governing equations. Rather, it learns statistical and structural regularities from data. The central scientific question is whether those learned regularities are robust enough to generalize beyond the specific distribution of training and test data.

10.3 Failure Modes

The blog identifies an important limitation: as the rollout extends further into the future, small discrepancies begin to emerge. Fine structures may blur, boundaries may soften, and subtle features can drift. The blog attributes this to the autoregressive mechanism: each prediction depends on the previous one, so errors accumulate over time.

Table 2. Observed and Expected Failure Modes

Failure Mode	Description	Likely Cause
Blurring	Fine structures become less sharp over time	Recursive averaging and model uncertainty

Failure Mode	Description	Likely Cause
Boundary Softening	Edges and transitions become less distinct	Accumulated prediction error
Drift	Predicted trajectory gradually deviates	Autoregressive compounding
Loss of Small-Scale Features	Fine physical details disappear	Limited resolution or latent compression
Physical Inconsistency	Prediction may not strictly conserve physical quantities	Lack of explicit conservation enforcement during inference

These failure modes are not unique to WALRUS. They are common in autoregressive learned simulations. Long-horizon prediction is difficult because even small one-step errors can become significant after repeated rollout.

---

11. Suggested Quantitative Evaluation Framework

The current implementation is primarily exploratory and qualitative. To strengthen the study as a formal research paper, future versions should include quantitative evaluation. Suggested metrics include:

Mean Squared Error

MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2

MSE measures the average squared difference between predicted and reference values.

Root Mean Squared Error

RMSE = \sqrt{MSE}

RMSE provides error magnitude in the same unit as the predicted field.

Mean Absolute Error

MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|

MAE is less sensitive to large outliers than MSE.

Structural Similarity Index

SSIM can compare structural similarity between predicted and reference field visualizations.

Spectral Error

Spectral analysis can evaluate whether the model preserves small-scale and large-scale spatial frequencies.

Energy Conservation Error

For systems where relevant physical quantities can be computed, energy conservation error can measure whether the model maintains physically plausible dynamics.

Divergence Error

For incompressible flow, divergence of the velocity field should remain close to zero. A divergence-based metric can evaluate physical consistency.

Rollout Stability

Rollout stability measures whether predictions remain bounded and plausible over long horizons.

Table 3. Recommended Evaluation Metrics

Metric	Purpose
MSE	Measures pointwise prediction error
RMSE	Measures error magnitude
MAE	Measures average absolute deviation
SSIM	Measures structural similarity
Spectral Error	Measures preservation of spatial frequencies
Conservation Error	Measures physical consistency
Divergence Error	Measures incompressibility consistency
Rollout Stability	Measures long-horizon behavior

12. Discussion

12.1 AI as a Complement to Classical Simulation

The experiment should not be interpreted as a claim that AI replaces physics. A more accurate interpretation is that AI expands the scientific computing toolkit. Traditional solvers remain essential because they are grounded in governing equations, numerical stability analysis, and validated physical principles. However, learned models can provide faster approximations, visual exploration, and surrogate predictions.

The author's blog states this clearly: the experiment suggests not a replacement for traditional physics, but an expansion of its toolkit.

12.2 Why Foundation Models Matter

A task-specific ML model may perform well only on the system it was trained on. A foundation model aims to learn broader representations across many systems. WALRUS is relevant because it is pretrained across nineteen scenarios spanning multiple physical domains.

If such models generalize effectively, they could reduce the need to train separate models for every new physical system. This would be significant for scientific workflows, where data generation is often expensive and domain-specific.

12.3 Practical Value of the Author's Implementation

The author's project has practical value because it demonstrates how a researcher or practitioner can run a modern physics foundation model in an accessible environment such as Google Colab. It moves the discussion from theory to implementation. The notebook shows how to install the framework, download checkpoints, use a dataset, and generate visual prediction output.

This implementation-oriented contribution is important for applied AI education. Many research papers describe models conceptually, but practitioners often need working examples that connect model architecture, code execution, and visual results.

12.4 Interpretability and Scientific Trust

A key challenge for learned physical simulation is interpretability. Numerical solvers are based on explicit equations and known assumptions. Learned models encode dynamics in model weights and latent representations. This makes it harder to determine whether a prediction is physically correct, statistically plausible, or merely visually convincing.

For scientific trust, future work should include:

- comparison with numerical baselines,

- explicit error metrics,
- physical conservation checks,
- uncertainty estimates,
- sensitivity analysis,
- out-of-distribution testing.

Without such validation, AI-generated simulation should be treated as exploratory rather than authoritative.

---

13. Limitations

This study has several limitations.

First, the implementation is exploratory. It demonstrates a prediction workflow but does not yet provide a full benchmark study.

Second, the analysis is primarily qualitative. The output is visually examined, but quantitative comparison against ground-truth solver outputs is not yet reported.

Third, the study focuses on one physical system: Rayleigh–Bénard convection. Additional systems would be required to evaluate broader generalization.

Fourth, the current paper does not verify conservation properties such as mass, momentum, or energy conservation.

Fifth, the autoregressive rollout approach is vulnerable to error accumulation. Long-horizon predictions may become less accurate even if short-term predictions appear plausible.

Sixth, the model's internal reasoning is not directly interpretable. It is difficult to determine whether the model has learned robust physical principles or dataset-specific correlations.

---

14. Future Work

Future work should extend the current implementation in several directions.

14.1 Quantitative Benchmarking

The next version should compare WALRUS predictions against reference simulation data using MSE, RMSE, MAE, SSIM, spectral error, and conservation-based metrics.

14.2 Baseline Model Comparison

The study should compare WALRUS with other scientific ML methods, including:

- Physics-Informed Neural Networks,
- Fourier Neural Operators,
- U-Net-based temporal predictors,
- ConvLSTM models,
- traditional reduced-order models,
- classical CFD solvers.

14.3 Conservation-Aware Evaluation

Future work should compute physical diagnostics such as energy, divergence, and flux consistency.

14.4 Long-Horizon Stability Testing

The model should be evaluated over different rollout lengths to determine when predictions begin to degrade.

14.5 Multi-Parameter Experiments

The experiment could be repeated across different Rayleigh numbers, boundary conditions, and initial states.

14.6 Hybrid Physics-AI Systems

A promising direction is to combine learned prediction with numerical solvers. For example, WALRUS could provide fast approximations, while a physics solver periodically corrects drift.

14.7 Uncertainty Quantification

Future models should estimate uncertainty so that users can distinguish high-confidence predictions from uncertain extrapolations.

---

15. Conclusion

This paper presented a research-style account of an applied AI experiment in fluid dynamics prediction. The author used WALRUS, a cross-domain physics foundation model, to predict the evolution of Rayleigh–Bénard convection from structured physical field data. The implementation demonstrates a data-driven alternative to traditional equation-first simulation: rather than solving the governing equations explicitly at every

step, the model learns spatiotemporal patterns and generates future states through autoregressive rollout.

The study shows that AI-based surrogate modeling can produce coherent visual predictions of fluid evolution. It also highlights the challenges of long-horizon learned simulation, including error accumulation, blurring, drift, loss of fine structures, and the need for physical validation.

The broader significance of this work lies in its practical demonstration of scientific machine learning. Foundation models such as WALRUS may become valuable tools for accelerating simulation, exploring complex systems, and supporting scientific discovery. However, their use in serious scientific or engineering contexts requires rigorous benchmarking, interpretability, uncertainty estimation, and conservation-aware evaluation.

The author's implementation is therefore best understood as a small but meaningful step toward a larger research direction: using artificial intelligence not merely to generate text or images, but to learn and predict the behavior of physical systems.

References

Pruseth, D. (2026). Rethinking Physics Simulation: Using AI to Predict Fluid Dynamics. Blog article.

Pruseth, D. (2026). Physics Simulation of Fluid Dynamics using WALRUS. GitHub notebook implementation.

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., & Hoyer, S. (2021). Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences, 118*(21), e2101784118.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2020). Fourier Neural Operator for Parametric Partial Differential Equations. arXiv:2010.08895.

PolymathicAI. (2025). WALRUS: A cross-domain foundation model for continuum dynamics.

Suggested Citation

Pruseth, D. (2026). *From Simulation to Prediction: Data-Driven Modeling of Fluid Dynamics Using Artificial Intelligence*.