Can machine learning replace numerical models for partial differential equations?

Christian Lessig
Otto-von-Guericke-Universitaet Magdeburg
Atmospheric dynamics

Coarse scale gas dynamics described by primitive equations (Navier-Stokes eqs.)

\[
\frac{dv}{dt} = -\left(\frac{1}{\rho}\right) \nabla p - g\left(\frac{r}{r}\right) + \left(\frac{1}{\rho}\right) \left[ \nabla \cdot \left( \mu \nabla v \right) + \nabla \left( \lambda \nabla \cdot v \right) \right]
\]

\[
c_v \frac{dT}{dt} + p \frac{d\alpha}{dt} = q + f
\]

\[
\frac{d\rho}{dt} + \rho \nabla \cdot v = 0
\]

\[
p = nT
\]
Atmospheric dynamics

Coarse scale gas dynamics described by primitive equations (Navier-Stokes eqs.)

\[
\frac{dv}{dt} = -\left(\frac{1}{\rho}\right)\nabla p - g\left(\frac{r}{r}\right) + \left(\frac{1}{\rho}\right) \left[ \nabla \cdot (\mu \nabla v) + \nabla (\lambda \nabla \cdot v) \right]
\]

\[
c_v \frac{dT}{dt} + p \frac{d\alpha}{dt} = q + f
\]

\[
\frac{d\rho}{dt} + \rho \nabla \cdot v = 0
\]

\[
p = nT
\]

Needs to be complemented with equations describing radiative transfer, cloud formation, interaction with soil, ocean, biosphere, ...
Atmospheric dynamics

Numerical solution:
- Primitive equation discretized using standard approaches (spectral, finite volume, finite elements)
- “Parametrizations” to model large range of other phenomena
- Two closure problems:
  - Turbulence of gas phase
  - Many phenomena do not even have equations or models are far too expensive in a global simulation
- Global simulation has billions of DOF
  - Exascale computing
Atmospheric dynamics

High societal relevance:
● Weather forecasting
● Climate projections
● Significant government and private investments in modeling and simulation
Atmospheric dynamics

High societal relevance:
- Weather forecasting
- Climate projections
- Significant government and private investments in modeling and simulation

Which role can machine learning play?
The plan

ECMWF machine learning roadmap
The plan

ECMWF machine learning roadmap

Two applications of machine learning speed up conventional modelling
How it is going

\[ L_2 [K] \]

\[ \text{temperature, 850 hPa} \]
How it is going

$L_2 [K]$ temperature, 850 hPa

days

IFS-HRES
How it is going

temperature, 850 hPa

L₂ [K] vs. days

- IFS-HRES
- GraphCast
How it is going

3-5 years of classical development time

IFS-HRES
GraphCast

L₂ [K] vs. days

temperature, 850 hPa
How it is going
How it is going
How it is going

Machine learning methods substantially outperform classical numerical schemes.
How it is going

19th WMO congress (last weekend)
What enables GraphCast, PanguWeather, …?
What enables GraphCast, PanguWeather, …?

All purely data-driven

- Trained on ERA5 reanalysis: post-processed observations
- Very large dataset (6 PB) readily amenable to machine learning
What enables GraphCast, PanguWeather, …?

All purely data-driven
- Trained on ERA5 reanalysis: post-processed observations
- Very large dataset (6 PB) readily amenable to machine learning
What enables GraphCast, PanguWeather, …?

All purely data-driven
● Trained on ERA5 reanalysis: post-processed observations
● Very large dataset (6 PB) readily amenable to machine learning

All use state-of-the-art machine learning models
● Transformer, graph neural networks, neural differential operators
● But all different ones
What enables GraphCast, PanguWeather, …?

All purely data-driven
- Trained on ERA5 reanalysis: post-processed observations
- Very large dataset (6 PB) readily amenable to machine learning

All use state-of-the-art machine learning models
- Transformer, graph neural networks, neural differential operators
- But all different ones

All by big industry labs
- Weather forecasting has clear metrics/benchmarks for comparison
- High societal impact
Dynamics are not really described by primitive equations
Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p(y|x; \alpha)$$
Theoretical formulation

- Atmosphere as stochastic dynamical system:
  
  \[ p(y|x; \alpha) \]

  \[
  \begin{align*}
  \text{future state} & \quad \text{auxiliary information} \\
  \text{initial condition} &
  \end{align*}
  \]

  \[
  x = (\zeta, \mu, T, z, \cdots)
  \]

  \[
  \alpha = (t_{\text{abs}}, \text{CO}_2, \cdots)
  \]
Theoretical formulation

- Atmosphere as stochastic dynamical system:

  \[ p(y|x; \alpha) \]

  › Forecasting, downscaling, interpolation: \[ p(y|x; \alpha) \]
Theoretical formulation

- Atmosphere as stochastic dynamical system:
  \[ p(y|x; \alpha) \]

  - Forecasting, downscaling, interpolation: \( p(y|x; \alpha) \)
  - Counterfactuals: \( p(y|x; \hat{\alpha}) \)
Theoretical formulation

- Atmosphere as stochastic dynamical system:

\[ p(y|x; \alpha) \]

- Forecasting, downscaling, interpolation: \( p(y|x; \alpha) \)

- Counterfactuals: \( p(y|x; \hat{\alpha}) \)

- Climate: \( p_\alpha(y) = \int p(y|x; \alpha) \, p(x) \, dx \)
Theoretical formulation

- Atmosphere as stochastic dynamical system:
  \[ p(y|x; \alpha) \] highly complex, in-stationary distribution

  › Forecasting, downscaling, interpolation: \( p(y|x; \alpha) \)
  
  › Counterfactuals: \( p(y|x; \hat{\alpha}) \)
  
  › Climate: \( p_\alpha(y) = \int p(y|x; \alpha) p(x) \, dx \)
Theoretical formulation

◦ Atmosphere as stochastic dynamical system:
  \[ p(y|x; \alpha) \]

◦ Numerical statistical atmospheric model:
  \[ \tilde{p}(\tilde{y}|\tilde{x}; \tilde{\alpha}) \]
Theoretical formulation

- Atmosphere as stochastic dynamical system:
  \[ p(y|x; \alpha) \]

- Numerical statistical atmospheric model:
  \[ \tilde{p}(\tilde{y}|\tilde{x}; \tilde{\alpha}) \approx p(y|x; \alpha) \]
Theoretical formulation

- Atmosphere as stochastic dynamical system:
  \[ p(y|x; \alpha) \]

- Numerical statistical atmospheric model:
  \[ \tilde{p}(\tilde{y}|\tilde{x}; \tilde{\alpha}) \approx p(y|x; \alpha) \]

  \[ \text{approx. future state} \quad \text{approx. initial condition} \]
Theoretical formulation

- Atmosphere as stochastic dynamical system:
  \[ p(y|x; \alpha) \]

- Numerical statistical atmospheric model:
  \[ \tilde{p}(\tilde{y}|\tilde{x}; \tilde{\alpha}) \approx p(y|x; \alpha) \]
  highly complex, instationary distribution
Theoretical formulation

- Atmosphere as stochastic dynamical system:
  \[ p(y|x; \alpha) \]

- Numerical statistical atmospheric model:
  
  \[ \tilde{p}(\tilde{y}|\tilde{x}; \tilde{\alpha}) \approx p(y|x; \alpha) \]

  very large neural network

  highly complex, instationary distribution
Theoretical formulation

- Atmosphere as stochastic dynamical system:
  \[ p(y|x; \alpha) \]

- Numerical statistical atmospheric model:
  very large neural network
  \[ \tilde{p}_\theta(\tilde{y}|\tilde{x}; \tilde{\alpha}) \approx p(y|x; \alpha) \]
  highly complex, instationary distribution
AtmoRep
AtmoRep

ERA5 reanalysis

$p_\theta(y|x)$

Historical observations

large scale machine learning

1950 1970 1990 2010

Climate projections

Impact analysis

Forecasting

Downscaling
AtmoRep

ERA5 reanalysis

\[ p_\theta(y|x) \]

large transformer
with 3.5 x 10^9 parameters
AtmoRep

ERAS reanalysis

$p_\theta(y|x)$

large transformer
with $3.5 \times 10^9$ parameters

Historical observations

1950 1970 1990 2010

ERA5 reanalysis

Climate projections

Impact analysis

Forecasting

Statistical analysis

Downscaling

applications

© Christian Lessig, 2023
AtmoRep

Historical observations

ERAS reanalysis

\[ p_\theta(y|x) \]

large transformer
with \(3.5 \times 10^9\) parameters

applications

- Forecasting
- Impact analysis
- Statistical analysis
- Downscaling

scientific insight
AtmoRep network architecture

- Transformer-based network architecture
  - Scales well to very large data-sets
  - Generative model (with decoder)
  - Effective for wide range of data sets
  - Attention maps provide (physical) interpretability
What is a token?

- Key property of transformer: set of tokens (=“small” inputs) is processed simultaneously
  - Language: token is word or sub-word
  - Images: token is small image patch (e.g. 16x16 pixels)
- Attention relates tokens
What is a token?
What is a token?

token: small neighborhood in space-time
What is a token?
What is a token?

- Flatland view:
What is a token?

- Flatland view:
What is a token?

- Flatland view:
Multiformer
Training objective

- Numerical statistical atmospheric model

\[ p_{\theta}(y|x, \alpha) \]
Training objective

- Numerical statistical atmospheric model

\[ p_\theta(y|x, \alpha) \]

- Training should model spatio-temporal relationship between arbitrary state \( x \) and \( y \)
Training objective
Training objective
Training objective
Training objective

masked token model: training to predict randomly masked information
Training objective

- Numerical statistical atmospheric model:

\[ p_\theta(y|x, \alpha) \]

approximate initial state
Training objective

- Numerical statistical atmospheric model:

\[ p_\theta(y|x, \alpha) \]

approximate initial state

=> BERT: encourage robust, probabilistic model by randomly applying distortions to input

---

Training objective

Masked token model:
Training to predict randomly masked information.

- Additive noise
- Coarsened input
Training objective

divergence, ml=96
Training objective

- BERT-style masked token model
  - Mixture of masking, noise-perturbation and coarsening to learn robust, probabilistic representation
  - Masking ratio is sampled up to, e.g., 0.5, 0.75
Pre-training results

![Pre-training results graph]

- Pre-training results graph showing the normalised vorticity distribution over different training days.
Pre-training results
Pre-training results

![Graph showing vorticity results for Berlin]

- Pre-training results for Berlin.
- The graph displays the vorticity values with Ensemble Member predictions in blue, Preds in red, and Targets in blue.
- The y-axis represents the number of occurrences at different vorticity levels.
Intrinsic Capabilities
Zero shot capabilities

- Numerical statistical atmospheric model:
  \[ p_\theta(y|x, \alpha) \]

- Model directly includes important applications: forecasting, downscaling, temporal interpolation, ...
Zero shot capabilities

- Numerical statistical atmospheric model:
  \[ p_\theta(y|x, \alpha) \]

- Model directly includes important applications: forecasting, downscaling, temporal interpolation, ...

- Training with random masking contains many of these applications as special cases
Zero shot capabilities

Training task: predict randomly masked neighborhoods in space-time
Zero shot capabilities

Training task:
predict randomly masked neighborhoods in space-time
Zero shot capabilities

Forecasting

Training task:
predict randomly masked neighborhoods in space-time
Zero shot capabilities

The three settings we explore for in-context learning

**Zero-shot**
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1. Translate English to French:
2. cheese => ........................................
```

**Traditional fine-tuning (not used for GPT-3)**
The model is trained via repeated gradient updates using a large corpus of example tasks.

```
1. sea otter => loutre de mer
```

---

Zero shot capabilities

Large language model: $p_\theta(y|x)$

The three settings we explore for in-context learning

Zero-shot
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

1. **Translate English to French:**
2. **cheese =>**

| task description | prompt |

Zero-Shot (0S)
is the same as one-shot except that no demonstrations are allowed, and the model is only given a natural language instruction describing the task. This method provides maximum convenience, potential for robustness, and avoidance of spurious correlations (unless they occur very broadly across the large corpus of pre-training data), but is also the most challenging setting. In some cases it may even be difficult for humans to understand the format of the task without prior examples, so this setting is in some cases “unfairly hard”. For example, if someone is asked to “make a table of world records for the 200m dash”, this request can be ambiguous, as it may not be clear exactly what format the table should have or what should be included (and even with careful clarification, understanding precisely what is desired can be difficult). Nevertheless, for at least some settings zero-shot is closest to how humans perform tasks – for example, in the translation example in Figure 2.1, a human would likely know what to do from just the text instruction.

Figure 2.1 shows the four methods using the example of translating English to French. In this paper we focus on zero-shot, one-shot and few-shot, with the aim of comparing them not as competing alternatives, but as different problem settings which offer a varying trade-off between performance on specific benchmarks and sample efficiency. We especially highlight the few-shot results as many of them are only slightly behind state-of-the-art fine-tuned models. Ultimately, however, one-shot, or even sometimes zero-shot, seem like the fairest comparisons to human performance, and are important targets for future work.

Sections 2.1-2.3 below give details on our models, training data, and training process respectively. Section 2.4 discusses the details of how we do few-shot, one-shot, and zero-shot evaluations.

Zero shot capabilities

Training task:
predict randomly masked neighborhoods in space-time

Forecasting
Zero shot capabilities

\[ \Delta t = 1h \]

Persistence

Zero-shot

Finetuned
Zero shot capabilities

Training task:
predict randomly masked neighborhoods in space-time

Temporal interpolation
Zero shot capabilities

Δt=1h

$L_2$ vs $\Delta t$

- Persistence
- Zero-shot
- Finetuned
Zero shot capabilities

Also:
- spatial interpolation (missing data)
- downscaling
- ...

Training task:
predict randomly masked neighborhoods in space-time
Model correction

- Numerical statistical atmospheric model:

\[ p_\theta(y|x, \alpha) \]
Model correction

- Numerical statistical atmospheric model:

\[ p_\theta(y|x, \alpha) \]

approximate initial state
Model correction

- Numerical statistical atmospheric model:

\[ p_\theta(y|x, \alpha) \]

approximate initial state
(training with masking and noise, lower resolution)
Model correction
Model correction

ERA5

α

vorticity

Embedding

Self-Attention

MLP

...
Model correction

IFS

\[ \alpha \]

\[ \text{Embedding} \rightarrow \text{Self-Attention} \rightarrow \text{MLP} \rightarrow \ldots \rightarrow \text{Self-Attention} \rightarrow \text{MLP} \rightarrow \text{Vorticity} \]

\[ ? \]
Model correction

references: ERA5

vorticity, ml=137, 10/01/2020, 12:00
Model correction

reference: IFS

vorticity, ml=137, 10/01/2020, 12:00
Model correction

Spectrum

ERA5
IFS

© Christian Lessig, 2023
Model correction

network input: ERA5, 1h prediction

vorticity, ml=137, 10/01/2020, 13:00
Model correction

network input: IFS, 1h prediction

vorticity, ml=137, 10/01/2020, 13:00
Model correction

network input: ERA5

network input: IFS
Model correction

Spectrum

- ERA5
- Input ERA5
- IFS
Model correction

Spectrum

- ERA5
- Input ERA5
- IFS
- Input IFS
Counterfactuals

- Numerical statistical atmospheric model:

\[ p_\theta(y|x, \alpha) \]
Counterfactuals

- Numerical statistical atmospheric model:

\[ p_\theta(y|x, \alpha) \]

\[ \alpha = (\text{year}, \text{day}, \text{hour}, \text{ml}, \theta, \phi, \text{res}) \]
Counterfactuals

- Numerical statistical atmospheric model:

  \[ p_\theta(y|x, \alpha) \]

  \[ \alpha = (\text{year, day, hour, ml, } \theta, \phi, \text{res}) \]

- Counterfactual: keep initial conditions \( x \) but manipulate the auxiliary information \( \alpha \)
Counterfactuals
Counterfactuals

$\alpha = 2005$
Counterfactuals

\[ \hat{\alpha} = 2018 \]
Counterfactuals

$p_{\theta}(y|x, \alpha)$

1h prediction

29/08/2005, 10:00
Counterfactuals

\[ p_\theta(y|x, \alpha) \]

1h prediction

29/08/2019, 10:00
Counterfactuals

\[ p_\theta(y|x, \hat{\alpha}) \]
Can machine learning replace numerical models for partial differential equations?
Can machine learning replace numerical PDEs?

Perhaps that is not the question: how can we solve problems with machine learning?
Can machine learning replace numerical PDEs?

Perhaps that is not the question: how can we solve problems with machine learning?

- Think of machine learning as complementary to existing approaches
  - Consider problems where existing approaches are sub-optimal, e.g. closure problems
- Integration of observational/measurement data
  - Growing exponentially in many domains
- Start machine learning with appropriate mathematical modeling
  - Find problems where machine learning fits rather than vice versa
- Develop domain specific but task independent neural networks
Questions
Medium range forecasting

- How to do global forecasts with a local model?
Medium range forecasting

- How to do global forecasts with a local model?
Medium range forecasting

- How to do global forecasts with a local model?
Medium range forecasting

- How to do global forecasts with a local model?
Medium range forecasting

- How to do global forecasts with a local model?
Medium range forecasting

- How to do global forecasts with a local model?

No exchange of information
Medium range forecasting

- How to do global forecasts with a local model?
Medium range forecasting

- How to do global forecasts with a local model?
Medium range forecasting

- How to do global forecasts with a local model?
Medium range forecasting

Video
Medium range forecasting

![Graph showing RMSE vs time step]

- sub-iterations = 0
Downscaling

- Generate higher-resolution field from coarser input
Downscaling
Downscaling
Bias correction

![Diagram of model architecture](image)
Bias correction
Outlook
Scaling

- Scale network to more levels, fields, 1950-...
- Include longer-range effects by training on SST, soil moisture
- Train across multiple steps to better capture long-range interactions
Scaling

- Scale network to more levels, fields, 1950-...
- Include longer-range effects by training on SST, soil moisture
- Train across multiple steps to better capture long-range interactions

How large does a numerical stochastic atmospheric model have to be to absorb ERA5?
Training on observations

- Neural networks are models that work well on heterogeneous and noisy data
- Fine-tune a pre-trained model with observations instead of training from scratch
Training on observations

- Neural networks are models that work well on heterogeneous and noisy data
- Fine-tune a pre-trained model with observations instead of training from scratch

Can one continuous update a model with observations? How to handle and propagate uncertainties?
GPT^x

- LLMs can generate runnable code (python, C++, ...)

GPT²

- LLMs can generate runnable code (python, C++, ...)

=> Model can autonomously gather new data and optimize its input

=> Model can fine-tune itself
GPT$^x$

- LLMs can generate runnable code (python, C++, ...)

=> Model can autonomously gather new data and optimize its input

=> Model can fine-tune itself

What is the equivalent for a stochastic atmospheric model?
Summary
AtmoRep

- Numerical statistical atmospheric model
  - Complementary to classical GCMs and ESMs
  - Represented by very large neural network
  - Very long training leads to continuous improvement
AtmoRep

- Numerical statistical atmospheric model
  - Complementary to classical GCMs and ESMs
  - Represented by very large neural network
  - Very long training leads to continuous improvement
- BERT-type training leads to versatile intrinsic capabilities
  - Forecasting, temporal interpolation, model correction, ...
  - Straightforward extension to various applications
AtmoRep

Historical observations

ERA5 reanalysis

$p_\theta(y|x)$

applications

Forecasting
Impact analysis
Statistical analysis
Downscaling

scientific insight
Model correction

\[ \Delta t = 1 \text{h} \]

Persistence ERA5
Zero-shot ERA5
Persistence IFS
Zero-shot IFS

\[ L_2 \]

96 105 114 123 137 ml
Counterfactuals

Histograms targets

[2018–2020] [1980–1982]
Counterfactuals

difference of histograms targets

Counterfactuals

histograms predictions

[2018–2020] [1980–1982]
Counterfactuals

initial conditions but

\[ \alpha_{\text{year}} = [1980, 1982] \]
Counterfactuals

initial conditions *but* \( \alpha_{\text{year}} = [1980, 1982] \)

[2018–2020]

initial conditions *but* \( \alpha_{\text{year}} = [2018, 2020] \)
Counterfactuals

Counterfactuals


Counterfactuals


Multiformer
Multiformer

Self-attention

\[ \sigma(QK^T)V \]
Multiformer

\[ \sigma(QK^T)V \]
Multiformer

\[ \sigma(QK^T)V \]
Multiformer

\[ \sigma(Q K^T) V \]
Multiformer

Self-attention

$\sigma(Q K^T) V$

© Christian Lessig, 2023
Multiformer

\[ \sigma(QK^T) V \]
Multiformer

\( \sigma \left( Q K^T \right) V \)
Multiformer

Self attention

\[ \sigma(QK^T) V \]

Cross attention

\[ \sigma(Q\zeta K^T_\mu) V_\mu \]
Multiformer

\[ \sigma(Q K^T) V \]

Self attention

\[ \sigma(Q_\zeta K_\mu^T) V_\mu \]

Cross attention
Multiformer

\[ \sigma \left( Q K^T \right) V \]

Self attention

\[ \sigma \left( Q \zeta K^T_\mu \right) V_\mu \]

Cross attention

geopotential

temperature

divergence

vorticity
Multiformer

- Plug-and-play of fields
  - Fields can be added/removed with limited (or no) computational effort
Multiformer

\[ \sigma(QK^T) V \]

Self attention

Cross attention

\[ \sigma(Q\zeta K^{T\mu}) V_\mu \]
Multiformer

Self attention

\[ \sigma(QK^T)V \]

Cross attention

\[ \sigma(Q\zeta K^T_\mu)V_\mu \]
Multiformer

◦ Plug-and-play of fields
  › Fields can be added/removed with very limited computational effort

◦ Cross-attention allows for explicit introspection of interaction between fields

◦ Different physical fields with different properties have separate latent spaces (and transformations for these)
AtmoRep data

- Vorticity
- Divergence
- Geopotential
- Temperature
AtmoRep data

Normalized spectra at 975 hPa

- vorticity
- divergence
- geopotential
- temperature

© Christian Lessig, 2023
ERA5 versus ImageNet

![Graph showing ERA5 versus ImageNet vorticity comparison. The graph plots vorticity against a parameter |ξ|. The blue line represents ImageNet, and the orange line represents vorticity.](attachment:era5_vs_imagenet.png)
ERA5 versus ImageNet

stream function $\leftrightarrow$ velocity potential $\leftrightarrow$ vector field $\leftrightarrow$ vorticity divergence
ERA5 versus ImageNet

\[\text{ImageNet} \approx \text{velocity}\]
ERA5 versus ImageNet

ImageNet

≈ Stream fct

|$\xi|$
AtmoRep data
Statistical loss

- Attention maps: