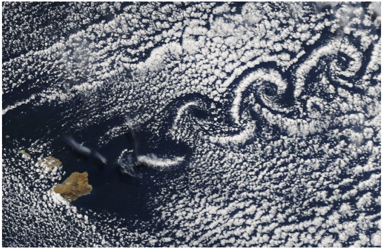
From Convolutions to Attention – Advanced Deep Learning Methods for Climate Data

Deep Learning Course for Climate Scientists

Alexander Fischer, Johannes Meuer

March 12, 2026

German Climate Computing Center (DKRZ)
Hamburg, Germany



Motivation

- Scientific ML models aim to learn complex physical processes from data
  - ▷ infilling missing data
  - ▷ super-resolution of coarse data (aka downscaling)
  - ▷ emulating medium- (weather) to long-range (climate) dynamics of physical fields
- Understanding the properties of physical fields and how they relate to model architecture is crucial for:
  - ▷ designing architectures that can capture key physical phenomena
  - ▷ improving generalization and physical consistency of predictions

Properties of Physical Fields in Scientific ML

- Many scientific ML tasks aim to emulate PDE-governed fields:
  - ▷ pressure, temperature, velocity, turbulence quantities
- **Spatial smoothness and gradients**
  - ▷ Many physical laws depend on spatial derivatives:
    $\nabla p, \quad \nabla T, \quad \nabla \cdot \boldsymbol{u}, \quad \Delta \boldsymbol{u}, \quad \nabla \times \boldsymbol{u}$
  - ▷ Accurate gradient representation is critical for stable dynamics
- **Nonlinear interactions**
  - ▷ PDEs often contain nonlinear advection terms: $\boldsymbol{u} \cdot \nabla \boldsymbol{u}$
  - ▷ Small-scale features interact with large-scale flow
- **Conservation laws**
  - ▷ Conservation of mass, momentum, energy
  - ▷ Violating these can lead to unphysical predictions and instability
- **Multi-scale structure**
  - ▷ Large coherent structures coexist with small-scale turbulence
  - ▷ Requires capturing long-range dependencies

Inductive Bias in ML Models for Physical Fields

- Structural assumptions built into the architecture
- Determines which patterns are more easy for a model to learn
  → if some bias is missing, model can only learn with enough data and capacity, but may still struggle to learn key physical relationships
- For PDE emulation, useful inductive biases include:
  - ▷ locality of interactions
  - ▷ translation equivariance
  - ▷ hierarchical spatial structure
  - ▷ global coupling across scales
- Different architectures encode these biases differently:
  - ▷ CNNs → local spatial operators
  - ▷ Vision Transformers → global interactions
  - ▷ Diffusion / Flow Matching → generative modeling of field distributions

Promoting Physical Consistency

Additional constraints can enforce or promote physical laws

- **Physics-informed loss terms**
  - ▷ PDE residual loss e.g. for Navier-Stokes momentum equation:

$$\mathcal{L}_{PDE} = ||\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nu \nabla^2 \boldsymbol{u}||^2$$

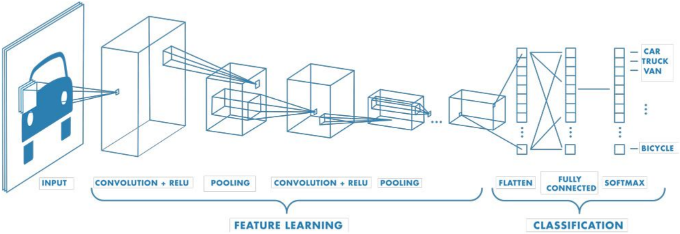
  - ▷ divergence-free constraint e.g. for incompressible flow:

$$\mathcal{L}_{div} = ||\nabla \cdot \boldsymbol{u}||^2$$

- **Spectral constraints**
  - ▷ enforce realistic turbulence spectra
- **Architectural approaches**
  - ▷ divergence-free parameterizations
  - ▷ Fourier Neural Operators

Convolutional Neural Networks (CNNs) i

- CNNs apply spatial convolutions to extract local features
- **Inductive biases**
  - ▷ locality of interactions (sliding window of convolution)
  - ▷ translation equivariance
  - ▷ hierarchical spatial feature extraction



Convolutional Neural Networks (CNNs) ii

Advantages

- efficient for grid-based data (operates directly on physical fields)
- good at capturing local operators (finite-difference stencils)
- parameter sharing improves generalization

Limitations

- limited receptive field for long-range interactions
- capturing multi-scale / global dynamics requires deep architectures

Self-Attention Mechanism

- Each token interacts with all other sequence tokens via attention
  → attention as *associative memory* mechanism
- Query, Key, Value are different linear projections of the input tokens:

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q, \quad \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K, \quad \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V$$

- Attention scores (scaled dot product, then softmax-weighted)

$$\boldsymbol{A} = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)$$

- Output is a weighted sum of values: $\boldsymbol{O} = \boldsymbol{A}\boldsymbol{V}$
- Interpretation for physical fields:
  - ▷ each image patch can interact with any other patch
  - ▷ enables modeling of long-range physical coupling

Transformer Block: MHSA and MLP Roles

A transformer block consists of 2 main components:

- Multi-Headed Self-Attention (MHSA)
- Feedforward MLP

MHSA layer

- multiple attention heads: $\mathrm{MHSA}(\boldsymbol{X}) = \mathrm{Concat}(\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_h)\,\boldsymbol{W}_O$
- each head learns different interaction patterns
- captures spatial correlations across the entire domain

MLP layer

- applied independently to each token: $\mathrm{MLP}(\boldsymbol{x}) = \boldsymbol{W}_2\,\sigma(\boldsymbol{W}_1\boldsymbol{x})$
- introduces nonlinear transformations
- mixes channel information (e.g. pressure–velocity coupling)

Vision Transformers: Tokenization of Physical Fields i

- Reminder: Transformers operate on *sequences of tokens* → need to convert physical fields into a **sequence of patch embeddings**
- For a 2D physical field (or vector field): $\boldsymbol{X} \in \mathbb{R}^{C \times W \times H}$
- Field is divided into $N = \frac{W \cdot H}{P^2}$ patches with $P \times P$ pixels each → flattened into a vector:

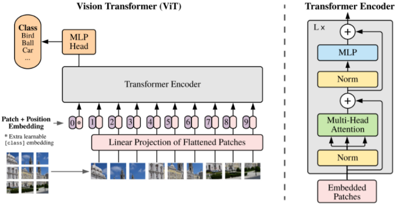
$$\boldsymbol{X} \quad \rightarrow \quad N \times \ \boldsymbol{X}_i \in \mathbb{R}^{P^2 C}$$

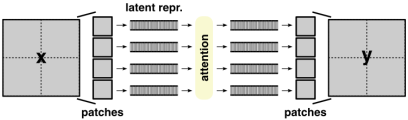
- A linear projection creates the token in some embedding space:

$$\boldsymbol{z}_i = (\boldsymbol{X}_i \boldsymbol{W} + \boldsymbol{b}) \in \mathbb{R}^D \quad \rightarrow \quad \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1^\top \\ \vdots \\ \boldsymbol{z}_N^\top \end{bmatrix} \in \mathbb{R}^{N \times D}$$

Vision Transformers: Tokenization of Physical Fields ii

- Attention is an *orderless* operation → add *positional embeddings* to encode/retain spatial information about patch locations in the grid





Advantages and Limitations of Vision Transformers

Advantages

- instant global receptive field → direct global interactions
- captures long-range interactions, e.g. teleconnections in climate data

Limitations

- weak spatial inductive bias
- requires large datasets due to non-physical embedding space
- patch tokenization may lose fine-scale gradients at the patch boundaries

Structure-Preserving Transformations in Physical Space i

- Vision Transformers project fields into an *arbitrary latent embedding space* → learn complex feature representations on these tokens:

$$\boldsymbol{X} \in \mathbb{R}^{C \times W \times H} \rightarrow n \times \ \boldsymbol{z}_i \in \mathbb{R}^{D} \rightarrow \boldsymbol{Z} \in \mathbb{R}^{N \times D}$$

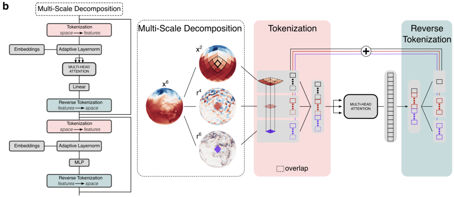
  - ▷ tokens represent abstract feature vectors rather than physical quantities
  - ▷ latent representations do a priori not correspond to physical fields → harder to learn & enforce physical structure/constraints throughout the network

- Alternative paradigm: directly learn **structure-preserving operators** that map fields to fields → each layer represents a transformation that keeps intermediate representations in the **physical state space**

Structure-Preserving Transformations in Physical Space ii

Advantages

- **Physical interpretability & consistency**
  - ▷ intermediate states remain valid fields (velocity, pressure, temperature)
  - ▷ stronger inductive bias → simplified learning of physical relationships
- **Multi-scale, residual deformations**
  - ▷ hierarchical operators can represent interactions between scales
  - ▷ resembles multigrid or spectral numerical methods


b
Multi-Scale Decomposition
Tokenization
space → features
Embeddings
Adaptive Layernorm
MULTI-HEAD ATTENTION
Linear
Reverse Tokenization
features → space
Tokenization
space → features
Embeddings
Adaptive Layernorm
MLP
Reverse Tokenization
features → space
Multi-Scale Decomposition
x^6
x^2
r^4
r^6
Tokenization
overlap
MULTI-HEAD ATTENTION
Reverse Tokenization