Transformer algorithm details

Building a machine learning climate emulator
Harvard Earth and Planetary Science 226, Eli Tziperman

The model first adds four zero rows so that the input maps can be divided evenly into 5×5 pieces. An initial convolution uses 64 different 5×5 filters (kernels); each filter combines the six input fields in one piece into a single number, so each piece is represented by 64 numbers. There are two transformer blocks, in which every piece uses information from all the pieces and then passes through two fully connected layers. A final (up)convolution assembles the processed pieces into next-month maps of SST, D20, and zonal wind stress.

1. State and model input

1.1 State, normalization, and calendar

Let $c \in \{1,2,3\}$ denote SST, D20, and zonal wind stress; let $M_{ij} \in \{0,1\}$ be the fixed Pacific mask on the 41×170 grid, $\Omega = \{(i,j) : M_{ij} = 1\}$, and $N = 5810$. From the first $8400 = 700 \times 12$ months, one scalar mean and population standard deviation are stored per field,

$$\mu_c = \frac{1}{8400N} \sum_{t=0}^{8399} \sum_{(i,j)\in\Omega} F_{tcij}, \qquad \sigma_c = \left[\frac{1}{8400N} \sum_{t=0}^{8399} \sum_{(i,j)\in\Omega} (F_{tcij} - \mu_c)^2 \right]^{1/2}. \tag{1}$$

At month t, define $S_{tcij} = (F_{tcij} - \mu_c)/\sigma_c$ on Ω and $S_{tcij} = 0$ outside Ω. Let $m_t \in \{0, \ldots, 11\}$ be the zero-based calendar month. Tensor dimensions below omit the batch axis. During deterministic inference, batch members are evaluated independently using the same learned parameters.

1.2 Patch tokens and position information

The six-channel grid input is

$$X_t[:,i,j] = \big(S_{t1ij}, S_{t2ij}, S_{t3ij}, \sin(2\pi m_t/12), \cos(2\pi m_t/12), M_{ij}\big)^\top \in \mathbb{R}^6. \tag{2}$$

Pad with four zero rows from 41 to 45 latitudes. Write the padded array as $\bar{X} \in \mathbb{R}^{6\times 45\times 170}$. A 5×5, stride-5 convolution with bias makes $9 \times 34 = 306$ nonoverlapping patch embeddings of length 64:

$$E_{rcd} = b^E_d + \sum_{q=1}^{6} \sum_{a=0}^{4} \sum_{b=0}^{4} W^E_{dqab}\, \bar{X}_{q,5r+a,5c+b}, \quad r = 0{:}8,\ c = 0{:}33,\ d = 1{:}64. \tag{3}$$

Next, flattening converts E_{rcd}, with dimensions $N_r \times N_c \times D = 9 \times 34 \times 64$, into dimensions $(N_r N_c) \times D = 306 \times 64$. To tell the transformer each patch's geographic location, the model learns a 64-number position vector for each of the 306 patches, $P \in \mathbb{R}^{306\times 64}$, and adds it to the corresponding patch representation: $T^{(0)}_{nd} = E_{nd} + P_{nd}$. Because each P_n is always added to the same patch—a fixed 5×5 set of grid points—training effectively turns it into a learned label for that geographic location.

2. Network building blocks

2.1 Self-attention

There are two blocks with separate parameters. Each has four heads of width $d_h = 16$ and a fully connected network with input and output lengths of 64 and one hidden layer of 128 neurons. No dropout is used, and the block uses pre-normalization. Given the patch matrix $T \in \mathbb{R}^{306\times 64}$ entering a block, apply its first layer normalization to obtain $\widetilde{T} = \mathrm{LN}_1(T)$. For head $h \in \{1,2,3,4\}$, let $W^h_Q, W^h_K, W^h_V \in \mathbb{R}^{16\times 64}$ be the learned query, key, and value projection matrices; $\top$ denotes transpose. The head first forms its query, key, and value matrices:

$$Q^h = \widetilde{T}(W^h_Q)^\top, \qquad K^h = \widetilde{T}(W^h_K)^\top, \qquad V^h = \widetilde{T}(W^h_V)^\top. \tag{4}$$

Here $Q^h, K^h, V^h \in \mathbb{R}^{306\times 16}$ are the query, key, and value matrices. The head then computes its attention weights and output:

$$A^h_{ij} = \frac{\exp\big(Q^h_i \cdot K^h_j/4\big)}{\sum_{k=1}^{306} \exp\big(Q^h_i \cdot K^h_k/4\big)}, \qquad H^h = A^h V^h. \tag{5}$$

Here $i,j,k \in \{1,\ldots,306\}$ index patches. The entry A^h_{ij} of the attention matrix $A^h \in \mathbb{R}^{306\times 306}$ is the weight assigned to patch j when updating patch i; k ranges over all patches in its denominator. Finally, $H^h \in \mathbb{R}^{306\times 16}$ is the output of head h. The divisor $4 = \sqrt{16}$ is exact. Every patch attends to all 306 patches, so the attention matrix has dimensions 306×306. The combined QKV projection uses a zero bias; the output projection uses the learned bias b_O:

$$Z = [H^1\ H^2\ H^3\ H^4]W_O^\top + \mathbf{1}b_O^\top, \qquad U = T + Z. \tag{6}$$

2.2 Layer normalization and feed-forward update

For any token vector $x \in \mathbb{R}^{64}$, layer normalization, for $d = 1, \ldots, 64$, is explicitly

$$\mathrm{LN}(x)_d = \gamma_d \frac{x_d - \bar{x}}{\sqrt{64^{-1}\sum_{k=1}^{64}(x_k - \bar{x})^2 + 10^{-5}}} + \beta_d, \qquad \bar{x} = 64^{-1}\sum_{k=1}^{64} x_k. \tag{7}$$

The feed-forward sublayer is also pre-normalized. With $\widetilde{U} = \mathrm{LN}_2(U)$,

$$R = \mathrm{GELU}(\widetilde{U}W_1^\top + \mathbf{1}b_1^\top)W_2^\top + \mathbf{1}b_2^\top, \qquad T_{\mathrm{out}} = U + R, \tag{8}$$

where $W_1 \in \mathbb{R}^{128\times 64}$, $W_2 \in \mathbb{R}^{64\times 128}$, and the activation function GELU operates elementwise:

$$\mathrm{GELU}(x) = x\Phi(x) = \frac{x}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right]. \tag{9}$$

3. Layer-by-layer evaluation

The two encoder blocks, final normalization, and decoder are used as follows.

3.1 Network evaluation and output

Let $\mathcal{B}_1$ and $\mathcal{B}_2$ denote the explicit calculation in Section 2, each with its own attention, feed-forward, and layer-normalization parameters:

$$T^{(1)} = \mathcal{B}_1(T^{(0)}), \qquad T^{(2)} = \mathcal{B}_2(T^{(1)}), \qquad T^{\mathrm{enc}} = \mathrm{LN}_f(T^{(2)}). \tag{10}$$

After the two blocks, apply a final layer normalization and reshape without changing order: $G_{drc} = T^{\mathrm{enc}}_{34r+c,d} \in \mathbb{R}^{64\times 9\times 34}$. The decoder is a biased 5×5 transpose convolution with stride 5. Because kernel size equals stride, reconstructed patches do not overlap; for $o = 1, 2, 3$,

$$Y^{\mathrm{pad}}_{o,5r+a,5c+b} = b^D_o + \sum_{d=1}^{64} W^D_{doab}\, G_{drc}, \quad a, b = 0{:}4. \tag{11}$$

This gives $Y^{\mathrm{pad}} \in \mathbb{R}^{3\times 45\times 170}$. The one-month standardized prediction is the leading grid,

$$Y_t = Y^{\mathrm{pad}}[:, 0{:}41, 0{:}170] \in \mathbb{R}^{3\times 41\times 170}. \tag{12}$$

There is no U-Net-like skip connection, class token, temporal attention, or autoregressive attention mask.

3.2 Dimensions and parameter accounting

stage	input	exact operation	output	parameters
patch embed	$6 \times 45 \times 170$	5×5 Conv2d, stride 5, bias	306×64	9,664
position	306×64	add learned P_{nd}	306×64	19,584
block 1	306×64	4-head MHA; $64 \to 128 \to 64$ two-layer fully connected network; norms	306×64	49,856
block 2	306×64	4-head MHA; $64 \to 128 \to 64$ two-layer fully connected network; norms	306×64	49,856
final norm	306×64	LayerNorm(64)	306×64	128
decode	$64 \times 9 \times 34$	5×5 ConvTranspose2d, stride 5, bias	$3 \times 45 \times 170$	4,803
crop	$3 \times 45 \times 170$	retain rows 0:40	$3 \times 41 \times 170$	0
			total trainable parameters	**133,891**

(13)

In each block, a 16,448-parameter cross-attention module and a 128-parameter normalization are registered but unused because cross-attention is disabled.

4. Deterministic monthly recurrence

Starting from a standardized state S_t, construct the model input described in Section 1 and evaluate the network to obtain $Y_t = f_\theta(S_t, m_t; M)$, with no output activation. During a deterministic free run, impose the ocean mask and advance the calendar:

$$S_{t+1,cij} = M_{ij}Y_{t,cij}, \qquad m_{t+1} = (m_t + 1) \bmod 12. \tag{14}$$

Rebuild the model input from the new state and calendar month and repeat the complete network. Saved ocean output is converted to physical units field by field:

$$\widehat{F}_{t+1,cij} = \sigma_c S_{t+1,cij} + \mu_c \quad ((i,j) \in \Omega). \tag{15}$$

Land is written as missing (NaN). The only state carried between forecast months is the three-channel standardized prediction; no hidden features are carried forward. The calendar advances analytically, while the mask and grid remain fixed.

5. Architecture at a glance

