# The Use of Pointwise Spectral Reinforcement in Transformers for Turbulence Generation

**Matthew Bentham, under the Supervision of Dr Christopher Chen**

Solar wind observations are primarily available as single-point time series, yet most turbulence-generation methods focus on spatial fields derived from direct numerical simulations (DNS), which are computationally costly and poorly matched to in-situ data. Rapid synthesis of realistic turbulent time series would enable new approaches to space-weather research, but their stochastic, multiscale, and long-range correlated nature makes them difficult to reproduce. Standard mean-squared-error (MSE) training suffers from exposure bias, causing long-horizon rollouts to collapse toward the mean.

Here, I introduce a Transformer-based model with *pointwise spectral reinforcement*, a physics-informed loss that penalises deviations from a prescribed spectral slope at every generation step. This approach preserves the target power-law spectrum while also capturing intermittency statistics and autocorrelation structure beyond what purely algorithmic spectral-shaping methods achieve. Results on WIND spacecraft magnetic field data show that the model generates sequences with realistic spectra, structure functions, and autocorrelations, while remaining computationally efficient for long rollouts.

This work demonstrates the feasibility of combining Transformer architectures with physics-informed constraints to generate physically faithful turbulent time series, opening new opportunities for synthetic solar-wind data generation and turbulence modelling.

## Introduction

The synthesis of realistic turbulent time series opens up numerous opportunities for applications in solar-wind research and forecasting. Methodologically, such models enable the exploration of how physics-informed constraints influence statistical fidelity, without incurring the substantial computational cost of running new simulations. They also make it possible to generate turbulence with tailored properties, allowing researchers to fill gaps in observational datasets or create synthetic backgrounds for comparison with rare solar events. For example, Magyar et al. (2024) demonstrate how synthetic time series can support plasma-frame variability analysis, but they extract the data from a higher-dimensional simulation.

Most existing machine-learning approaches to turbulence generation in plasma are trained on multi-dimensional simulation outputs, such as direct numerical simulations (DNS) of reduced magnetohydrodynamics (MHD). For instance, Fukami et al. (2018) present a convolutional autoencoder-based model trained on DNS data to generate turbulent inflow conditions with realistic spatio-temporal statistics. Kim & Lee (2019) employ generative adversarial networks (GANs) trained on DNS fields to synthesise turbulent boundary conditions that replicate key statistical features of true turbulence. Although such datasets are valuable for reconstructing spatial turbulence fields, they are less directly applicable to the one-dimensional in-situ time series obtained from spacecraft. Furthermore, their high computational cost limits both the diversity and duration of available training data. Models trained solely on simulation outputs are also inherently constrained to reproduce the physical regimes and statistical properties embedded in those simulations, potentially overlooking behaviours observed

in real measurements (Duraisamy et al., 2021).

Recent developments have introduced GAN-based generators and diffusion models for turbulence-field synthesis (e.g. Li et al., 2024), some of which can reproduce intermittency and rare-event statistics with high fidelity. However, these techniques are typically applied in spatial contexts, requiring large and densely sampled training datasets, which can be computationally intensive. In addition, their propensity to overfit to the structural patterns present in the training data can limit their ability to generalise to new flow regimes or alternative measurement formats (Duraisamy et al., 2021).

Transformer neural networks have shown strong scalability for long-sequence modelling and the flexibility to capture both local and global statistical structures in time (Vaswani et al., 2017; Zeng et al., 2023). This makes them excellent candidates for turbulence generation, where long-range correlations coexist with small-scale fluctuations. Unlike recurrent architectures, Transformers allow efficient parallel training and can maintain performance across a wide range of context lengths, a feature particularly valuable in generative settings (Child et al., 2019).

By integrating physics-informed loss functions—such as spectral reinforcement terms to enforce target power-law spectra—models can be steered toward statistically faithful outputs rather than collapsing toward the mean (Raissi et al., 2019; Li et al., 2024). With the addition of small, controlled stochastic perturbations, the model can generate physically consistent, arbitrary-length realisations from short observational seeds, preserving statistical fidelity across both short and long-time horizons.

To date, no published work combines Transformer-based architectures with physics-informed loss constraints for 1D turbulence time series generation in a heliophysical context. Existing spatial turbulence synthesis models (whether GAN, diffusion, or autoregressive Transformer-based) are typically trained on multi-dimensional simulation data and are evaluated on spatial snapshots rather than continuous 1D in-situ measurements.

By directly targeting the statistical format of spacecraft magnetic field data, the proposed approach bridges this gap, offering a method that is both physically grounded and applicable to real solar wind observations.

## Spectral Reinforcement

A model trained exclusively with a mean-squared-error (MSE) objective struggles to capture fine-scale structures and stochastic variability. In turbulent time series, the inherent randomness provides little incentive under MSE minimisation for the network to deviate from predicting either the most recent value or the global mean. As illustrated in [FIGURE], an MSE-only model, when used in an autoregressive setting, rapidly collapses toward the mean rather than sustaining realistic fluctuations. Alternative loss formulations, such as mean absolute error (MAE), have been explored, but while they can encourage greater deviation from the mean, they often introduce training instabilities and lead to rapid divergence (Zhang & Sabuncu, 2018).

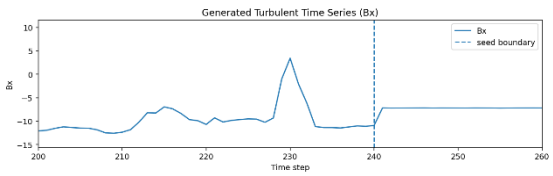



*Figure 1, an example generation attempt showcasing near immediate regression to the mean, as is typical for MSE driven methods.*

A further limitation of MSE is its bias toward fitting large-scale features before small-scale ones. Empirical studies have shown

that neural networks tend to resolve small-scale components only after achieving near-perfect reconstruction of the dominant large-scale modes, an outcome driven by the gradient descent optimisation process (Rahaman et al., 2019). In the context of stochastic turbulent signals, where the large scales cannot be perfectly predicted, this dynamic results in persistent neglect of small-scale structure, leaving MSE-trained models incapable of reproducing the full multiscale character of the data.

A way to address both of these issues is to bring in physics-informed loss constraints. By combining the MSE term with an additional loss that explicitly rewards the target behaviour, we can push the model to start capturing small-scale structure. A spectral loss, for example, can make the network sensitive to high-frequency fluctuations, which are of small amplitude in turbulence theory. At the same time, it naturally penalises long-term drift, any sustained drift in the signal would show up as excess power at low frequencies, and the loss would push against that.

The method used here takes a novel approach, particularly well-suited to iterative single-point generation. At each step, the loss function compares the spectrum of the existing signal to that of the signal with the next predicted point appended and penalises any deviation. This "spectral reinforcement" strategy has been shown to be mathematically significant, and to reliably drive the emergence of desired behaviours in both deterministic algorithms and machine-learning–based models.

Whereas previous studies have incorporated Fourier-domain features (Fukami et al., 2018) or matched structure function scaling (Johnson et al., 2023), our approach goes further by directly targeting the log–log spectral slope itself, imposing a penalty for any deviation from the prescribed inertial-range scaling law.

## Mathematical Formalism

For a context sequence of length $T$ with constant timestep $\Delta t > 0$:

$$z_n \equiv z[n] = z[0], z[1], z[2], \dots, z[T-1] \in \mathbb{R}$$

consider appending a prediction $z[T] = p$. Let the two different sequence lengths be $L \in \{T, T+1\}$, and let their DFT basis be $\omega_L = e^{-\frac{2\pi i}{L}}$, with frequency grid:

$$f_k^L = \frac{k}{L\Delta t}, k = 1, \dots, \left\lfloor \frac{L}{2} \right\rfloor.$$

The original context has DFT coefficients:

$$X_T[k] = \sum_{n=0}^{T-1} z[n]\omega_T^{kn}$$

The extended sequence has coefficients:

$$X_{T+1}[k] = \sum_{n=0}^{T} z[n]\omega_{T+1}^{kn}$$

$$= \sum_{n=0}^{T-1} z[n]\omega_{T+1}^{kn} + p\omega_{T+1}^{kn} = S_k + p\gamma_k$$

$$S_k = \sum_{n=0}^{T-1} z[n]\omega_{T+1}^{kn},$$

$$\gamma_k = \omega_{T+1}^{kn}, \text{where } |\gamma_k| = 1$$

Note the DFT is decomposed into a part dependent only on the context ($S_k$) and a part dependent on only the prediction ($p\gamma_k$).

Define their respective periodograms as:

$$P_L[k] = c_L|X_L[k]|^2, c_L = \left(\frac{2\Delta t}{L}\right)^2$$

$$P_{T+1}[k] = c_{T+1}(S_k + p\gamma_k)(S_k^* + p\gamma_k^*)$$

$$\gamma_k^* S_k = \rho_k e^{i\phi} \text{ where } \rho_k = |S_k|$$

$$P_{T+1}[k] = c_{T+1}(S_k S_k^* + p\gamma_k S_k^* + p\gamma_k^* S_k + p^2\gamma_k^*\gamma_k)$$

$$= c_{T+1}\left(\rho_k^2 + 2\rho_k \cos(\phi)\, p + p^2\right)$$

And defining the log-Periodogram:

$$y_L[k] = \log\left(P_L[k]\right)$$

Using the principal that in turbulence the PSD takes a power law form, we can link periodograms to the frequencies via the following relationship:

$$x_k^L = \log(f_k^L), \qquad y_k^L \approx \alpha_L x_k^L + \beta_L$$

$$k \in \mathcal{K}_L = \left\{1, \dots, \left\lfloor \frac{L}{2} \right\rfloor\right\}$$

Define the following statistical values:

$$N^L = \sum_{k \in \mathcal{K}_L} 1, \qquad S_x^L = \sum_{k \in \mathcal{K}_L} x_k^L,$$

$$S_{xx}^L = \sum_{k \in \mathcal{K}_L} \left(x_k^L\right)^2, \qquad D^L = N^L S_{xx}^L - \left(S_x^L\right)^2$$

This gives OLS weights:

$$a_k^L = \frac{N^L x_k^L - S_x^L}{D^L}, \qquad b_k^L = \frac{S_{xx}^L - S_x^L x_k^L}{D^L},$$
$$k \in \mathcal{K}_L$$

And therefore, fitted functionals:

$$\alpha_L = \sum_{k \in \mathcal{K}_L} a_k^L y_k^L, \qquad \beta_L = \sum_{k \in \mathcal{K}_L} b_k^L y_k^L$$

This gives the parameter shift values, the effect of adding the prediction to the sequence:

$$\Delta\alpha(p) = \sum_{k \in \mathcal{K}_{T+1}} a_k^{T+1} \log\left(c_{T+1}\left(\rho_k^2 + 2\rho_k \cos(\phi)\, p + p^2\right)\right) - \sum_{k \in \mathcal{K}_T} a_k^T \log\left(c_T |X_T[k]|^2\right)$$

$$\Delta\beta(p) = \sum_{k \in \mathcal{K}_{T+1}} b_k^{T+1} \log\left(c_{T+1}\left(\rho_k^2 + 2\rho_k \cos(\phi)\, p + p^2\right)\right) - \sum_{k \in \mathcal{K}_T} b_k^T \log\left(c_T |X_T[k]|^2\right)$$

The loss function is then defined as:

$$\mathcal{L}_{spec}(p) = \sqrt{\Delta\alpha(p)^2 + \Delta\beta(p)^2}$$

The second term in each parameter shift is independent of $p$, meaning the overall loss takes the form of a sum of logarithms of quadratic expressions. This is exactly what emerges when we evaluate the loss landscape for a given prediction, as illustrated in the figure below.

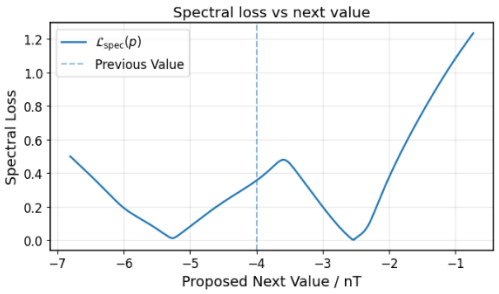



*Figure 2, the loss landscape for a predicted point using the spectral reinforcement loss. The twin minima are in line with the predicted log quadratic form.*

Differentiating with respect to $p$ shows that the gradient scales as $T^{-1/2}$, so even for long sequences the gradient remains appreciable, making this approach effective for contexts spanning hundreds of points. While the loss landscape shows distinct minima, it isn't immediately obvious that repeatedly generating points by minimising this loss would actually yield a signal with the target spectrum. To verify this, I designed an iterative Semi-Deterministic Algorithm (SDA) aimed at producing a spectrally consistent signal starting from an initial seed using only this principal.

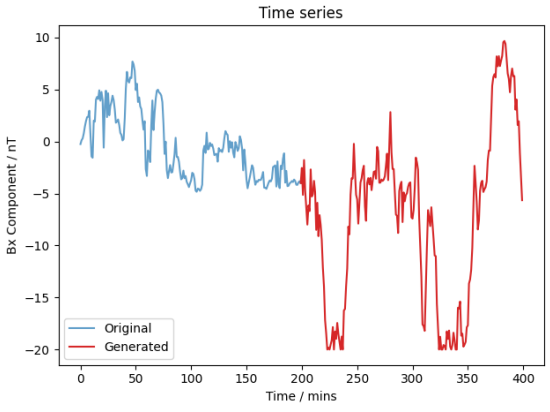



*Figure 3, an example seed being continued via pointwise spectral loss minimisation using the SDA. Values are clipped to remain within a range of 40 nT for ease of visualization, as this makes little difference to the spectrum.*

The SDA applies the loss function described above to evaluate the parameter space,

assigning each candidate point a probability weight obtained by inverting the loss and applying a SoftMax. The next point is then randomly selected from this distribution of candidates.

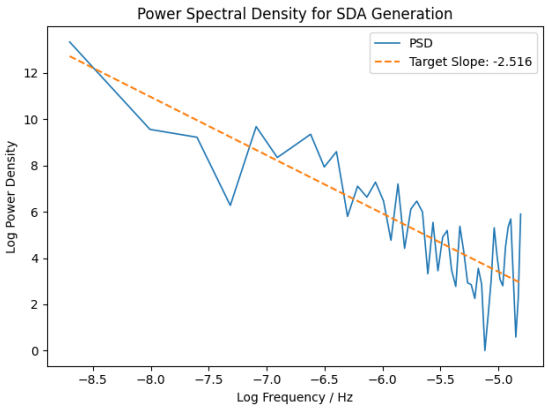



*Figure 4, the PSD of the signals generated via the SDA, with the target line of the seed PSD fitted annotated. The generated PSD shows a tight following of the target line, signalling that the SDA can be used to maintain spectra in signal generation.*

The generated spectrum displays a clear linear trend that closely follows the target, demonstrating that the spectral loss can effectively preserve the intended spectral shape across a multi-step rollout. However, despite producing a spectrally consistent signal, this approach has notable shortcomings: it fails to capture essential turbulent characteristics such as intermittency and realistic autocorrelation, consistent with findings that spectral constraints alone are insufficient for full turbulence realism (Frisch, 1995). The spectral loss alone is therefore insufficient for generating physically accurate turbulence and must be combined with additional constraints or modelling strategies.

# Use of the Spectral Loss in Transformers

## Transformer theory

Transformers are a relatively new class of deep learning architecture, first introduced for sequence modelling in natural language processing (NLP) (Vaswani et al., 2017). In contrast to recurrent neural networks (RNNs), which process inputs sequentially and are hindered by vanishing gradients and restricted temporal context (Hochreiter & Schmidhuber, 1997), transformers operate on the full sequence in parallel. This enables them to capture long-range dependencies, an area where many other sequence models struggle, and makes them well-suited to representing the multiscale, long-memory dynamics characteristic of turbulence.

The input time series is first projected into a higher-dimensional latent space, producing a sequence of input tokens $\mathbf{x}_t \in \mathbb{R}^d$. This embedding stage is learned during training and serves to re-express the raw inputs, in my case, features such as multi-lag increments, spectral descriptors, and positional encodings, in a form that makes their relationships easier to exploit. In NLP, this latent space is often interpreted as capturing the semantic meaning of words; in turbulence generation, it instead allows the model to combine physically relevant features (e.g., intermittency measures at various scales, local gradients, phase relations) into a structured representation.

At the core of the transformer is the self-attention mechanism, which enables tokens to interact with one another and dynamically weight their mutual influence (Vaswani et al., 2017). Each token is linearly projected into three learned vectors: a query $\mathbf{q}_i = \mathbf{x}_i W_Q$, a key $\mathbf{k}_i = \mathbf{x}_i W_K$, and a value $\mathbf{v}_i = \mathbf{x}_i W_V$, where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_k}$ are trainable matrices. For a given token $i$, its

contextual influence from token j is determined by the scaled dot-product attention weight:

$$\alpha_{ij} = \mathrm{softmax}\left(\frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}}\right)$$

The updated token representation is then a linear sum of itself and other tokens:

$$\mathbf{x}_i \rightarrow \mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij} \mathbf{v}_j$$

Physically, this weighting allows the model to decide which timesteps in the context carry the most predictive information for the current output. For example, two adjacent points in a turbulent time series may receive very high mutual attention if they exhibit a large increment, indicating the onset of an intermittency burst, while distant points could be emphasised to preserve long-range phase coherences.

Following attention, each token passes through a feed-forward network (a small multi-layer perceptron applied independently to each token) to allow interaction between latent dimensions, as in attention each dimension of the token vector is updated independently. These two blocks, self-attention and feed-forward, are wrapped in residual connections and layer normalisation to stabilise optimisation and are repeated multiple times to give the model sufficient capacity to capture the complex, multiscale dynamics of turbulence.

Two key adaptations are required for sequential generation. First, the attention mechanism is multi-headed: rather than computing a single set of attention weights, the model splits the embedding into $H$ subspaces and applies attention in each. This not only increases computational efficiency through parallelism but also enables increased fidelity through variety, together they form a richer, multi-perspective context.

Second, for autoregressive generation, each token must only be influenced by tokens at earlier timesteps. This is enforced through causal masking: the attention matrix is modified so that $\alpha_{ij} = 0$ whenever $i > j$, preventing any leakage of future information into the token. Without this constraint, the model could learn to exploit information it would not have in a real generation problem, biasing its predictive performance on unseen data.

# Practical Implementation

## Data Selection and Preprocessing

The limited access to high-performance computing resources constrains the design of the model. Consequently, architectural choices have been made to control computational cost during training while maintaining sufficient capacity to model the statistical structure of turbulence.

The training data are drawn from the WIND spacecraft, a dedicated solar wind monitoring mission stationed at the L1 Lagrange point for over two decades. WIND provides well-calibrated measurements of the interplanetary magnetic field vector, particle fluxes, and other plasma parameters. Its long operational history ensures a large, homogeneous dataset suitable for training a statistically rich model. The spacecraft's fixed position at L1 simplifies filtering of transient space weather phenomena, notably coronal mass ejections (CMEs).

CME intervals are excluded from the dataset for two reasons:

1. The scarcity of CME events relative to the background solar wind means the model would lack sufficient training data to learn meaningful patterns for these regimes.

2. Including CMEs would increase the complexity of the loss landscape without providing benefit to the intended goal of modelling the common, quasi-stationary solar wind.

Data is sourced from the WIND satellites Magnetic Field Investigation (NASA/SPDF CDAWeb, 2024), specifically the H0 MFI (non-processed Magnetic Field Investigation) dataset, covering the years 2004–2024 inclusive. WIND remained at L1 throughout this period, eliminating radial-distance biases. Of the 7,671 days in this range, 1,382 are removed based on two criteria:

- Overlap with days listed in the CME catalogue (Richardson & Cane ICME List, 2024).
- Presence of extreme magnetic field events, defined as |B| exceeding 50 nT.

The remaining dataset consists of 1-minute cadence vector magnetic field measurements in GSE coordinates. Each day contains 1,440 measurements, with each component on the order of a few nanotesla. The training set is constructed using 200-point context windows with a target one step ahead, yielding approximately 1,410 usable training examples per day after masking all non-physical (NaN) values. This results in a total of 7,740,129 training examples.

## Effective Sample Size and Model Capacity

The number of statistically independent examples is smaller than the raw count due to temporal correlations. An estimate is obtained using:

$$N_{eff} = \frac{N_{raw}}{\tau_{corr}}$$

where $\tau_{corr}$ is the autocorrelation timescale in samples. From autocorrelation analysis, $\tau_{corr} \approx 15$ minutes, yielding $N_{eff} \approx 2.5 \times 10^5$.

In generative modelling of turbulence, overfitting is less critical than in forecasting tasks: the aim is not to generalise to unseen inputs but to produce realistic samples that reproduce the correct statistical behaviour. This, together with the large $N_{eff}$, permits the use of a comparatively large-capacity model without significant overfitting risk.

## Model Architecture and Input Features

A standard multi-head attention Transformer is employed. A Longformer variant, with sparse, low-dimensional attention (Beltagy et al., 2020), was considered but rejected, as global attention is advantageous for capturing the long-range correlations and spectral structure characteristic of turbulence.

Each input is a 200-point context window with 9 features:
1–3. Magnetic field vector components $(B_x, B_y, B_z)$
4–6. First-order increments $(\Delta B_x, \Delta B_y, \Delta B_z)$
7–9. Log-scaled increments:

$$L(x) = \text{sign}(x) \log\left(\frac{|x|}{\epsilon} + 1\right)$$

where $\epsilon$ is a noise-floor parameter.

This transformation is smooth, continuous, and approximates a shifted logarithm for $|x| \gg \epsilon$. It provides two key advantages:

- Scale sensitivity: Encodes relative increment magnitudes over a wide dynamic range, down to the noise floor, preserving small-scale structure in heavy-tailed turbulent increment distributions.
- Noise control: $\epsilon$ sets a lower scale limit, filtering out inconsequential variations and giving the function a tighter operational range. The Nyquist-

frequency oscillation amplitude was selected for this purpose.

## Feature Normalisation

All features are globally normalised via Z-score scaling over the entire dataset. Global scaling retains the relative magnitudes of fluctuations across different samples, avoiding the need for auxiliary networks to handle magnitude scale variability. While global normalisation technically introduces a minimal data leak, the 18 parameters (mean and standard deviation per feature) are computed from the full dataset, this leakage is negligible relative to the 7 million training examples and provides no practical predictive advantage to bias the model.

## Positional Encoding

Transformers are inherently permutation-invariant and do not encode temporal order without explicit positional information. Here, fixed sinusoidal positional encodings (Vaswani et al., 2017) are added to the input features, mapping the discrete time index $t$ within the context window to a $\mathrm{d_{model}}$-dimensional vector:

$$\mathrm{PE_{t,2k}} = \sin\left(\frac{\mathrm{t}}{\lambda_\mathrm{k}}\right), \qquad \mathrm{PE_{t,2k+1}} = \cos\left(\frac{\mathrm{t}}{\lambda_\mathrm{k}}\right)$$

with wavelengths $\lambda_\mathrm{k}$ from the smallest resolvable scale (~2 samples) up to the full context length. This allows the model to infer a tokens position from the unique positional encoding signature it carries.

The exact dimensions of the model are listed below, with a short explanation of why these choices were taken:

- Sequence length - 240 minutes at 1 token per minute. This keeps computation feasible, as transformer complexity scales quadratically with sequence length. A lower sequence length leaves more capacity for higher-fidelity internal dimensions. In physical terms, 240 tokens correspond to 4 hours of WIND data; with a typical solar wind speed of 300 km/s, the largest spatial feature the model can capture is roughly $4 \times 10^6 km$.
- Input dimension - 96. Sufficient to embed all nine physical features along with a positional encoding into a rich latent space. A dimension of 64 was tested but found to significantly underfit.
- Feedforward dimension - 384. Following common transformer practice, this is set to roughly four times the input dimension to give the model full representational capacity in its intermediate layers.
- Number of heads - 8. This yields a per-head dimension of 12, large enough to average out high-frequency noise while still allowing each head to specialise in distinct feature subspaces.
- Number of layers - 8. Fewer layers led to significant underfitting; 8 was the minimal depth that maintained predictive performance.
- Dropout - 0.1. Kept intentionally low because rare intermittent bursts in the turbulence are critical to preserve. A higher dropout risks erasing these events from the learned representation. Overfitting is not a major concern in this regime; dropout here mainly serves to encourage some long-range dependency.

The model was trained for 23 epochs, stopping when the loss curves began to plateau. Training followed a curriculum learning schedule. Initially, the network was trained using pure mean absolute error (MAE) loss on the three magnetic field components. MAE was chosen over mean squared error (MSE) to encourage the model to take risks on extreme values, even when uncertain about the sign of the increment. In a stochastic setting like this, heavy penalties for incorrect predictions, as in MSE, can overly flatten the loss landscape. For example, when faced with a heavy-tailed

distribution whose peak is offset from zero, an MSE-trained model is incentivised to systematically underpredict increments. MAE mitigates this bias (Zhang & Sabuncu, 2018), albeit at the cost of increased instability. In the context of turbulence, however, such instability is far less problematic, as bursty, unpredictable events are an intrinsic part of the underlying dynamics.

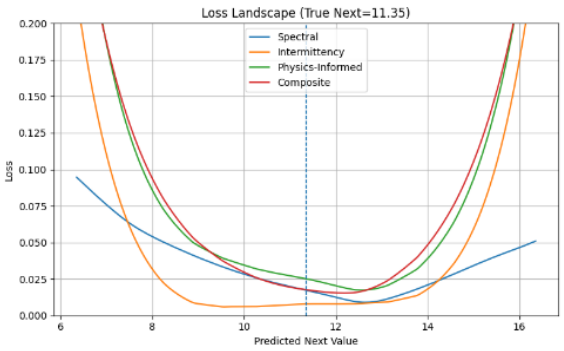



*Figure 5, the loss landscape for the next predicted point. The spectral and intermittency losses are combined in a weighted sum to produce the physics informed loss. The physics loss is then combined with a pure MSE loss using the alpha parameter to set their relative scales.*

Over the course of training, the loss weighting was gradually shifted from pure MAE toward a greater emphasis on the physics-informed component. This physics loss combined the spectral loss with an intermittency loss. The intermittency term was designed to match statistical moments of the increment distribution, improving the model's sensitivity to its heavy-tailed nature and enhancing its ability to reproduce realistic turbulent variability.

For a context sequence of length T, let the vector samples be:

$$\mathbf{B}_n \equiv \mathbf{B}[n] = \left(B_x[n], B_y[n], B_z[n]\right) \in \mathbb{R}^3, \quad n = 0,1,\dots,T-1$$

With constant timestep $\Delta t > 0$.

Consider appending a prediction $\mathbf{B}[T] = \mathbf{p} \in \mathbb{R}^3$.

Let the two sequence lengths be $L \in \{T, T+1\}$.

Denote by $\mathbf{B}^T$ the original context and by $\mathbf{B}^{T+1}$ the extended sequence.

Now fix a set of lags:

$$\Lambda = \{l_1, \dots, l_{N_\Lambda}\} \subset \mathbb{N}$$

And a set of moment orders:

$$Q = \{q_1, \dots, q_{N_\Lambda}\} \subset \mathbb{N}$$

Define the l-lag vector increments by:

$$\Delta_l \mathbf{B}_n = \boldsymbol{B}^L_{n+l} - \boldsymbol{B}^L_{n+l}$$

The q[th] order structure function is then:

$$S_q^L(l) = \frac{1}{L-1} \sum_{n=0}^{L-l-1} \|\Delta_l \mathbf{B}_n\|_2^q$$

To allow scales to be reasonably consistent without delicate weightings, taking the logs:

$$\Upsilon_q^L(l) = \log\left(S_q^L(l)\right)$$

Giving the parameter shifts for a specific **p**:

$$\Delta\Upsilon_q(l;\boldsymbol{p}) = \Upsilon_q^{T+1}(l) - \Upsilon_q^T(l)$$

Choosing specific weightings:

$$\omega_{l,q} = q l^{-\alpha}$$

So that we can define the intermittency loss:

$$\mathcal{L}_{\text{int}}(p) = \sum_{l\in\Lambda}\sum_{q\in Q} \omega_{l,q} \left|\Delta\Upsilon_q(l;\boldsymbol{p})\right|$$

Finally, the composite loss is defined by:

$$\mathcal{L}_{\text{total}}(p) = \alpha\left(\mathcal{L}_{\text{int}}(p) + \sum_i \mathcal{L}_{\text{spec},i}(p)\right) + \mathcal{L}_{\text{MSE}}(p)$$

By gradually increasing the $\alpha$ parameter, the model is allowed to first capture the large-scale structure before being pushed to resolve small-scale features through the physics loss. As shown in the physics-loss validation curve below, the loss begins to plateau after epoch 15, coinciding with $\alpha$ reaching its maximum value. This suggests

that the physics loss had converged, and the model had reached a minimum in its loss landscape.

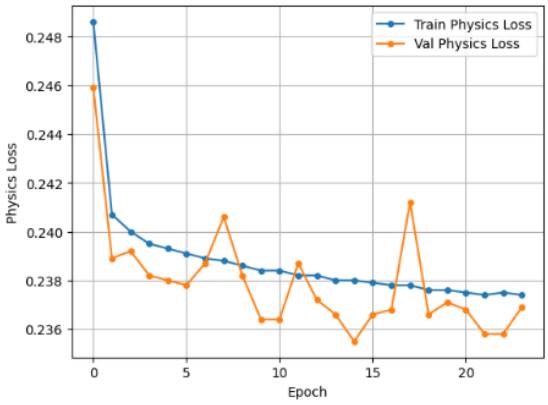



*Figure 6, the training plot for the physics-informed loss over many epochs. From epoch 15, the valuation loss has flattened out indicating convergence. The continued downward slope of the training loss indicates a small degree of overfitting.*

## Rollout

A persistent challenge for autoregressive generative models is the rapid accumulation of errors. Because each new prediction is fed back into the model as part of the input sequence, even small inaccuracies can compound rapidly, especially in systems where the next state depends strongly on the most recent value (Ranzato et al., 2015).

In my final trained model, the first-step predictions align extremely well with the target, matching both intermittency and spectrum. However, without any form of regularisation, exposure to its own outputs causes errors to grow quickly.

The spectral loss mitigates large-scale drifts in real space, and MSE training provides some stabilisation against long-term offset. Still, the model tends to produce a steepening of the power spectral density (PSD) over time. This happens because the generator focuses on preserving the current spectrum, so any gradual loss of high-frequency power, particularly vulnerable due to the sensitivity of these oscillations, will cause a steepening the slope.

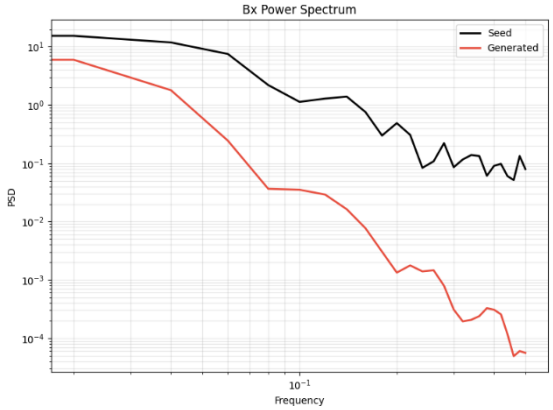



*Figure 7, the PSD of a generated sequence using the model and no noise-correction. Note that the high frequencies begin to sag due to long term drift of the spectrum.*

One way to counteract this is to inject a small amount of random noise into the output. The noise is drawn from a distribution specifically chosen so that its increments match the distribution of increments in the seed data. The magnitude must be carefully controlled: large enough to replenish lost variability at small scales, but not so large that it overwhelms the model's predictions and devolves the output into a random walk.

# Results

Much of the following analysis and the accompanying plots are based on aggregate statistics from one hundred randomly selected $(seed, generated)$ sequence pairs. Because the model is stochastic, the quality of individual realisations can vary, and this aggregation provides a more reliable picture of the model's overall behaviour.

## Visual inspection of generated vs. real time series

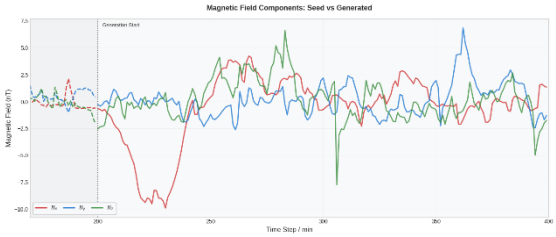
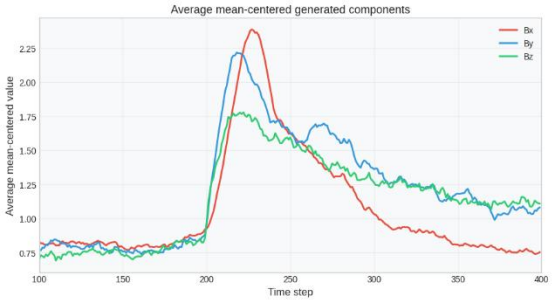



*Figure 8, generated sequences for all three components. The end of the seed sequence is shown (<200 min), as well as 200 minutes of generated sequence. Note the 'shock' as the model reacts to the noisy conditions in the Bx component.*

For the final model, I have presented here an example set of generated magnetic field time series from the dataset. The "ground truth" sequence has little direct influence on the generated output, an outcome of the fact that modelling turbulence advecting past a fixed point is inherently stochastic, unlike the spatio-temporal reconstructions explored in other studies. Consequently, no temporal alignment between truth and generated data is expected. What matters instead is that, at first glance, the turbulence appears physically plausible: it should avoid being overly smooth or excessively erratic, and it should maintain approximate stability around the seed's magnitude.




A notable feature in this example is a pronounced "shock" in the $B_x$ component, where the value departs sharply from its prior context before settling. This behaviour, which appears in many runs, likely stems from the high noise level in the seed data. Enforcing a linear spectral slope only works reliably when the seed spectrum is already close to linear; if it is not, the model makes aggressive corrections to bring the spectrum back toward the trained target. In the time domain, this manifests as an unphysical bulge. The fact that the spectrum subsequently recovers suggests that, once the desired slope is re-established, the system is reasonably stable and self-correcting. By doing a superposed analysis of the deviations from the mean, we can see the shock tendencies, also noting that Bx recovers quickest, followed by By and Bz.

## Autocorrelation analysis

One straightforward way to generate low-quality synthetic turbulence is via a random walk. When the steps are sampled from the correct distribution, this can even reproduce the target increment distribution and power spectrum. However, such a process has no phase coherence, and its autocorrelation function (ACF) reflects this lack of organisation. By contrast, our model typically produces an ACF that remains reasonably close to that of the seed data in most samples, suggesting a degree of phase structure not explicitly enforced by the physics loss. This points to the MSE component learning meaningful internal organisation directly from the data.

Examining the average ACF over many generated samples provides a clearer picture of the model's behaviour. Across the ensemble, the generated ACF never deviates from the seed's ACF by more than 0.1, indicating consistently good structural reproduction. The averaged ACF is noticeably smoother than that of the real data, likely a sign of underfitting, which is unsurprising given the model's scale. The curve also differs in shape, with the model exhibiting stronger correlations around lag 10 than the seed data. This again points to underfitting, with the model favouring short-range dependencies, which are more consistent and easier to learn, over the more variable long-range dependencies it tends to underweight.

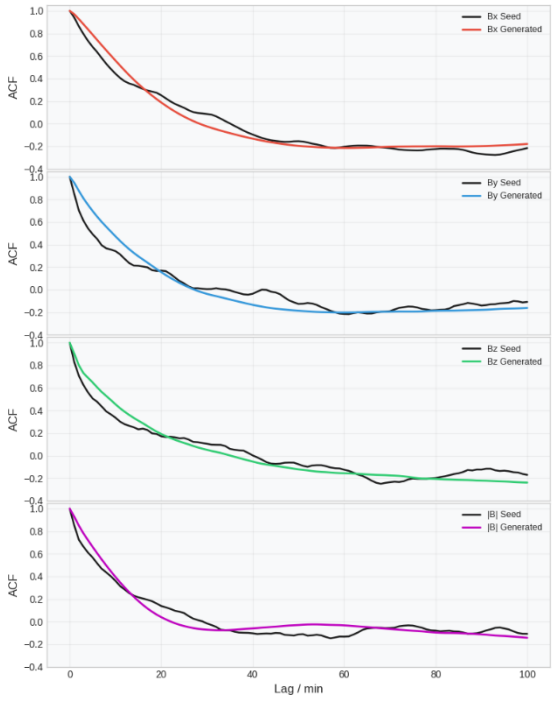



*Figure 9, autocorrelation functions taken from 1000 (seed, generated) samples. There is a strong matching between the seed and generated, but the model still consistently overweights the corrolation at short lags across all components.*

A potential concern with the noise injection step is that it might overwhelm the model's predictions, reducing the output to little more than a random walk. To assess this, I compared the model's output to a simple random walk tuned to match the correct increment distribution and spectrum. Comparing their ACFs makes the benefit of the model clear: even with noise injection, it retains much of its phase structure, whereas the random walk shows none. This highlights a key advantage of the proposed method over modified random-walk–based generators.

## Spectral conformity

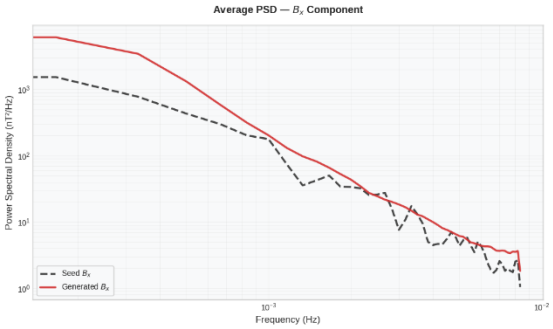



*Figure 10, the Bx PSD for the generated signal, averaged over 1000 (seed, generated) pairs. Note that the spectrum contains more power in the generated sequence across all frequency bins, but escpecially at the largest frequencies.*

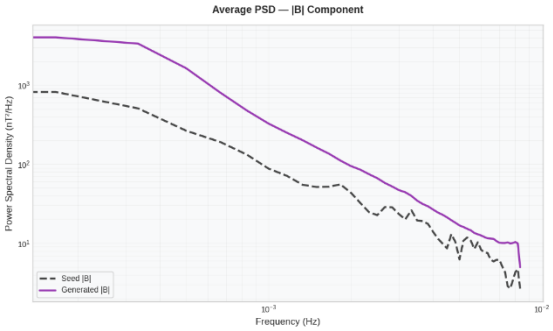



*Figure 11, the PSD for the magnetic field magnitude. This contains significantly more energy across all frequency bins, likely due to the accumulation of excess energy in each component.*

The model exhibits reasonable spectral conformity particularly at the higher frequencies. This is unsurprising as it was directly trained into the loss function, and the noise injection helps to match the target. The average spectrums are shown below. This shows a continual overestimate at low estimates, although particularly at low frequencies. This is partially due to the noise injection affecting all scales, not just the small ones, which drives the intercept of the

spectrum up, and causing a slight overestimate of power, due to the injection of energy as noise. This is particularly apparent in the magnetic field magnitude plot, where excess energy that was injected individually to all three components accumulates.

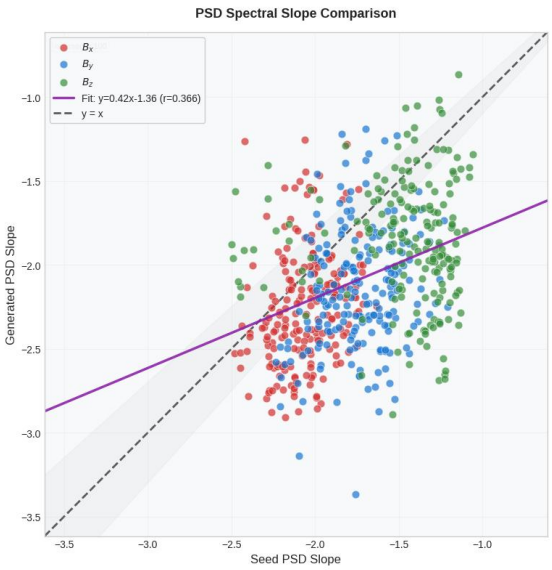



*Figure 12, a scatter plot of the seed PSD slope, and corresponding generated PSD slope. Perfect alignment would result in all points lying on the y=x line. However, the model continually produces a steeper slope than its seed, indicating a tendency for the spectrum to drift.*

The plot of seed slope against generated slope shows a very weak correlation, with an r-value of just 0.369, and a slope of 0.4. This is partially due to the random nature of the problem, but also due to the fact that the model has been trained to maintain spectrum only point to point, so errors can quickly grow. The model has also only been trained on spectra with slopes in the range of standard turbulence, which is why none of the slopes diverge outside of the 'working range' of slopes. This means that the turbulence remains realistic, even if it diverges from the seed behaviour.

## Intermittency and higher-order statistics

The model demonstrates a limited ability to reproduce intermittency. Visually, the increment distributions are noticeably too Gaussian, even after noise injection. Structure function analysis confirms this, the model consistently overestimates the magnitude of the second-order structure function and similarly overestimates the fourth-order structure function at larger lags.

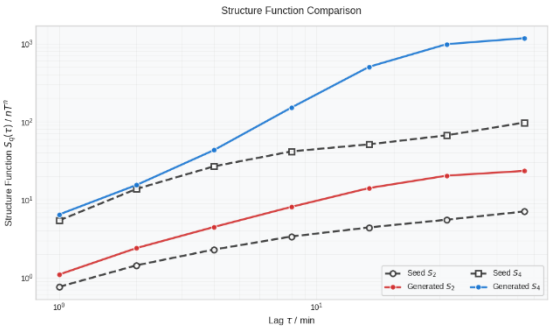



*Figure 13, a comparison of the structure functions at different lags. The model overestimates the second order structure function at all lags, but manages to recreate the fourth order well at small lags, only diverging around lag 4*

## Model Analysis

Overall, the model shows strong potential for generating realistic turbulence, producing time series with broadly accurate spectra and autocorrelation, and demonstrating an ability to learn deep structural relationships in the data. Its main shortcoming lies in reproducing correct intermittency behaviour, particularly over long ranges. A promising way forward would be to adapt the model to predict a probability distribution rather than a single value, effectively integrating the noise injection process into training. This could be achieved by training with a negative log-likelihood loss, which would both formalise and validate the noise injection procedure.

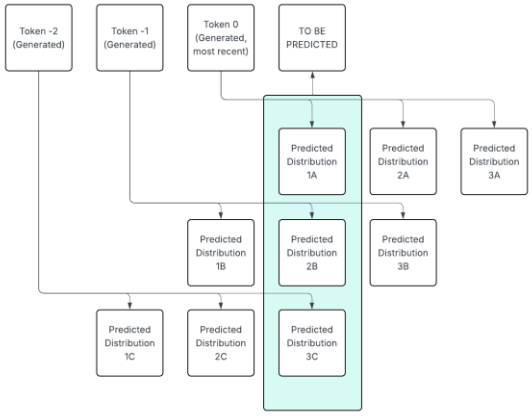



*Figure 14, a flowchart detailing the procedure for multipoint distribution rollout. Here it is illustrated with 3 tokens but could be arbitrarily more (limited only by model fidelity). This would involve combining several predictions in a Bayesian form and randomly selecting from the resulting distribution.*

Such a formulation would also enable informed multi-step rollouts, allowing for more stable enforcement of physics-based loss terms, including the possibility of explicitly encoding target intermittency distributions. Because the outputs would be full distributions, each point prediction could be formed as a Bayesian combination of multiple distributions from previous steps, potentially improving stability and physical realism over long horizons, as illustrated in figure 14.

# References


Beltagy, I., Peters, M.E. and Cohan, A., 2020. Longformer: The long-document transformer. *arXiv preprint* arXiv:2004.05150.

Child, R., Gray, S., Radford, A. and Sutskever, I., 2019. Generating long sequences with sparse transformers. *arXiv preprint* arXiv:1904.10509.

Duraisamy, K., 2021. Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence. *Physical Review Fluids*, 6(5), p.050504.

Farge, M., Schneider, K. and Kevlahan, N., 1997. Non-Gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthogonal wavelet basis. *Physics of Fluids*, 9(8), pp.2483–2501. https://doi.org/10.1063/1.869351.

Frisch, U., 1995. *Turbulence: The legacy of A. N. Kolmogorov*. Cambridge University Press.

Fukami, K., Fukagata, K. and Taira, K., 2019. Synthetic turbulent inflow generator using machine learning. *Physical Review Fluids*, 4(6), p.064603.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural Computation*, 9(8), pp.1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Johnson, R., Boubrahimi, S.F., Bahri, O. and Hamdi, S.M., 2023. Physics-Informed Neural Networks for Solar Wind Prediction. In: Rousseau, J.J. and Kapralos, B. (eds) *Pattern Recognition, Computer Vision, and Image Processing*. ICPR 2022 International Workshops and Challenges. Lecture Notes in Computer Science, vol 13645. Springer, Cham, pp.273–286. https://doi.org/10.1007/978-3-031-37731-0_21.

Kim, J. and Lee, C., 2020. Deep unsupervised learning of turbulence for inflow generation at various Reynolds numbers. *Journal of Computational Physics*, 406, p.109216.

Kiyani, K., Chapman, S.C. and Hnat, B., 2009. Intermittency, scaling, and the Fokker–Planck approach to turbulence. *Physical Review E*, 79(3), p.036109. https://doi.org/10.1103/PhysRevE.79.036109.

Li, T., Buaria, D., Pumir, A., Xu, H. and Yeung, P.K., 2024. Synthetic Lagrangian turbulence by generative diffusion models. *Nature Machine Intelligence*, 6(4), pp.393–403.

Magyar, N., Chane, E., Daldorff, L., Innocenti, M.E., Narita, Y. and Verscharen, D., 2024. Solar wind data analysis aided by synthetic modeling: A better understanding of plasma frame variations from temporal data. *Astronomy & Astrophysics*, 688, A74.

NASA Goddard Space Flight Center, Space Physics Data Facility, 2025. *Coordinated Data Analysis Web (CDAWeb)*. NASA. Available at: https://cdaweb.gsfc.nasa.gov/ [Accessed 15 August 2025].

Patil, A., Viquerat, J. and Hachem, E., 2023. Autoregressive transformers for data-driven spatiotemporal learning of turbulent flows. *APL Machine Learning*, 1(4).

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F.A., Bengio, Y. and Courville, A., 2019. On the spectral bias of neural networks. In: *International Conference on Machine Learning*. PMLR, pp.5301–5310.

Raissi, M., Perdikaris, P. and Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, pp.686–707. https://doi.org/10.1016/j.jcp.2018.10.045.

Ranzato, M., Chopra, S., Auli, M. and Zaremba, W., 2015. Sequence level training with recurrent neural networks. *arXiv preprint* arXiv:1511.06732.

Richardson, I. and Cane, H., 2024. Near-Earth interplanetary coronal mass ejections since January 1996. V3, Harvard Dataverse. https://doi.org/10.7910/DVN/C2MHTH.

Sreenivasan, K.R. and Antonia, R.A., 1997. The phenomenology of small-scale turbulence. *Annual Review of Fluid Mechanics*, 29, pp.435–472. https://doi.org/10.1146/annurev.fluid.29.1.435.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Zeng, A., Chen, M., Zhang, L., Xu, Q., Sun, J. and Zhou, Y., 2023. Are transformers effective for time series forecasting?. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9), pp.11121–11129.

Zhang, Z. and Sabuncu, M., 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems*, 31.

Zhu, X., Stevens, R.J.A.M., Shishkina, O., Verzicco, R., Grossmann, S., Lohse, D. and Xia, K.Q., 2019. Nu~Ra1/2 scaling enabled by multiscale wall roughness in Rayleigh–Bénard turbulence. *Journal of Fluid Mechanics*, 869.