# Artificial Intelligence generated surrogate model of plasma turbulence

B. Clavier[1], D. Zarzoso[1], D. del-Castilllo-Negrete[2] and E. Frénod[3]

[1]Aix-Marseille Université, CNRS, Centrale Med, M2P2 UMR 7340, Marseille

[2] Oak Ridge National Laboratory, Oak Ridge, TN 37831-8071, United States of America

[3] Université Bretagne Sud, LMBA UMR 6205, Vannes

Turbulent transport plays a major role in confinement degradation, thereby limiting the performance of current and future fusion devices. Modelling turbulent transport requires long-time simulations which are limited by the computational resources available. One way to overcome this limitation is by using surrogate models that are computationally cheaper to evaluate. In this presentation we apply Artificial Intelligence (AI) methods to construct a surrogate model of plasma edge turbulence described by the Hasegawa-Wakatani (HW) system [1] and use the model to perform fast, long-time turbulent transport computations [2,3]. The proposed surrogate model is based on the combination of a convolutional variational auto-encoder (CVAE) [4] and a deep neural network (DNN). A convolutional network is used to encode snapshots of computed turbulence states into a reduced latent space, and a DNN is trained to reproduce the time evolution of turbulence in the latent space. Once the autoencoder is trained, new turbulence sates are obtained by decoding the latent space dynamics generated by the DNN.

Essentially, the CVAE encodes high-dimensional data in a low dimension space, and reconstructs it with high fidelity, thus it consists of two parts: an encoder and a decoder. The low dimension space in which the data is encoded is called the latent space. The CVAE maps a sample of the original data (i.e. a snapshot of turbulence in our case) to a probability distribution in the latent space. Representing data as distributions instead of vectors leads to a more continuous organization of the data in latent space, and it makes the auto-encoder robust against noise. For the plasma turbulence problem of interest, we use convolutional networks for the encoder and the decoder. Convolutional networks are well-suited for handling of 2D or 3D data sets, such as turbulent snapshots of density and electric potential. Once trained, a CVAE is very efficient at generating new data. By exploring the latent space in new regions, or by moving from one known point to another, and applying the decoder, one can obtain new data that evolves continuously. However, one needs still to find a way to move in latent space that will create dynamics similar to the training data after decoding. For this purpose, we implement a DNN coupled to the latent space. The DNN learns to reproduce the evolution of

the encoded data in the latent space and is then used to generate a new evolution that can be decoded to obtain new turbulence. Fig.1 represents a schematic representation of the model.
This AI model has been tested on the relatively simple and well-known Hasegawa-Wakatani 2D fluid drift wave turbulence model

$$\partial_t n + [\phi, n] = C(\phi - n) - \kappa \partial_y \phi - \mu \nabla^2 n \quad (1)$$

$$\partial_t \Omega + [\phi, \Omega] = C(\phi - n) - \mu \nabla^2 \Omega \quad (2)$$

where $n$ is the density perturbation, $\phi$ the electric potential and $\Omega = \nabla^2 \phi$ the vorticity. The three free parameters are the adiabaticity parameter $C$, the drive of the instability $\kappa$, and the hyper-diffusion coefficient $\mu$. For our study, we set $C = 1$, $\kappa = 1$ and $\mu = 10^{-3}$.

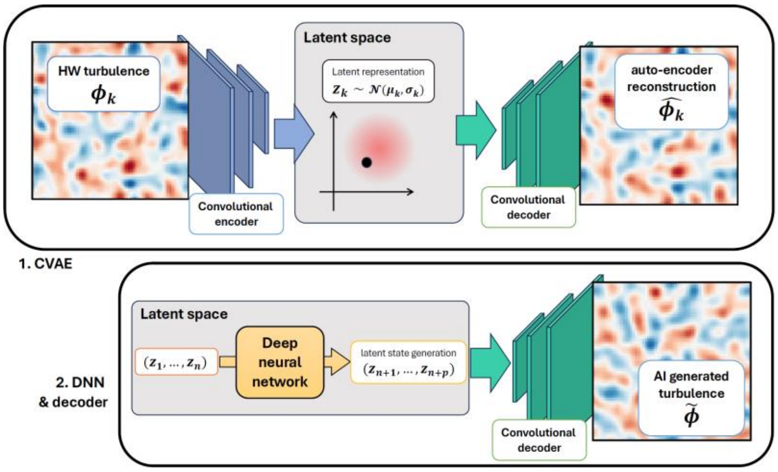


Fig 1. Schematic representation of the AI surrogate model

The Hasegawa-Wakatani set of equation was integrated with a pseudo-spectral code to generate a training set of $N_s = 8000$ saturated turbulence snapshots of the electric potential $\phi$. The CVAE was trained to encode and reconstruct the given turbulent snapshots by minimizing the loss function:

$$Loss_{CVAE} = \sum_{i=1}^{8000} w_\phi \left\| \phi_i - \widehat{\phi}_i \right\|^2 + w_{\nabla\phi} \left\| \nabla\phi_i - \nabla\widehat{\phi}_i \right\|^2 + w_{\mathbb{KL}} \mathbb{KL}(\mathcal{N}(\mu_i, \sigma_i), \mathcal{N}(0,1)) \quad (3)$$

where $\widehat{\phi}_i$ is the reconstruction of $\phi_i$ by the auto-encoder, and $\mathcal{N}(\mu_i, \sigma_i)$ is the probability distribution of the encoded potential $\phi$ in latent space. The first two terms of the loss functions penalize reconstruction errors, while the third term, called the Kullback-Leibler divergence, enforces the regularity of the latent space. The hyper-parameters $w_\phi$, $w_{\nabla\phi}$ and $w_{\mathbb{KL}}$ are weights allowing to better capture the dynamics of both the electric potential and its gradient.
After training the CVAE, an encoding of the training dataset in latent space is produced, i.e. a set of $N_s = 8000$ latent vectors $(z_1, z_2, \ldots, z_{N_s})$. We train a DNN to predict the evolution of vectors in latent space, by minimizing the loss function