Data-driven modeling of near-wall turbulence using β-variational autoencoder, transformers, and adversarial loss

Niccolò Tonioni, Mohammad Umair, Lionel Agostini, Franck Kerhervé, Laurent Cordier, Ricardo Vinuesa

Turbulence, Heat and Mass Transfer 11

Data-driven modeling of near-wall turbulence using β-variational autoencoder, transformers, and adversarial loss

N. Tonioni[1], M. Umair[2,3], L. Agostini[1], F. Kerhervé [1], L. Cordier[1] and R. Vinuesa[3]

[1]*Curiosity Team, Pprime Institute, CNRS, Université de Poitiers, ISAE-ENSMA, Bât. H2 - Bd. Marie & Pierre Curie TSA 51124, 86073, Poitiers, niccolo.tonioni@univ-poitiers.fr*

[2]*Laboratoire des Écoulements Géophysiques et Industriels (LEGI), UMR 5519, CNRS, BP 53, 38041 Grenoble CEDEX 09, France*

[3]*FLOW, Engineering Mechanics, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden*

Abstract – A machine learning framework is developed for data-driven modeling of near-wall turbulence. It employs a hybrid generative architecture, combining β-Variational Autoencoders and Generative Adversarial Networks (β-VAE-GAN), to extract dominant flow features into a low-dimensional latent space. A decoder-only Transformer models the temporal evolution within this space. The approach is validated on two-dimensional flow snapshots from direct numerical simulations of a minimal turbulent channel flow ($Re_\tau = 200$). Results demonstrate that the framework accurately predicts the latent space dynamics, preserving the system's chaotic characteristics. Furthermore, the predicted latent states enable accurate reconstruction of the main low-velocity streak (within one Lyapunov time) and, importantly, the dominant momentum transport mechanisms governing the flow.

1. Introduction

Wall-bounded turbulent flows are ubiquitous in both engineering and natural systems, playing a central role in applications ranging from energy and aerospace technologies to atmospheric and environmental sciences. Despite decades of intensive research, predicting and controlling these flows remains a formidable challenge due to their inherent complexity (Agostini *et al.* 2014). This complexity arises from the presence of a shear layer induced by the confining wall, where chaotic flow structures span a wide range of spatial and temporal scales. At higher Reynolds numbers, these scales become increasingly disparate, amplifying the nonlinear interactions between large outer-layer structures and the smaller, energetic near-wall streaks. This pronounced multiscale behavior at high Reynolds numbers presents major obstacles for both experimental characterization and numerical modeling (Marusic *et al.* 2010).

In this context, data-driven reduced-order models (ROMs) have emerged as a complementary approach to understanding near-wall turbulence by extracting dominant features and capturing their evolution in a low-dimensional manifold, simplifying the flow while preserving essential dynamics (Berkooz et al. 1993). However, traditional ROMs—often built on linear decomposition techniques like Proper Orthogonal Decomposition—are limited in their ability to capture the nonlinear interactions and rich dynamics of wall-bounded turbulent flows in a compact form. As a result, recent research has increasingly turned to machine learning, which offers powerful tools for developing nonlinear ROMs. For example, Solera-Rico *et al.* (2024) introduced a data-driven framework that combines a β-variational autoencoder (β-VAE) for feature extraction with a transformer network for temporal prediction, demonstrating strong performance on a flow past angled plates.

Building on this foundation, the present study adapts and applies this machine learning methodology as an initial step towards data-driven modeling of turbulent channel flows. A novel hybrid architecture is introduced, combining β-VAE and Generative Adversarial Network (GAN) for unsupervised feature extraction. These extracted features are then used as inputs to a transformer-based model that predicts their temporal evolution. As a proof of concept, this study analyze two-dimensional (x–z) flow snapshots at $y^+ = 14$, taken from simulations of a minimal channel at a friction Reynolds number of $Re_\tau = 200$. This computationally tractable domain is designed to contain a single dominant low-velocity streak, simplifying the flow morphology while retaining key near-wall dynamics, making it a suitable testbed for developing and evaluating data-driven modeling approaches.

2. Dataset Generation and Preparation

2.1 Direct Numerical Simulation of Minimal Channel Flow

The dataset was generated through Direct Numerical Simulations (DNS) of homogeneous plane turbulent channel flow at a friction Reynolds number $Re_\tau = u_\tau h/\nu = 200$, where u_τ is the friction velocity, h is the channel half-height, and ν is the kinematic viscosity. The size of the computational domain was selected following the work of Jiménez and Moin (1991) on the *minimal flow unit*, and is $0.6\pi h \times 2h \times 0.18\pi h$ long in the streamwise (x), wall-normal (y), and spanwise (z) direction, respectively. The periodic boundary condition was employed in the streamwise and spanwise directions, while no-slip boundary condition was imposed at the upper and lower walls ($y = \pm h$).

The simulations were performed using the highly-fidelity GPU based code `SOD2D` (Gasparino *et al.* 2024). The code is based on the Spectral-Element formulation of the Navier-Stokes equations, and utilizes Gauss-Lobatto-Legendre (GLL) nodes distribution inside each element. For the present case, the computational domain was discretized using $12 \times 36 \times 12$ elements of polynomial order 5, in the streamwise, wall-normal and spanwise directions, respectively. The element size was uniform in the streamwise and spanwise directions and stretched in the wall-normal direction following a power law distribution, yielding maximum resolutions of 8.9, 2.7, and 0.4–4.4 wall units based on ν and u_τ.

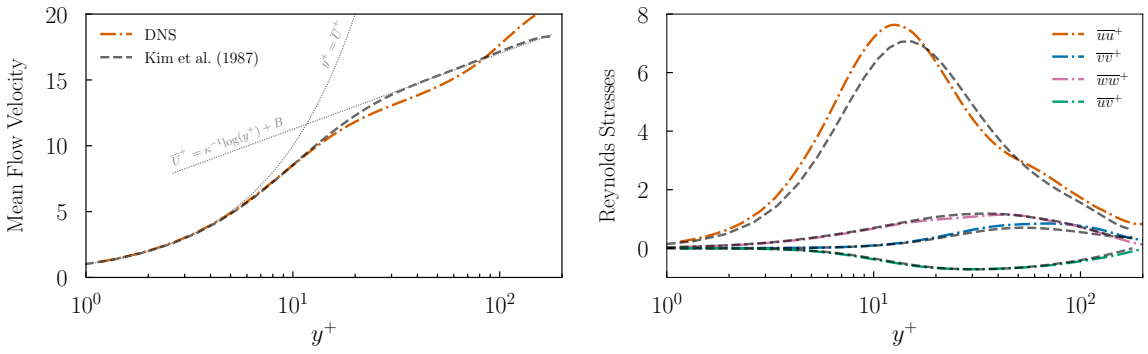

Figure 1: Streamwise mean velocity (left) and turbulent stress profiles (right) in the minimal channel flow. Colored lines show results from the present DNS at $Re_\tau = 200$, and dashed gray lines indicate reference data from full channel simulation at $Re_\tau = 180$ (Kim *et al.* 1987) . Stresses are normalized by u_τ^2, and wall-normal distance by u_τ and ν.

The solution was advanced in time using the fourth-order Runge-Kutta method with a fixed time-step of $10^{-3}\ h/U_b$, where U_b is the bulk velocity. The statistical profiles were obtained using a total of $8,800$ snapshots covering a time window of about $8,400$ inner time-units, separated by approximately 1 inner time-unit based on ν and u_τ. The resulting flow field was spectrally interpolated to a regular grid of resolution $\Delta x^+ = 8$, $\Delta z^+ = 5$, and $\Delta y^+ = 1 - 5$. To ensure the grid resolution was sufficiently refined, the mean flow profiles were compared with the reference DNS results of Kim *et al.* (1987) at $Re_\tau = 180$, shown in Figure 1.

2.2 Extraction and Preparation of 2D Flow Snapshots for Model Input

Two-dimensional (x–z) snapshots of the velocity fluctuation components (u, v, w) were extracted from the validated 3D dataset at the fixed wall-normal location $y^+ = 14$, corresponding to the peak of the streamwise Reynolds stress $\overline{uu}$ (Figure 1). A total of $8,800$ snapshots were extracted from the lower wall-adjacent plane and $7,800$ from the upper plane. The final $1,000$ snapshots from the upper plane were excluded due to intermittent relaminarization events observed near the wall at those time instants. Each snapshot, denoted $\mathbf{S}$, is a tensor of shape $(N_x, N_z, N_C = 3)$, with $N_x \times N_z = 48 \times 24$ grid points.

This set of $N_{tot} = 16,600$ snapshots was partitioned into three subsets for model development and evaluation:

- **Training set**: The first $7,050$ snapshots of each plane ($\approx 85\%$), $N_{train} = 14100$, used for optimizing model parameters.
- **Validation set**: The subsequent 880 snapshots from the lower plane and 750 from the upper-plane ($\approx 10\%$), $N_{val} = 1,630$, used for hyperparameter selection, monitoring training progress against overfitting, and implementing early stopping criteria.
- **Test set**: The final $N_{test} = 870$ snapshots from the lower plane ($\approx 5\%$), strictly reserved for evaluating the performance of the finally trained models (results presented in Section 4).

During the training phase, the data for each velocity component within each snapshot was normalized channel-wise to the range $[0, 1]$. For evaluation on the test set, model predictions were rescaled back to their original physical dimensions to allow for quantitative error assessment.

3. Reduced-Order Modeling Approach

With the dataset of near-wall flow snapshots extracted and preprocessed, a learning framework is introduced to construct a nonlinear reduced-order model of the flow dynamics.

3.1 Latent Feature Extraction via β-VAE with Adversarial Loss

To extract informative low-dimensional latent features ζ from high-dimensional flow snapshots $\mathbf{S}$ (defined in Section 2.2), a hybrid generative architecture is proposed, combining the strengths of β-Variational Autoencoders and Generative Adversarial Networks. This formulation enables simultaneous optimization of three objectives: reconstruction fidelity, disentanglement (independency) of latent variables, and statistical consistency between reconstructed and reference distributions. The foundational components of the proposed architecture are briefly summarized below:

- **β-VAE**: A modification of the standard Variational Autoencoder (VAE), the β-VAE introduces a scaling factor β to the Kullback–Leibler (KL) divergence term in the objective function (Higgins *et al.* 2017). This adjustment provides a mechanism to balance the trade-off between reconstruction fidelity and the degree of disentanglement in the latent space. Higher values of β promote higher disentanglement, however at the possible expense of reconstruction fidelity.

- **GANs**: Generative Adversarial Networks are composed of two networks—a generator G and a discriminator D—that are trained in opposition to one another (Goodfellow *et al.* 2014). The generator $G(\boldsymbol{\zeta})$ maps latent vectors $\boldsymbol{\zeta} \sim p_{\boldsymbol{\zeta}}$ (e.g., standard Gaussian) to reconstructions $\tilde{\mathbf{S}}$, while the discriminator D attempts to distinguish between real samples $\mathbf{S} \sim p_{\mathbf{S}}$ and generated samples $\tilde{\mathbf{S}}$. This adversarial training encourages the generator to produce outputs that are statistically consistent with the real data distribution.

The combined β-VAE-GAN framework (illustrated in Figure 2) consists of three components: an encoder E, a decoder/generator G, and a discriminator D. The encoder and generator are jointly optimized, while being trained adversarially against the discriminator. The training objective integrates terms from both the β-VAE and the GAN formulations into the following adversarial loss:

$$\mathscr{L}_{EG} = \mathbb{E}_{\mathbf{S}\sim p_{\mathbf{S}}}\Big[\underbrace{d(\mathbf{S},\tilde{\mathbf{S}}) + \beta\, D_{\mathrm{KL}}\big(q_{\mathbf{w}_E}(\boldsymbol{\zeta}|\mathbf{S}) \,\|\, p_{\boldsymbol{\zeta}}\big)}_{\beta\text{-VAE terms}} - \alpha \underbrace{\log\big(D(\tilde{\mathbf{S}})\big)}_{\text{GAN term}}\Big], \tag{1}$$

$$\mathscr{L}_{D} = \mathbb{E}_{\mathbf{S}\sim p_{\mathbf{S}}}\Big[-\log\big(1 - D(\tilde{\mathbf{S}})\big)\Big] + \mathbb{E}_{\mathbf{S}\sim p_{\mathbf{S}}}\Big[-\log\big(D(\mathbf{S})\big)\Big], \tag{2}$$

Here, β and α are user-defined hyperparameters that control the trade-off between reconstruction accuracy ($d(\cdot,\cdot)$), latent disentanglement ($D_{\mathrm{KL}}(\cdot)$), and statistical fidelity ($\log D(\cdot)$). During training, two separate optimizers update the networks parameters: one for the combined EG loss (Equation 1) and another for the discriminator (D) loss (Equation 2).

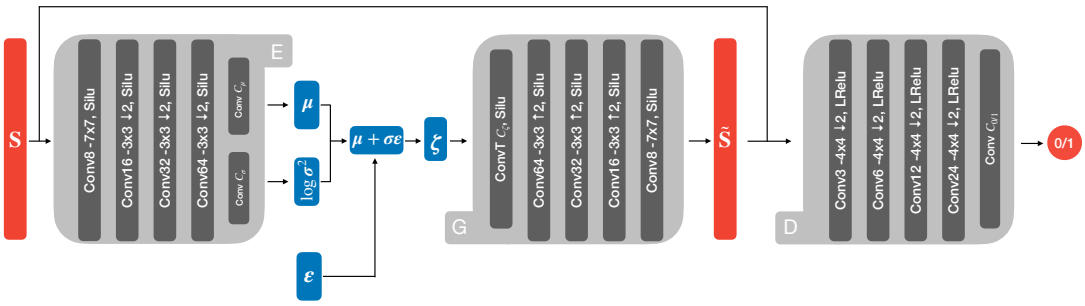

Figure 2: β-VAE-GAN architecture. The encoder (E), generator (G), and discriminator (D) use convolutional layers. Each Conv$C$ layer applies $p \times p$ filters with C output channels, where p is the filter kernel size. Downward arrows ($\downarrow 2$) denote strided down-sampling, while upward arrows ($\uparrow 2$) denote Lanczos up-sampling. Silu (*sigmoid linear unit*) and LRelu (*leaky ReLU*) are used as activation functions. The layers labelled `Conv` C_μ and `Conv` C_σ project their inputs to the mean and variance of the latent variables, respectively, whereas `ConvT` C_ζ projects the sampled latent variable back to the encoder's final layer dimensionality.

3.2 Temporal Modeling via Unidirectional Transformer

To model the temporal dynamics encoded within the low-dimensional latent features $\boldsymbol{\zeta}$ extracted by the β-VAE-GAN (Section 3.1), a decoder-only Transformer architecture is employed. This architecture is inspired by the Generative Pre-trained Transformer (GPT) framework (Radford *et al.* 2018), which is well-suited for this task due to its inherent autoregressive capabilities for

sequence modeling. However, standard GPT models are designed for text sequence generation, necessitating adaptations for multivariate time series forecasting. The goal here is to forecast the entire latent state vector $\boldsymbol{\zeta}_{t+1}$ based on a sequence of k previous states, i.e., given the input sequence $\{\boldsymbol{\zeta}^{t-k+1}, \ldots, \boldsymbol{\zeta}^{t}\}$, predict $\boldsymbol{\zeta}^{t+1}$.

Two primary modifications distinguish this model from a standard GPT architecture. First, the input embedding layer, typically used to map discrete tokens (like words) to vectors, is replaced by a shallow convolutional neural network (CNN). This input CNN processes the sequence of latent vectors $\{\boldsymbol{\zeta}^{t-k+1}, \ldots, \boldsymbol{\zeta}^{t}\}$ to extract local temporal features within each latent variable's history and to capture potential cross-channel dependencies between different latent variables over the input window. Second, the output layer is adapted for time series value prediction. While standard GPT models commonly use a final linear layer followed by a softmax activation to predict probabilities over a vocabulary, this model omits the softmax step. Instead, a final linear projection layer directly outputs the predicted time series values for the next latent state vector $\boldsymbol{\zeta}^{t+1}$. Apart from these input and output modifications, the core structure consists of stacked Transformer blocks adhering to the decoder-only design. Each block primarily utilizes a unidirectional (causal) self-attention mechanism, maintaining the required autoregressive property.

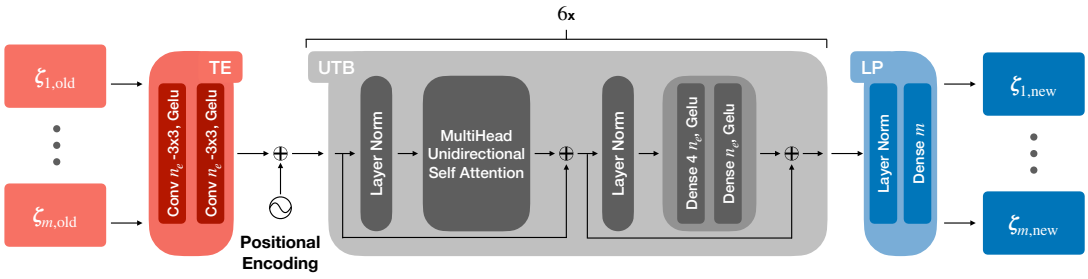

Figure 3: Transformer architecture. The time embedding (TE) layer consists of two convolutional layers with 3×3 filters and Gelu (*Gaussian error linear unit*) activation functions. The number of filter channels is set to n_e, which determines the dimensionality of the encoded representation fed into the unidirectional transformer block (UTB). The latter uses a multihead, unidirectional self-attention mechanism combined with a shallow dense feedforward network. This transformer block is repeated six times. Finally, a dense latent projection (LP) layer maps the transformer's output to the latent trajectory space.

4. Results

The proposed reduced-order modeling framework is evaluated using the test dataset introduced in Section 2.2.

4.1 Temporal Modeling on the Extracted Low-Dimensional Manifold

The latent variables encoded by the β-VAE-GAN (Section 3.1) define a low-dimensional manifold capturing the dominant flow dynamics. Despite the reduced dimensionality, the system's chaotic behavior is preserved, and Transformer forecasts are expected to retain its dynamical features. This preservation is assessed via the dominant Lyapunov exponent (λ), estimated from the divergence of nearby trajectories using Rosenstein *et al.* (1993).

Figure 4 (left) presents a representative divergence plot for latent variable ζ_6. The linear region of the logarithmic divergence curve yields estimates for both the ground truth ($\lambda_{gt} = 0.2094$) and predicted ($\lambda_p = 0.1928$) trajectories, indicating strong agreement and suggesting effective preservation of the system's chaotic nature. Across all latent variables, the maximum absolute deviation was $|\lambda_{gt} - \lambda_p| = 0.0423$ (observed for ζ_2), confirming that the Transformer preserves and captures the underlying chaotic dynamics.

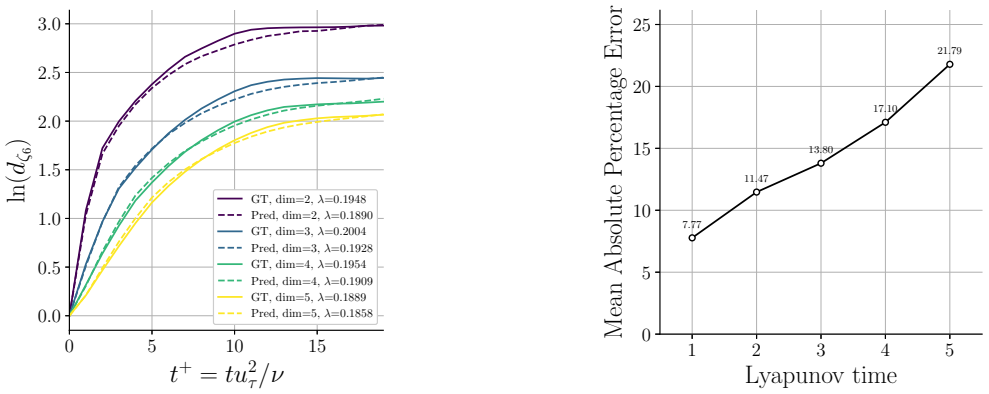

Figure 4: Latent space forecasting performance. Divergence plot of latent variable ζ_6 used to estimate the Lyapunov exponent (left). Solid and dashed lines represent ground truth (from encoded snapshots) and predicted trajectories, respectively. Colors indicate different embedding dimensions as in Rosenstein *et al.* (1993). Mean absolute percentage error (MAPE) between predicted and true latent trajectories as a function of Lyapunov time t/τ_λ (right).

Prediction accuracy within the latent space is further assessed in Figure 4 (right), which reports the mean absolute percentage error (MAPE) between ground truth and forecasted trajectories as a function of Lyapunov time, t/τ_λ. Here, $\tau_\lambda = \langle \lambda^{-1} \ln(d_{max}/d_0) \rangle$ is the characteristic chaotic time of the system; $\ln(d_{max}/d_0)$ represents the maximum logarithmic divergence, and $\langle \cdot \rangle$ denotes averaging over all latent variables. Within one Lyapunov time ($t/\tau_\lambda = 1$), the MAPE remains below 8%. Despite the inevitable divergence characteristic of chaotic systems, the Transformer maintains strong predictive accuracy with errors increasing gradually to approximately 11% and 14% at $t/\tau_\lambda = 2$ and 3, respectively. Figure 5 shows the time evolution of two representative latent coordinates, ζ_3 and ζ_6, illustrating the predictive behavior.

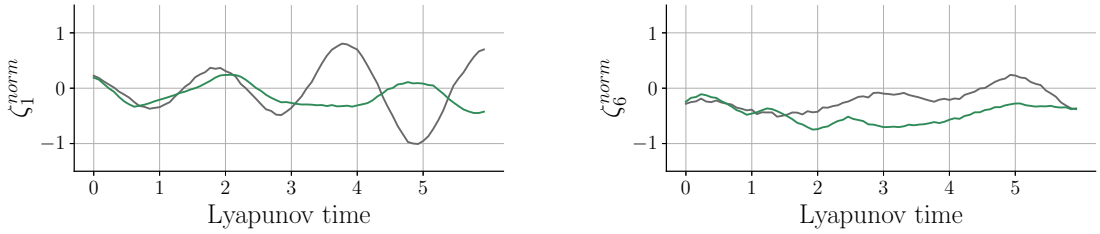

Figure 5: Time evolution of two representative latent coordinates: ζ_1 (left) and ζ_6 (right). Gray and green lines denote ground truth (from encoded snapshots) and predicted trajectories, respectively.

In both cases, predicted trajectories closely follow the ground truth initially, before diverging in a manner consistent with the underlying chaotic behavior. These results confirm that, as expected, the predictive horizon of the Transformer is ultimately limited by the system's chaotic nature. Nevertheless, long-term forecasting may be enhanced through data assimilation strategies — a promising direction for future research, though not pursued in the present study.

4.2 Flow Modeling: Decoding the Forecasted Latent Variables

The forecasted latent variables over a prediction horizon of one Lyapunov time are fed into the decoder to reconstruct the full flow field. This allows us to assess the framework's performance in the original high-dimensional space.

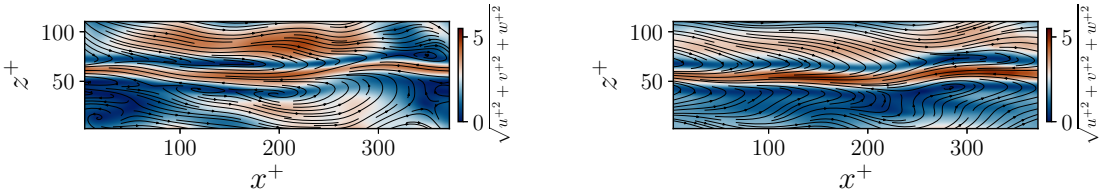

Figure 6: Instantaneous fluctuation velocity magnitude at $t/\tau_\lambda = 1$. Ground truth from DNS (left). Prediction obtained by decoding the Transformer-forecasted latent variables (right).

Figure 6 shows snapshots of the instantaneous velocity fluctuation magnitude, comparing the ground truth (DNS) with the predicted fields (decoded from forecasted latents) at $t/\tau_\lambda = 1$. The reconstruction captures the central low-speed streak accurately, indicating that the dominant near-wall structure is preserved. However, discrepancies are observed in the secondary features closer to the boundaries. Nevertheless, the normalized root mean squared error (NRMSE) between ground truth and prediction is 0.0890, which represents a modest deviation relative to the flow's fluctuation amplitude.

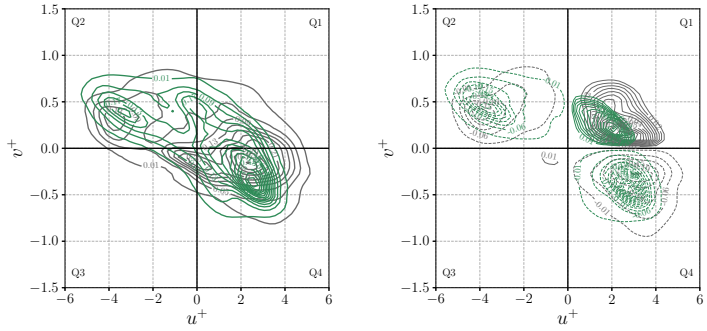

Figure 7: Comparison of ground truth and predicted statistics over a forecast horizon of $t/\tau_\lambda = 1$. Joint probability density function $p(u,v)$ (left). Weighted distribution $uv\,p(u,v)$ (right). Gray and green curves correspond to ground truth (from DNS) and prediction, respectively.

A statistical comparison further validates the quality of the predicted flow fields. Figure 7 presents the joint probability density function $p(u,v)$ and the Reynolds-stress-weighted form $uv\,p(u,v)$. The distributions exhibit good qualitative and quantitative agreement, achieving a

Wasserstein distance of 0.5589. Notably, the weighted distributions align well in the Q2 (ejection) and Q4 (sweep) quadrants, demonstrating that the model captures the significant mechanisms of momentum transfer in near-wall turbulence. Despite this success, limitations remain. Minor peak misalignments persist across the distributions, and higher discrepancies are observed in the Q1 (outward interaction) quadrant, which represents weaker energetic events. Therefore, while the reconstructions capture essential flow physics, achieving higher quantitative accuracy remains a target. The previously mentioned potential of data assimilation techniques offers a promising avenue to address these remaining discrepancies in future work.

References

L. Agostini and M. A. Leschziner. On the influence of outer large-scale structures on near-wall turbulence in channel flow. *Physics of Fluids*, 26.7, 2014.

G. Berkooz, P. Holmes, J.L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. of Fluid Mechanics*, 25(1):539-575, 1993.

L. Gasparino, F. Spiga, O. Lehmkuhl. SOD2D: A GPU-enabled Spectral Finite Elements Method for compressible scale-resolving simulations. *Computer Physics Communications*, 297: 109067, 2024.

I.J. Goodfellow, J. Pouget-Abadie, M. Mirza., B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets. *Advances in neural information processing systems*, 17, 2014.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Matthew, M. Shakir, A. Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

J. Kim, P. Moin, R. Moser. Turbulence statistics in fully developed channel flow at low Reynolds number. *J. of Fluid Mechanics*, 177:133-166, 1991.

J. Jiménez and P. Moin. The minimal flow unit in near-wall turbulence. *J. of Fluid Mechanics*, 225:213-240, 1991.

I. Marusic, R. Mathis, N. Hutchins. High Reynolds number wall turbulence. Annual Review of Fluid Mechanics, 42, 245-274, 2010. *Ann. Rev. of Fluid Mechanics*, 42: 245-274, 2010.

A. Radford, K. Narasimhan, T. Saliman and I. Sutskever. Improving language understanding by generative pre-training. *OpenAI*, 2018.

M. T. Rosenstein, J. J. Collins, C. J. De Luca. A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, 65.1-2: 117-134, 1993.

A. Solera-Rico, C. Sanmiguel Vila, M. Gómez-López, Y. Wang, A. Almashjary, S. T. Dawson, R. Vinuesa. β-Variational autoencoders and transformers for reduced-order modeling of fluid flows. *Nature Communications*, 15.1-1361, 2024.

Acknowledgements

This work was supported by the French government program "Investissements d'Avenir" (EUR INTREE, ANR-18-EURE-0010 and LABEX INTERACTIFS, ANR-11-LABX-0017-01) and performed using GENCI–IDRIS HPC resources (Grant 20XX-AD011015409). RV and MU acknowledge the financial support from ERC grant no. 2021-CoG-101043998, DEEPCONTROL. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council.