Efficient Vision Transformer-based Surrogate for Scalable Pressure Prediction in Incompressible Turbulent Flows

Anonymous Authors[1]

Abstract

Pressure estimation in turbulent flows is crucial to model and understand the evolution of turbulence, boundary-layer behavior, and energy distribution. Unlike compressible flow, pressure is not a thermodynamic state variable in incompressible flows, rather pressure acts as a kinematic constraint governed by the Pressure Poisson Equation (PPE) and keeps the flow divergence-free. The PPE is generally solved via traditional numerical solvers, which are often prohibitively expensive for high-resolution pressure fields. Advanced deep learning approaches offer a compelling alternative; however, data-driven pressure prediction remains challenging due to the high dimensionality and inherent multiscale dynamics of turbulent flows. Conventional machine learning architectures typically impose strong locality assumptions and fixed receptive fields, limiting their ability to capture long-range pressure interactions and generalize beyond training distribution. To address these challenges, we adopt a Vision Transformer (ViT)-based architecture that explicitly models nonlocal dependencies through self-attention and enables patch-based representations suited for multiscale turbulence. We propose a hybrid loss function that incorporates a pressure-gradient consistency loss to better enforce physically meaningful pressure fluctuations. While effective, a consequential downside of standard ViT is that the computational cost scales up quadratically as the number of patches grows, hindering scalability to large flows. We therefore introduce a physics-guided adaptive patching mechanism that dynamically decides the patch size based on the richness of information and predefined reduced sequence length, substantially improving computational efficiency. Experimental results on oceanic incompressible stratified turbulence demonstrate that the designed ViT framework accurately predicts turbulent pressure fields, and that the adaptive patching-based ViT model achieves competitive accuracy at a significantly reduced computation cost. These results highlight the potential of physics-guided scalable transformer models for pressure prediction across diverse application domains, including aeroacoustics, atmospheric science, and oceanic turbulence where incompressible flow arises.

1. Introduction

In turbulent flows, pressure is a fundamental driver of fluid motion and plays a central role in governing the flow dynamics (Hamba, 1999). It regulates the flow separation and energy redistribution across different length scales and spatial locations within a flow (Gotoh & Nakano, 2003). Pressure gradients, especially the adverse ones, impacts boundary layer behavior and stability in fluid flow (Hamba, 1999). Failure to accurately estimate the pressure may lead to compromised performance in diverse engineering applications, even leading to hazardous conditions in aviation or deep sea mining (Gopalakrishnan Meena et al., 2024).

In this work, we consider incompressible stratified turbulence, mostly observed in geophysical flows such as atmospheric boundary layers and oceanic flow (Gopalakrishnan Meena et al., 2024)(Riley & DeBruynkops, 2003). In such flows, pressure is not a thermodynamic state variable and can not be defined through the equation of state. Pressure instead takes the role of a constraint that ensures the flow is divergence-free. Pressure is strongly coupled to the velocity field by the Pressure Poisson Equation (PPE). Thus, pressure can be estimated by solving the momentum equation followed by the continuity equation in an operator splitting scheme. However, numerical solution to the PPE involves integrating over the large spatial flow field and iterative refinement. Machine learning–based surrogate models can offer an efficient alternative by enabling fast and scalable pressure inference from readily available flow variables, thereby facilitating performance improvement in

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

real-world applications involving turbulent flows. Surrogate modeling of pressure can accelerate incompressible flow solvers, as it would significantly reduce the computational need of iterative pressure–velocity coupling schemes.

The flow becomes anisotropic and layered in incompressible stratified turbulence (Gopalakrishnan Meena et al., 2024). In such setting, vertical motions are relatively suppressed by buoyancy, however, horizontal motions remain strong. Thus, in horizontal direction pressure exhibits fine-scale and intermittent fluctuations while in vertical directions pressure varies slowly. Strong horizontal shear and vortical motions create sharp pressure gradients, making the pressure prediction tasks more challenging for surrogates. Most prior work on machine learning (ML)-based modeling of pressure seeks to predict a particular term of the PPE to solve the PPE. However, the literature lacks direct pressure prediction from velocity through a fully surrogate model. For general turbulence modeling tasks, studies have largely relied on local operators such as convolutional neural networks (CNNs) and graph neural networks (GNNs)(Pfaff et al., 2020). For problems requiring long-range dependency modeling, such as high-precision pressure prediction on a large grid, local methods face inherent limitations due to fixed receptive fields (Xu et al., 2025). Even when hierarchical architectures like U-Net (Ronneberger et al., 2015), that attempt to fuse global information via skip-connections, are employed in a standalone fashion, they often do not help to capture global context. Vision Transformer (ViT) addresses this problem by leveraging patch-based representation and self-attention to model global interactions across the entire input (Dosovitskiy et al., 2021). Recent works have adopted ViTs for the standard neural turbulence simulation task. Further, ViTs offer computational benefit compared to CNNs that rely on local operations that may depend more on memory bandwidth and are not very suitable to exploit the massive arithmetic throughput of modern graphics processing units (GPUs) (Dosovitskiy et al., 2021).

The central motivation of this work is to enable pressure prediction capability for large-scale 3D turbulence data in an effective way. To this end, first we develop a ViT model that accepts large 3D volumes and performs uniform patching to process inputs. We design a hybrid loss function for the ViT training that combines pressure gradient loss with the regular voxel-level Mean Squared Error (MSE) loss. This hybrid loss ensures that sharp changes in pressure are well-captured by the model. While ViTs become an obvious choice for large-scale turbulence data, their quadratic computational and memory complexity with respect to input sequence size N makes them impractical for large-scale turbulent flows with millions to billions of grid points, especially in resource constraint settings (Zhang et al., 2024). To address this, we propose a novel physics-guided adaptive patching mechanism that selectively emphasizes critical regions while reducing focus in less informative regions. We attribute the success of such a technique to the fact that different spatial regions contribute unequally to system behavior. While areas with strong gradients, nonlinear interactions, or localized phenomena require fine-grained resolution, smoother regions can be represented more coarsely without loss of fidelity. To achieve this, we propose to use Potential Vorticity (PV), a physical property of the fluid flow, to identify regions of potential nonlinear interactions and selecting smaller size patches where absolute value of PV is high.

Comprehensive experiments are conducted on an incompressible stably stratified turbulence dataset representing ocean dynamics. Experimental results show that our uniform patching-based ViT model with gradient consistency loss achieves very low MSE, indicating highly accurate pressure prediction. Finally, detailed analysis reveals that our adaptive patching-based ViT predicts pressure with a reasonable accuracy at a significantly lower computational budget. In particular, our approach enables pressure prediction with a sequence length reduced by a factor of 3. Such reduction in sequence length brings greater computational gain due to the involved $O(N^2)$ complexity.

2. Related Works

ViTs for Turbulence: Due to the inherent benefit and success in diverse scientific domains, Transformers are widely been used in turbulence (Hu et al., 2025; Li et al., 2023; Guo, 2026). However, it's use was limited to model 2D flows (Xu et al., 2025). A large number of work has focused on predicting the dynamics, denoting the next state of the same flow variables in the temporal regime (Hu et al., 2025). Recent transformer-based work explored the viability of Transformers in 3D flows, however, within the scope of temporal prediction or high-resolution reconstruction of same flow variables (Dang et al., 2022; Liu et al., 2025). Instead, in this work, we aims to develop a surrogate model for a correlated flow variable prediction from other flow variables in 3D, thus enabling large-scale, high-resolution, and high-dimensional turbulence modeling.

Long Sequence Issue and Adaptive Approaches: The quadratic complexity of transformers in terms of sequence length has motivated substantial research on improving computational performance when dealing with long-sequences. Most approaches address scalability by approximating attention through sparsity, locality, or low-rank assumptions (Beltagy et al., 2020; Zaheer et al., 2020; Choromanski et al., 2021), typically requiring architectural modifications. A separate line of work leverage hierarchical spatial representations inspired by adaptive mesh refinement (AMR) and quadtree (or octree) decompositions, long studied in numerical simulation and imaging. These processes allocate high

resolution to information-rich regions while using coarser resolution elsewhere (Berger & Colella, 1989)(Samet, 1984). Recent transformer-based models adopt similar ideas either by modifying the attention mechanism (Tang et al., 2022; Lin et al., 2022), or by designing hierarchical architectures that process information at multiple resolutions (Chen et al., 2021). Multiscale and operator-learning approaches improve efficiency through coarse-to-fine representations or spectral compression. However, they often rely on fixed hierarchies and struggle with localized, non-stationary structures typical of turbulent flows (Pfaff et al., 2020; Li et al., 2021). Recent adaptive patching methods operate at the tokenization stage by dynamically adjusting patch size (Zhang et al., 2024) or discarding patches (Xu et al., 2025) based on local information content, while leaving the transformer architecture and attention formulation unchanged. Recenlt, entropy-based adaptive patching has been proposed, however, not suitable for multichannel data (Choudhury et al.). Also, they mostly consider 2D scenarios, leaving the high-dimensional 3D case unexplored. Our physics-guided adaptive patching builds on this idea, enabling efficient ViT modeling for large 3D flow domains.

3. Preliminaries

3.1. Incompressible Navier-Stokes Equations

The dimensionless incompressible Navier-Stokes equations for continuity and momentum are given by

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \frac{1}{Re} \nabla^2 \boldsymbol{u}, \tag{2}$$

Here, $\boldsymbol{u} = (u, v, w)$ is the velocity vector in a Cartesian coordinate system, $\boldsymbol{x} = (x, y, z)$, evolving in time, t. Re is the Reynolds number, and p is the pressure of the flow field. An operator splitting scheme can be used to obtain a momentum equation, a pressure correction and a velocity correction:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} + \boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n = -\nabla p^n + \frac{1}{Re} \nabla^2 \boldsymbol{u}^n \tag{3}$$

$$p^{n+1} = p^n + \delta p \tag{4}$$

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\Delta t} = -\nabla \delta p. \tag{5}$$

where, for simplicity of exposition, we use a first order explicit scheme for the momentum equation. The Poisson equation is obtained by taking the divergence of Eq. 5 and requiring $\nabla \cdot \boldsymbol{u}^{n+1} = 0$:

$$\nabla^2 \delta p = \frac{1}{\Delta t} \nabla \cdot \boldsymbol{u}^*. \tag{6}$$

We described the basic operator splitting scheme above. In other iterative pressure-based schemes such as the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) (Patankar, 1980), at each time step, the solver iteratively updates the velocity and pressure fields. Computing the pressure update via the PPE (Eq. (6)) is time-consuming since commonly available numerical solvers, such as Conjugate Gradient preconditioned with symmetric Gauss-Seidel, are slow to converge. It is thus of interest to develop a surrogate model that can help us compute the solution much faster.

3.2. Vision Transformer and Self-Attention

ViT considers an image $x \in \mathbb{R}^{H \times W}$ as a bag-of-words where each word corresponds to an image patch. All the image patches are independently embedded into a corresponding patch embedding or token, thus making $x \in \mathbb{R}^{N \times d}$, where N is the number of patches and d is the embedding dimension. Leveraging a sequence of multi-head self-attention blocks, ViT learns the relation between the tokens and translates the sequence of tokens to a new representation space.

Given three sets of projected vectors (from input), the queries $\{\mathbf{q}_i\}_{i=1}^{N_q}$, keys $\{\mathbf{k}_i\}_{i=1}^{N_k}$, and values $\{\mathbf{v}_i\}_{i=1}^{N_v}$ (assuming $N_k = N_v$), the attention mechanism computes a weighted average of the values dynamically as follows:

$$\mathbf{z}_i = \sum_{j=1}^{N_v} h(\mathbf{q}_i, \mathbf{k}_j)\, \mathbf{v}_j, \tag{7}$$

where $\mathbf{q}_i, \mathbf{k}_j, \mathbf{v}_j \in \mathbb{R}^{1 \times d}$ and $h(\cdot)$ in general refers to a weighting function defines the importance of each value to the output. A common choice for $h(\cdot)$ is the scaled dot-product attention with softmax:

$$h(\mathbf{q}_i, \mathbf{k}_j) = \frac{\exp\left(\mathbf{q}_i \mathbf{k}_j^\top / \tau\right)}{\sum_s \exp\left(\mathbf{q}_i \mathbf{k}_s^\top / \tau\right)}, \tag{8}$$

where the scaling factor τ is set to $\tau = \sqrt{d}$. Given the sequence length is N, the complexity of the attention matrix is $O(N^2)$. Same attention process is repeated for multiple times with different weight matrices and referred as multi-head self-attention.

4. Methodology

While learning from high-resolution 3D data, traditional methods such as CNN-based approaches struggle to model the long-range interactions effectively. In contrast, self-attention mechanism allows transformers to capture the global context (Dosovitskiy et al., 2021). Leveraging that fact, first we design a vision transformer model for turbulence data with standard uniform patching and our proposed hybrid loss function that incorporates pressure-gradient consistency loss. Subsequently, we propose a novel physics-guided adaptive patching mechanism to reduce the number of patches (sequence length). Such adaptive patching technique directly addresses the central issue in ViT model,

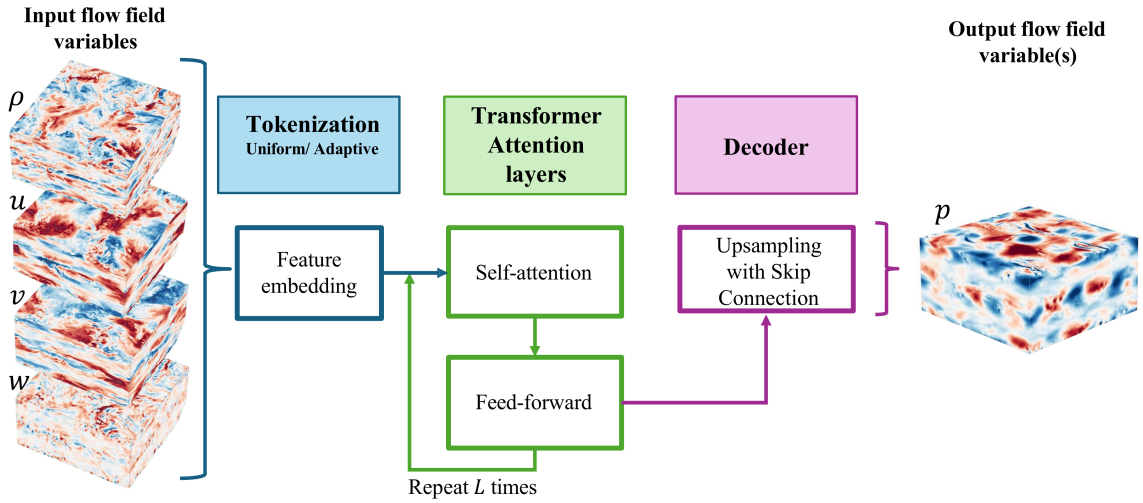

Figure 1. An overview of the Vision Transformer architecture used to predict the pressure based on raw DNS output including velocity vector components (u, v, w) and density ρ fluctuations.

the quadratic complexity of self-attention with regard to sequence length.

4.1. ViT-based Neural Surrogate

This section describes the transformer architecture and the proposed hybrid loss function used to train the model. We opt for an encoder-decoder architecture that uses vision transformer as the backbone (Dosovitskiy et al., 2021)(Hatamizadeh et al., 2022). Figure 1 presents an overview of the architecture used.

Transformer Encoder: Following the standard ViT formulation(Dosovitskiy et al., 2021), we extracts patches uniformly from 3D turbulence data $x \in \mathbb{R}^{Z\times Z\times Z}$ as non-overlapping cubic patches of size p^3. Our contribution in the next section, presents a novel way to adaptively patch the data instead of using uniform patch size to improve compute efficiency. The patches are flattened to construct feature vectors which are then individually transformed to patch embedding using a linear projection, making $x \in \mathbb{R}^{N\times d_{in}}$, where $N = [\frac{Z}{p}]^3$ is the number of patches and d_{in} is the patch embedding dimension. As transformers do not inherently track the spatial positions of the tokens, to make the representation position-aware, patch embeddings are concatenated with a learnable position embedding. We pass this final input token sequence through L transformer layers to obtain the final representation r_l, where l refers to the l-th transformer layer.

Each transformer layer T_l implements a multi-head self-attention mechanism where quieries, keys, and values are computed from the same source as following:

$$q_i = u_i W_q,\ k_i = u_i W_k,\ v_i = u_i W_v, \tag{9}$$

where $u_i \in \mathbb{R}^{1\times d_{in}}$ is the input vector and $\{\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v\} \in \mathbb{R}^{d_{in}\times d}$ are projection matrices learned during training. We adopt scaled dot-product for self-attention as described in Eqn. 2.

Decoder: The decoder follows a hierarchical U-Net style design similar to UNETR that progressively reconstructs high-resolution volumetric predictions from multi-scale transformer representations. Starting from the deepest transformer representation, the decoder performs a sequence of upsampling and fusion stages.

For skip connection, intermediate feature embeddings are extracted from selected Vision Transformer blocks and first projected from the token space to spatial 3D feature maps using learned projection matrices. Then, deconvolution layer upsamples and aligns the feature maps with the target volumetric resolution of the corresponding decoder module. The decoder block at that level (e.g. 3,6,9) first upsamples the feature maps and then integrates the encoded feature maps obtained through skip connection. These combined features are further refined by lightweight convolution blocks. Finally, a convolutional output head maps the final decoder output to the desired prediction shape. This design enables effective integration of global contexts with localized volumetric structure, which is necessary for accurate 3D dense prediction tasks.

Physics-guided Loss: The pressure differences between consecutive spatial locations significantly control the behavior of the turbulent flows, making the accurate estimation of pressure gradient a crucial aspect in the surrogate modeling. Motivated by the implication of pressure gradient, we propose a hybrid loss function that considers the standard Mean Squared Error (MSE) loss along with an auxiliary pressure

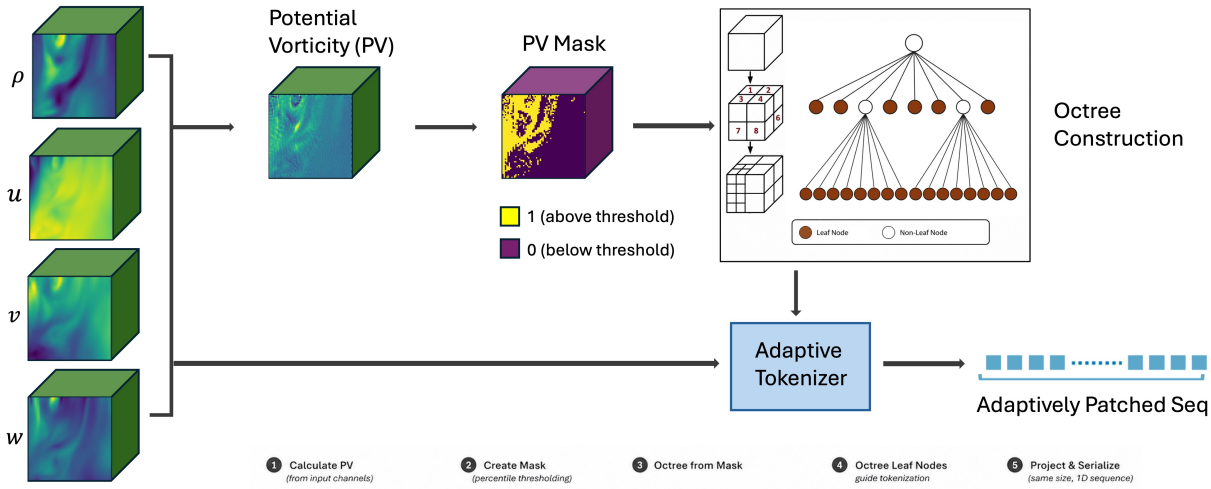

Figure 2. An overview of the potential vorticity-guided adaptive patching pipeline: (1) calculates potential vorticity based on the input channels, (2) creates the mask by percentile-based thresholding, (3) generates the octree based on the mask, (4) octree leaf nodes guides the tokenization of input channels, (5) projected to same size and serializes the patches in a sequence. Taking only the leaf nodes allows reduced number of patches. We present the front 2D slice for simplicity, however, we experiment on full 3D volumes.

gradient consistency loss as follows:

$$\mathcal{L} = \sum_{i,j,k} [(P_{i,j,k} - \hat{P}_{i,j,k}) + \alpha(\nabla P_{i,j,k} - \nabla \hat{P}_{i,j,k})] \quad (10)$$

where α is the coefficient of the pressure gradient loss term, P is the ground truth pressure, $\hat{P}$ is the predicted pressure, and i, j, k denotes the spatial location in the volume with respect to a 3D Cartesian coordinate system. Such gradient consistency loss ensures that the model respects the smaller scale fluctuations in pressure field. We estimate the true pressure gradient from the ground truth pressure while the predicted pressure gradient comes from the predicted pressure field. The pressure field is already discretized with uniform mesh of points in each spatial direction. Thus, we compute the gradient based on the central differences for each dimensions (x,y,z) of the the 3D volumes as follows:

$$\frac{P_{i+1,j,k} - P_{i-1,j,k}}{2}, \frac{P_{i,j+1,k} - P_{i,j-1,k}}{2}, \frac{P_{i,j,k+1} - P_{i,j,k-1}}{2}$$

where i, j, k denote the spatial location considering a Cartesian coordinate system in 3D.

4.2. Physics-guided Adaptive Patching

While standard ViTs divide the images uniformly into equal-sized patches, Adaptive Patching (AP) mechanism allows splitting an image into patches of different sizes. We propose a physics-guided approach to dynamically decide the patch sizes based on the richness of the information. The central motivation is to allow large size patches in low information region in contrast to having smaller patches in information rich regions. Large patches significantly reduces the length of the patch sequence N as $N = [\frac{Z}{p}]^3$ where Z is the volume dimension and p is the patch size.

Potential Vorticity-based Adaptive Patching Decision: Conventional adaptive patching mechanism in images leverages the edge information to estimate information density in a spatial region and partitions the image with different patch size based on edge map. In particular, they deploy the Canny edge detection algorithm with certain threshold to extract edge map. While such edge-based approach is suitable for 2D image classification or segmentation tasks, for turbulence like 3D multi-channel scientific data the definition of edge and the threshold become critical and unreliable.

As we want to predict pressure from multi-channel input, it is crucial to consider all the available channels. Thus, we propose to use a physical attribute Potential Vorticity (PV) that is a function of the considered input channels, all components of velocity $\boldsymbol{u} = (u, v, w)$ and density ρ. For a turbulent flow PV is defined as follows:

$$\Pi = \omega \cdot \nabla \rho \quad (11)$$

where $\omega = \nabla \times \boldsymbol{u}$ is the vorticity and ρ is the density.

Patch Extraction: We construct a PV mask $M_{pv} \in \{0, 1\}^{Z \times Z \times Z}$ by thresholding the PV value Π with a certain percentile threshold P_t. In particular, we take the absolute values of the Π and then estimate the percentile value V_t based on P_t. Finally, we apply the percentile thresholding as follows to create the PV mask M_{pv}.

$$M_{pv} = \begin{cases} 1 & if\ abs\ (\Pi) > V_t, \\ 0 & else, \end{cases} \quad (12)$$

The mask M_{pv} is recursively partitioned in a octree structure. Let O^h denotes a node at depth h. Then the node O^{h+1} at

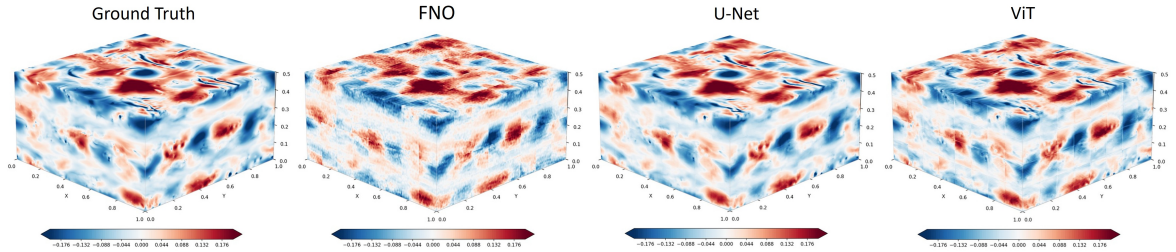

Figure 3. Predicted pressure across a snapshot *close* to the training regime of size $512 \times 512 \times 256$ and using 4^3 patch size. Left-to-Right: Ground truth, FNO prediction, UNet prediction, and our ViT prediction. We inference over 64^3 inputs to cover the full snapshot.

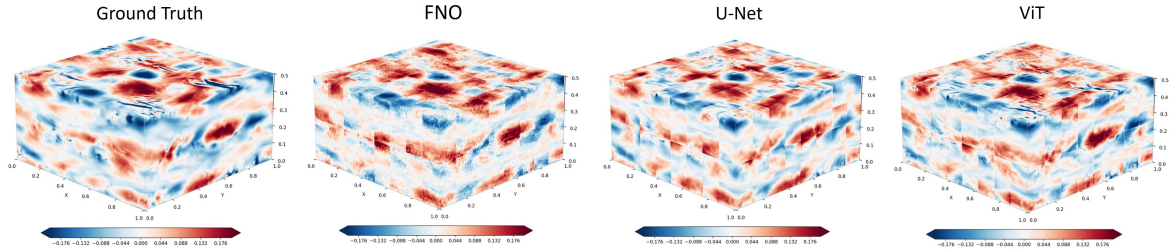

Figure 4. Prediction results across a *distant* snapshot of size $512 \times 512 \times 256$ and using 4^3 patch size. Left-to-Right: Ground truth, FNO prediction, UNet prediction, and our ViT prediction. We inference over 64^3 inputs to cover the full snapshot.

depth $h+1$ is defined as follows:

$$O^{h+1} = \begin{cases} O^h & if \sum_i D_i \leq v \; or \; h = H \\ \{O^h_{FTL}, O^h_{FTR}, & \\ O^h_{FBL}, O^h_{FBR}, & if \sum_i D_i > v \; or \; h \leq H \\ O^h_{BTL}, O^h_{BTR}, & \\ O^h_{BBL}, O^h_{BBR}\} & \end{cases} \tag{13}$$

where H is the maximum depth of the octree, v is the criterion for partition, and $O^h_{FTL}, O^h_{FTR}, O^h_{FBL}, O^h_{FBR}, O^h_{BTL}, O^h_{BTR}, O^h_{BBL}, O^h_{BBR}$ are the child nodes at depth h representing Front-Top-Left, Front-Top-Right, Front-Bottom-Left, Front-Bottom-Right, Back-Top-Left, Back-Top-Right, Back-Bottom-Left, Back-Bottom-Right octants, respectively. $\sum_i D_i$ defines the the total number of masked pixel in the current volume.

In uniform patching, generally all the patches are concatenated in a row-wise fashion to construct the 1D patch sequence. In adaptive patching, the leaf nodes from the final octree represents the patches and only those leaf nodes need to be serialized in a sequence. We use Morton Z-order curve to construct the patch sequence as it keeps the geometrically affine patches closer in the sequence. It is important to note that at this point all patches are not of same size, rather patches at same octree depth have same patch size. Thus we need to explicitly project all patches to the same size p_e which we refer as the effective patch size. If the constructed sequence length for any input is less than the desired length L, then we pad them with patches containing zeros. At the end, the model receives an input sequence of patches $x_p \in \mathbb{R}^{L \times {p_e}^3}$ which is further embedded to $x_s \in \mathbb{R}^{L \times d}$.

Figure 2 shows the full adaptive patching pipeline.

5. Experimental Setup

5.1. Implementation Details

We use the widely used UNETR model (Hatamizadeh et al., 2022) as our ViT backbone. UNETR uses series of Transformer layers as encoder and employ skip connection in the decoder. We use embedding dimension of 768, number of Transformer layers 12, and the number of head for self-attention is 12. Deconvulation layers are used as decoder and skip connections are established at depth $3, 6, 9$. We use 70 percentile as the threshold for PV masking. We use a learning rate of 0.00001 with weight decay. For 64^3 we use a batch size of 4, while for 32^3 inputs with use a batch size of 8. However, the effective batch size is multiplied by 40, as we use 40 GPUs distributed over 5 nodes for training through Distributed Data Parallel (DDP) from PyTorch. Experiments were conducted on a supercomputing cluster. Each node in the cluster consists of one 64-core AMD EPYC CPU and 8 GPUs (64 GB each), organized into 4 MI250X cards with two GPUs per card. Crucial libraries include PyTorch v3.10, ROCm v6.2.4.

5.2. Datasets

The dataset is obtained from decaying homogeneous SST flow simulations as described in (Riley & DeBruynkops, 2003). The flow field is initialized by Taylor–Green vortices and perturbed with a low-level noise (10% of initial energy) to trigger instabilities for the flow to evolve into

Table 1. Evaluation of ViT model trained on 32^3 input and 64^3 input over test snapshots and the patch size is 4^3.

Vol.	Seq Len	MSE	RMSE	PSNR	SSIM
32^3	512	5.3e-04	2.3e-02	35.4	83.1
64^3	4096	**1.7e-04**	**1.3e-02**	**40.2**	**94.9**

Table 2. Evaluating generalization performance of the ViT model over distant snapshots. Model input is 64^3 and patch size is 4^3.

Method	MSE	RMSE	PSNR	SSIM
FNO	1.5e-03	3.97e-02	31.60	63.10
UNet	1.1e-03	3.23e-02	33.39	70.28
ViT	**1.0e-03**	**3.18e-02**	**33.53**	**74.11**

turbulence. The flow is initially laminar, then transitions to turbulence, and finally becomes increasingly more stratified as it decays. Following the analysis of (Gopalakrishnan Meena et al., 2024), we use the flow field initialized with Reynolds and Froude numbers $Re3200$ and $Fr = 4$. The spatial resolution of the data in three directions is $(n_x, n_y, n_z) = (512, 512, 256)$, with gravity in the z direction. At each instant in time, the following fields are extracted for training the model: the velocity in three spatial directions, $\boldsymbol{u} = (u, v, w)$, density, ρ, and pressure, p. The time regime used for training is between nondimensional time $t = [10, 15]$, where the flow transitions from laminar to turbulent regime and starts decaying (see Fig. 6 of (Gopalakrishnan Meena et al., 2024) for temporal evolution of the flow field). For the physics-guided adaptive patching, the potential vorticity (PV) is computed using velocity and density: $\Pi = \omega \cdot \nabla\rho$, where $\omega = \nabla \times \boldsymbol{u}$.

6. Results and Discussions

We first show the effectiveness and capability of uniform patching-based ViT in pressure prediction. We compare the uniform ViT with traditional approaches such as Fourier Neural Operator (FNO) (Li et al., 2020) and U-Net (Ronneberger et al., 2015). Then we discuss the performance trade-off achieved by adaptive patching mechanism and its implication. We compare our adaptive patching results with existing edge density-based adaptive patching approach (Zhang et al., 2024). Finally. we conduct ablation studies to understand the contribution of other components such as proposed gradient consistency loss.

6.1. Evaluation on In-Distribution Test Data

Figure 3 demonstrates that ViT model with standard uniform patching and our hybrid loss accurately predicts the pressure and captures the variation in pressure gradient. ViT and UNet perform competitively, while FNO shows spectral bias and fails to capture sharp gradients. We conduct experiments with two different input setting: 32^3 input and 64^3 input with patch size 4. This leads to sequence length of 512 and 4096 for 32^3 input and 64^3, respectively. Table 1 shows the results on held out test snapshots of volume $512 * 512 * 256$. It clearly shows that model learned from 64^3 attains better performance across different evaluation metrics. The MSE and RMSE loss are substantially lower than the 32^3 counterpart while the PSNR and SSIM scores are much higher indicating strong similarity with the ground truth pressure. Also, using large sub-cubes to iteratively cover a big volume is beneficial to capture vortices and avoid more discontinuity. Fig. 3 in Appendix shows that qualitatively also larger sub-cubes yield better predictive accuracy.

6.2. Generalization to Distant Snapshots

To test generalization performance of the model we select data points far away from the training regime in time and analyze how it performs with the evolved turbulent flow. Table 2 shows the results on a out-of-distribution snapshot and it is evident that prediction via ViT-based surrogate provides substantial performance gain over FNO and UNet.

6.3. Performance Gain via Adaptive Patching

Considering the better predictive accuracy with 64^3 input model and the corresponding high computational need (sequence length $N = 4096$ with 4^3 patches) with uniform patching, we apply our adaptive patching mechanism to this setting with the aim of reducing computational overhead. We set our target reduced sequence length to 1331 which is smaller by a factor of 3 compared to the required 4096 length when uniform patching is used. This reduced length implies that we can not have all 4^3 patches, rather we will have a mix of 4^3, 8^3, 16^3, and 32^3. The octree formation will dynamically decide which octant to divide starting from the 64^3 input. More specifically, as described in the methodology potential vorticity will guide which octant may have more correlation guiding information. At the end, all patches are projected to the minimum patch size, in this case 4^3.

Figure .5 shows the comparison of pressure prediction by the uniform patching method and the proposed adaptive patching method. It is clear that the adaptive patching approach successfully preserves the dominant pressure fluctuations and the overall spatial organization of the pressure field, even at substantially reduced sequence length. However, Table 3 reveals that the adaptive patching strategy introduces a trade-off compared to uniform patching. By dynamically adjusting patch sizes based on the local information content, adaptive patching significantly reduces the sequence length and, consequently, the overall computational cost of the model. However, this reduction often comes at the expense of a modest loss in predictive accuracy, as small-

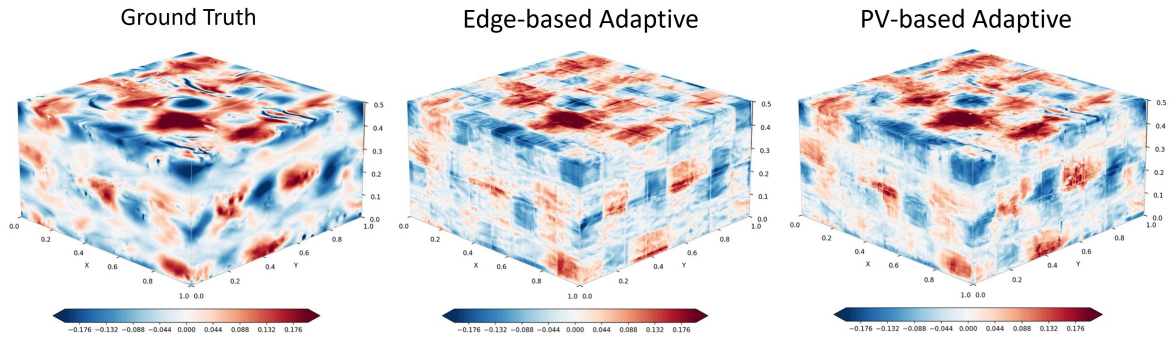

Figure 5. Predicted pressure across a full snapshot of size $512 \times 512 \times 256$ and using 4^3 patch size. (Left) prediction with uniform patching 4096 seq. length, and (Right) prediction with adaptive patching only 1331 seq. length.

Table 3. Evaluation of PV-based adaptive patching compared to uniform and edge-based patching on test data. While it reduces computational overhead, it needs to trade-off the accuracy with uniform version. Please refer to Fig. 5 that shows the overall pressure fluctuations are captured even with a reduced sequence length.

Patching	Sq L.	Speed	RMSE	PSNR	SSIM
Uniform	4096	1×	**1.3e-02**	**40.2**	**94.9**
Edge-based	1331	3×	3.9e-02	30.7	65.4
PV-based	1331	**3**×	2.7e-02	33.9	81.1

Table 4. Generalization performance of proposed adaptive patching approach. Even with reduced sequence length and computational overhead, PV-based adaptive patching achieves better generalization than uniform and edge density-based adaptive approaches.

Patching	Sq L.	Speed	RMSE	PSNR	SSIM
Uniform	4096	1×	3.2e-02	33.5	74.1
Edge-based	1331	3×	5.2e-02	29.3	45.5
PV-based	1331	**3**×	**2.5e-02**	**35.6**	**83.9**

scale details may be lost by larger patches. Despite this compromise, the results in Table 4 suggest that potential vorticity-guided adaptive patching achieves superior generalization performance on distant unseen data. The failure of uniform patching could be attributed to its over-reliance on input structure. Thus PV-based adaptive patching offers an effective balance of efficiency and fidelity, retaining the prominent physical structures of the pressure field while enabling scalable inference. Such model can effectively be deployed in real-world scenario where inference time is crucial or in places where training time is limited by computational resources.

6.4. Effect of Gradient Consistency Loss

We perform an ablation study by removing the gradient consistency loss and keeping only the MSE loss. We do the experiments for both the uniform and adaptive patching version and observe performance gain. Evaluation on the Distant data in table 5 reveals that the gradient consistency loss helps to achieve generalization by capturing the pressure gradient dynamics.

Table 5. Assessing the impact of Gradient consistency loss on 64^3 input variant over OOD snapshot and the patch size is 4^3 leading to 4096 sequence length with uniform patching.

Loss	MSE	RMSE	PSNR	SSIM
MSE	1.10e-03	3.32e-02	33.17	72.75
MSE+GC	**1.01e-03**	**3.18e-02**	**33.53**	**74.11**

7. Conclusion

In this work, we demonstrated that accurate and scalable pressure prediction in turbulent flows can be achieved by combining transformer-based architectures with physics-guided design principles. By leveraging the global context modeling capability of ViTs, the proposed framework effectively captures long-range pressure interactions that are difficult to represent with conventional locality-driven machine learning approach. The introduction of a pressure-gradient consistency loss further improves physical fidelity by enforcing meaningful pressure variations, while the proposed adaptive patching mechanism significantly reduces computational complexity with moderate accuracy compromise in certain cases. Reduced sequence length allows training and inference with large batch sizes and enables scalability in high-dimensional large-scale turbulence data. Current limitations include CPU-based processing of adaptive patches that restricts full quadratic performance gain. The strong generalization performance achieved by the proposed adaptive patching-based ViT highlights its potential for large-scale, real-world, and real-time pressure prediction tasks and provides a practical alternative to expensive numerical solvers across a wide range of scientific applications.

References

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

Berger, M. J. and Colella, P. Local adaptive mesh refinement for shock hydrodynamics. *Journal of computational Physics*, 82(1):64–84, 1989.

Chen, C.-F. R., Fan, Q., and Panda, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021.

Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., et al. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

Choudhury, R., Kim, J., Park, J., Yang, E., Jeni, L. A., and Kitani, K. Faster vision transformers with adaptive patches. In *The Fourteenth International Conference on Learning Representations*.

Dang, Y., Hu, Z., Cranmer, M., Eickenberg, M., and Ho, S. Tnt: Vision transformer for turbulence simulations. *arXiv preprint arXiv:2207.04616*, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Gopalakrishnan Meena, M., Liousas, D., Simin, A. D., Kashi, A., Brewer, W. H., Riley, J. J., and de Bruyn Kops, S. M. Machine-learned closure of urans for stably stratified turbulence: connecting physical timescales & data hyperparameters of deep time-series models. *Machine Learning: Science and Technology*, 5(4):045063, 2024.

Gotoh, T. and Nakano, T. Role of pressure in turbulence. *Journal of statistical physics*, 113(5):855–874, 2003.

Guo, Z. Physics-informed transformer operator for the prediction of three-dimensional turbulence. *arXiv preprint arXiv:2601.19351*, 2026.

Hamba, F. Effects of pressure fluctuations on turbulence growth in compressible homogeneous shear flow. *Physics of Fluids*, 11(6):1623–1635, 1999.

Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H. R., and Xu, D. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 574–584, 2022.

Hu, X., Zhang, J., Yan, K., Wan, T., and Zheng, X. Physics-informed transformer for efficient fluid dynamics predictions. In *International Conference on Wireless Artificial Intelligent Computing Systems and Applications*, pp. 356–368. Springer, 2025.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Li, Z., Kovachki, N. B., and Azizzadenesheli, K. Burigede liu, kaushik bhattacharya, andrew stuart, and anima anandkumar. fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, volume 2, pp. 4, 2021.

Li, Z., Shu, D., and Barati Farimani, A. Scalable transformer for pde surrogate modeling. *Advances in Neural Information Processing Systems*, 36:28010–28039, 2023.

Lin, K., Li, L., Lin, C.-C., Ahmed, F., Gan, Z., Liu, Z., Lu, Y., and Wang, L. Swinbert: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17949–17958, 2022.

Liu, X., Zhang, Y., Guo, T., Li, X., Song, D., and Yang, H. A multi-scale hybrid attention swin-transformer-based model for the super-resolution reconstruction of turbulence. *Nonlinear Dynamics*, pp. 1–30, 2025.

Patankar, S. *Numerical Heat Transfer and Fluid Flow*. Taylor & Francis, 1980. ISBN 978-0-89116-522-4.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and Battaglia, P. Learning mesh-based simulation with graph networks. In *International conference on learning representations*, 2020.

Riley, J. J. and DeBruynkops, S. M. Dynamics of turbulence strongly influenced by buoyancy. *Physics of Fluids*, 15 (7):2047–2059, 2003.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Samet, H. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260, 1984.

Tang, S., Zhang, J., Zhu, S., and Tan, P. Quadtree attention for vision transformers. In *International Conference on Learning Representations*, 2022.

Xu, Z., Liu, J., Chen, K., Chen, Y., Hu, Z., and Ni, B. Amr-transformer: Enabling efficient long-range interaction for complex neural fluid simulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5804–5813, 2025.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Zhang, E., Lyngaas, I., Chen, P., Wang, X., Igarashi, J., Huo, Y., Munetomo, M., and Wahib, M. Adaptive patching for high-resolution image segmentation with transformers. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2024.

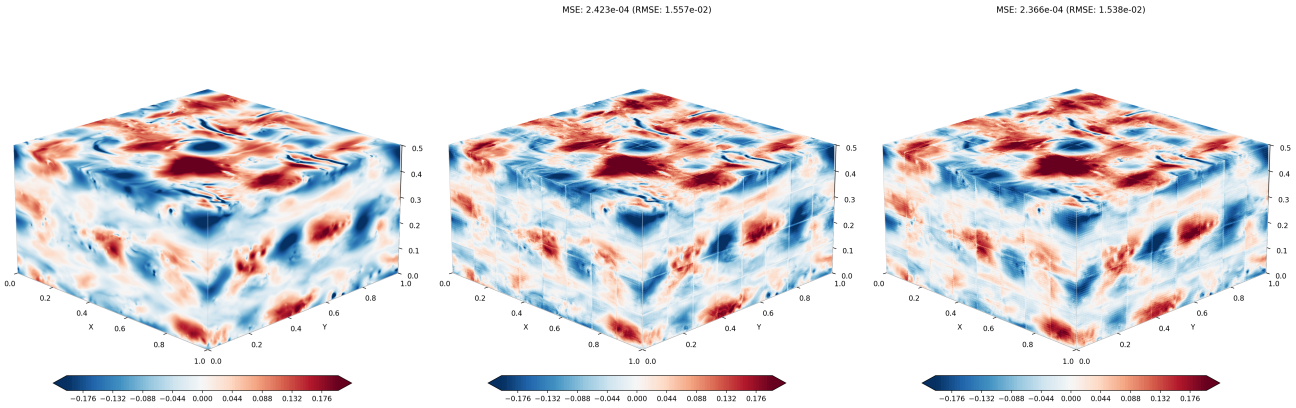

Figure 6. (Left) Ground Truth (Middle) Training with MSE loss only, and (Right) Training with MSE + Gradient consistency loss. The visualizations have been extracted from an intermediate epoch of training.

A. Evaluation Metrics

Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are widely used pixel-wise metrics. They quantify the average of the squared and square-rooted difference between prediction and ground truth. Thus, they can provide an idea of absolute reconstruction error. While MSE and RMSE captures large deviations, they do not consider perceptual or structural similarity. Peak Signal-to-Noise Ratio (PSNR) captures reconstruction quality in logarithmic scale relative to the maximum possible signal value. Thus, PSNR provides an interpretable measure of overall accuracy. Structural Similarity Index Measure (SSIM) evaluates similarity in terms of local structure and contrast, making it effective for assessing the retention of spatial patterns in prediction space. While lower MSE and RMSE are expected, higher PSNR and SSIM values indicate better reconstruction quality. Finally, we measure the computational overhead in terms of GPU hours and wall clock time for 1 epoch covering the whole dataset.

B. Guideline to Select Hyperparameters

For the gradient consistency loss coefficient (discussed in Section cite 4.1) we use 0.1. We search over [0.1, 0.2, 0.5, 1.0] and find 0.1 as optimal. This shows a relative lower weight with respect to the main MSE loss is helpful.

For the percentile threshold (discussed in Section cite 4.2), we search over [50, 70, 90] and find 70 as optimal percentile threshold. Our investigation reveals that higher cut-off such as 90 lacks enough masked point to effectively divide big patches, while low threshold such 50 retains more data and leads to sub-optimal patching decision. Thus, it would be legitimate to set the threshold based on reduced sequence length so that the tokenizer gets sufficient guidance from the selected points using the threshold.

We also compare our method with other adaptive patching policy using flow-related physical attributes such as Kinetic Energy or individual channels info. However, they fail to outperform our proposed Potential Vorticity-guided adaptive patching method.

C. Qualitative Comparison of Gradient Consistency Loss

Fig. 6 shows that even at early stage of training (e.g. 15 epoch) gradient consistency loss helps to identify smaller blobs of sharp gradient change in pressure.

D. Impact of Skip Connections in Decoder

Skip connections in decoder plays a crucial role in ViTs (Hatamizadeh et al., 2022), especially where multi-scale structures are important. Turbulence data is inherently multi-scale and that criteria significantly impact the behavior of the flow variables. Skip connections shortcuts the transformer encoder representations from intermediate layer with high-resolution to the decoder output. This design propagates the large-scale structure and fuse it with small-scale features. Fig. 7 shows

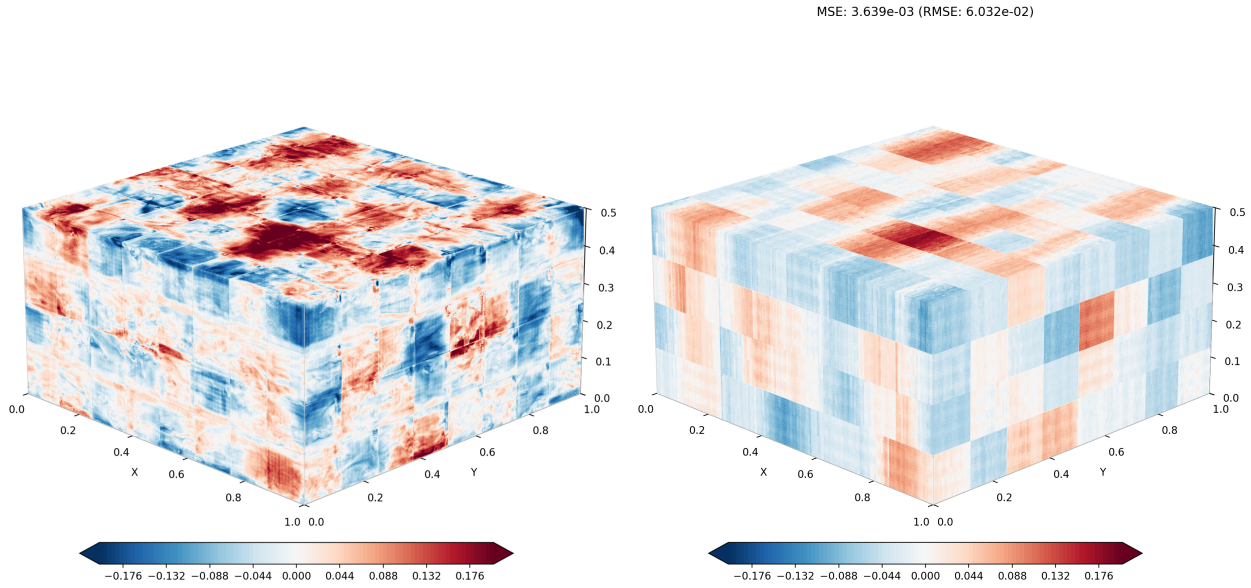

Figure 7. Predicted pressure (Left) decoder with skip connections, and (Right) no skip connection in the decoder.

that such skip connections are highly important to capture the dynamics of turbulence.