Engineering Applications of Artificial Intelligence 149 (2025) 110463

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Research paper

Dynamic feature capturing in a fluid flow reduced-order model using attention-augmented autoencoders

Alireza Beiki, Reza Kamali *

School of Mechanical Engineering, Shiraz University, Shiraz, Iran

ARTICLE INFO

Keywords:
Reduced-order modeling
Convolutional autoencoders
Adaptive attention mechanisms
Time series forecasting in fluid dynamics
Flow field reconstruction

ABSTRACT

This study looks into how adding adaptive attention to convolutional autoencoders can help reconstruct flow fields in fluid dynamics applications. The study compares the effectiveness of the proposed adaptive attention mechanism with the convolutional block attention module approach using two different sets of datasets. The analysis encompasses the evaluation of reconstruction loss, latent space characteristics, and the application of attention mechanisms to time series forecasting. Combining adaptive attention with involution layers enhances its ability to identify and highlight significant features, surpassing the capabilities of the convolutional block attention module. This result demonstrates an increase of over 20% in the accuracy of reconstruction. Latent space analysis shows the adaptive attention mechanism's complex and flexible encoding, which makes it easier for the model to represent different types of data. The study also looks at how attention works and how it affects time series forecasting. It shows that a new method that combines multi-head attention and bidirectional long-short-term memory works well for forecasting over 5 s of futures of flow fields. This research provides valuable insights into the role of attention mechanisms in improving model accuracy, generalization, and forecasting capabilities in the field of fluid dynamics.

1. Introduction

Fluid flow analysis is crucial in several areas of technology, such as automotive, aerospace, and weather forecasting. Understanding fluid dynamics improves vehicle aerodynamics, fuel efficiency, and performance in the automotive industry. In aviation, accurate understanding of airflow around aircraft components is necessary to ensure stability, minimize drag, and optimize lift. Turbine design, medical diagnostics, pollution management in energy systems, biomedical engineering, and environmental research are among areas where fluid flow analysis contributes to advancements.

Fluid flow dynamics are governed by nonlinear partial differential equations, which lack analytical solutions. In order to solve these equations, numerical approaches are required. The numerical solution of equations for task decision problems, such as optimization and flow control, can be a time-consuming process to achieve acceptable accuracy. One potential strategy for addressing this matter involves the utilization of reduced-order models (ROMs), which aim to decrease the dimensions of the problem (Lucia et al., 2004; Taira et al., 2017). As a result, they can effectively minimize the computational time required for numerical solutions. This makes them very suitable for resolving task-decision problems.

Reduced-order models are divided into two types, namely intrusive and non-intrusive. Intrusive reduced-order models, which are directly derived from the physical system's governing equations, frequently use projection-based approaches such as Proper Orthogonal Decomposition (POD) (Lumley, 1967) to minimize the computational complexity of high-fidelity models. In contrast, non-intrusive reduced-order models are equation-free. Instead, they rely on data-driven strategies to model the behavior of the full-order system (Taira et al., 2017; Xu and Duraisamy, 2020).

Proper orthogonal decomposition and dynamic mode decomposition (DMD) (Schmid, 2010) are two methods for reducing high-dimensional data. The linearity of these methods makes them unsuitable for dealing with advection-dominated problems. Therefore, advanced nonlinear methods are needed to accurately represent the underlying dynamics of such problems. A reducing-order model based on deep learning has been developed by many researchers to utilize the nonlinear feature extraction capabilities of deep learning on high-dimensional datasets (Han et al., 2019; Geneva and Zabaras, 2020; Hasegawa et al., 2020a). Because of significant improvements in computational capacity and the availability of datasets, deep learning's application to fluid dynamics has grown in prominence over the past several years (Brunton et al., 2020). Many fluid flow applications have

* Corresponding author.
E-mail addresses: alireza.beiki@shirazu.ac.ir (A. Beiki), rkamali@shirazu.ac.ir (R. Kamali).

https://doi.org/10.1016/j.engappai.2025.110463
Received 14 February 2024; Received in revised form 21 January 2025; Accepted 26 February 2025
Available online 11 March 2025

benefited from this data-driven approach, such as flow control (Tang et al., 2020; Rabault et al., 2019), turbulence modeling (Duraisamy et al., 2019), and super-resolution (Fukami et al., 2019, 2021).

In the context of deep learning-based reduced-order models, the autoencoder plays a crucial role in dimensional reduction in deep learning-based reduced-order models. The encoder part of the autoencoder converts high-dimensional space to low-dimensional, namely latent variables, and extracts the most dominant features in the snapshots. The decoder then reconstructs the original snapshots from the latent variables (Brunton et al., 2020). Several studies showed that non-linear autoencoders could produce non-linear modes and had a lower error rate than POD modes (Murata et al., 2020; Xu et al., 2023). This higher accuracy is due to the autoencoder's ability to capture more complicated aspects of the data with the non-linear activation function.

In contrast to POD modes, autoencoder modes lack orthogonality and pose challenges to interpretability. A hierarchical convolutional autoencoder was employed by Fukami et al. (2020) to organize modes based on their contributions. The findings of their study demonstrated that the modes derived from a hierarchical convolutional autoencoder exhibit more influence on the flow field than those obtained from a standard convolutional autoencoder.

Another study applied a β-Variational Autoencoder (β-VAE) architecture to generate a robust, non-linear, and nearly orthogonal latent space for forecasting time series (Solera-Rico et al., 2024). Furthermore, the researchers revealed that the Long Short-Term Memory (LSTM) and easy attention transformer models outperform the Koopman with Non-linear Forcing (KNF) model. To accomplish this, the transformer model obtained an internal representation with distinctive frequency components. They also studied the sensitivity of hyperparameters in the VAE architectures (Wang et al., 2024). To achieve this, they used turbulent flow around a wall-mounted cylinder as a test case to find optimal values. Variational autoencoders have applications beyond mode decomposition. It has been used by researchers to identify the transonic buffet's most prominent characteristics (Wang et al., 2023). They constructed a novel physics-assisted variational autoencoder network that integrates the buffet state as a label within the classifier. This modification significantly improves the model's capacity to accurately capture characteristics that are particularly associated with buffet settings.

The discrete empirical interpolation and autoencoder neural networks were proposed for accurate predictions across a low-dimensional subspace (Moni et al., 2024). The approach partitions data into distinct groups, allowing for accurate predictions of underlying dynamics. Validation on two cases showed equivalent accuracy compared to a full-order model and is faster to evaluate. The method has potential for multidisciplinary design and optimization, making it a cost-effective alternative to traditional computational methods.

Maulik et al. (2021) combined recurrent neural networks (RNNs) and convolutional autoencoders. In order to capture the temporal behavior of the system, RNNs are used. Convolutional autoencoders, on the other hand, capture spatial features and patterns in the data. Parametric models were created by using a control parameter as input to LSTM. Their method proved to be more accurate than the POD-Galerkin method. Xu and Duraisamy (2020) used a multi-level neural network to create a parametric model as an alternative approach. Convolutional autoencoders were used for dimension reduction and temporal convolutional neural networks for dynamic evolution of latent spaces. For parametrization, a third-level neural fully connected network was used. In contrast to previous works where dimension reduction model and dynamic model were separate (pipeline models), Wu et al. (2021) introduced a joint-model that combines both models. Their method has a lower error value due to a decrease in error propagation.

The Latent Dynamics Network architecture (LDNets), a family of neural networks, uses data to learn and predict the evolution of space-dependent fields (Regazzoni et al., 2024). LDNets automatically encode the system state using latent scalar variables, without the need for an autoencoder. It generates output fields in a meshless manner, making training lightweight and improving generalization ability even in time-extrapolation regimes. Raj et al. (2023), investigating three reduced-order models, demonstrated that the deep learning-based methods, LSTMs, and temporal convolutional neural networks (TCNNs) performed better than the DMD in the more chaotic case. Qu et al. (2023) propose a nonlinear decomposition technique that utilizes dynamic mode decomposition and a non-linear mode decomposition autoencoder. A convolutional encoder network, a fully connected inner network, a convolutional shared decoder network, and several fully connected individual decoder networks comprise the approach. The non-linear decoders concentrate a significant amount of energy, leading to well-organized low-dimensional encodes. The authors proved the suitability of their approach for dynamic modeling and achieved superior reconstruction accuracy with a reduced number of modes.

1.1. Contributions

In dimension reduction, Convolutional Neural Networks (CNN) have been widely used to extract hierarchical features. They are, however, limited in their ability to handle spatial hierarchies and global features. Traditional CNNs are constrained by fixed kernel sizes, making it challenging to capture information at multiple scales. A multi-scale CNN was employed by Hasegawa et al. (2020b) in order to address these limitations. By incorporating layers with different kernel sizes, these architectures can extract features at different spatial resolutions. As a result, the model is able to identify intricate details and larger structures in the input data more easily. Nevertheless, multi-scale approaches can result in redundant information flow, e.g., similar low-level features can be extracted multiple times at different levels.

Convolutional neural networks have become powerful tools for extracting features. However, they can be limited in their performance and ability to generalize. One major drawback is that they treat all parts of an input image equally, ignoring potential variations in importance. This uniform approach can lead to loss of information and a lack of interpretability. To overcome this limitation, attention mechanisms have been proposed (Guo et al., 2022). These mechanisms focus on the most important regions of an image while disregarding irrelevant parts, improving the accuracy, robustness, and interpretability of CNNs. Additionally, attention mechanisms assist in minimizing computational costs and memory usage. In this context, Wu et al. (2021) implemented a self-attention mechanism into a joint model to improve fluid field forecasting. Although attention mechanisms have been successfully implemented in a variety of disciplines, including polyp segmentation (Sushama and Menon, 2023), additional research is needed to obtain a comprehensive understanding and refine attention mechanisms in the context of convolutional autoencoders and mode decomposition in fluid mechanics.

In this regard, our main contributions are:

- Implemented an involutional layer and multidimensional channel-wise and spatial-wise attention to mitigate the reconstruction loss in convolutional autoencoders. The addition of the involutional layer enhances the ability to capture spatial relationships, while the multidimensional CBAM improves attention mechanisms across channels, height, and width, thus increasing the process of selecting features. Our methodology aims to integrate these two components in order to effectively overcome the constraints of traditional convolutional autoencoders. This integration promises to provide a more sophisticated and efficient method for reducing reconstruction losses.
- we have made progress by creating an LSTM-attention model. This unique approach aims to explore the future dynamics of fluid flow fields with increased accuracy and comprehensibility. The objective of our work is to expand the limits of forecasting approaches with the goal of achieving exceptional performance and reliable predictions in intricate fluid flow scenarios.

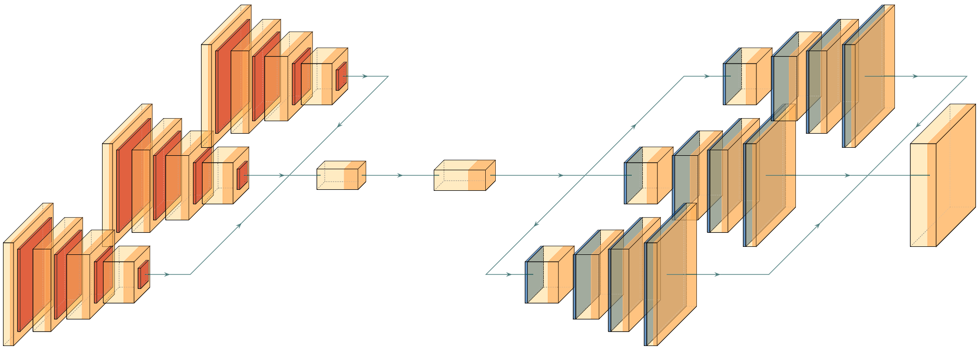
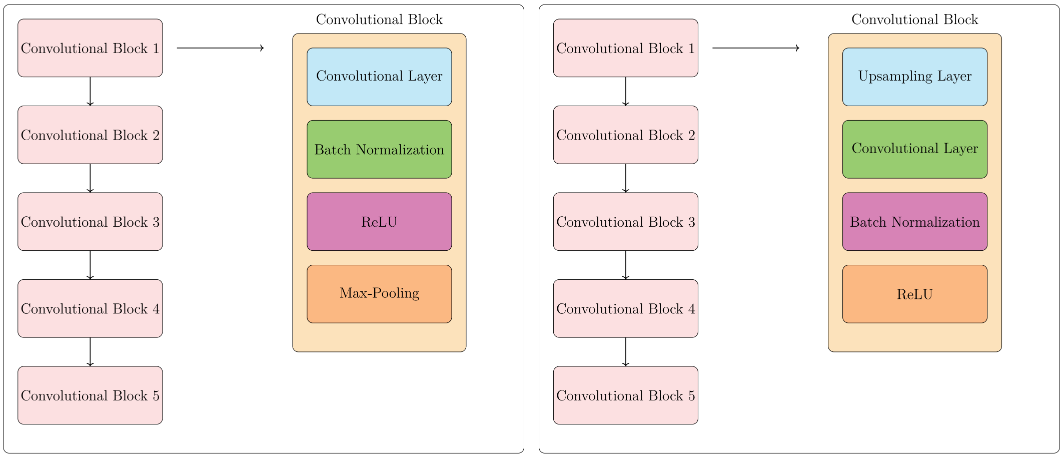


Fig. 1. Schematics of the base model in the present study.



(a) Encoder

(b) Decoder

Fig. 2. Architecture of convolutional blocks in encoders and decoders.

In the next section, we will provide an elaborate analysis of our suggested approach, explaining the distinctive elements and methods entailed. In addition, we will present the dataset used in this study, including an analysis of its features and significance to our methodology. Next, we will examine the outcomes derived from our experiments, emphasizing significant discoveries and performance indicators. Finally, we will end the investigation by summarizing the contributions, evaluating the implications of the findings, and suggesting possible avenues for further research.

2. Mode decomposition methodology

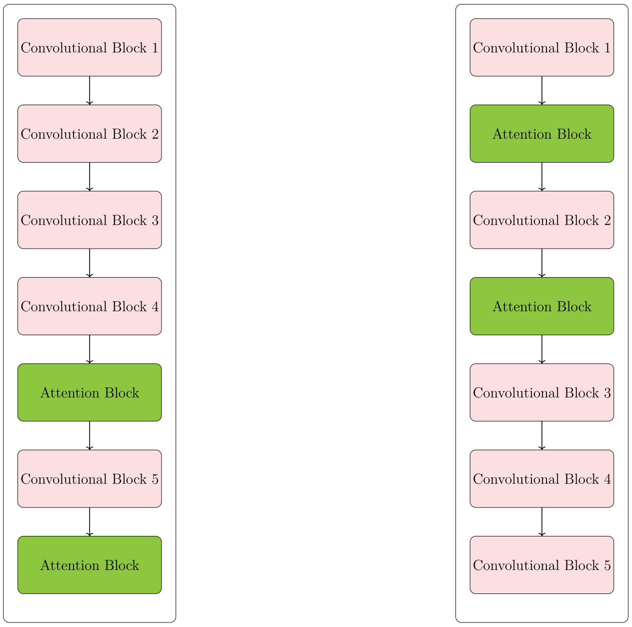
The Convolutional Autoencoder (CAE) is a highly effective unsupervised learning model widely used to reduce dimensionality in fluid mechanics problems. It accomplishes this by compressing data into a lower-dimensional latent space and subsequently reconstructing the original data from the compressed representation. The present study proposes using an augmented autoencoder that integrates the advantages of convolutional neural networks (CNNs) with attention mechanisms. Therefore, the model has the capability to extract features that have complementary strengths. The encoder for the base model, as depicted in Fig. 1 and influenced by Hasegawa et al. (2020b), has three convolutional branches with varying kernel sizes of (3×3), (5×5), and (9×9). The branches merge at the bottleneck to form a latent space. Furthermore, the decoder performs a similar function to the encoder, but it is specifically designed to increase the resolution of the latent spaces. The decoder utilizes three upsampling convolutional branches, each employing a distinct kernel size of (3×3), (5×5), and (9×9).

2.1. *Encoder-decoder convolutional branch*

The convolutional branch of the encoder uses five convolutional layers followed by batch normalization, ReLU activation, and max-pooling, which is shown in Fig. 2(a). With the help of the convolutional branch, local features are extracted from the input data. In each layer, the number of channels gradually increases in order to capture higher-level features. The convolutional layer in CNNs is composed of small filters that are applied across the input to extract local patterns. As a result of this local feature extraction, CNNs are capable of recognizing low-level features in inputs. Through multiple layers, CNNs can learn higher-level features by combining lower-level features. By using this hierarchical representation, the model is able to encode complex patterns in the latent space.

Moving over to the decoder, each convolutional branch consists of five convolutional blocks, as illustrated in Fig. 2(b). Each block consists of an upsampling layer, a convolutional layer, a batch normalization layer, and a ReLU activation function. The last stage is merging all results and feeding them into a convolutional layer with a kernel size of three, which aids in reconstructing the inputs. This decoder architecture ensures a comprehensive capture and combination of features at various scales, contributing to the faithful reconstruction of the input data.

To improve the accuracy of reconstruction, attention mechanisms were selectively incorporated into the model, building upon the earlier disclosed encoder–decoder architecture. More precisely, attention blocks were incorporated into the last two levels of the encoder and the initial two layers of the decoder, as shown in Fig. 3. Two attention models were used for comparison: one employing Channel-wise



(a) Encoder (b) Decoder

Fig. 3. Position of attention blocks in the encoder and decoder parts of the convolutional autoencoder.

Attention Mechanism (CBAM) and the other employing an adaptive attention block. This augmentation attempts to enhance the encoding and decoding processes by selectively prioritizing important areas of the input data, utilizing either CBAM's channel and spatial attention or the flexibility provided by the involution layer in the adaptive attention block. Adding attention mechanisms to the hierarchical feature extraction done by convolutional layers makes the representation of the input data in the latent space more accurate and aware of its surroundings. The deliberate integration of attention mechanisms in the model serves the purpose of maximizing both the quality of clustering and the accuracy of reconstruction. This approach offers a comprehensive method for learning features and representing data within the autoencoder architecture.

2.1.1. *Attention block: Channel-wise attention mechanism*

The Channel-wise Attention Mechanism (CBAM), proposed by Woo et al. (2018), has been effectively included in the augmented autoencoder architecture to strengthen its ability to capture important features and improve overall reconstruction accuracy, as shown in Fig. 4. CBAM operates by dynamically adjusting feature maps across channels, giving priority to information based on its importance. CBAM consists of two separate attention modules, namely the Channel Attention Module (CAM) and the Spatial Attention Module (SAM), which enhance the encoding and decoding processes. This module's capabilities can be expressed in the following way:

$$F' = M_c(F) \otimes F \tag{1}$$

$$F'' = M_s(F') \otimes F' \tag{2}$$

Following the refinement process, we obtain F'', with $\otimes$ representing element-wise multiplication. The channel attention values are then broadcast along spatial dimensions and reciprocally. The features derived from the spatial attention module undergo element-wise multiplication with the input feature map, as illustrated in Fig. 4.

The Channel Attention Module evaluates the significance of each channel by generating a channel-wise attention map, as illustrated in Fig. 5. This process involves applying global pooling to the feature map, followed by two fully connected layers featuring a gating mechanism. The resulting attention map is then multiplied element-wise with the original feature map, amplifying informative channels while suppressing less relevant ones. This mechanism enables the model to selectively concentrate on crucial patterns and features throughout both the encoding and decoding phases. Notably, as highlighted by Woo et al. (2018), simultaneous integration of average-pooled and maximum-pooled features during spatial information aggregation has been identified as a strategy yielding superior results. It is noteworthy that the formula governing this attention mechanism is based on the following formula:

$$\begin{aligned} M_c(F) &= \sigma[\mathrm{MLP}(\text{Avg-Pool}(F)) + \mathrm{MLP}(\text{Max-Pool}(F))] \\ &= \sigma[W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))] \end{aligned} \tag{3}$$

Where the sigmoid function is indicated by σ. Besides, Multi-Layer Perceptron (MLP) weights W_0 and W_1 are shared for both inputs, and W_0 is followed by the ReLU activation function.

Simultaneously, the Spatial Attention Module evaluates the significance of spatial locations within each channel, as illustrated in Fig. 6. It employs a similar mechanism, utilizing global max pooling and average pooling, followed by two fully connected layers. The obtained attention map is multiplied element-wise with the original feature map, enhancing the representation of spatially critical regions, as formulated as follows:

$$\begin{aligned} M_s(F) &= \sigma[f^{7\times7}(\text{Avg-Pool}(F); \text{Max-Pool}(F))] \\ &= \sigma[f^{7\times7}([F^s_{avg}; F^s_{max}])] \end{aligned} \tag{4}$$

Where $f^{7\times7}$ represents a convolution operation with a filter size of 7×7 and σ stands for the sigmoid function.

By incorporating both channel-wise and spatial attention mechanisms, CBAM allows the autoencoder to adaptively attend to the most relevant information, leading to improved feature extraction and reconstruction accuracy.

2.1.2. *Attention block: Adaptive attention mechanism*

Fig. 7 illustrates our proposal of a novel attention mechanism, which involves an involution layer followed by traversal of a multidimensional CBAM. The design is improved by including a skip connection that adds the attention-processed features to the original input. This distinctive attention block attempts to enhance the encoding and decoding processes by utilizing the versatility offered by the involution layer and the focus on both channel-wise and spatial-wise aspects facilitated by CBAM.

The involution layer, as the initial step, allows the model to dynamically adjust its receptive field, capturing intricate patterns across different spatial scales. This adaptability is especially crucial for scenarios where diverse feature sizes are present in the input data. After the involution layer, the multidimensional CBAM is used, which adds attention mechanisms that work both channel-wise and spatial-wise. This dual attention approach enables the model to emphasize critical channels and spatial regions concurrently, enhancing the discrimination of salient features. Moreover, the incorporation of a skip connection ensures that the attention-processed features are not isolated but seamlessly integrated with the original input. This strategy fosters a more holistic consideration of both attention-processed and unaltered information, contributing to a comprehensive and refined representation of the input data in the latent space.

Proposed by Li et al. (2021), the involution layer has garnered significant attention for its adeptness in capturing intricate spatial dependencies with remarkable efficiency, diverging from the constraints imposed by predefined filter sizes. A pivotal distinction between involution layers and conventional convolutional layers lies in their methodologies for spatial feature extraction. While conventional convolutional layers adhere to fixed filters, involution layers dynamically determine filter sizes, adapting to the distinct characteristics of each

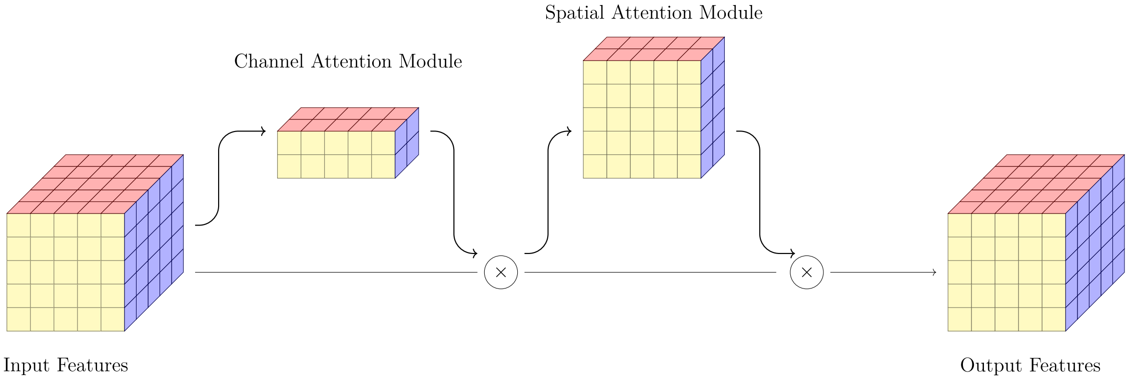


Fig. 4. Schematics of channel-wise attention mechanisms (CBAM), proposed by Woo et al. (2018).

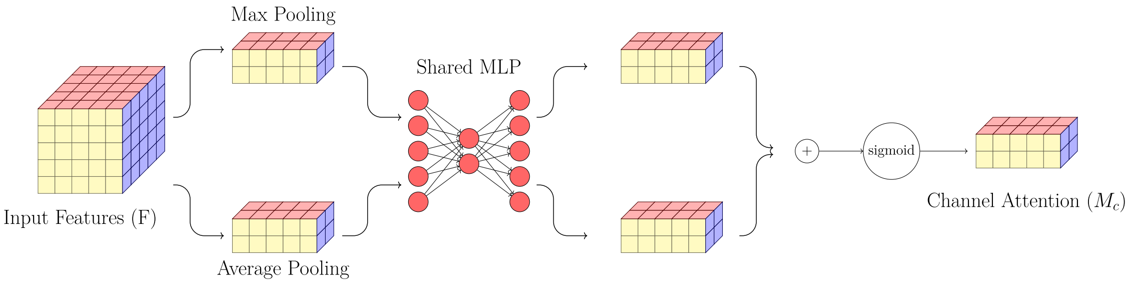


Fig. 5. Architecture of channel attention mechanisms in CBAM.

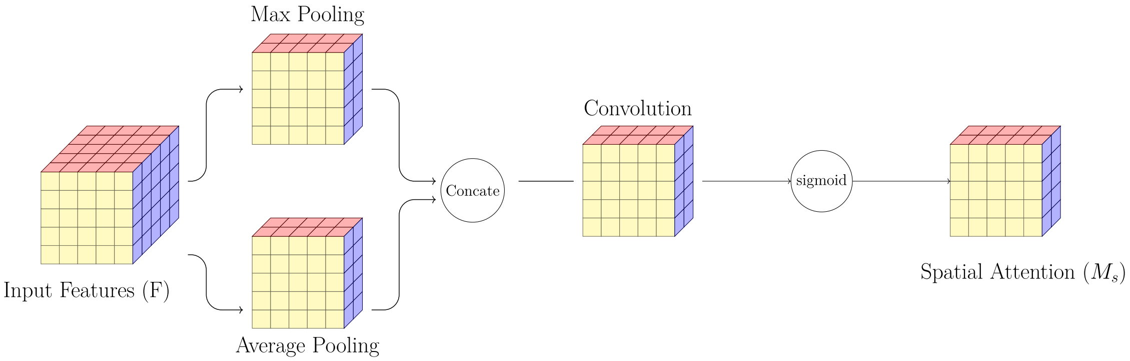


Fig. 6. Architecture of spatial attention mechanisms in CBAM.

channel. This inherent flexibility empowers the layer to aptly capture diverse spatial patterns, particularly advantageous in scenarios where features exhibit significant variation in scale. The involution layer offers notable advantages, including heightened adaptability to different input data characteristics, augmented flexibility in learning spatial dependencies, and a diminished reliance on manually tuned filter sizes.

We utilized the Multidimensional Channel Attention Module (MCAM) to enhance feature selection inside our attention mechanism. This module, seen in Figs. 8, 9, 10, and 11, was integrated after the involution layer. Although standard CBAM is effective at extracting features on a per-channel basis, it tends to neglect the underlying interconnections among the channel, height, and width dimensions. MCAM addresses this disparity by expanding attention processes to incorporate spatial dimensions, including both height and width, in addition to the channel dimension. This multidimensional technique is accomplished by utilizing three specialized divisions, each concentrating on channel, height, and width separately, to capture detailed patterns across different spatial scales.

The MCAM utilizes trainable convolutional layers for each dimension, dynamically modifying filter sizes according to the distinct attributes of channel, height, and width. MCAM provides a more thorough examination of essential features in the input data by combining attention from these three aspects. Using sigmoid activation functions and batch normalization also makes sure that attention weights are adjusted, which leads to a more complex and context-aware representation in the latent space. The incorporation of channel, height, and width considerations in this attention mechanism offers a flexible method for extracting features. This method shows potential for adapting to various input data characteristics and achieving better performance in identifying intricate patterns within neural network architectures.

3. Modes' dynamic methodology

To forecast future sequences using the latent space representation derived from a convolutional autoencoder, we suggest a reliable combination of multi-head attention and bidirectional LSTM procedures. The latent space, obtained by encoding sequential data using

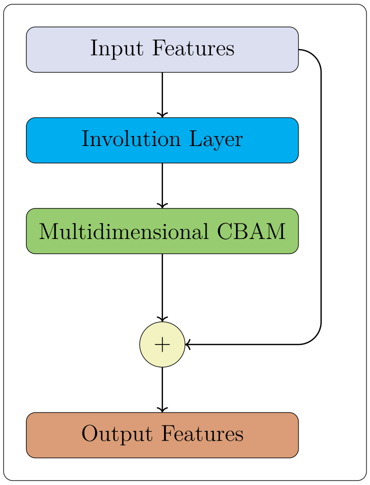


Fig. 7. Architecture of the proposed adaptive attention block.

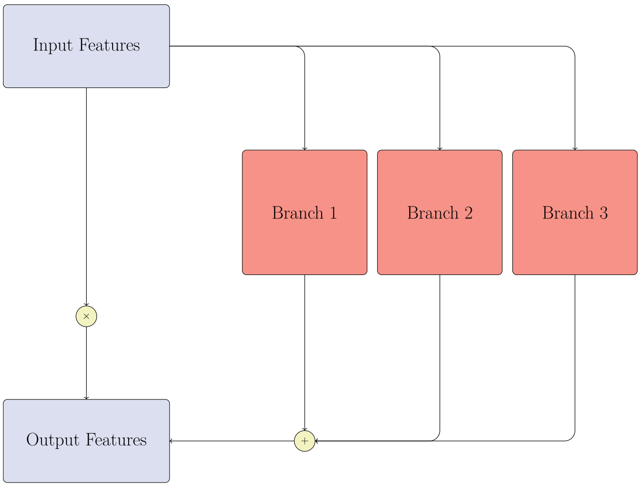


Fig. 8. Architecture of the multidimensional convolutional attention block.

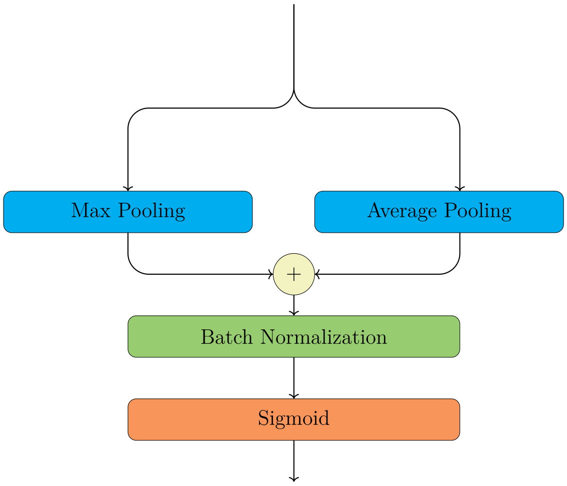


Fig. 9. Branch 1 of the multidimensional convolutional attention block.

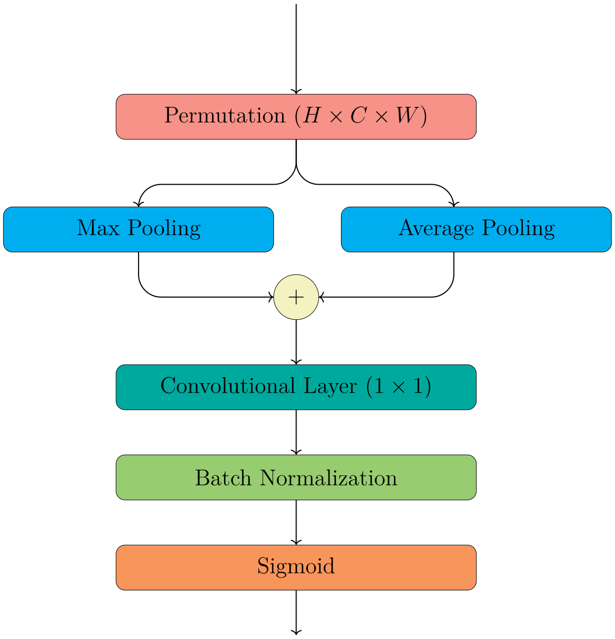

Permutation $(H \times C \times W)$
Max Pooling
Average Pooling
+
Convolutional Layer (1×1)
Batch Normalization
Sigmoid

Fig. 10. Branch 2 of the multidimensional convolutional attention block.

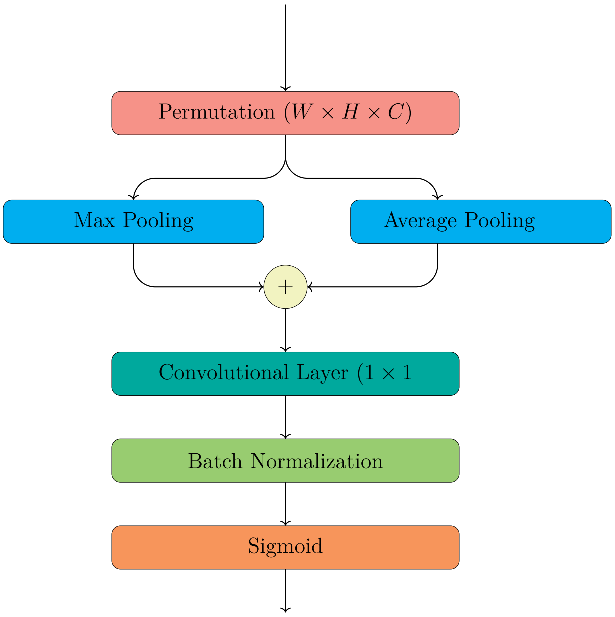

Permutation $(W \times H \times C)$
Max Pooling
Average Pooling
+
Convolutional Layer $(1 \times 1$
Batch Normalization
Sigmoid

Fig. 11. Branch 3 of the multidimensional convolutional attention block.

a convolutional autoencoder, contains essential information regarding the underlying patterns. In order to utilize the predictive capabilities inherent in this hidden dimension, we propose an innovative method that merges the advantages of multi-head attention and bidirectional LSTMs. Multihead attention facilitates the model's ability to concentrate on several aspects of the hidden space concurrently, hence providing a more sophisticated comprehension of intricate interconnections. Simultaneously, the bidirectional LSTM architecture analyzes the underlying information in both forward and backward directions, capturing extensive temporal connections. The combination of these two components creates a powerful forecasting framework that is capable of

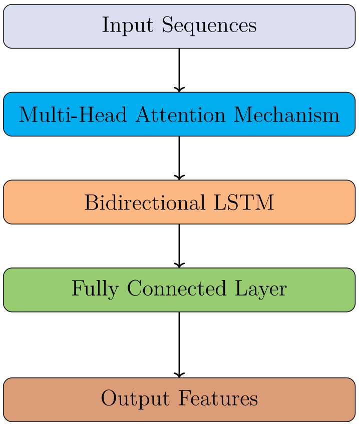


Fig. 12. Proposed fusion of multi-head attention and bidirectional LSTM for forecasting future sequences based on convolutional autoencoder latent space.

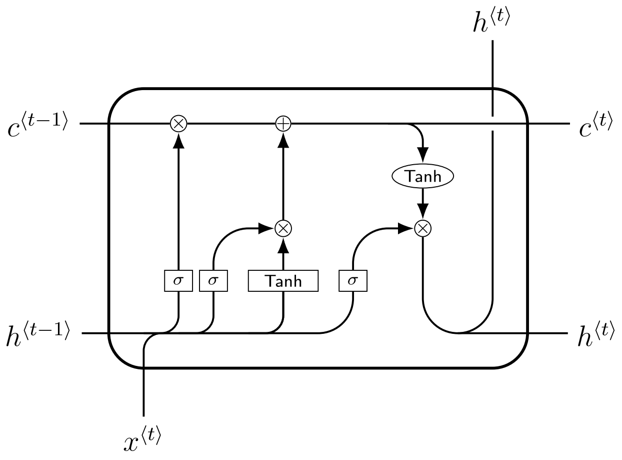


Fig. 13. Long-Short-Term Memory structure.

effectively managing complex sequential data. Fig. 12 depicts the suggested amalgamation, demonstrating the incorporation of multi-head attention with bidirectional LSTM for improved predictive modeling.

3.1. Long-Short Term Memory (LSTM)

LSTMs are highly effective at modeling sequential data because they overcome the vanishing gradient problem in traditional recurrent neural networks. LSTMs use a complex gating mechanism to selectively retain or discard information over time (Fig. 13). There are three main gates in LSTMs: the input gate (i_t), the forget gate (f_t), and the output gate (o_t). These gates are responsible for regulating the flow of information and controlling the memory state within the LSTM cell. By using sigmoid activation functions, gates determine how much information can pass through by producing values between 0 and 1.

3.1.1. Input gate

Input gates determine how much candidate memory should be added to the cell state (C_{t-1}). It takes into account the current input (x_t) and the previous hidden state (h_{t-1}). Cell state is updated based on sigmoid activation, which helps determine the relevance of new input information. Values close to 0 indicate that the new input is mostly ignored, while values close to 1 indicate that the new input is important to the current state of the cell.

3.1.2. Forget gate

For the current time step, the forget gate controls how much information is retained from the previous cell state (C_{t-1}). Additionally, it takes into account the current input (x_t) and the previous hidden state (h_{t-1}). Based on the output of the gate, which ranges from 0 to 1, the previous state of the cell is either forgotten or retained. A value close to 0 implies that most of the previous cell state is forgotten, while a value close to 1 implies most of the previous cell state is kept.

3.1.3. Output gate

The output gate controls how much the hidden state (h_t) is influenced by the cell state (C_t) in the current time step. The current input (x_t) is taken into account as well as the previous hidden state (h_{t-1}). To produce the final output of the LSTM cell, the hidden state is scaled by this gate's output. When the gate's value is close to 1, the cell state has a strong influence on the output, while when it is close to 0, its influence is diminished.

Through the use of these gating mechanisms, LSTMs are able to retain and propagate relevant information while discarding irrelevant or noisy information. As a result, LSTMs can capture long-range dependencies and overcome issues like the vanishing gradient problem that plagues traditional RNNs. The key equations of LSTM operations include:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{5}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{6}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{7}$$

$$\tilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{8}$$

$$C_t = f_t \odot \tilde{C}_{t-1} + i_t \odot \tilde{C}_t \tag{9}$$

$$h_t = o_t \odot \tanh(C_t) \tag{10}$$

3.1.4. Bidirectional LSTMs

Bidirectional LSTMs, in addition to regular LSTMs, analyze the input sequence by considering both the forward and backward directions. The bidirectional structure of the model enables it to gather information from both previous and future time steps, hence improving its capacity to comprehend and depict the sequential relationships within the data. The main concept is to combine the hidden states obtained from both the forward and backward passes, resulting in a more inclusive representation of the input sequence.

By employing bidirectional LSTMs, the model is able to not only comprehend long-range dependencies but also utilize knowledge from future contexts, hence enhancing its effectiveness in representing sequential data. Bidirectional LSTM operations entail the processing of the input sequence in both forward and backward directions, followed by the concatenation of the resulting hidden states.

$$\overrightarrow{h_t} = \text{Forward LSTM}(x_t, \overrightarrow{h_{t-1}}) \tag{11}$$

$$\overleftarrow{h_t} = \text{Backward LSTM}(x_t, \overleftarrow{h_{t+1}}) \tag{12}$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{13}$$

Here, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ represent the hidden states from the forward and backward passes, respectively. The final hidden state h_t is obtained by concatenating these two hidden states.

3.2. Self-attention

In recent years, the utilization of self-attention has brought about a significant transformation in the fields of natural language processing and computer vision. By employing this method, models are able to effectively capture complex relationships and contextual dependencies

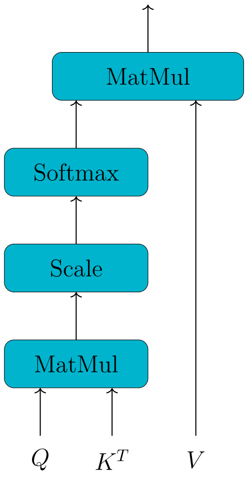


Fig. 14. Schematics of scaled-dot product attention.

among elements included in a sequence. The self-attention mechanism is defined as follows (Vaswani et al., 2017):

$$\text{Attention}(Q,K,V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V \tag{14}$$

Q, K, and V are querkey, key, and value matrices, respectively, which are outputs of three linear layers with the same input. The structure of the self-attention mechanism is shown in Fig. 14.

3.3. Multi-head attention

In multi-head attention, multiple self-attention mechanisms are used simultaneously to focus on different aspects of input data (Fig. 15). The model is able to emphasize different aspects of input during processing by learning different attention weights for each "head". The final result is obtained by concatenating the outputs of all attention functions and passing them through a linear layer.

With scaled dot-product attention, the input sequence can be transformed into query, key, and value vectors to compute the attention mechanism within each head. Attention scores are calculated by computing the dot product between query and key vectors. This is scaled by the square root of the key vector dimension. The resulting attention scores are then softmaxed to obtain weights, which are used to linearly combine the value vectors, producing the output for that head. These operations can be formulated as follows:

The formula for multi-head attention is as follows:

$$\text{MultiHead(Q,K,V)} = \text{Concat}(head_1, head_2, \dots, head_n)W^o \tag{15}$$

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{16}$$

Where $head_i$ represents the output of the ith attention head, h is the total number of attention heads, and Wo is a learnable weight matrix applied after concatenation to obtain the final output.

4. Benchmarks

In order to evaluate the capabilities of our novel deep learning approach for producing a reduced-order model, we present a study involving two test cases. The following describes these test cases in more detail.

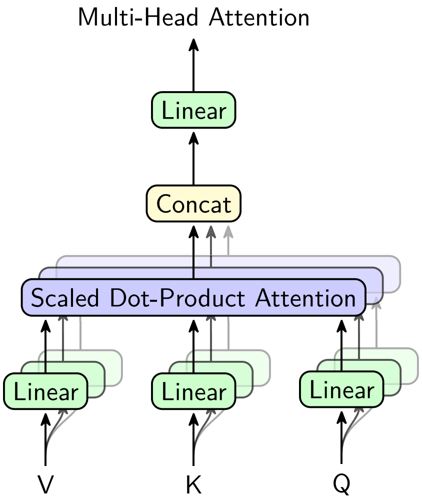


Fig. 15. Schematics of multi-head attention.

4.1. Flow over a cylinder at a Reynolds number of 100

As a first example, we investigate flow over a circular cylinder at a Reynolds number of 100, as depicted in Fig. 16. To generate a dataset, an area of sampling with a given dimension is chosen for extracting velocity components in the x and y directions. With second-order discretization for convective terms and first-order discretization for temporal terms, OpenFOAM numerically solves the incompressible Navier–Stokes equations to simulate flow over a cylinder. In addition, a flow field with a time step of 0.001 was simulated for 100 s, and 2500 snapshots from the sampling area were extracted for the following 25 s.

4.2. Isotropic turbulence

In the second test case, the dataset employed in this study is derived from a direct numerical simulation of forced isotropic turbulence conducted on a 1024^3 periodic grid. The simulation was executed using a pseudo-spectral parallel code provided by Johns Hopkins Turbulence Databases (Minping et al., 2012). During the simulation, the viscous term was analytically integrated using an integrating factor, while other terms were handled through a second-order Adams–Bashforth scheme. To create a representative subset, we extracted 3000 snapshots, each with a size of 256 × 256, to capture the velocity components (u, v, and w) at the mid-plane of the simulation, encompassing a temporal range from 0 to 10 s.

5. Results and discussion

This section examines the evaluation of an adaptive attention mechanism to enhance the reconstruction of flow fields by utilizing a convolutional autoencoder. The investigation is conducted on two distinct datasets and compared with the CBAM approach employed by Woo et al. (2018). In this study, the size of the latent space was fixed to 10 for the flow over a cylinder case and 100 for the isotropic turbulence case. Although, We divided the dataset into 70% for training and 30% for testing.

The training process for all models was consistently carried out using PyTorch as the deep learning framework, Adam as the optimizer, and Mean Squared Error (MSE) as the loss function. In addition, a stopping criterion was developed to terminate the training process if no improvement was observed within the initial 100 epochs. The implementation of this standardized training approach guarantees a uniform and equitable comparison among the models being evaluated.

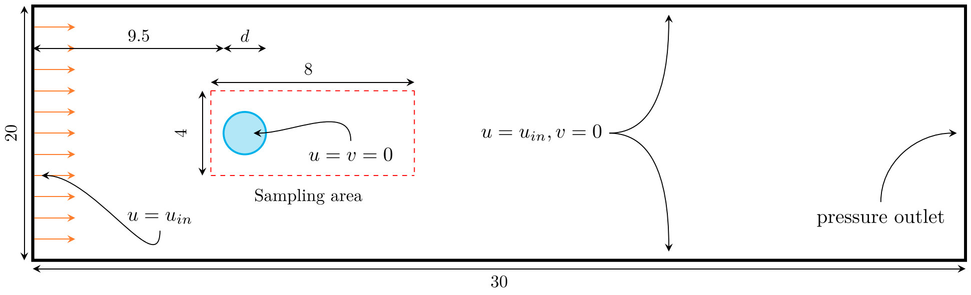


Fig. 16. Schematic of flow field over a cylinder and sampling area.

Table 1
The convolutional autoencoder's structure was designed for fluid flow over a cylinder.

Encoder		Decoder	
Layer	Output shape	Layer	Output shape
Encoder Input	(192, 384, 2)	Decoder Input	10
1st Conv. and ReLU	(192, 384, 8)	2nd Dense layer	4608
Batch Normalization	(192, 384, 8)	Reshape layer	(6, 12, 64)
1st Max Pooling	(96, 192, 8)	1st Upsampling	(12, 24, 64)
2nd Conv. and ReLU	(96, 192, 16)	6th Conv. and ReLU	(12, 24, 64)
Batch Normalization	(96, 192, 16)	Batch Normalization	(12, 24, 64)
2nd Max Pooling	(48, 96, 16)	2nd Upsampling	(24, 48, 64)
3th Conv. and ReLU	(48, 96, 32)	7th Conv. and ReLU	(24, 48, 32)
Batch Normalization	(48, 96, 32)	Batch Normalization	24, 48, 32)
3th Max Pooling	(24, 48, 32)	3th Upsampling	48, 96, 32)
4th Conv. and ReLU	(24, 48, 64)	8th Conv. and ReLU	(48, 96, 16)
Batch Normalization	(24, 48, 64)	Batch Normalization	(48, 96, 16)
4th Max Pooling	(12, 24, 64)	4th Upsampling	(96, 192, 16)
5th Conv. and ReLU	(12, 24, 64)	9th Conv. and ReLU	(96, 192, 8)
Batch Normalization	(12, 24, 64)	Batch Normalization	(96, 192, 8)
5th Max Pooling	(6, 12, 64)	5th Upsampling	(192, 384, 8)
Flatten layer	4608	10th Conv.	(192, 384, 2)
1st Dense layer	10		

Table 2
The convolutional autoencoder's structure was designed for isotropic turbulence.

Encoder		Decoder	
Layer	Output shape	Layer	Output shape
Encoder Input	(256, 256, 3)	Decoder Input	100
1st Conv. and ReLU	(256, 256, 8)	2nd Dense layer	4096
Batch Normalization	(256, 256, 8)	Reshape layer	(8, 8, 64)
1st Max Pooling	(128, 128, 8)	1st Upsampling	(16, 16, 64)
2nd Conv. and ReLU	(128, 128, 16)	6th Conv. and ReLU	(16, 16, 64)
Batch Normalization	(128, 128, 16)	Batch Normalization	(16, 16, 64)
2nd Max Pooling	(64, 64, 16)	2nd Upsampling	(32, 32, 64)
3th Conv. and ReLU	(64, 64, 32)	7th Conv. and ReLU	(32, 32, 32)
Batch Normalization	(64, 64, 32)	Batch Normalization	(32, 32, 32)
3th Max Pooling	(32, 32, 32)	3th Upsampling	(64, 64, 32)
4th Conv. and ReLU	(32, 32, 64)	8th Conv. and ReLU	(64, 64, 16)
Batch Normalization	(32, 32, 64)	Batch Normalization	(64, 64, 16)
4th Max Pooling	(16, 16, 64)	4th Upsampling	(128, 128, 16)
5th Conv. and ReLU	(16, 16, 64)	9th Conv. and ReLU	(128, 128, 8)
Batch Normalization	(16, 16, 64)	Batch Normalization	(128, 128, 8)
5th Max Pooling	(8, 8, 64)	5th Upsampling	(256, 256, 8)
Flatten layer	4096	10th Conv.	(256, 256, 3)
1st Dense layer	100		

Furthermore, we obtained the hyperparameters for our models via grid search. Table 1 presents the detailed results for the hyperparameters for the flow over a cylinder, while Table 2 presents the results for isotropic turbulence.

5.1. Reconstruction loss analysis

From the analysis of the root mean square error (RMSE), as defined in Eq. (17), of several convolutional autoencoder models applied to flow over a cylinder and isotropic turbulence, it is evident that attention mechanisms play a vital role in improving the performance of these models. This is demonstrated in Figs. 17 and 18.

$$\mathrm{RSME} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2} \tag{17}$$

The reconstruction loss is a crucial measure to evaluate how well the reconstructed output matches the input data. It offers useful insights into the effectiveness of various attentional methods. The experiment revealed that the autoencoder without an attention mechanism exhibited the lowest performance. This suggests that the model encountered

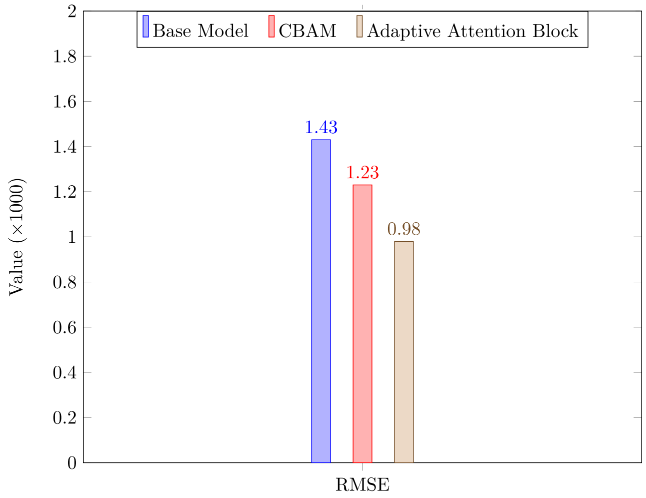


Fig. 17. Root mean square error of reconstructing the flow field for flow over a cylinder.

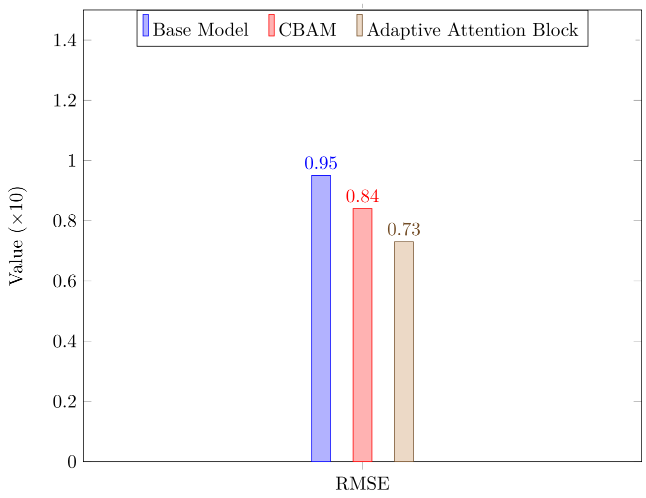


Fig. 18. Root mean square error of reconstructing the flow field for isotropic turbulence.

difficulties in capturing complex patterns and features during the encoding and decoding stages of the input data. Due to the inclusion of an attention mechanism, the autoencoder with CBAM attention partially addresses the issue. Compared to the standard autoencoder, this mechanism lets the model focus on only the important spatial and channel-wise information. This results in less reconstruction loss. Nevertheless, the adaptive attention mechanism surpasses both equivalents in terms of reconstruction loss. This implies that the adaptive attention mechanism is better at capturing and highlighting significant features in the input data, resulting in a more accurate reconstruction.

Analysis of the saliency map provides additional evidence of the efficacy of these attention mechanisms in Figs. 19 and 20 for flow over a cylinder and isotropic turbulence, respectively. According to the saliency maps, both the CBAM and adaptive attention methods focus on key elements of the flow field, such as the boundary layer and wake regions. Nevertheless, the adaptive attention mechanism offers a higher level of concentration on these regions, especially in cases where notable flow characteristics become apparent. This more focused attention emphasizes the higher capacity of adaptive attention to capture detailed spatial characteristics, which leads to improved reconstruction performance in comparison to CBAM.

The enhanced efficacy of the adaptive attention mechanism compared to CBAM can be attributed to the increased flexibility and adaptability introduced by the involution layer. Involution layers have the ability to alter their receptive fields according to the input, enabling the model to accommodate different patterns in the data and provide a representation that is more aware of the context. The integration of involution with the multidimensional CBAM enhances the attention mechanism, allowing the model to concentrate on spatial and channel-wise characteristics more effectively and flexibly. This flexibility is beneficial in situations where the significance of certain characteristics differs in the input data, leading to a more efficient process of encoding and decoding.

Fig. 21 depicts the variations over time in the magnitude of velocity for different models. All of the models provide remarkable accuracy for the flow over a cylinder. Nevertheless, while analyzing the case of isotropic turbulence, the baseline model exhibits a certain divergence from the outcomes obtained using computational fluid dynamics (CFD). The integration of attention processes, including adaptive attention, significantly reduces this disparity, which is particularly noticeable in peak regions. Furthermore, Figs. 22 and 23 display contour plots illustrating the magnitude of velocity at $t = 115$ seconds for flow over a cylinder and $t = 5$ seconds for isotropic turbulence, together with an error metric defined as the absolute difference between the predicted velocity and the velocity obtained using computational fluid dynamics ($V_{CFD} - V_{\text{predicted}}$). The findings demonstrate a uniform decrease in error across all areas when attention strategies are incorporated.

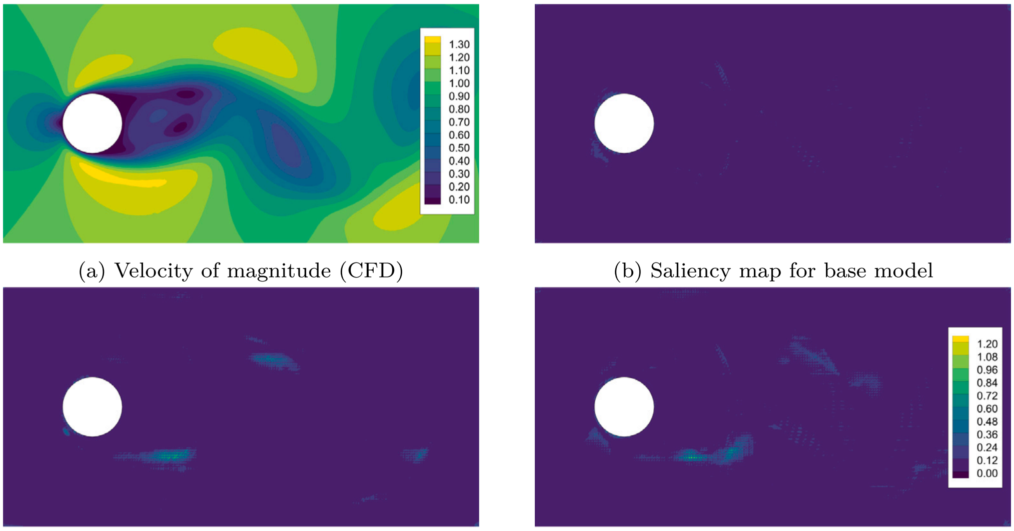
5.2. Latent space analysis

Fig. 24 illustrates the correlation analysis within the latent representations of convolutional autoencoders applied to flow over a cylinder. Although identical results were found for isotropic turbulence, only the correlation for flow over a cylinder is provided in this section for clarity. A perfect correlation is obtained for the base model, demonstrating a highly linear and deterministic link between latent dimensions. This shows a rigorous and consistent encoding of information, which raises the possibility of overfitting to specific patterns. Overfitting may impair the model's adaptability to diverse or noisy data, limiting its generalization capabilities.

In contrast, the addition of attention mechanisms, notably the adaptive attention block, significantly decreases the overall values of these correlations. This result means that the model, particularly when paired with an attention mechanism, can extract more distinguishing features from the input dataset. The lower correlation indicates a more nuanced and adaptive encoding, implying that the attention mechanism improves the model's capacity to detect and prioritize key aspects of the data. This improvement in how latent space is represented with attention mechanisms, especially adaptive attention, shows that the model can handle a wider range of data patterns. This supports its robustness and suggests that it could work better on different datasets.

An analysis of the standard deviations in latent dimensions for both the autoencoder and the attention-augmented autoencoders offers useful insights on the variability and spread of the learned representations. Fig. 25 illustrates the distribution of the standard deviation of the latent space for flow over a cylinder. The results indicate that the base model is notable for its smaller standard deviation, implying that it tends to provide a more restricted and condensed representation of the input data in its latent space. The lesser variability observed suggests that the base model has a reduced tendency to capture a wide range of complex or nuanced features, potentially leading to a more concentrated but restricted representation. Furthermore, the model's lowest maximum standard deviation indicates a reduced likelihood of capturing outliers or extreme fluctuations in the data.

Conversely, the CBAM attention mechanism is notable for having the biggest maximum standard deviation, which suggests its ability to encompass a wider variety of features in the latent space. The



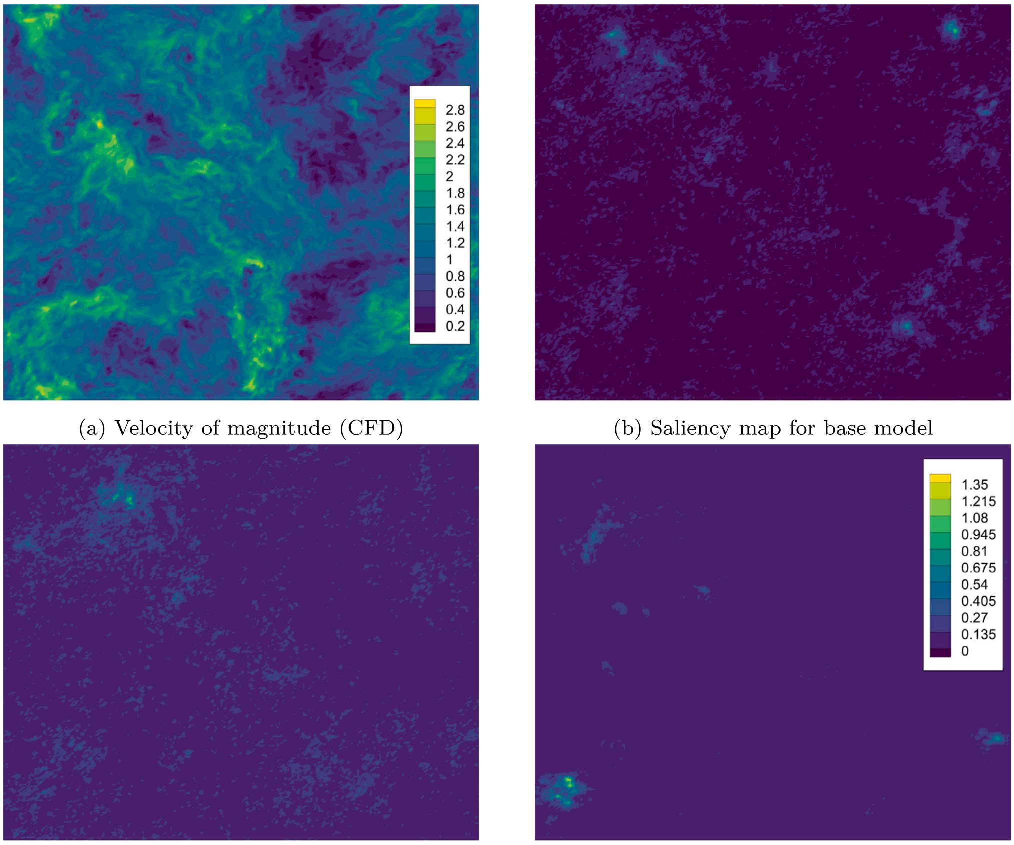
(a) Velocity of magnitude (CFD)

(b) Saliency map for base model

(c) Saliency map for CBAM

(d) Saliency map for adaptive attention block

Fig. 19. Comparison of saliency maps for different models applied to flow over a cylinder at t = 105 s.



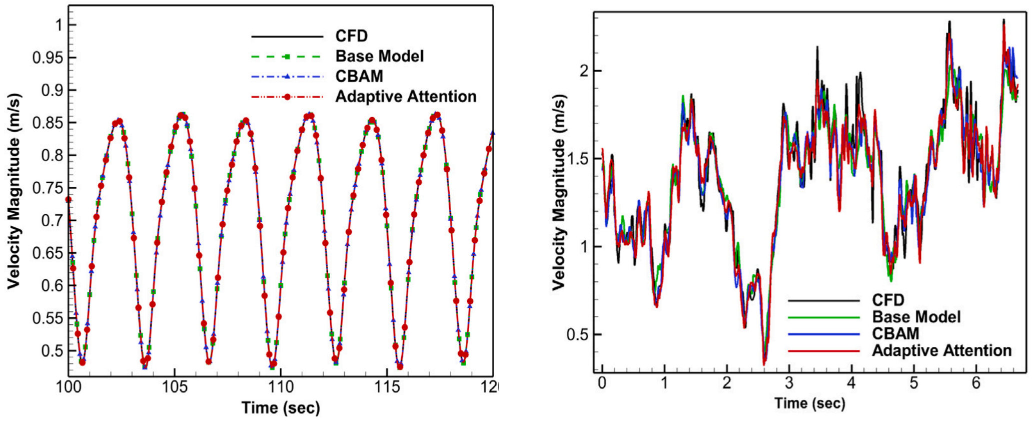
(a) Velocity of magnitude (CFD)

(b) Saliency map for base model

(c) Saliency map for CBAM

(d) Saliency map for adaptive attention block

Fig. 20. Comparison of saliency maps for different models applied to isotropic turbulence at t = 2 s.



(a) Flow over a cylinder (b) Isotropic Turbulence

Fig. 21. Variation of velocity magnitude over time at a particular point.

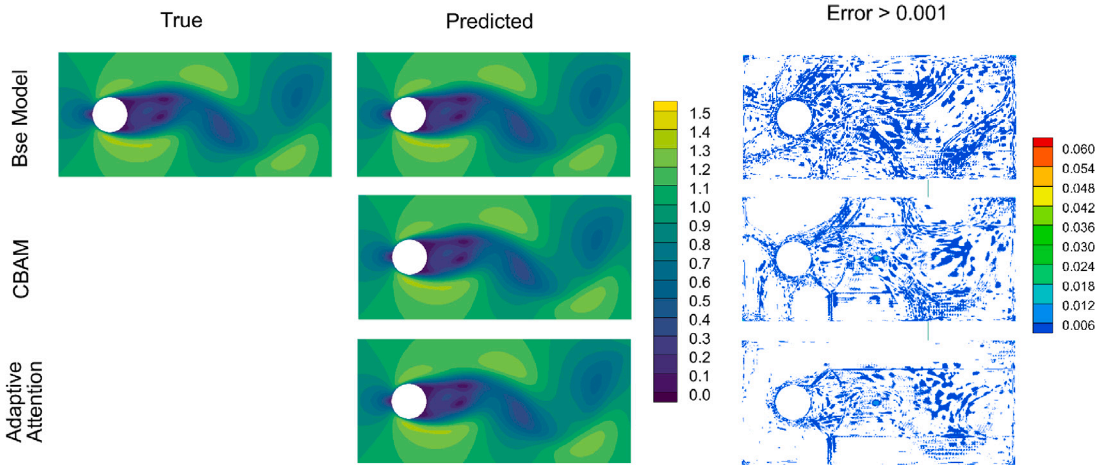


Fig. 22. Velocity magnitude contours and error maps for flow over a cylinder for different models.

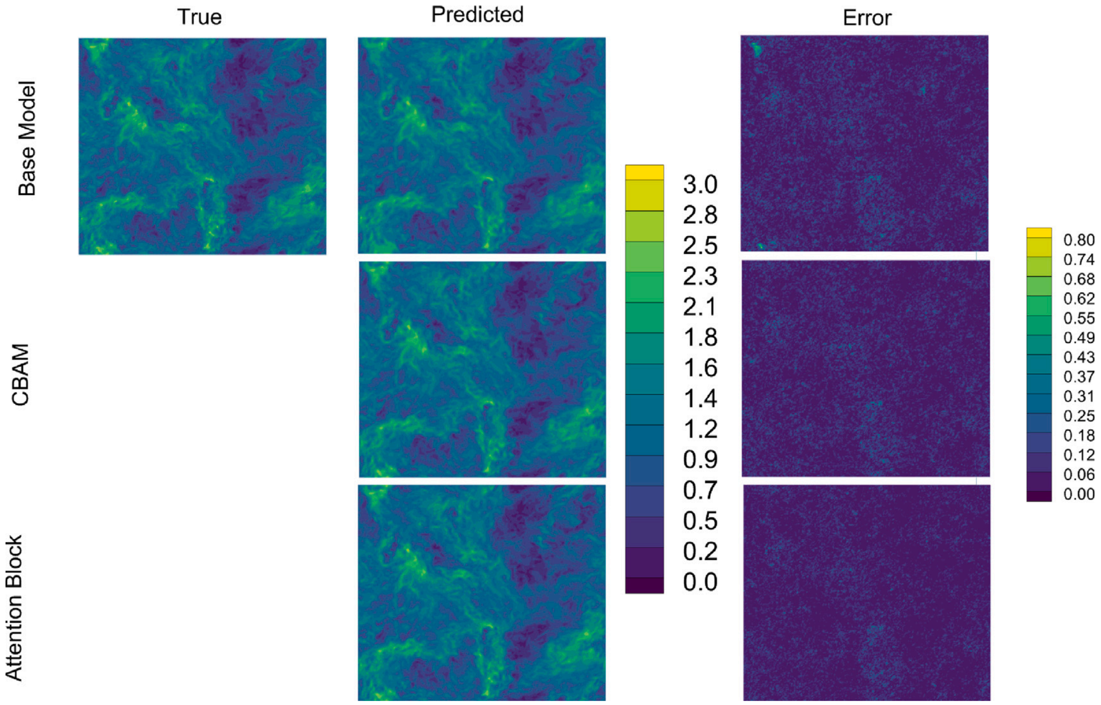
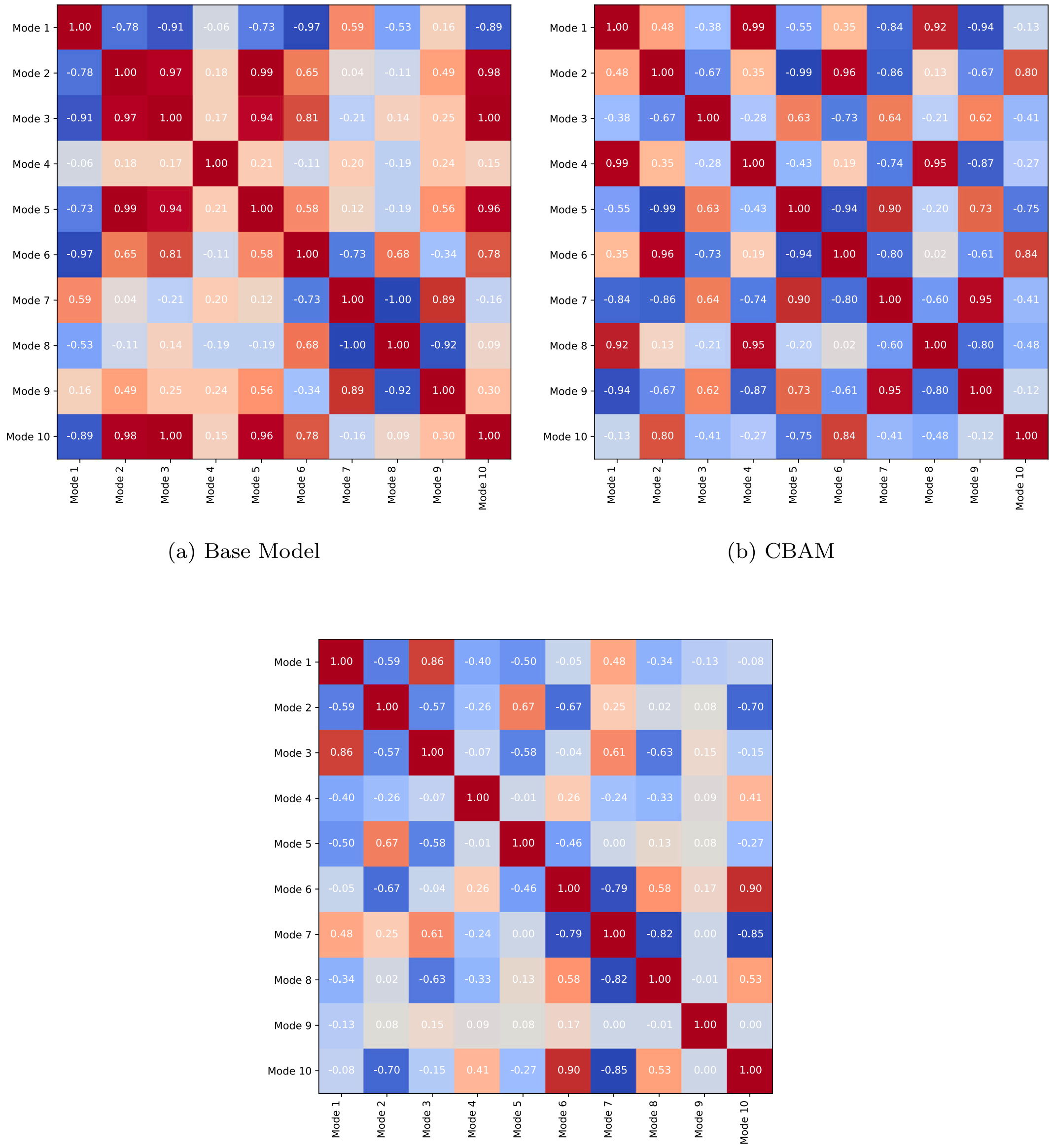


Fig. 23. Velocity magnitude contours and error maps for isotropic turbulence for different models.



(a) Base Model

(b) CBAM

(c) Adaptive Attention

Fig. 24. Correlation analysis of latent space for different models.

fact that there are data points in the boxplot that are outside of the lower quartile (Q1) and upper quartile (Q3) supports the idea that CBAM lets different and maybe unique characteristics be included. Nevertheless, the adaptive attention block model seems to achieve a harmonious equilibrium. The lack of points beyond Q1 and Q3 indicates that the adaptive attention block successfully captures a comprehensive representation without excessively expanding the latent space. This demonstrates that the selected attention mechanisms are well integrated, leading to a more refined feature extraction operation.

Furthermore, Fig. 26 illustrates the distribution of the standard deviation of isotropic turbulence. The significant variation in the standard deviation of the latent space distribution in the adaptive attention model indicates that these attention mechanisms result in a broader and more diversified range of features being recorded in the learned representations. In addition, the outlier points above the third quartile (Q3) in all models suggest that, irrespective of the attention mechanism employed, there are cases where the latent space demonstrates unexpectedly large standard deviations.

A K-means clustering was performed on the latent space representations of autoencoders to conduct the clustering study. KMeans clustering is a commonly employed technique for separating data points into ‘k’ clusters based on their similarity. Here, it was utilized to categorize latent space samples into clusters, with the number of clusters determined as an input parameter (in this case, 3 classes). K-means offers a direct method for evaluating the effectiveness of the autoencoder in categorizing data points into separate clusters in the latent space.

Moreover, the choice of the silhouette score for clustering analysis was based on its effectiveness in measuring the quality of clustering. The silhouette score measures the level of cohesion and separation among clusters, offering an indication of the clarity and distinctiveness of the clusters. A higher silhouette value suggests better clustering performance.

In the examination of flow around a cylinder in Fig. 27, the base model exhibited a relatively high silhouette score, indicating significant cohesiveness and separation within clusters. Interestingly,

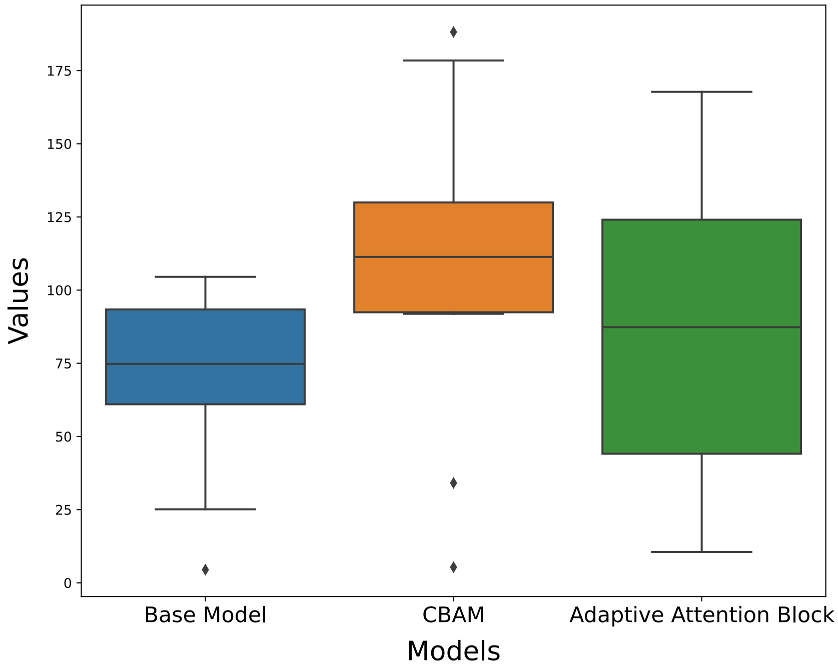


Fig. 25. Standard deviation analysis of latent space for flow over a cylinder.

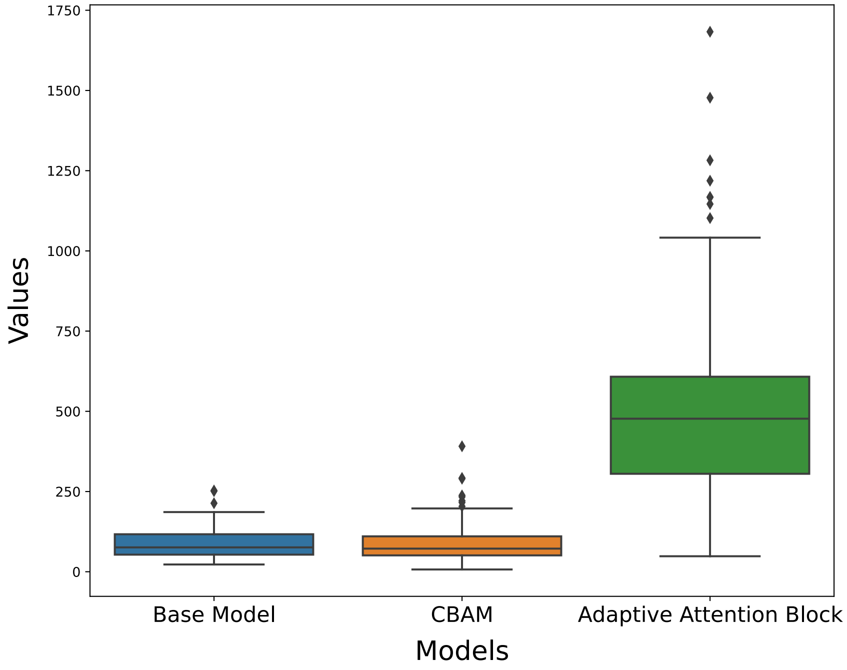


Fig. 26. Standard deviation analysis of latent space for isotropic turbulence.

both the CBAM and adaptive attention mechanism types demonstrated marginally lower silhouette scores. Although it may appear contradictory at first, it is vital to recognize that the clustering analysis focuses on the arrangement of data points in the latent space rather than on the measure of reconstruction loss. Both CBAM and the adaptive attention mechanism performed better than the basic model in terms of reconstruction loss. This suggests that although attention methods may not improve clustering performance in this dataset, they do contribute to enhancing the accuracy of the reconstructed data.

In Fig. 28, an alternative scenario occurred for isotropic turbulence. The silhouette score of the basic model was notably lower, suggesting a clustering structure that is less well-defined. However, it is worth noting that both the CBAM and adaptive attention mechanism types have shown enhancements in the silhouette score, with corresponding values of 0.14 and 0.18. Consequently, in the context of isotropic turbulence, attention processes have a favorable impact on the quality of clustering. This can potentially improve the autoencoder's capacity to accurately identify and group data points.

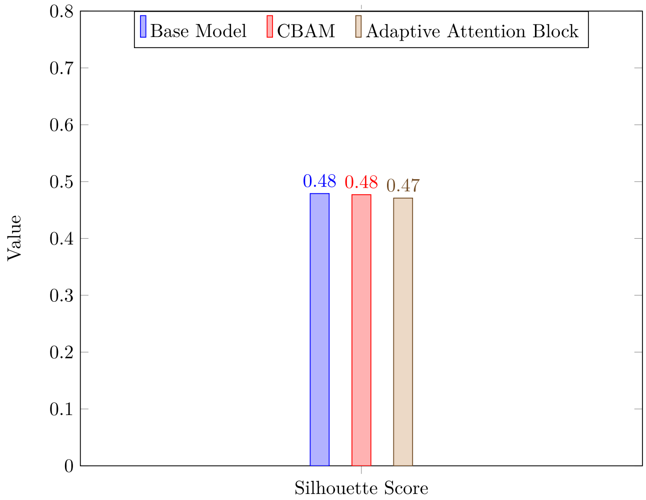


Fig. 27. Silhouette score evaluation of the autoencoder's latent space using K-means clustering for flow over a cylinder.

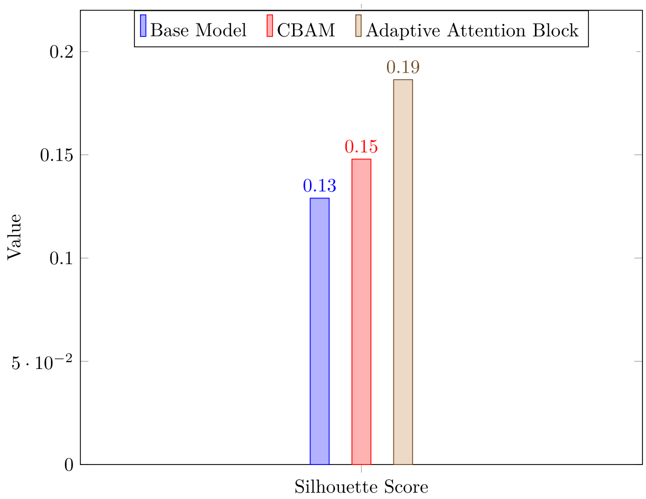


Fig. 28. Silhouette score evaluation of the autoencoder's latent space using K-means clustering for isotropic turbulence.

The unpredictable variation in the influence of attention processes on clustering performance across the two datasets highlights the fact that these improvements are very dependent on the specific characteristics of the datasets. This necessitates a more thorough investigation of the attributes of each dataset and the particular attributes that attention mechanisms give priority to. Although the flow over a cylinder does not show a substantial improvement in clustering performance with attention, the inclusion of attention mechanisms proves to be particularly advantageous for isotropic turbulence. This comprehensive comprehension highlights the need to take into account dataset-specific attributes when including attention processes in autoencoder structures, providing useful perspectives for enhancing model efficiency based on the inherent qualities of the data.

5.3. Latent space's dynamics forecasting

This section will utilize the latent spaces of a convolutional autoencoder as input for three different models: LSTM, bidirectional LSTM, and the proposed technique seen in Fig. 12. The proposed method combines bidirectional LSTM with a multi-head attention mechanism, enabling temporal analysis. Subsequently, the decoder will transform the projected latent space back to its original dimensions in order to incorporate it into the prediction.

The present study employed a temporal sequence comprising 60 previous latent spaces to forecast latent spaces one step forward. Recursive methods are employed to forecast 1000 upcoming time intervals following the training of models for flow over cylinder scenarios and 5 for isotropic turbulence. To assess the models, the predicted velocity magnitude is compared at point A ($x = 1.5$, $y = 0$) for flow over a cylinder located in the wake region and positioned at the center of the plane for isotropic turbulence. In this study, we tuned all models using the grid search method to optimize performance and then compared their optimal states.

The choice of model architecture plays a pivotal role in achieving accurate predictions for both short-term and long-term trends in time series forecasting. In Fig. 29, we present a comparative analysis of LSTM, bidirectional LSTM, and multi-head bidirectional LSTM for time series forecasting, focusing on the variation of velocity magnitude over time in the context of flow over a cylinder and isotropic turbulence. Notably, the LSTM model demonstrated limitations in effectively forecasting future sequences. The inherent challenge with LSTMs stems from their struggle to capture long-range dependencies in sequential data, leading to suboptimal performance in capturing intricate patterns and relationships.

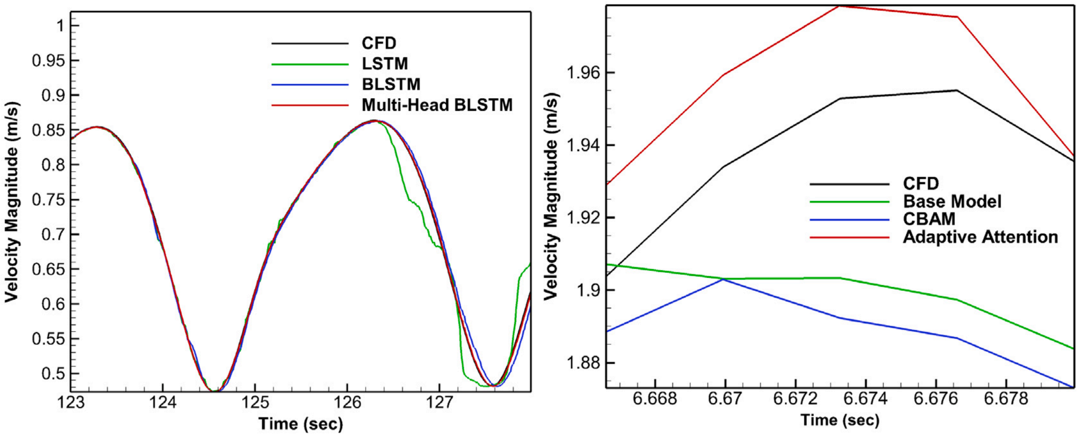
In response to these limitations, we explored the bidirectional LSTM architecture, which processes sequences in both forward and backward directions. This bidirectional processing enables the model to consider information from both past and future time steps. While bidirectional LSTM exhibited improvements over its unidirectional counterpart, it still faced challenges in handling complex dependencies.

In contrast, the combination of multihead attention followed by bidirectional LSTM emerged as a promising solution for enhancing future forecasting tasks. Multihead attention, recognized for its ability to simultaneously focus on different parts of the input sequence, provided the model with a nuanced understanding of complex relationships within the data. This attention mechanism effectively highlighted crucial features in the input, enabling the subsequent bidirectional LSTM to comprehensively capture both short- and long-term dependencies. The bidirectional processing further augmented the model's ability to leverage information from both past and future contexts. The synergistic effect of multihead attention and bidirectional LSTM proved instrumental in overcoming forecasting challenges encountered by traditional LSTMs, leading to significantly improved predictive performance. These findings underscore the efficacy of integrating attention mechanisms with bidirectional processing for robust future sequence forecasting across diverse applications.

6. Conclusion

In conclusion, the evaluation of an adaptive attention mechanism for enhancing the reconstruction of flow fields through a convolutional autoencoder provides valuable insights into the effectiveness of attention mechanisms in the context of fluid dynamics. The analysis of reconstruction loss, particularly in the comparison between models with CBAM and adaptive attention mechanisms, revealed that the adaptive attention mechanism outperforms its counterparts. This superiority can be attributed to the increased flexibility and adaptability introduced by the involution layer, enabling the model to capture and highlight significant features more accurately. The visualizations of velocity variations over time and contour plots further supported the idea that attention mechanisms contribute to better accuracy in reconstructing flow fields, especially in the case of isotropic turbulence.

The latent space analysis shed light on the model's ability to represent and generalize information. The adaptive attention mechanism exhibited nuanced and adaptive encoding, as indicated by lower correlations between latent dimensions. The standard deviation analysis



(a) Flow over a cylinder (b) Isotropic Turbulence

Fig. 29. Forecasting the future of flow fields by different models.

demonstrated that the adaptive attention mechanism strikes a balance between capturing a comprehensive representation and avoiding excessive expansion of the latent space. Additionally, the clustering analysis, while showing inconsistent results across datasets, highlighted the dataset-specific impact of attention mechanisms on the quality of clustering. The findings emphasize the importance of considering dataset characteristics when incorporating attention mechanisms into autoencoder structures.

Furthermore, the exploration of latent space dynamics forecasting using different models revealed that the combination of multihead attention and bidirectional LSTM proved to be a promising solution for accurate time series forecasting. This approach, by simultaneously focusing on different parts of the input sequence and processing information in both forward and backward directions, overcame the limitations of traditional LSTM and bidirectional LSTM models. The results underscore the potential of attention mechanisms to enhance the forecasting capabilities of models in dynamic systems, showcasing their versatility in capturing complex relationships within sequential data.

The present work has successfully shown the efficacy of the adaptive attention mechanism. However, further investigation is required to comprehensively understand alternative attention mechanisms across a broader range of flow conditions. Furthermore, this study demonstrates that short-term prediction is more effective for flow over a cylinder compared to isotropic turbulence due to the superior ability of attention mechanisms to detect patterns in periodic flows over time. Our idea for how to overcome this limitation involves the use of time derivatives or hybrid models that integrate attention mechanisms with stochastic models such as Gaussian processes or ensemble methods.

CRediT authorship contribution statement

Alireza Beiki: Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation. **Reza Kamali:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

Brunton, S.L., Noack, B.R., Koumoutsakos, P., 2020. Machine learning for fluid mechanics. Annu. Rev. Fluid Mech. 52, 477–508.

Duraisamy, K., Iaccarino, G., Xiao, H., 2019. Turbulence modeling in the age of data. Annu. Rev. Fluid Mech. 51, 357–377.

Fukami, K., Fukagata, K., Taira, K., 2019. Super-resolution reconstruction of turbulent flows with machine learning. J. Fluid Mech. 870, 106–120.

Fukami, K., Fukagata, K., Taira, K., 2021. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. J. Fluid Mech. 909, A9.

Fukami, K., Nakamura, T., Fukagata, K., 2020. Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data. Phys. Fluids 32 (9).

Geneva, N., Zabaras, N., 2020. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. J. Comput. Phys. 403, 109056.

Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., Zhang, S.-H., Martin, R.R., Cheng, M.-M., Hu, S.-M., 2022. Attention mechanisms in computer vision: A survey. Comput. Vis. Media 8 (3), 331–368.

Han, R., Wang, Y., Zhang, Y., Chen, G., 2019. A novel spatial-temporal prediction method for unsteady wake flows based on hybrid deep neural network. Phys. Fluids 31 (12).

Hasegawa, K., Fukami, K., Murata, T., Fukagata, K., 2020a. CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers. Fluid Dyn. Res. 52 (6), 065501.

Hasegawa, K., Fukami, K., Murata, T., Fukagata, K., 2020b. Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. Theor. Comput. Fluid Dyn. 34, 367–383.

Li, D., Hu, J., Wang, C., Li, X., She, Q., Zhu, L., Zhang, T., Chen, Q., 2021. Involution: Inverting the inherence of convolution for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12321–12330.

Lucia, D.J., Beran, P.S., Silva, W.A., 2004. Reduced-order modeling: new approaches for computational physics. Prog. Aerosp. Sci. 40 (1–2), 51–117.

Lumley, J.L., 1967. The structure of inhomogeneous turbulent flows. In: Atmospheric Turbulence and Radio Wave Propagation. Nauka, pp. 166–178.

Maulik, R., Lusch, B., Balaprakash, P., 2021. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. Phys. Fluids 33 (3).

Minping, W., Chen, S., Eyink, G., Meneveau, C., Johnson, P., Perlman, E., Burns, R., Li, Y., Szalay, A., Hamilton, S., 2012. Forced Isotropic Turbulence Data Set (Extended). Johns Hopkins Turbulence Databases.

Moni, A., Yao, W., Malekmohamadi, H., 2024. Data-driven reduced-order modeling for nonlinear aerodynamics using an autoencoder neural network. Phys. Fluids 36 (1).

Murata, T., Fukami, K., Fukagata, K., 2020. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. J. Fluid Mech. 882, A13.

Qu, J., Zhao, Y., Cai, W., 2023. Nonlinear dynamic mode decomposition from time-resolving snapshots based on deep convolutional autoencoder. Phys. Fluids 35 (6).

Rabault, J., Kuchta, M., Jensen, A., Réglade, U., Cerardi, N., 2019. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. J. Fluid Mech. 865, 281–302.

Raj, N.A., Tafti, D., Muralidhar, N., 2023. Comparison of reduced order models based on dynamic mode decomposition and deep learning for predicting chaotic flow in a random arrangement of cylinders. Phys. Fluids 35 (7).

Regazzoni, F., Pagani, S., Salvador, M., Dede', L., Quarteroni, A., 2024. Learning the intrinsic dynamics of spatio-temporal processes through Latent Dynamics Networks. Nat. Commun. 15 (1), 1834.

Schmid, P.J., 2010. Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech. 656, 5–28.

Solera-Rico, A., Sanmiguel Vila, C., Gómez-López, M., Wang, Y., Almashjary, A., Dawson, S.T., Vinuesa, R., 2024. β-Variational autoencoders and transformers for reduced-order modelling of fluid flows. Nat. Commun. 15 (1), 1361.

Sushama, G., Menon, G.C., 2023. Attention augmented residual autoencoder for efficient polyp segmentation. Int. J. Imaging Syst. Technol. 33 (2), 701–713.

Taira, K., Brunton, S.L., Dawson, S.T., Rowley, C.W., Colonius, T., McKeon, B.J., Schmidt, O.T., Gordeyev, S., Theofilis, V., Ukeiley, L.S., 2017. Modal analysis of fluid flows: An overview. Aiaa J. 55 (12), 4013–4041.

Tang, H., Rabault, J., Kuhnle, A., Wang, Y., Wang, T., 2020. Robust active flow control over a range of Reynolds numbers using an artificial neural network trained through deep reinforcement learning. Phys. Fluids 32 (5).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wang, Y., Solera-Rico, A., Vila, C.S., Vinuesa, R., 2024. Towards optimal β-variational autoencoders combined with transformers for reduced-order modelling of turbulent flows. Int. J. Heat Fluid Flow 105, 109254.

Wang, J., Xie, H., Zhang, M., Xu, H., 2023. Physics-assisted reduced-order modeling for identifying dominant features of transonic buffet. Phys. Fluids 35 (6).

Woo, S., Park, J., Lee, J.-Y., Kweon, I.S., 2018. Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 3–19.

Wu, P., Gong, S., Pan, K., Qiu, F., Feng, W., Pain, C., 2021. Reduced order model using convolutional auto-encoder with self-attention. Phys. Fluids 33 (7).

Xu, J., Duraisamy, K., 2020. Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics. Comput. Methods Appl. Mech. Engrg. 372, 113379.

Xu, Y., Sha, Y., Wang, C., Cao, W., Wei, Y., 2023. Comparative studies of predictive models for unsteady flow fields based on deep learning and proper orthogonal decomposition. Ocean Eng. 272, 113935.