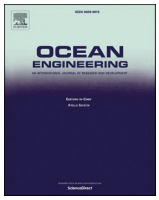
Ocean Engineering 318 (2025) 120205

Contents lists available at ScienceDirect

Ocean Engineering

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



Research paper

Twins-PIVNet: Spatial attention-based deep learning framework for particle image velocimetry using Vision Transformer

Yuvarajendra Anjaneya Reddy *, Joel Wahl, Mikael Sjödahl

Department of Fluid and Experimental Mechanics, Luleå University of Technology, Sweden

ARTICLE INFO

Keywords:
Particle image velocimetry
Deep learning
Vision transformer
Self-attention
Optical flow estimation

ABSTRACT

Particle Image Velocimetry (PIV) for flow visualization has advanced with the integration of deep learning algorithms. These methods enable end-to-end processing, extracting dense flow fields directly from raw particle images. However, conventional deep learning-based PIV models, which predominantly rely on convolutional architectures, are limited in their ability to utilize contextual information and capture dependencies between pixels across sequential images, impacting the prediction accuracy. We introduce Twins-PIVNet, a deep learning framework for PIV optical flow estimation that leverages a spatial attention-based Vision Transformer architecture. Its self-attention mechanism captures multi-scale features of particle motion, significantly improving the dense flow field estimation. Trained on synthetic PIV datasets covering a wide range of flow conditions, Twins-PIVNet has been evaluated on both synthetic and experimental datasets, demonstrating superior accuracy and performance. In comparative studies, Twins-PIVNet outperforms existing optical flow and conventional methods, achieving accuracy improvements of 51% for backstep flow, 42% for DNS-turbulence case, and 33% for surface quasi-geostrophic flow. Additionally, it also exhibits strong generalization on experimental PIV data, demonstrating robustness in handling real-world PIV uncertainties. Twins-PIVNet has faster inference times compared to other PIV models, offering a balance between complexity, efficiency, and performance.

1. Introduction

Advancements in Particle Image Velocimetry (PIV), a non-intrusive optical flow visualization technique has been intricately tied to advancements in hardware and computational strategies (Scharnowski and Kähler, 2020). PIV utilizes tracer particles to measure instantaneous velocity fields of flows through the comparison of time-resolved image sequences. Cross-correlation and optical flow estimation are the two established methods used to analyze PIV images (Willert et al., 2007; Westerweel, 1995; Horn and Schunck, 1981a). Cross-correlation methods segment images into patches or sub-windows to compute displacement vectors from intensity shifts in cross-correlation maps, offering stability, but often yielding low-resolution results that require both pre- and post-processing procedures (Scarano and Riethmuller, 1999). Optical flow methods, based on variational formulation optimize an energy functional, balancing data fidelity with regularization to generate smooth, high-resolution flow fields while preserving motion boundaries (Ruhnau et al., 2005; Horn and Schunck, 1981b; Lucas and Kanade, 1981). Although optical flow methods provide adaptable and dense high-resolution velocity estimations (at the pixel level), they have limitations such as sensitivity to noise in the images, errors proportional to particle displacements and velocity gradients affecting robustness, and computational demands (Scharnowski and Kähler, 2020; Willert et al., 2007; Liu et al., 2015).

The application of deep learning methodologies to PIV has emphasized the development of end-to-end algorithms that utilize optical flow estimation strategies, thus eliminating the need for pre- and post-processing steps. Early work by Rabault et al. showcased the potential of fully connected convolutional neural networks (CNNs) trained on synthetic data for PIV applications (Rabault et al., 2017). Lee et al. introduced PIV-DCNN, a deep CNN model with four cascaded levels to improve spatial resolution by directly mapping image patches to velocity vectors without explicit model assumptions (Lee et al., 2017). In 2019, Cai et al. developed PIV-NetS (Cai et al., 2019a), a CNN-based optical flow estimator inspired by the FlowNetS architecture (Dosovitskiy et al., 2015), marking the first global fluid motion estimator for PIV using CNNs. Cai et al. further extended their work by introducing PIV-LiteFlowNet-en (Cai et al., 2019b), which was based on

* Corresponding author.
E-mail address: yuvarajendra.anjaneya.reddy@ltu.se (Y.A. Reddy).

https://doi.org/10.1016/j.oceaneng.2024.120205
Received 3 October 2024; Received in revised form 16 December 2024; Accepted 21 December 2024
Available online 26 December 2024

the more efficient LiteFlowNet (Hui et al., 2018), a CNN-based optical flow estimator. Subsequently, RAFT (Recurrent All-pairs Field Transforms) (Teed and Deng, 2020), a CNN-based recurrent model, offered improved accuracy and computational efficiency for optical flow estimation. Models such as LightPIVNet (Yu et al., 2021), RAFT-PIV (Lagemann et al., 2021a), and DeepTR-PIV (Yu et al., 2023) extended RAFT's capabilities, using CNNs and ConvGRUs (Convolutional Gated Recurrent Units) to extract per-pixel features, compute multi-scale correlation volumes, and iteratively refine predictions. While these models rely on supervised learning with reference velocity fields (ground truth), unsupervised methods have also been explored. Zhang et al.'s UnLiteFlowNet-PIV (Zhang and Piggott, 2020) and Lagemann et al.'s URAFT-PIV (Lagemann et al., 2021b) utilized photometric, smoothness, and consistency losses in their cost functions to bypass reference velocity fields. Zhang et al. also introduced UnPWCNet-PIV (Zhang et al., 2023a) based on PIV-PWCNet (Zhang et al., 2023b), which integrated pyramidal processing, feature warping, and cost volume in a recurrent framework.

For supervised learning approaches, the nature of PIV data significantly influences training. Real-world PIV data presents challenges such as noise, occlusions, internal reflections, out-of-plane motion, and complex flow dynamics characterized by multi-scale structures, rotation, divergence, and evolving vortices. These factors can degrade the performance of models trained solely on synthetic datasets. To accurately predict dense optical flow, feature extractors in the CNNs must have a large receptive field or enough deep layers to capture all relevant information (Luo et al., 2016). However, most CNNs being inherently local, utilize smaller kernels, and operations like pooling and normalization reduce dimensionality, limiting the receptive field and potentially missing important contextual details. Recent advances in spatial attention-based neural networks, inspired by Vision Transformers (ViT) (Dosovitskiy, 2020) and the Transformer architecture (Vaswani, 2017), have improved optical flow estimation. These models use multi-head self-attention mechanisms to focus on relevant motion areas, mitigating the influence of irrelevant regions and enhancing flow field estimation (Zhu et al., 2019). Han et al.'s ARaft-FlowNet (Han and Wang, 2023), integrates the attention mechanism into RAFT, enhancing performance on noisy PIV datasets. This model employs the probability-based CBAM (Convolutional Block Attention Module) (Woo et al., 2018) to compute residual attention across spatial and multi-channel domains. Yu et al. introduced DeepST-CC (Yu et al., 2024), which combines the Swin Transformer (Liu et al., 2021) with cross-correlation strategies within the RAFT framework, improving robustness. The Swin Transformer enhances contextual information through hierarchical feature representation using shifting windows. Wang et al. proposed GMA-PIV (Wang et al., 2023), incorporating the GMA (Global Motion Aggregation) architecture (Jiang et al., 2021), which captures spatial dependencies between features and performs global aggregation with attention to motion features from a two-frame input. Despite these advancements in prediction accuracy, challenges remain. ViTs are constrained by their high computational complexity, which scales quadratically with the number of pixels in the input. This, along with their extended inference times, limits their practicality for real-time applications.

To address these issues, we introduce Twins-PIVNet for dense optical flow estimation in PIV, which builds on the concepts from the refined attention-based Vision Transformer architecture, Twins-SVT (Chu et al., 2021a), and the current state-of-the-art, RAFT-PIV's (Lagemann et al., 2021a) structure. We replace the traditional encoder used for feature extraction with a soft attention encoder. Inspired by VideoFlow (Shi et al., 2023), and FlowFormer (Huang et al., 2022) models for efficient optical flow estimation using ViTs, we utilize only the first two stages of the Twins-SVT-L architecture and omit the classification layers. This facilitates high-level feature extraction addressing the long-range dependencies.

Section 2 details the architecture, while Section 3 provides an overview of the training hyperparameters and the hardware used in training. It also summarizes the datasets for training and testing, along with benchmarking methods for performance evaluation. Section 4 offers a comprehensive discussion and results, including a detailed comparison of Twins-PIVNet with other benchmark PIV methods on both synthetic and real-world datasets. This section also examines the computational costs, and model's performance across various conditions and uncertainties, emphasizing its strengths and limitations. Section 5 concludes the findings and contributions of this work.

2. Proposed model

Twins-PIVNet is a modified version of the RAFT-PIV architecture, as illustrated in Fig. 1(a). It retains the overall architecture of RAFT256-PIV (Lagemann et al., 2021a). However, the layers related to feature extraction have been redesigned to incorporate a ViT architecture.

2.1. Feature extraction

Twins-PIVNet employs a computationally efficient twin-transformer-

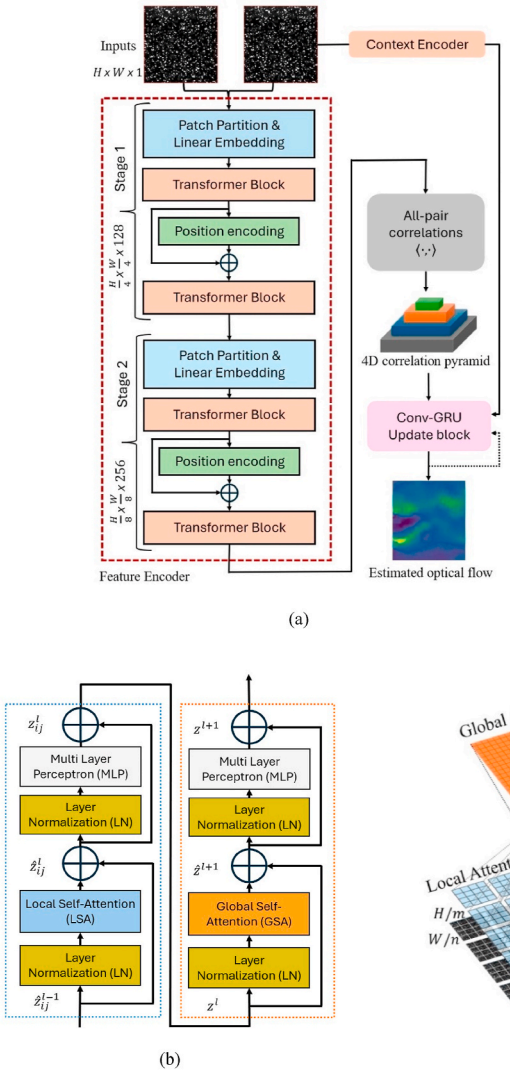


Fig. 1. (a) Twins-PIVNet network architecture proposed in this paper, which includes a Vision Transformer-based feature encoder, a 4D all-pairs correlation volume, and an iterative update module based on a Convolutional Gated Recurrent unit (ConvGRU). **(b)** Illustration of Transformer Block structure. **(c)** Schematic of Spatially Separable Self-Attention (SSSA), incorporating Locally grouped Self-Attention (LSA) and Global Sub-sampled Attention (GSA).

based feature encoder, E_F, that extracts multi-scale features and patterns in the initial stages of training due to their minimal inductive biases and inductive priors (Chu et al., 2021a). This feature encoder comprises *Patch Partition* and *Linear Embedding* blocks, *Transformer block*s, and spatial *Position Encoding* blocks deployed in two stages (see Fig. 1(a)). Additionally, a context encoder (similar in structure to E_F) processes the first input image to generate a context feature vector. This contextual information connects the dominating correlation patterns with specific image features that are computed in the all-pairs correlation volume from RAFT-PIV.

2.1.1. Patch Partition and Embedding

The feature encoder in Twins-PIVNet processes images with dimensions H x W, by dividing them into non-overlapping patches of size m x n. This results in $\frac{H}{m}$x$\frac{W}{n}$ patches. These patches are then passed through convolution operations to extract features, producing tensors of shape $\frac{H}{m}$x$\frac{W}{n}$xd, where d denotes the feature dimension. These tensors are subsequently flattened and transformed into feature vectors. The 2D convolution function '*torch.nn.Conv2d*' from the PyTorch library (Paszke et al., 2019) is used for these operations, with a kernel and stride size of [4, 4] pixels in Stage 1 and [2, 2] pixels in Stage 2, as depicted in Fig. 1 (a). This results in outputs with sizes $\frac{H}{4}$x$\frac{W}{4}$ x 128 and $\frac{H}{8}$x$\frac{W}{8}$ x 256, where the feature dimensions are 128 and 256 respectively. It is important to note that all the functions used in the feature encoder are from the PyTorch version 1.6 library (Paszke et al., 2019).

2.1.2. Transformer block with dual attention

The embedded patches pass through transformer blocks that include two types of attention: Locally grouped Self-Attention (LSA) and Global Sub-sampled Attention (GSA), forming a Spatially Separable Self-Attention (SSSA) mechanism. Each transformer block begins with a Layer Normalization (LN) module ('*torch.nn.LayerNorm*'), followed by either LSA or GSA, whose outputs are normalized and processed with a Multi-Layer Perceptron (MLP) module. As illustrated in Fig. 1(b), residual connections are applied after each attention mechanism and MLP module to ensure stability and prevent vanishing gradients. The MLP module consists of two fully connected layers and a GELU (Gaussian Error Linear Unit) activation function ('*torch.nn.GELU*'), enabling the model to learn more complex non-linear representations.

In the LSA module, the feature maps are divided into small groups, and self-attention (*local attention*) is computed independently (restricted to individual groups) to capture fine-grained features and small patterns. Within each group, a unit for Self-attention is defined by linearly projecting tokens into three separate vectors: a query vector (Q), a key vector (K), and a value vector (V), and calculating an attention score using the equation (Vaswani, 2017):

$$Self-attention\,(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{1}$$

Here, QK^T represents the dot product of the query and key vectors, which measure the similarity between patches. The scaling factor is equal to the dimensionality of the key vectors (feature dimensions). Softmax was computed using the function, '*torch.nn.Softmax*'. However, limiting attention to local windows restricts global information exchange, which is essential for tasks requiring dense predictions.

To facilitate cross-group information exchange across the whole image, the GSA module introduces *global attention* by computing attention between representative tokens of local windows (Chu et al., 2021a). This is achieved through spatial sub-sampling of feature maps after self-attention computation in GSA, using the '*torch.nn.Conv2d*' function with kernel and stride sizes of [8, 8] pixels for Stage 1 and [4, 4] pixels for Stage 2 respectively. This enables effective information exchange across the whole image, without the computational burden of full attention. Consequently, the combined dual-attention mechanism, SSSA (see Fig. 1(c)), allows Twins-PIVNet to effectively balance local and global feature dependencies, avoiding the computational complexity of large matrix multiplications, restricted receptive fields, and cyclic operations, as seen in Swin Transformers based approaches.

2.1.3. Position Encoding

Transformer architectures inherently lack access to the spatial order of non-overlapping patches. To address this, a conditional Position Encoding Generator (PEG) is used to preserve spatial locality and add conditional position information to each patch embedding (Chu et al., 2021b). This PEG dynamically generates conditions based on the input data, allowing it to handle inputs of varying lengths and effectively tackle translation invariance issues (Chu et al., 2021a). At its core, PEG uses 2D depthwise convolution, using the '*torch.nn.Conv2d*' function with a kernel size of [4, 4] pixels, and 'groups' settings that match the feature dimensions: 128 for Stage 1 and 256 for Stage 2.

This PEG is integrated after the first Transformer Block in each stage. By the end of Stage 2, for input particle images I_1 and, I_2 of resolution (H x W x 1), the output from each feature encoder E_F is of the shape ($\frac{H}{8}$x$\frac{W}{8}$ x 256). These outputs are high dimensional feature vectors, $f(I_n)$, that encapsulate patterns learned through adjustable weights from the input data during training, and are defined as:

$$f(I_n) = E_F(I_n) : \mathbb{R}^{H\text{x}W} \rightarrow \mathbb{R}^{\frac{H}{8}\text{x}\frac{W}{8}\text{x}256}, \tag{2}$$

where, 256 is the number of channels or the feature dimension d, and $n = 1, 2$, represents input particle images I_1 and, I_2 respectively.

2.2. All-pairs correlational volume

The visual similarity between dense image feature vectors $f(I_1)$ and $f(I_2)$, is computed using an all-pairs full correlational volume, as implemented in RAFT-PIV (Lagemann et al., 2021a). The 4D correlational volume C, of shape H x W x H x W, is obtained by computing the inner product between all pairs of extracted feature vectors:

$$C(f(I_1), f(I_2)) = C_{pqkl} = \sum_h (f(I_1))_{pqd} \cdot (f(I_2))_{kld} \tag{3}$$

Here, (p, q) and (k, l), represent the height and width indices of feature vectors $f(I_1)$ and $f(I_2)$, respectively, with d as the feature dimension. A four-layer correlation pyramid is then built by sequentially average pooling the last two dimensions of each layer, using kernel sizes and stride lengths of 1, 2, 4, and 8, respectively. This approach retains high-resolution information for small motions in the lower levels and captures long-range dependencies in the higher layers of the pyramid.

2.3. Iterative updates

The update block from RAFT-PIV, based on RAFT (Teed and Deng, 2020) architecture's iterative optimization module, is used to decode flow predictions from the correlation volume and context information from I_1. Initially, the predicted flow is set to zero. A correlation lookup maps each pixel in I_1 to its estimated new location in I_2, within a 4-pixel radius neighborhood across all layers of the correlation pyramid. A ConvGRU (Convolutional Gated Recurrent Unit) block then maintains high pixel reconstruction of the flow field by iteratively enhancing the accuracy. This block uses the last estimated flow, correlation values from the pyramid, and the flow hidden state to generate a new hidden state, which is then passed through two convolution layers for flow updates. Finally, an upsampling module refines the flow over 16 iterations to produce a high-resolution output. The convex upsampler is used as in RAFT256-PIV (Lagemann et al., 2021a) to upscale the estimated flow field from 1/8th to the original input resolution of the particle images.

3. Experiments

3.1. Training details and hyperparameters

The proposed Twins-PIVNet model was developed in Python language using the PyTorch deep learning framework (Paszke et al., 2019). The model was initialized with random weights, and the AdamW optimizer (Loshchilov, 2017) was employed with an initial learning rate of 12.5e-5. Gradient clipping was applied to limit the gradient norm to a maximum of 1.0, ensuring stable training. The OneCycleLR (Smith, 2018) strategy was employed to dynamically adjust the learning rates (LR), and a batch size of 5 was used during training. The initial LR was estimated by running an LR range test, which involved increasing the learning rate while monitoring the model's performance over one epoch in evaluation mode. The training process was supervised based on the l_1 distance between the predicted and reference optical flow fields across all iterations (Lagemann et al., 2021a). The cost or loss function incorporated exponentially increasing weights and sequence loss, defined as,

$$\mathscr{L} = \sum_{i=1}^{N} \gamma^{N-i} \left\| f_{ref} - f_{pred,i} \right\|_1, \tag{3}$$

where, f_{ref} and $f_{pred,i}$, represent the reference and predicted optical flow for the i^{th} iteration, respectively, while N denotes the total number of recurrent steps in the ConvGRU block, which is 16 in our case. γ is a constant, that assigns higher weights to later predictions exponentially and is set to 0.8. The training was concluded once the model showed no further improvement in the validation accuracy. The Average End Point Error (AEPE) (Dosovitskiy et al., 2015), representing the mean Euclidean distance between final predictions and reference optical flow over all pixels, was used as the evaluation metric.

$$AEPE = \left\| f_{pred} - f_{ref} \right\|_1. \tag{4}$$

The model was trained and evaluated on a system with an Intel Xeon Silver 4210 CPU at 2.20 GHz and two NVIDIA Quadro RTX 8000 GPUs. Training times were 21 h for the Problem Class I dataset and 32 h for the Problem Class II dataset, as described in Section 3.3.

3.2. Evaluation methods

This section outlines the methodologies employed for a thorough evaluation of the proposed Twins-PIVNet architecture using publicly accessible PIV datasets, highlighting its effectiveness compared to well-established PIV techniques. We evaluate the accuracy by analyzing its AEPE on the same test datasets against recognized PIV benchmarks, including cross-correlation methods, variational optical flow algorithms, and deep learning-based approaches. The conventional PIV technique, WIDIM (Window Deformation Iterative Multigrid) (Scarano and Riethmuller, 1999) with a two-pass configuration (interrogation windows of sizes 32 x 32 and 16 x 16 pixels), was chosen to be the baseline of cross-correlation methods, given its precision and reliability in recent evaluations (Lee et al., 2017; Cai et al., 2019a, 2019b; Hui et al., 2018; Yu et al., 2021, 2023; Lagemann et al., 2021a; Zhang and Piggott, 2020). The multi-resolution Horn-Schunck (HS) method (Horn and Schunck, 1981a), featuring six pyramid levels and two warping stages per level, represents the variational optical flow method in our comparison. Additionally, we include leading deep learning methods inspired by traditional PIV approaches, such as correlation peak finding and brightness constancy strategies. These include PIV-DCNN (Lee et al., 2017), a four-level deep CNN, which uses a 64 x 64 pixels interrogation window or patch size with a step size of 8 pixels and predicts a displacement vector at the center of the interrogation window; PIV-LiteFlowNet-en (Hui et al., 2018), an enhanced version of a multi-level pyramid structure-based network for dense per-pixel optical flow estimation; and both variants of RAFT-PIV (Lagemann et al., 2021a).

3.3. Datasets

Optical flow-based PIV neural networks trained with supervised learning rely on accurate reference velocity or displacement fields. These reference flow fields are commonly obtained from experimental data, high-fidelity direct numerical simulations (DNS), or analytical solutions. Recently, synthetic PIV datasets, combining these methods, have become popular for developing and benchmarking end-to-end PIV models. These datasets allow precise control over parameters such as particle diameter, displacement, and seeding density, enabling the creation of diverse and customized training sets while ensuring accurate flow fields.

We train and evaluate Twins-PIVNet using the publicly available synthetic PIV data from Cai et al. (2019b), specifically the Problem Class I and Problem Class II datasets from RAFT-PIV (Lagemann et al., 2021a). Problem Class I dataset contains 14,150 pairs of particle images and flow fields at a resolution of 256 x 256 obtained from computational fluid dynamics (CFD) simulations (Cai et al., 2019b). This dataset includes scenarios such as uniform channel flow, flow over a circular cylinder, flow around a backward-facing step, channel flow (John Hopkins Turbulence Databases) referred to as JHTDB-channel (Cai et al., 2019a), and sea surface simulations driven by the SQG (Surface Quasi-Geostrophy) model, which capture features like turbulent structures, separation zones, and recirculation areas. We used an 8:1:1 split in datasets for training, validation, and testing, with particle displacements ranging from 0 to 10 pixels and particle diameters from 1 to 4 pixels.

The Problem Class II dataset was developed by Lagemann et al. (2021a) using the same ground-truth flow fields as in the Problem Class I dataset while generating new particle image pairs with realistic image conditions. Images were generated focusing on replicating realistic experimental imaging conditions with non-uniform intensities, large maximum particle displacements, Gaussian noise, reduced SNR (signal-to-noise ratio), and particle densities. This dataset consists of 19,000 training pairs and 1000 validation pairs at 256 x 256 resolution. For further details on the Problem Class II dataset, refer to the RAFT-PIV article (Lagemann et al., 2021a). It is important to note that separate neural networks were trained for the Problem Class I and II datasets.

To evaluate the generalization capabilities of Twins-PIVNet, we employed experimental PIV data from a transitional turbulent boundary layer (TBL) over a flat plate (Zaki, 2013; Li et al., 2008; Perlman et al., 2007) and DNS data from a turbulent wavy channel flow (TWCF) (Rubbert et al., 2019). The TBL case covered the laminar, transitional, and turbulent stages of boundary layer development, while the TWCF case, featuring both favorable and adverse pressure gradients, was used to study the evolution of turbulent scales influenced by varying local pressure gradients (Lagemann et al., 2021a). The TBL dataset has a resolution of 256 x 3296 pixels, while the TWCF dataset is 2160 x 2560 pixels, providing high spatial resolution to analyze finer-scale turbulent structures and gradient changes in detail (Lagemann et al., 2022). Importantly, these datasets were not used during training or validation, offering a robust assessment of the neural network's ability to generalize. Efficient data loading during training was facilitated using the NVIDIA Data Loading Library (DALI) with a TFRecord-based pipeline, reducing loading delays (Aach et al., 2023).

4. Results and discussion

In this section, we present the results from various evaluations and comparisons to assess the performance of Twins-PIVNet on publicly available PIV data.

4.1. Evaluation of Problem Class I dataset

Table 1 shows the AEPE in pixels for Twins-PIVNet and other

Table 1
Average End Point Error (AEPE) values in pixels for individual test flow cases from Problem Class I dataset.

Methods	Back-step (20)	Cylinder (100)	JHTDB-channel (30)	DNS-turbulence (100)	SQG (75)	Average inference time [sec]
WIDIM	0.034	0.083	0.084	0.304	0.457	0.86
HS-Optical Flow	0.045	0.070	0.069	0.135	0.156	2.06
PIV-DCNN	0.049	0.100	0.117	0.334	0.479	2.23
PIV-LiteFlowNet-en	0.033	0.049	0.075	0.122	0.126	0.13
RAFT256-PIV	0.016	0.014	0.165	0.072	0.095	0.08
RAFT32-PIV	0.004	0.011	0.013	0.022	0.021	8.29
Twins-PIVNet	**0.013**	**0.012**	**0.092**	**0.056**	**0.091**	**0.08**

The test images are of 256 x 256 resolution, with the number of instances tested shown in parentheses. The lowest AEPE values in the table represent the best performance, and the results for Twins-PIVNet are highlighted in bold. The last column shows the computation time for a single instance, measured in seconds on the same machine.

evaluation methods, averaged across individual test datasets for each flow scenario. A total of 1400 pairs of randomly selected Problem Class I instances with 256 x 256 pixels resolution were tested. The conventional cross-correlation-based WIDIM (Scarano and Riethmuller, 1999) method performs well with standard flows but struggles with complex scenarios like DNS-turbulence or SQG cases due to its averaging approach, which leads to sparse flow field reconstructions and inadequate detail in small-scale flows. The deep learning-based PIV-DCNN (Lee et al., 2017) method, which also relies on cross-correlation, shows similar performance limitations but avoids additional input and output processing. The Horn-Schunck (Horn and Schunck, 1981a) optical flow method offers improved performance across all flow types by providing denser flow field information, though at the cost of higher execution times. Among deep learning methods, the enhanced PIV-LiteFlowNet-en (Hui et al., 2018) delivers dense reconstructions faster and handles common PIV uncertainties more effectively but still falls short in resolving fine-scale motions.

Twins-PIVNet demonstrates superior accuracy compared to other established deep learning architectures listed in Table 1. The self-attention module in Twins-PIVNet enables efficient feature extraction from input particle images by filtering out background details and stagnant zones, thereby minimizing the impact of uncertainties on the dense flow field predictions. The spatially separable self-attention algorithm (SSSA) further enhances contextual information utilization by avoiding spatial averaging, typical of cross-correlation methods, addressing issues related to diminishing receptive fields. Another self-attention-based end-to-end PIV architecture, ARaft-FlowNet (Han and Wang, 2023), trained on the same Problem Class I, exhibits similar trends, but Twins-PIVNet has a 19% improvement in accuracy. It is important to note that, ARaft-FlowNet was tested on a dataset with a different distribution, making comparisons vague since we lack access to its checkpoints or architecture. Overall, the present architecture shows a 27% improvement over pre-trained RAFT256-PIV and a 41% improvement over PIV-LiteFlowNet-en on the same test datasets.

The higher AEPE values observed in Twins-PIVNet and other deep learning methods on complex flows (e.g., JHTDB-channel, DNS-turbulence, and SQG) may be attributed to multi-scale turbulent flow characteristics and low-resolution input images. These methods often use spatial downsampling (to include more contextual information), which can lead to the loss of fine-scale information. To address this issue, architectures like RAFT32-PIV, by default, divide the images into finite patches with a fixed interrogation window of 32 x 32 pixels, thus avoiding spatial down sampling and preserving high-resolution details, leading to state-of-the-art accuracy on public PIV datasets. Similarly, DeepST-CC (Yu et al., 2024) (not used in this study) employs self-attention modules with shifting window operations (Swin transformer) and integrates a cross-correlation strategy to generate a rough initial flow field, which is then refined by a flow update module to enhance accuracy. However, both approaches significantly raise computational costs during training and inference compared to other methods (refer to the last column of Table 1). To ensure a fair runtime comparison, both the RAFT-PIV architectures have been modified to remove parallelization.

4.2. Evaluation of Problem Class II dataset

The evaluation of the Problem Class II dataset offers valuable insights into the architecture's ability to handle real-world PIV data, which often contains uncertainties and errors from experimental apparatus or data processing. Table 2 reports the AEPE values for Twins-PIVNet and two pre-trained RAFT-PIV (Lagemann et al., 2021a) variants. Twins-PIVNet, trained from scratch on 19,000 newly modified PIV data pairs with the same ground truths as the Problem Class I datasets, showing similar trends to Problem Class I results but with a 2-3-fold increase in AEPE values. This increase can be attributed to factors such as lower signal-to-noise ratio (SNR), increased maximum particle displacements, the presence of Gaussian noise, and inadequate particle densities in the input particle images, all of which negatively impact optical flow estimation. Twins-PIVNet combines strengths from both RAFT32-PIV—by capturing most of the finer flow structures through its self-attention module—and RAFT256-PIV, which processes more spatial information by downsampling the input resolution. Despite showing a 38% improvement in accuracy over the RAFT256-PIV architecture, Twins-PIVNet still underperforms compared to RAFT32-PIV. The superior performance of RAFT32-PIV is attributed to its multi-pass sub--windowing strategy, which avoids the loss of detail associated with spatial downsampling and upsampling.

Fig. 2 illustrates a qualitative comparison of the horizontal displacement components for a DNS turbulence test case from the Problem Class I dataset. The left column shows the ground truth and optical flow predictions, while the right column displays absolute error maps, with lower intensities indicating better model performance. The absolute error is obtained by subtracting the model predictions from the ground truth values. The error maps and AEPE values in Fig. 2 demonstrate that RAFT32-PIV excels in accurately resolving finer, small-scale structures compared to the other architectures. The RAFT256-PIV architecture tends to overestimate displacement magnitudes, likely due to the extensive contextual information gained from downsampling. Though the displacement predictions of both Twins-PIVNet and RAFT256-PIV are smoothed due to the feature encoding and decoding processes, arising from downsampling inputs to 1/8th resolution during

Table 2
Average End Point Error (AEPE) values for flow cases from Problem Class II test dataset with 256 x 256-pixel resolution images and reference flow fields.

Methods	Back-step	Cylinder	JHTDB-channel	DNS-turbulence	SQG
RAFT256-PIV	0.068	0.069	0.242	0.209	0.267
RAFT32-PIV	0.011	0.016	0.030	0.044	0.047
Twins-PIVNet	**0.035**	**0.031**	**0.175**	**0.121**	**0.167**

The lowest AEPE values in the table represent the best performance, and the results for Twins-PIVNet are highlighted in bold.

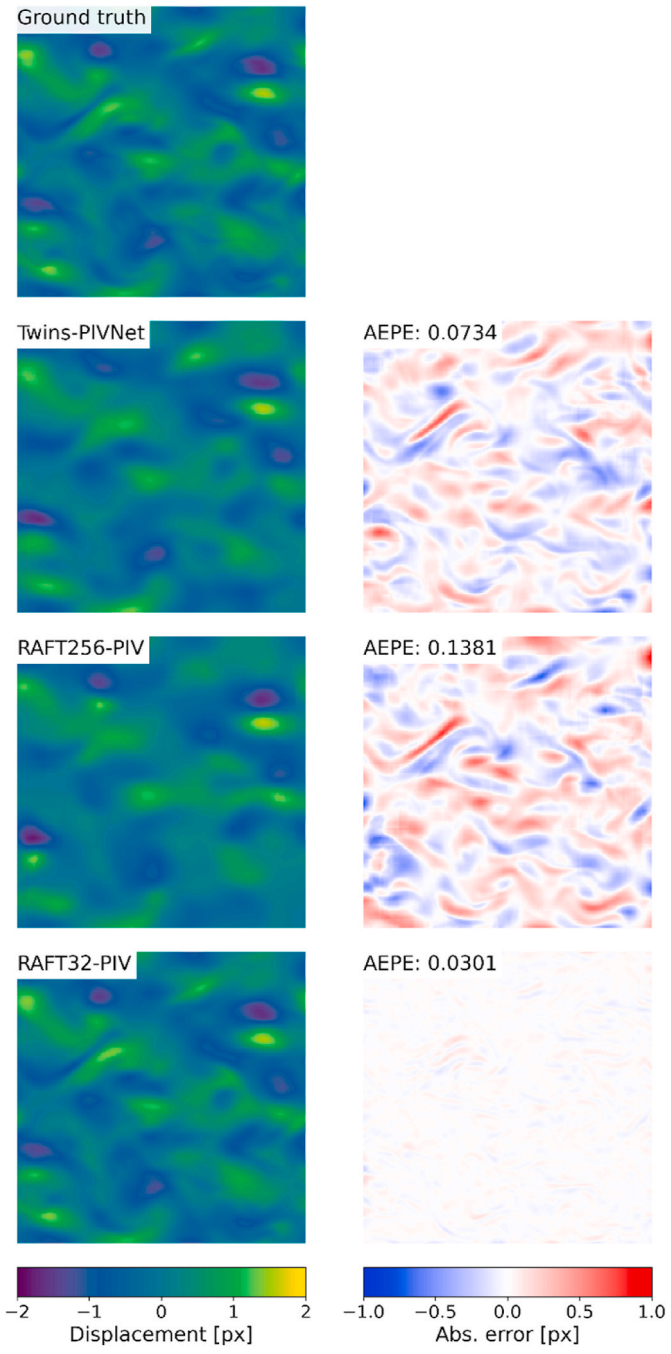


Fig. 2. Evaluation of DNS turbulence test case (256 x 256 pixels) from synthetic PIV dataset, Problem Class I: The figure compares horizontal displacement predictions from various neural network architectures trained on Problem Class I datasets. The left column shows predictions, while the right column presents absolute errors obtained by subtracting predictions from ground truth, with lighter contours indicating higher accuracy of the method.

feature extraction and subsequent upsampling after optical flow estimation, the Twins-PIVNet exhibits relatively lower AEPE and less intense error maps. This improvement can be again attributed to the self-attention module in the feature extraction stage, which effectively focuses on relevant particles and adapts the weights to better learn feature representations by considering contextual information on both global and local levels each time. However, in the iterative update stage, all of these models in Table 2 are equally constrained by the limited spatial context information available in the ConvGRU-based recurrent module, which reduces performance in regions with occluded particles.

4.3. Evaluation of DNS and experimental PIV data

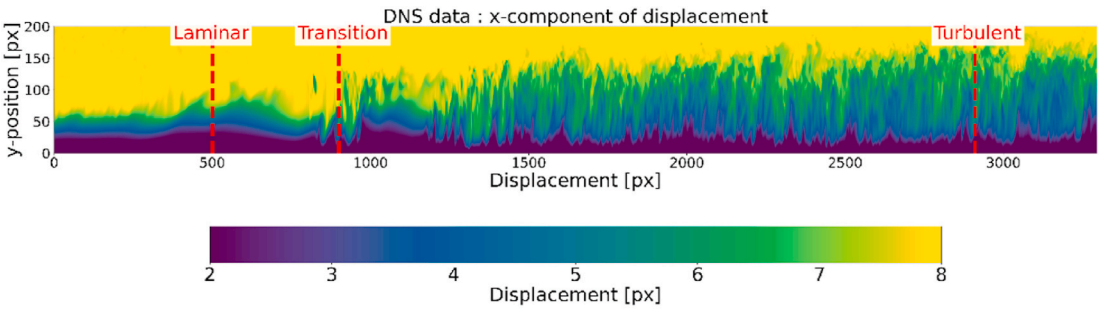
Despite comprehensive testing of end-to-end PIV deep learning algorithms using synthetic data, validating their performance with real-world data is crucial for their practical application. We first evaluate these models using synthetic images of size 256 x 3296 pixels, and direct numerical simulation (DNS) data of a zero-pressure-gradient transitional turbulent boundary layer (TBL) over a flat plate as reference values. Fig. 3(a) depicts the x-directional displacement component, illustrating the boundary layer's transition from laminar to turbulent states. Displacement data from three specific locations (highlighted by red dotted lines) are analyzed to compare the Twins-PIVNet and two variants of RAFT-PIV architectures across laminar, transitional, and turbulent regions. Fig. 3(b) shows that for the laminar profile, the predictions from all the models closely collapse on the ground truth data. However, around 75 pixels in the y-direction, where particle velocities approach freestream values and shear gradients diminish, both RAFT256-PIV and Twins-PIVNet demonstrate decreased accuracy due to increased flow recirculation and chaotic behavior. During the transition phase, flow instability generates turbulent spots that disrupt the laminar flow. In the fully turbulent region, the flow is characterized by intense irregular fluctuations, adverse pressure gradients, and high energy dissipation rates, presenting challenges for deep learning-based models. Twins-PIVNet tends to average predictions in both transition and turbulent regions, as shown in Fig. 3(b), while RAFT256-PIV captures trends but does so inaccurately, resulting in errors in displacement component estimation. These inaccuracies can be attributed to out-of-plane motion, high-mixing rates, spatial averaging, and smoothing effects, explaining the discrepancies observed in boundary layer regions near freestream velocities. Conversely, RAFT32-PIV, which avoids downsampling of inputs, demonstrates better predictive accuracy and closely aligns with DNS results. Although RAFT32-PIV performs better than Twins-PIVNet or RAFT256-PIV, it incurs more computational cost and inference times, making it unsuitable for real-time applications.

Fig. 4 presents a comparison of flow field predictions by deep learning models for TWCF data (turbulent wavy channel flow experimental PIV dataset with a resolution of 2160 x 2560 pixels) against the conventional Pascal PIV method (Lagemann et al., 2022). This dataset includes both adverse and favorable pressure gradients, effectively capturing the evolution of turbulent scales in the presence of varying local pressure gradients (Lagemann et al., 2021a). The left column depicts horizontal displacement components, while the right column shows vertical components. Twins-PIVNet outperforms RAFT32-PIV, which displays significant distortion spots in both components, and RAFT256-PIV, which produces grainy, less smooth predictions. Despite the absence of pre- or post-processing steps that are common in conventional cross-correlation methods, these deep learning models achieve high-quality results due to iterative updates following optical flow estimation. Nonetheless, on a broader spectrum, their performance is

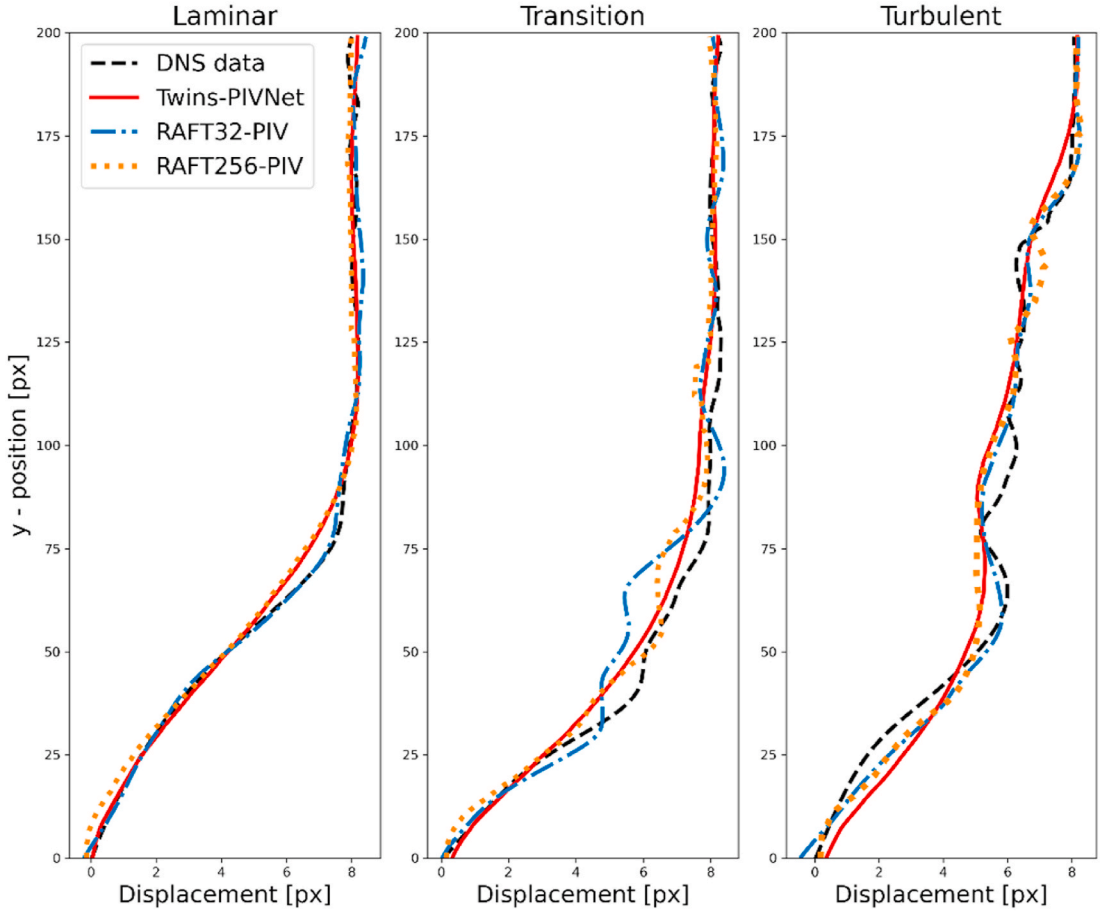
Slightly lower than that of the conventional method, likely due to the '*distribution shift*' (Lagemann et al., 2022) challenge that neural networks face when tested on new datasets outside their training scope. It is worth noting that the input images for both TBL and TWCF were divided into patches (typically 256 x 256 non-overlapping patches) for analysis in Twins-PIVNet to avoid 'out of memory' issues during inference (a method inspired by RAFT-PIV (Lagemann et al., 2021a) to handle large input image resolution). These image patches were still downsampled, similar to the approach used for Problem Class I and II datasets, and later the outputs from the update block were upsampled to original resolutions.

4.4. Computational cost

In Vision Transformers for optical flow estimation, computing self-attention requires more model parameters and resources than traditional convolution-based feature extractors (Chu et al., 2021a). The



(a)



(b)

Fig. 3. (a) Test Case – Turbulent Boundary Layer Flow (TBL) – 256 x 3296 pixels – Horizontal displacement field from Direct Numerical Simulations (DNS), representing the reference flow field. The red dotted lines indicate the laminar, transition, and turbulent regions, each with its respective label. **(b)** The figure compares horizontal displacement profiles at the three locations in Fig. 3 (a), corresponding to laminar, transition, and turbulent flow regimes, shown by dotted red lines. The laminar regime predictions closely match the ground truth, with minor discrepancies in boundary layer regions. In transition and turbulent regimes, with irregular fluctuations and adverse pressure gradients, RAFT32-PIV offers better predictions compared to Twins-PIVNet and RAFT256-PIV, which suffer from smoothing and spatial averaging effects. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

complexity of self-attention for input images with resolution H x W and dimension d is generally $\mathcal{O}(H^2W^2d)$ (Chu et al., 2021a). The Twins-PIVNet model mitigates this complexity by employing a two-stage self-attention mechanism, comprising Local Self-Attention (LSA) and Global Self-Attention (GSA), collectively termed Spatially Separable Self-Attention (SSSA), as previously discussed.

For LSA, the input feature maps are divided into m x n subwindows, with H divisible by m and W divisible by n. The computational cost for each sub-window with $\frac{HW}{mn}$ elements would then be $\mathcal{O}\left(\frac{H^2W^2}{m^2n^2}d\right)$. When defining $k_1 = \frac{H}{m}$, and $k_2 = \frac{W}{n}$, the total cost becomes $\mathcal{O}(k_1k_2HWd)$. In GSA, a single representative summarizes the feature information across all m x n subwindows, with a computational cost of $\mathcal{O}(mnHWd) = \mathcal{O}\left(\frac{H^2W^2}{k_1k_2}d\right)$. By leveraging LSA and GSA in a manner similar to separable convolution (depth-wise + point-wise convolutions), achieving a total computational cost of $\mathcal{O}\left(k_1k_2HWd + \frac{H^2W^2}{k_1k_2}d\right)$ (Chu et al., 2021a).

The Twins-PIVNet model, with 11.6 million parameters and fewer FLOPS (floating operations per second), is more efficient than the Swin-

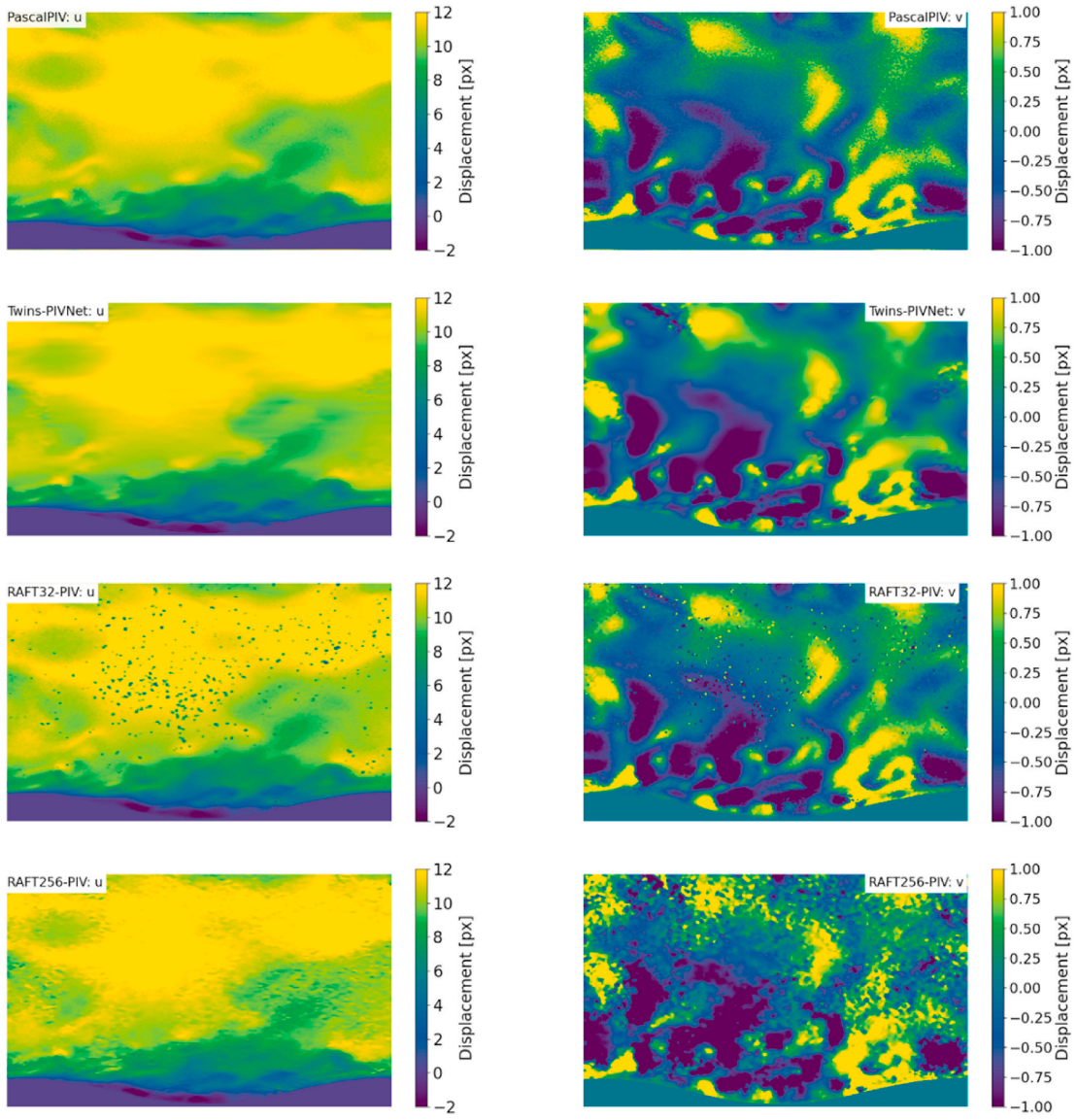


Fig. 4. Test Case – Turbulent Wavy Channel Flow (TWCF) – 2160 x 2560 pixels from experimental PIV dataset: A visual comparison of displacement contours among the classical correlation-based PIV method, Pascal-PIV, our architecture, Twins-PIVNet, and the RAFT-PIV architectures. The left column shows the horizontal displacement components, while the right column shows the vertical components. Twins-PIVNet provides predictions, that are closely comparable to Pascal-PIV, despite these cases not being part of the training data.

transformer-based architecture (which has 12.1 million parameters) and comparable to RAFT256-PIV (5.3 million parameters). It is important to note that the Swin-transformer model used for this parameter's comparison is our implementation and differs from DeepST-CC; it was not used for training or testing. Albeit having more parameters due to the self-attention module, Twins-PIVNet demonstrates faster or comparable run times when testing 256 x 256 pixel image pairs, as shown in the last column of Table 1. In terms of inference times, Twins-PIVNet and RAFT256-PIV are about 100 times faster than RAFT32-PIV, with Twins-PIVNet also demonstrating an overall 27% improvement in accuracy over RAFT256-PIV. It's crucial to note that no parallelization or optimization was applied to any of these models, and the existing parallelization in the RAFT-PIV architecture was removed for a fair comparison. This highlights Twins-PIVNet as a computationally efficient alternative to other attention-based deep learning models for PIV while achieving similar performance to the current state-of-the-art model, RAFT-PIV.

Optical flow measurement techniques, leveraging computer vision and deep learning methodologies, have shown significant potential for analyzing complex hydrodynamic flow phenomena (Li et al., 2023; Okbaz et al., 2023). These include wave-structure interactions, as well as velocity and pressure measurements, which are typically validated using non-intrusive measurement techniques (PIV) and high-fidelity simulations (Li et al., 2022). Building on these advancements, Twins-PIVNet offers an efficient and accurate solution for PIV analysis, outperforming other self-attention-based transformer models. The network has sub-second inference times, which can be further benefit from optimization in a production environment (Chu et al., 2021a). The computational speed makes Twins-PIVNet well-suited for real-time applications, such as digital twin integration, feedback control in dynamically changing experiments, predictive modeling, and simulations. Leveraging the transfer learning capabilities of ViTs, Twins-PIVNet can effectively manage distribution shifts, enabling it to learn from smaller, application-specific datasets (Steiner et al., 2021; Zhou et al., 2021). This makes it adaptable to new PIV setups with minimal retraining, supporting a wide range of engineering scenarios with high transferability. The multi-head self-attention mechanism in Twins-PIVNet can also be parallelized for further efficiency in training and inference

time (Zhou et al., 2021). This flexibility opens possibilities for analyzing complex flow fields, making Twins-PIVNet particularly useful for applications involving intricate geometries, such as flow around turbine blades, ship hulls, or fluidized beds. Moreover, the self-attention design enhances Twins-PIVNet's robustness to occlusions, noise, and natural or adversarial perturbations, significantly improving its generalizability against data drift during deployment stages. Studies have shown that deep learning models trained on synthetic PIV data can still perform well with real-world data (Reddy et al., 2024), and Twins-PIVNet builds on this by demonstrating strong performance across both experimental and Direct Numerical Simulation (DNS) datasets. This resilience, combined with its low-latency performance, positions Twins-PIVNet as an attractive solution for high-demand, real-world engineering applications requiring fast, reliable, and adaptive PIV analysis.

5. Conclusions

In this study, we introduced Twins-PIVNet, a deep-learning model that leverages a self-attention-based feature extractor for end-to-end optical flow estimation in PIV. Our results demonstrate that Twins-PIVNet achieves substantial computational efficiency and comparable or superior performance to state-of-the-art deep recurrent optical flow networks that lack self-attention mechanisms. By incorporating a refined Vision Transformer within the feature extractor, Twins-PIVNet adeptly captures essential features through both local and global attention modules, facilitating the extraction of fine-grained details and broad contextual information. This dual attention mechanism provides a notable advantage over traditional Vision Transformers utilizing shifting windows, as it enhances contextual information sharing and reduces computational effort.

Twins-PIVNet was rigorously trained on publicly available synthetic datasets and evaluated extensively on both synthetic and real-world experimental data. Among deep neural networks that employ downsampling and upsampling of input and output images, Twins-PIVNet demonstrates a significant performance improvement, achieving a 40% enhancement over baseline models, 27% over RAFT256-PIV, and 19% over ARaft-FlowNet. Although multi-pass approaches, such as RAFT32-PIV and DeepST-CC, which do not downsample inputs, achieve higher accuracy on public PIV datasets, they come at the cost of significantly increased training and inference times, as well as greater computational expenses. Therefore, Twins-PIVNet represents a balanced trade-off, offering high-resolution optical flow estimations, while maintaining lower computational demands compared to more computationally intensive architectures. This makes it a robust and efficient option for practical PIV applications, bridging the gap between accuracy and computational efficiency in complex fluid dynamics analyses.

CRediT authorship contribution statement

Yuvarajendra Anjaneya Reddy: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Joel Wahl:** Writing – review & editing, Supervision, Conceptualization. **Mikael Sjödahl:** Supervision, Software, Resources, Project administration, Funding acquisition, 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.

References

Aach, M., Inanc, E., Sarma, R., Riedel, M., Lintermann, A., 2023. Large scale performance analysis of distributed deep learning frameworks for convolutional neural networks. J. Big Data 10 (1), 96.

Cai, S., Zhou, S., Xu, C., Gao, Q., 2019a. Dense motion estimation of particle images via a convolutional neural network. Exp. Fluid 60, 1–16.

Cai, S., Liang, J., Gao, Q., Xu, C., Wei, R., 2019b. Particle image velocimetry based on a deep learning motion estimator. IEEE Trans. Instrum. Meas. 69 (6), 3538–3554.

Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., et al., 2021a. Twins: revisiting the design of spatial attention in vision transformers. Adv. Neural Inf. Process. Syst. 34, 9355–9366.

Chu, X., Tian, Z., Zhang, B., Wang, X., Shen, C., 2021b. Conditional Positional Encodings for Vision Transformers arXiv preprint arXiv:2102.10882.

Dosovitskiy, A., 2020. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale arXiv preprint arXiv:2010.11929.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., et al., 2015. Flownet: learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2758–2766.

Han, Y., Wang, Q., 2023. An attention-mechanism incorporated deep recurrent optical flow network for particle image velocimetry. Phys. Fluids 35 (7).

Horn, B.K., Schunck, B.G., 1981a. Determining optical flow. Artif. Intell. 17 (1–3), 185–203.

Horn, B.K., Schunck, B.G., 1981b. Determining optical flow. Artif. Intell. 17 (1–3), 185–203.

Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K.C., Qin, H., et al., 2022. Flowformer: a transformer architecture for optical flow. In: European Conference on Computer Vision. Springer Nature Switzerland, Cham, pp. 668–685.

Hui, T.W., Tang, X., Loy, C.C., 2018. Liteflownet: a lightweight convolutional neural network for optical flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8981–8989.

Jiang, S., Campbell, D., Lu, Y., Li, H., Hartley, R., 2021. Learning to estimate hidden motions with global motion aggregation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9772–9781.

Lagemann, C., Lagemann, K., Mukherjee, S., Schröder, W., 2021a. Deep recurrent optical flow learning for particle image velocimetry data. Nat. Mach. Intell. 3 (7), 641–651.

Lagemann, C., Klaas, M., Schröder, W., 2021b. Unsupervised recurrent all-pairs field transforms for particle image velocimetry. In: 14th International Symposium on Particle Image Velocimetry, vol. 1, 1.

Lagemann, C., Lagemann, K., Mukherjee, S., Schröder, W., 2022. Generalization of deep recurrent optical flow estimation for particle-image velocimetry data. Meas. Sci. Technol. 33 (9), 094003.

Lee, Y., Yang, H., Yin, Z., 2017. PIV-DCNN: cascaded deep convolutional neural networks for particle image velocimetry. Exp. Fluid 58, 1–10.

Li, Y., Perlman, E., Wan, M., Yang, Y., Meneveau, C., Burns, R., et al., 2008. A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. J. Turbul. (9), N31.

Li, J., Kong, X., Yang, Y., Yang, Z., Hu, J., 2022. Optical flow based measurement of flow field in wave-structure interaction. Ocean Eng. 263, 112336.

Li, J., Kong, X., Yang, Y., Yang, Z., Hu, J., 2023. Computer vision-based measurement of wave force on the rectangular structure. Ocean Eng. 270, 113624.

Liu, T., Merat, A., Makhmalbaf, M.H.M., Fajardo, C., Merati, P., 2015. Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images. Exp. Fluid 56, 1–23.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al., 2021. Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022.

Loshchilov, I., 2017. Decoupled Weight Decay Regularization arXiv preprint arXiv:1711.05101.

Lucas, B.D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In: IJCAI'81: 7th international joint conference on Artificial intelligence, 2, pp. 674–679.

Luo, W., Li, Y., Urtasun, R., Zemel, R., 2016. Understanding the effective receptive field in deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 29.

Okbaz, A., Aksoy, M.H., Kurtulmuş, N., Çolak, A.B., 2023. Flow control over a circular cylinder using vortex generators: particle image velocimetry analysis and machine-learning-based prediction of flow characteristics. Ocean Eng. 288, 116055.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al., 2019. Pytorch: an imperative style, high-performance deep learning library. Adv. Neural Inf. Process. Syst. 32.

Perlman, E., Burns, R., Li, Y., Meneveau, C., 2007. Data exploration of turbulence simulations using a database cluster. In: Proceedings of the 2007 ACM/IEEE Conference on Supercomputing, pp. 1–11.

Rabault, J., Kolaas, J., Jensen, A., 2017. Performing particle image velocimetry using artificial neural networks: a proof-of-concept. Meas. Sci. Technol. 28 (12), 125301.

Reddy, Y.A., Wahl, J., Sjödahl, M., 2024. Experimental dataset investigation of deep recurrent optical flow learning for particle image velocimetry: flow past a circular cylinder. Meas. Sci. Technol. 35 (8), 085402.

Rubbert, A., Albers, M., Schröder, W., 2019. Streamline segment statistics propagation in inhomogeneous turbulence. Phys. Rev. Fluids 4 (3), 034605.

Ruhnau, P., Kohlberger, T., Schnörr, C., Nobach, H., 2005. Variational optical flow estimation for particle image velocimetry. Exp. Fluid 38, 21–32.

Scarano, F., Riethmuller, M.L., 1999. Iterative multigrid approach in PIV image processing with discrete window offset. Exp. Fluid 26, 513–523.

Scharnowski, S., Kähler, C.J., 2020. Particle image velocimetry-classical operating rules from today's perspective. Opt Laser. Eng. 135, 106185.

Shi, X., Huang, Z., Bian, W., Li, D., Zhang, M., Cheung, K.C., et al., 2023. Videoflow: exploiting temporal cues for multi-frame optical flow estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12469–12480.
Smith, L.N., 2018. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv: 1803.09820.
Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L., 2021. How to train your vit? data, augmentation, and regularization in vision transformers. arXiv preprint arXiv:2106.10270.
Teed, Z., Deng, J., 2020. Raft: recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. Springer International Publishing, pp. 402–419.
Vaswani, A., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst.
Wang, J., Sista, H., Hu, H., He, P., Hu, H., 2023. A novel deep learning based approach for particle image velocimetry with global motion aggregation. In: AIAA AVIATION 2023 Forum, p. 4357.
Westerweel, J., 1995. Digital Particle Image Velocimetry: Theory and Application.
Willert, C., Wereley, S.T., Kompenhans, J., 2007. Particle Image Velocimetry: a Practical Guide.
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.
Yu, C., Bi, X., Fan, Y., Han, Y., Kuai, Y., 2021. LightPIVNet: an effective convolutional neural network for particle image velocimetry. IEEE Trans. Instrum. Meas. 70, 1–15.
Yu, C., Fan, Y., Bi, X., Kuai, Y., Chang, Y., 2023. Deep dual recurrence optical flow learning for time-resolved particle image velocimetry. Phys. Fluids 35 (4).
Yu, C., Chang, Y., Liang, X., Liang, C., Xie, Z., 2024. Deep learning for particle image velocimetry with attentional transformer and cross-correlation embedded. Ocean Eng. 292, 116522.
Zaki, T.A., 2013. From streaks to spots and on to turbulence: exploring the dynamics of boundary layer transition. Flow, Turbul. Combust. 91, 451–473.
Zhang, M., Piggott, M.D., 2020. Unsupervised learning of particle image velocimetry. In: High Performance Computing: ISC High Performance 2020 International Workshops, Frankfurt, Germany, June 21–25, 2020, Revised Selected Papers, vol. 35. Springer International Publishing, pp. 102–115.
Zhang, W., Dong, X., Sun, Z., Xu, S., 2023a. An unsupervised deep learning model for dense velocity field reconstruction in particle image velocimetry (PIV) measurements. Phys. Fluids 35 (7).
Zhang, W., Nie, X., Dong, X., Sun, Z., 2023b. Pyramidal deep-learning network for dense velocity field reconstruction in particle image velocimetry. Exp. Fluid 64 (1), 12.
Zhou, H.Y., Lu, C., Yang, S., Yu, Y., 2021. Convnets vs. transformers: whose visual representations are more transferable?. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2230–2238.
Zhu, X., Cheng, D., Zhang, Z., Lin, S., Dai, J., 2019. An empirical study of spatial attention mechanisms in deep networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6688–6697.