PIVNO: Particle Image Velocimetry Neural Operator

Jie Xu[*2], Xuesong Zhang[* †1], Jing Jiang[2] and Qinghua Cui[†3,4]

[1]Beijing University of Posts and Telecommunications
[2]Beijing Union University
[3]Peking University
[4]Wuhan Sports University
[1]xuesong_zhang@bupt.edu.cn
[3]cuiqinghua@hsc.pku.edu.cn

Abstract

Particle Image Velocimetry (PIV) aims to infer underlying velocity fields from time-separated particle images, forming a PDE-constrained inverse problem governed by advection dynamics. Traditional cross-correlation methods and deep learning-based feature matching approaches often struggle with ambiguity, limited resolution, and generalization to real-world conditions. To address these challenges, we propose a PIV Neural Operator (PIVNO) framework that directly approximates the inverse mapping from paired particle images to flow fields within a function space. Leveraging a position informed Galerkin-style attention operator, PIVNO captures global flow structures while supporting resolution-adaptive inference across arbitrary subdomains. Moreover, to enhance real-world adaptability, we introduce a self-supervised fine-tuning scheme based on physical divergence constraints, enabling the model to generalize from synthetic to real experiments without requiring labeled data. Extensive evaluations demonstrate the accuracy, flexibility, and robustness of our approach across both simulated and experimental PIV datasets. Our code is at https://github.com/ZXS-Labs/PIVNO.

1 Introduction

Particle Image Velocimetry (PIV) is a computer vision-based metrology widely used in various scientific and engineering fields, *e.g.*, physics [1–3], materials [4–6], life sciences [7–9], engine designs [10, 11], and locomotion inspection in tissue engineering [12]. By dispersing tracer particles into the fluid under measurement, PIV employs high-frame-rate cameras (typically 10^3 to 10^5 fps) and high-repetition-rate laser sources (up to 10^4 Hz) to capture particle image sequences and calculate the particle displacements, providing a discrete observation of the underlying motion field for further flow dynamic analysis. The motion of these tracer particles in an incompressible fluid can be modeled by the advection–diffusion equation [13–16]

$$\frac{\partial I}{\partial t} + \nabla \cdot (\mathbf{u} I) = D \nabla^2 I, \tag{1}$$

where $I(\vec{x}, t)$ represents the observed scalar field (e.g., particle intensity), $\mathbf{u}$ is the underlying velocity field, and D is the diffusion coefficient. Under the assumption of no source terms, negligible diffusion ($D = 0$), and divergence-free flow, (1) simplifies to the pure advection equation in the operator form

$$\mathcal{B}[u](\vec{x}, t) = f(\vec{x}, t), \quad \forall \vec{x} \in X, \quad \text{where } \mathcal{B} : u \mapsto u \cdot \nabla I, \quad f(\vec{x}, t) = \frac{\partial I}{\partial t}(\vec{x}, t) \tag{2}$$

[*]Equal contributions.
[†]Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

In the context of PIV, One is given $I(\vec{x}, t_1)$ and $I(\vec{x}, t_2)$—two noisy, discretely sampled particle images at successive time steps—and intends to infer the latent velocity field $\mathbf{u}$ that satisfies this the partial differential equation (PDE). This makes PIV fundamentally a *PDE-constrained inverse problem*, where the forward evolution is governed by physical transport dynamics, but the goal is to invert that process and recover the control variable $\mathbf{u}$ from its outcomes. Traditional PIV methods [17–21] generally rely on local cross-correlation matching techniques, but suffer from several limitations. First, the information within a single matching window is often insufficient to distinguish completely textureless particle clusters, and this issue worsens when the particle distribution is relatively uniform in the flow. Second, the size of the matching window directly affects both the spatial resolution and the accuracy of the estimated velocity field, leading to an inherent trade-off that is difficult to balance. In recent years, deep learning-based methods [22–27, 13, 28] have been proposed to represent particle image features effectively in high-dimensional latent spaces, thus reducing matching ambiguities. Moreover, the consecutive layers in deep neural networks offer larger receptive fields capable of capturing broader spatial context while preserving spatial resolution and matching accuracy. However, the inverse problem remains fundamentally ill-posed: the observed particle images are discrete, noisy, and of limited resolution, hence the numerical solution to the governing PDE is neither unique nor stable. In other words, the inverse operator of $\mathcal{B}$ in Eq.(2) may not exist.

Algorithmically, existing PIV approaches [26, 25, 27, 29, 30] solve the PDE in (2) via the construction of a cost-volume (CV) based on which the displacement array can be inferred. Nevertheless, the resolution of the CV is up to the affordable computational resources and once trained the PIV models' scalability to different measurement requirements is very limited. Inspired by the discretization invariant neural operator (NO) method, recently developed in the field of computational physics [31–35] as efficient PDE solvers,we inspect the PIV inverse problem through the lens of operator learning. Specifically, we introduce a PIV *neural operator*(PIVNO) framework that directly approximates the inverse map $\mathcal{G} : (I_{t_1}, I_{t_2}) \mapsto \mathbf{u}$, bypassing the necessity of CVs.Particularly, we devise a position informed Galerkin-type attention operator, whose approximation properties correspond to the classical Petrov–Galerkin projections[36]. This design enables our model to perceive global motion patterns while remaining numerically efficient and resolution-agnostic.

The sim2real transferring is another challenge confronting PIV models, let alone obtaining sufficient labeled data for supervision itself is expensive and often impractical in real experiments. Existing methods [37, 22–24] are primarily trained on synthetic datasets, which, despite offering perfect ground-truth flow, fail to cover all variability of real flow conditions as well as particle image qualities. Consequently, when applied to unseen flow regimes or lighting conditions, these models suffer from poor generalization. To address this, we incorporate self-supervised fine-tuning constrained by the physical incompressibility of the flow, enforcing the divergence-free conditions, to adapt the model to unlabeled real data. This strategy bridges the domain gap between synthetic and real experiments.

Finally, practical fluid experiments often demand localized flow analysis at varying resolutions [38–41]. However, the region of interest is rarely known *a priori*, and existing PIV networks are designed for uniform global inference. This limitation prevents adaptive refinement in high-shear or boundary-layer regions, restricting their utility for fine-scale investigations. In this work, we overcome this challenge by designing our operator-based architecture to support resolution-adaptive inference over arbitrary subdomains, enabling detailed flow field predictions.

In summary, our contributions can be summarized as follows:

- We propose the PIV Neural Operator, a neural operator framework that directly maps particle image pairs to flow fields, offering a function space-level approximation of fluid dynamics and improving estimation accuracy.
- We incorporate a self-supervised fine-tuning mechanism based on physical divergence constraints, enabling domain adaptation from synthetic training to real-world testing without labeled data.
- PIVNO enables resolution-adaptive flow inference over arbitrary subdomains, allowing fine-grained analysis in critical regions, which is essential for practical experimental fluid mechanics [42–44].

2 Related Work

Cost-Volume-Based PIV. Deep learning-based PIV has attracted significant attention recently. Early studies [37, 22–24] were primarily based on convolutional neural networks (CNNs) for supervised

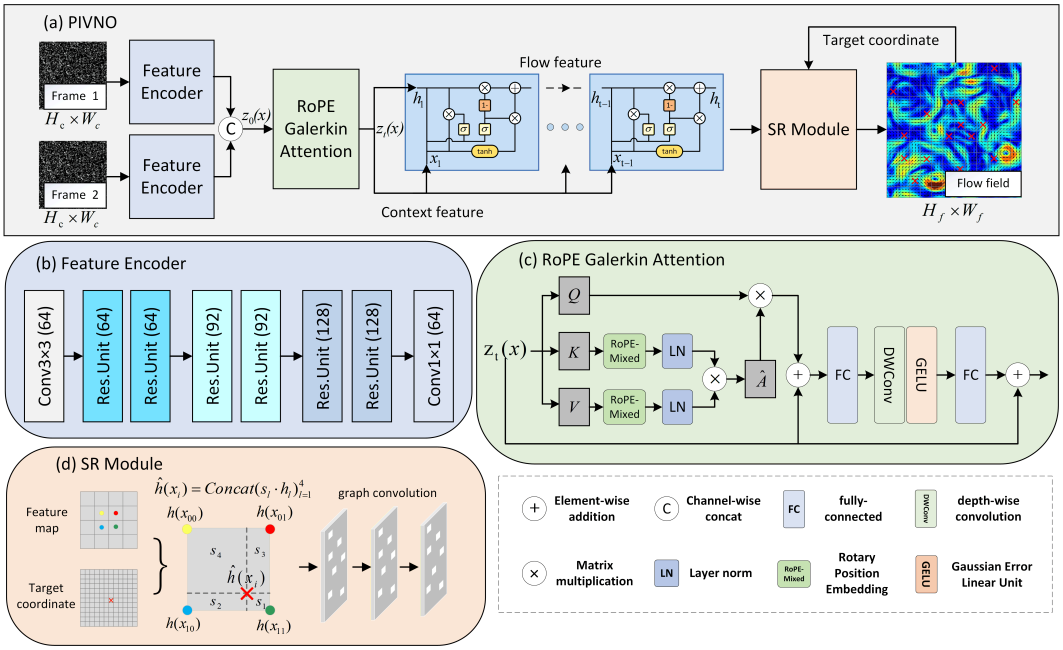

Figure 1: (a) The PIVNO framework, where each module works collaboratively to enable efficient inference from the (possible low-resolution) inputs to desired high-resolution flow fields. First, the Feature Encoder (b) extracts image features from the input pair. Next, the RoPE Galerkin Attention module (c) maps image feature functions to flow field functions via a Galerkin-style projection into the following Conv-GRU modules, where the coarse flow field features undergo iterative refinement. Finally, the SR Module (d) generates high-resolution flow field predictions through Continuous-Scale Flow Velocimetry, detailed in section 3.2.

training to estimate flow fields from images but achieved limited accuracy in flow field estimation due to their inefficiency in global motion awareness. Later work [26, 25, 27, 29, 30] proposed a cost-volume-based similarity computation to explicitly model matching relationships between image pairs, effectively improving displacement estimation accuracy. However, the construction of a four-dimensional cost volume incurs a huge computational burden and memory footprint on one hand, and more importantly freezes the possible matching accuracy and resolution determined by the grid size and dimensions of the cost volume. Therefore, cost-volume-based CNNs once trained cannot afford to solve different flow measurement problems with various accuracy requirements.

Generalization Challenges in PIV. Supervised learning methods depend on large labeled datasets. Due to the variability in flow conditions and particle image quality, these methods often generalize poorly to unseen flows or lighting setups. To address this, recent studies [13, 28, 45–48] have explored unsupervised optical flow algorithms for fluid flow estimation. These approaches optimize a cost function with physical constraints to estimate motion fields, offering better adaptability across diverse scenarios. However, they are highly sensitive to image quality; noise or lighting variations can significantly degrade robustness and accuracy.

3 Prticle Image Velocimetry Neural Operator

3.1 Framework of PIVNO

The goal of the proposed PIVNO is to mathematically establish a mapping $\mathcal{G} : A \to U$, where A and U represent the Hilbert spaces of particle image function pairs and flow fields, respectively. Specifically, we define: $A \triangleq \{a = [I_1, I_2] \in \mathbb{R}^2 | I_1, I_2 : \mathcal{X} \to \mathbb{R}, I_1(x) = I_2(x - u(x))\}$ to stress that the flow function $u(x)$ relates the input particle grayscale image pair, and $U \triangleq u : \mathcal{X} \to \mathbb{R}^2\}$ represents the function space of output flow fields. $\mathcal{X} \subset \mathbb{R}^2$ is the domain of the images. To approximate the mapping operator $\mathcal{G}$, we parameterize it with a neural network $\mathcal{G}_\theta$ and optimize the

parameters θ using supervised training. Given n observations $\{a_j, u_j\}_{j=1}^n$, where a_j are sampled from a distribution μ over A and $u_j = \mathcal{G}(a_j)$, the training process minimizes the empirical risk:

$$\min_{\theta} \frac{1}{n} \sum_{j=1}^{n} \|u_j - \mathcal{G}_\theta(a_j)\|_U^2, \tag{3}$$

where $\|\cdot\|_U$ represents the norm in the space U.

As shown in fig. 1(a), PIVNO depicts a structured workflow to transform particle image pairs into accurate flow field estimates. It begins with a feature encoder that extracts localized image features as the foundational representation. The RoPE-GA module implements a Petrov–Galerkin-style projection that maps image feature functions to flow field representations. Subsequently, the Conv-GRU module iteratively optimizes the flow field features using the contextual features. Finally, the SR module reconstructs the refined features as a continuous-scale flow field representation, allowing for accurate arbitrarily scaled flow velocity estimates.

3.1.1 Feature Encoder

The core of flow field estimation lies in accurately matching local features; thus, extracting discriminative features is critical for matching precision. As illustrated in fig. 1(b), we design a feature encoder to extract local features from the input image pair (I_1, I_2) and project them into a high-dimensional latent space. To enhance the encoder's representation capability, we integrate six residual units, each implementing a residual convolution operation defined as:

$$\mathrm{P}(I)(x) = \int_{N(x)} \kappa(x-y) I(y)\, dy + I(x), \tag{4}$$

where $\kappa(x-y)$ denotes the 3×3 convolution kernel defined over the local neighborhood $N(x)$ of position x, and $I(x)$ is the residual connection term. After the encoder extracts the spatial features from the image pair I_1 and I_2, the joint feature $z_0(x)$ is obtained through concatenation:

$$z_0(x) = \mathrm{Concat}(\mathrm{P}(I_1), \mathrm{P}(I_2)), \tag{5}$$

where Concat merges the two feature maps along the channel dimension; consequently, $z_0(x)$ now contains the temporal evolution of the spatial features.

3.1.2 RoPE Galerkin Attention

To construct a operator mapping from the image feature space to the flow field space, we propose the RoPE Galerkin Attention (RoPE-GA) module illustrated in fig. 1(c). [36] presents a general discussion on the parallelism between the finite element methods and its proposed GA module, but lacks of the positional embedding treatment for practical problems. Since the absolute positions are utmost important for particle tracking, we modify the positional encoding scheme of [49] to adapt to the Galerkin attention mechanism, leading to the enhanced GA matrix with Positional Encoding (PE) and frequency modulations:

$$\hat{A}_{j_1 j_2} = \mathrm{Re}\left[k_{j_1}^* v_{j_2} e^{i\left(p_n^x(\theta_{j_1}^x - \theta_{j_2}^x) + p_n^y(\theta_{j_1}^y - \theta_{j_2}^y)\right)} \right], \tag{6}$$

where p_n^x and p_n^y denote the spatial coordinates of position n along the x- and y-axes, k_{j_1} and v_{j_2} are the components of the key and value along channel dimensions j_1 and j_2, and θ_j^x and θ_j^y are learnable frequency parameters. See Supplementary Material Section A for detailed derivations and discussions.

The operator form of GA reads:

$$(\mathcal{A}[z])(x) \approx \sum_{j=1}^{d_z} \left(\sum_{k=1}^{d_k} \mathcal{K}_j[z](y_k) \mathcal{V}_j[z](y_k) \mathcal{Q}_j[z](x_j) \right), \tag{7}$$

where $\mathcal{K}_j[z](y_k)$ represents the evaluation of the input function $z(x)$ after the linear operator $\mathcal{K}$ at coordinate y_k; the index j means the j-th dimention of $\mathcal{K}(z)$. The same applies to $\mathcal{V}$ and $\mathcal{Q}$, and $\mathcal{A}$ denotes the integral kernel operator that refines $z(x)$ through the so-call basis update in [50]. However, this process is solely linear, lack of nonlinear expression capability.

In contrast, our RoPE-GA in Eq.(6) introduces frequency modulation to each channel j with different learnable frequency parameter θ_j. This treatment is equivalent to applying the sinusoidal activation [51] to the correlated features K and V after PE.

RoPE-GA in Eq.(6) attends to the correlation among the evaluated basis functions, $i.e.$, along the channel dimension. In order to enhance the spatial aggregation capability, we implant a 3×3 depthwise convolution between the first fully connected layer and the GELU activation function in the feed-forward network. Previous studies have shown that this operation helps capture finer spatial positional information [52–55].

3.1.3 GRU-based flow refinement

RoPE-GA produces coarse flow features $z_t(x)$ by mapping along the channel dimension (see Supplementary Materials Figure 1), but it does not explicitly organize spatial information; this often leads to locally inconsistent estimates. Because spatial continuity is critical for accurate motion estimation, merely stacking additional RoPE-GA layers is insufficient. We therefore introduce an iterative refinement mechanism in the spatial domain.

We adopt a convolutional GRU (Conv-GRU) for this purpose. Unlike RAFT [56], which uses image features as the context provider, our method performs recurrent refinement directly on the flow features, as illustrated in fig. 1(a). Specifically, the RoPE-GA features $z_t(x)$ serve two roles: they initialize the hidden state h_0 and, at every iteration, they are provided as the input feature c_t to supply local neighborhood information. The Conv-GRU then updates hidden states by convolving over spatial neighborhoods, enabling iterative propagation and correction of local flow evidence. Mechanistically, the update and reset gates regulate information exchange between the current input and the previous hidden state, allowing differential nonlinear mappings across multiple local subspaces. At each iteration, the module fuses the current features with the previous estimate and adjusts both the direction and magnitude of the predicted motion through locally sensitive convolutions and gating. In this way, the Conv-GRU functions not as a temporal recurrence for sequences but as a memory-equipped, differentiable spatial transformation that progressively approximates the true flow field within local regions. In practice, we find that a small number of refinement steps suffices: five iterations achieve near-optimal performance (see Supplementary Material Section B). The operations of the Conv-GRU module can be expressed as:

$$y_t = \sigma(\text{Conv}_{3\times3}([h_{t-1}, c_t], W_z)), \tag{8}$$

$$r_t = \sigma(\text{Conv}_{3\times3}([h_{t-1}, c_t], W_r)), \tag{9}$$

$$\tilde{h}_t = \tanh(\text{Conv}_{3\times3}([r_t \odot h_{t-1}, c_t], W_h)), \tag{10}$$

$$h_t = (1 - y_t) \odot h_{t-1} + y_t \odot \tilde{h}_t. \tag{11}$$

where σ and $\tanh$ denote the sigmoid and hyperbolic tangent activation functions, respectively; $\text{Conv}_{3\times3}$ denotes a 3×3 convolution; W_z, W_r, and W_h are the learnable convolutional weights; and $\odot$ represents element-wise multiplication.

3.2 Continuous-Scale Flow Velocimetry

To achieve accurate super-resolution reconstruction of the flow field, the SR module (fig. 1(d)) is designed by combining random sampling, continuous interpolation, and graph convolution. The first step is adapted from SRNO [50], which enhances the scale generalizability of our model:

$$\hat{h}(x_i) = \text{Concat}(s_l \cdot h_l)_{l=1}^{4} \tag{12}$$

where x_i represents the target coordinate, and $l \in \{00, 01, 10, 11\}$ denotes the coordinates of the four neighboring points of x_i . s_l is the diagonal area of the neighboring grid point's coordinates, and h_l is the feature vector of the neighboring points. The final feature vector $\hat{h}(x_i)$ is obtained by concatenating the weighted feature vectors $s_l \cdot h_l$.

The resulting feature $\tilde{h}(x_i)$ is then input into the subsequent graph convolution layers. Since the features of sampled points already incorporate positional information through the RoPE-GA module, the graph convolution [57–59] can effectively capture the spatial correlations between the randomly sampled points. By supervising on the randomly sampled points,whose results come from the graph convolution performing local information fusion, we are actually enforce PIVNO to attend to the

local correlation in fluids, enhancing the completeness and robustness of the feature representation. Finally, the graph convolution projects the high-dimensional features back into the PIV flow field solution space $u(x)$ via a 3×3 convolution:

$$u(\tilde{h})(x) = \int_{R(x)} \kappa(x-y)\tilde{h}(y)\, dy, \tag{13}$$

where $\kappa(x-y)$ represents the 3×3 graph convolution kernel defined over the set of randomly sampled points $R(x)$.

3.3 Self-Supervised Fine-Tuning

We further propose a self-supervised fine-tuning scheme to adapt the simulation-based pre-trained models to real experimental data. The fine-tuning strategy employs a variational optical flow method consisting of three components: a data term, a smoothing term, and a divergence regularization term. The self-supervised loss function is defined as:

$$L_P(u) = L_d(u) + \lambda_s L_s(u) + \lambda_d L_{\text{div}}(u), \tag{14}$$

where L_d represents the data term, modeling the similarity of the image pairs, L_s and L_{div} are the spatial smoothing and divergence regularization terms respectively, and λ_s and λ_d are their respective weights.

Data Term:

$$\text{corr}(x) = \frac{\sum_{y \in N(x)} \langle I_1(y), \hat{I}_1(y) \rangle}{\sqrt{\sum_{y \in N(x)} \|I_1(y)\|^2} \cdot \sqrt{\sum_{y \in N(x)} \|\hat{I}_1(y)\|^2}} \tag{15}$$

$$L_d(I_1, I_2) = \sigma\left(1 - E_x(\text{corr})\right) \tag{16}$$

where $\hat{I}_1(x) = I_2(x - u(x))$ is the warped image compared against I_1, and $u(x)$ represents the flow field prediction at the spatial location x. $N(x)$ denotes the set of all pixel points within the sliding window. $\langle I_1(x), \hat{I}_1(x) \rangle$ is the dot product of the real image and the predicted image, and $\|I_1(x)\|$ and $\|\hat{I}_1(x)\|$ represent the magnitudes of the real image and the predicted image within the window, respectively. The function $\sigma(z) = (z^2 + \epsilon^2)^\gamma$ is the Charbonnier penalty function, used to smooth the error term z, where γ controls the degree of smoothing, and E_x denotes the expectation over all the positions in the images.

Smoothing Term:

$$L_s(u) = \sigma\left(\nabla^2 u(x)\right) \tag{17}$$

where $\nabla^2 u(x)$ is the second derivative of $u(x)$ (the Laplacian operator), which measures the local variation of the underlying flow field.

Divergence Term:

$$L_{\text{div}}(u) = \sigma\left(\nabla \cdot u(x)\right) \tag{18}$$

where $\nabla \cdot u(x)$ is the divergence of $u(x)$, which enforces the incompressibility constraint for fluid flow, constraint the divergence of the flow.

4 Experiments

The experimental evaluation comprises three synthetic datasets and three real-world PIV challenge tasks. Initially, supervised training and benchmark testing are conducted on Synthetic Datasets 1 and 2. Training samples are generated by uniformly sampling downsampling factors within the range of $1\times$ to $4\times$, allowing the model to generalize across varying output resolutions after a single training phase. Subsequently, self-supervised fine-tuning is performed on Synthetic Dataset 3 and on the three real-world PIV benchmarks to enhance cross-domain adaptability. Since ground-truth flow fields are unavailable for real-world benchmarks, direct quantitative evaluation is inherently infeasible. The selected real-world PIV tasks are chosen for their established and credible evaluation protocols, enabling reliable qualitative comparison.

In addition, ablation experiments are conducted to analyze the contribution of key modules and loss components. **Table 3** specifically presents the ablation results of the self-supervised loss terms and

Table 1: This table presents the Average Endpoint Error (AEE) on synthetic dataset 1, where the error unit is set to pixels per 100 pixels for easier comparison. All methods are evaluated using a fixed input and output resolution of 256^2. The last four rows show the performance of our method at different input resolutions (e.g., $64^2 \times 4$ represents an input of 64^2 with 4× upsampling to achieve the 256^2 output). The Params column indicates the number of parameters in millions (M).

Methods	Backstep	Cylinder	JHTDB Channel	DNS turbulence	SQG	Params. (M)
Farneback [60]	8.5	8.3	14.1	37.8	33.2	-
PIV-DCNN [22]	4.9	7.8	11.7	33.4	47.9	8.40
PIV-LiteFlowNet [23]	5.6	8.3	10.4	19.6	20.0	6.25
PIV-LiteFlowNet-en [23]	3.3	4.9	7.5	12.2	12.6	5.59
UnPwcNet-PIV [13]	8.2	7.1	13.4	21.5	25.2	9.37
UnLiteFlowNet-PIV [13]	9.4	6.9	8.4	15.0	17.3	5.38
OFVNetS [61]	15.0	1.6	25.0	8.3	22.3	-
OFVNetS-HS [61]	13.7	4.7	32.7	7.0	18.9	-
PIV-RAFT [25]	1.6	1.4	13.7	9.3	11.7	5.31
ARaft-FlowNet [29]	3.1	2.0	8.3	9.6	9.8	-
PIVNO($256^2 \times 1$)	1.9	**0.8**	**1.7**	**3.5**	**2.5**	**2.52**
PIVNO($128^2 \times 2$)	**0.9**	1.1	2.8	4.7	4.1	
PIVNO($64^2 \times 2$)	1.0	1.4	2.9	5.0	4.2	
PIVNO($64^2 \times 4$)	2.7	3.1	11.3	18.0	18.0	

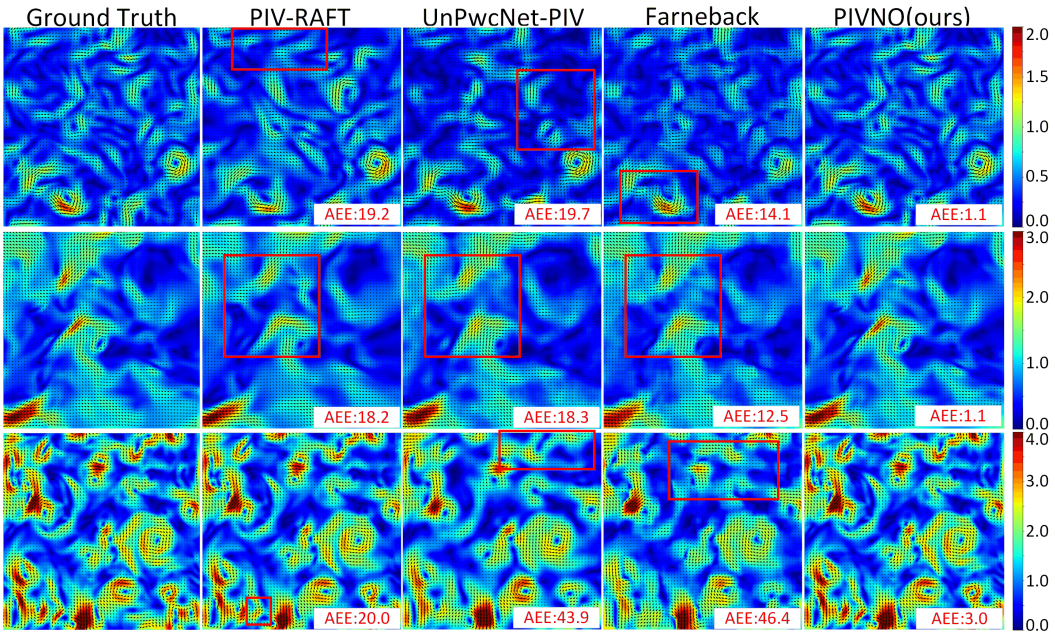

Figure 2: Visualization of three flow fields: DNS turbulence (first row), JHTDB channel flow (second row), and SQG (third row).The rectangles indicate over-smoothing or even failure artifacts existing in the outputs of comparative methods, while PIVNO faithfully recovers the flow.

simultaneously summarizes the overall performance of the self-supervised strategy on both synthetic and real-world datasets. It is therefore presented separately to highlight its central role in validating the effectiveness of the proposed self-supervised strategy. **Section 4.3** further investigates the influence of architectural components. Beyond the main results, we also conduct extended analyses, including statistical robustness evaluation (maximum error and standard deviation), cross-domain fine-tuning generalization experiments, and zero-shot resolution generalization studies. All of these additional results are provided in the Supplementary Material for completeness. Comprehensive details of the network architecture, dataset configurations, implementation methods, and hyperparameter settings are also included therein.

Table 2: The AEE on synthetic dataset 2. Representative baselines include the unsupervised UnLiteFlowNet and the self-supervised PIV-RAFT.

Methods	Backstep	Cylinder	JHTDB Channel	DNS turbulence	SQG
UnLiteFlowNet-PIV [24]	12.3	7.9	14.5	22.5	21.6
UnLiteFlowNet32-PIV [24]	40.9	65.9	41.9	44.3	40.1
PIV-RAFT [25]	6.4	5.2	22.8	19.7	24.9
PIVNO($256^2 \times 1$)	4.5	**3.4**	**4.7**	**4.4**	**6.6**
PIVNO($128^2 \times 2$)	4.1	3.6	8.8	9.6	13.2
PIVNO($64^2 \times 2$)	**3.4**	5.1	8.8	10.2	13.9
PIVNO($64^4 \times 4$)	4.7	6.3	19.6	21.7	27.9

Table 3: Impact of loss term combinations. The table shows AEE and divergence results on SPID (sim) and solid body rotation (real).

L_d	L_s	L_{div}	**SPID**		**Solid Body Rotation Flow**	
			AEE	**div.**	**AEE**	**div.**
×	×	×	1.62	0.23	0.64	458
✓	×	×	5.00	-8036.29	0.24	-2676
×	✓	×	4.20	-12.46	5.43	88
×	×	✓	3.72	**-0.09**	5.45	**7.08**
✓	✓	×	1.10	-6639.26	0.23	-2072.67
✓	✓	✓	**0.60**	-13.00	**0.17**	-1424.37

4.1 Evaluation on Synthetic Datasets

Synthetic Dataset 1: This PIV dataset [23] contains five classic flow field cases commonly used for training and benchmarking PIV algorithms. As shown in table 1, the proposed PIVNO model consistently outperforms state-of-the-art (SOTA) methods across all evaluation metrics at a resolution of 256^2. Its advantage is especially notable in complex flow regimes such as DNS turbulence, JHTDB channel flow, and SQG sea surface flow. A key strength of PIVNO lies in its training strategy: the model is trained once on samples uniformly downsampled by factors from $1\times$ to $4\times$, enabling robust generalization across multiple upsampling scales during inference. Notably, even with low-resolution inputs (e.g., 64^2), PIVNO produces high-resolution outputs with accuracy comparable to models using full-resolution inputs. Furthermore, when the output resolution is twice that of the input, PIVNO still achieves superior velocity field estimation, demonstrating strong robustness and adaptability in low-resolution settings.

Additionally, fig. 2 presents visual comparisons. The first row displays DNS turbulence, featuring rich small-scale vortical structures and multi-scale turbulence interactions, demonstrating high complexity and dynamic characteristics. Comparatively, only PIVNO effectively captures both local features and global relationships. The second row illustrates JHTDB channel flow, marked by shear effects, stratified velocity gradients, boundary confinement, and turbulent transition behavior. Remarkably, PIVNO accurately captures these boundary layer features. The third row shows SQG sea surface flow, including nonlinear interactions between large-scale background fields and small-scale disturbances, mainly exhibiting two-dimensional quasi-geostrophic characteristics. In summary, PIVNO handles these complex dynamics with precision.

Synthetic Dataset 2: To evaluate the model's performance under large displacement and high noise conditions, we used a synthetic dataset from [25], which simulates large particle displacements, low particle density, and significant noise—providing a suitable testbed for assessing the robustness of PIV algorithms. As shown in table 2, models like UnLiteFlowNet-PIV, which rely on photometric loss, suffer in high-noise settings due to disrupted motion feature extraction. PIV-RAFT also struggles with large displacements due to limitations in its local correlation-based approach. In contrast, PIVNO maintains stable prediction performance even under such challenging conditions.

Synthetic Dataset 3: To evaluate the impact of the three loss terms (L_d, L_s, and L_{div}) on flow field estimation, we conducted ablation experiments using the SPID dataset [62], which simulates real-world conditions such as noise, particle distribution, and out-of-plane motion. We progressively removed each loss term and evaluated the changes in AEE and divergence metrics. As shown in table 3, we observed that using any single loss function in isolation resulted in a significant performance degradation, with increased AEE and anomalous divergence values, indicating that a single loss term cannot effectively constrain the model's fine-tuning. In contrast, the combination of all three loss terms yielded substantial performance gains, demonstrating their synergistic effect in optimizing flow field estimation. The consistent performance deterioration when omitting any loss function further validates the effectiveness of this fine-tuning strategy.

4.2 Generalizability on Real PIV Challenges

Solid Body Rotation Flow: To evaluate the generalization capability of the fine-tuning strategy in real-world scenarios, we selected the solid body rotation flow [63] as a classical benchmark due to its theoretical clarity, uniform vorticity, and strict adherence to solid body rotation. table 3 quantifies the impact of fine-tuning, showing that combining all three loss terms (L_d, L_s, and L_{div}) significantly

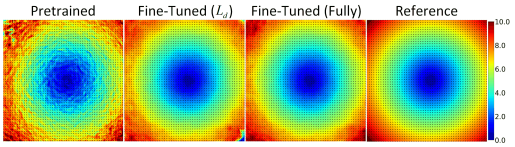

Figure 3: Comparison of the flow estimation results in the solid body rotation test.

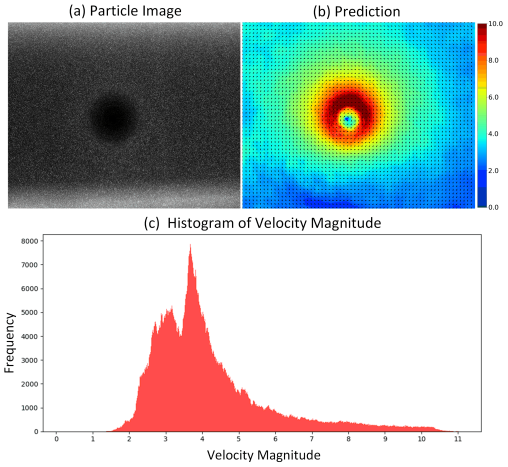

Figure 4: (a) Strong vortex particle image; (b) Estimated flow field via proposed method; (c) Velocity magnitude histogram. PIVNO accurately estimates flow and avoids "peak locking".

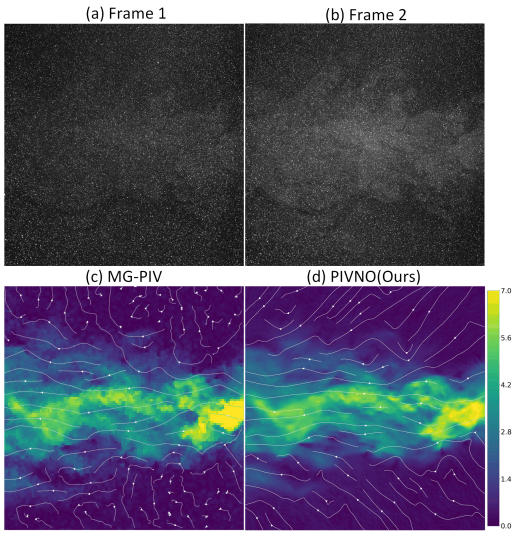

Figure 5: Comparison of turbulent round jet velocity fields: (a) and (b) are the original particle images with significant illumination variation; (c) is the fluid vector field estimated using the multigrid PIV method, appearing blurry and discontinuous; (d) is the fluid vector field estimated using PIVNO, exhibiting fair structures and continuity. The white lines indicate significant motion trajectories.

reduces AEE and improves divergence, reinforcing the model's robustness. fig. 3 illustrates this effect by comparing predictions with theoretical values, where the pre-trained model fails to capture rotational characteristics, showing flow discontinuities near the rotation center. While using only L_{div} mitigates some errors, it still yields boundary anomalies and lacks smoothness. In contrast, the full fine-tuning strategy enables precise particle displacement prediction, closely aligning with the theoretical solution and ensuring high-quality flow estimation across both boundary and rotation center regions. These findings highlight the critical role of the three-loss synergy in enhancing model generalization and robustness for flow prediction.

Strong Vortex: We use PIVNO to process the vortex flow field images recorded by the German Aerospace Center in the DNW-LLF large wind tunnel [64]. The motion field contains complex characteristics such as high velocity gradients, particle density loss, size variations and small particles, making it ideal for testing the robustness of our method. Experimental results show that our approach accurately reconstructs intricate velocity distributions even under low particle density and steep velocity gradients, as illustrated in fig. 4(b). In the experimental images, particle sizes are less than two pixels, causing grayscale distributions to suffer from pixelation effects, which hinder precise subpixel-scale localization. During displacement measurement, this blurring effect biases measured values toward integer pixel positions rather than forming a continuous distribution, leading to the so-called peak-locking effect [65] and reducing accuracy. Our method effectively mitigates this issue.

Turbulent Jet: We evaluated the turbulent round jet dataset from Delft University of Technology [66], focusing on high-gradient regions and flow continuity. As shown in fig. 5(a) and (b), significant lighting changes between the two frames challenged flow field estimation. To obtain the velocity field, the dataset provider uses a multi-grid PIV method (fig. 5(c)). However, due to resolution limits of the grid-based approach, it struggles to resolve fine-scale flow structures, causing visible blurring artifacts. In the experiment, fluid is injected from middle-right toward middle-left, while the upper and lower sections are expected to flow inward due to pressure differences. Nevertheless, the multi-grid PIV method shows irregular and discontinuous flow in these regions, failing to preserve

Table 4: Ablation comparisons on Synthetic Dataset 1.

RoPE-Mixed	DWConv	GA	GRU	GCN	Uniform	Backstep	Cylinder	JHTDB Channel	DNS turbulence	SQG
			×		10.73	15.55	4.12	8.95	20.01	14.96
				×	4.04	3.07	**0.78**	1.88	3.94	2.90
×	×	×			6.68	2.30	0.95	1.93	3.85	3.06
×	×				3.31	2.43	1.08	1.73	3.57	2.61
×					4.03	2.03	0.84	1.84	3.67	2.73
	×				3.40	2.56	0.87	1.71	3.50	2.57
					3.26	**1.91**	0.79	**1.68**	**3.47**	**2.54**

expected motion coherence. In contrast, our method (fig. 5(d)) captures the central flow trend in both upper and lower regions more effectively, yielding a more structured and continuous velocity field.

4.3 Ablation Studies

We conducted ablation studies on Synthetic Dataset 1 to assess the importance of each module in PIVNO. As shown in Table 4, removing any single component degrades performance. Notably, the GA and GRU modules have the most significant impact when being removed, indicating their essential roles in the overall architecture. Other components such as RoPE-Mixed and DWConv also contribute consistently. These results validate the necessity of the full model design. More ablation experiments can be found in the supplementary material (Case B).

5 Conclusion

We propose PIVNO, a neural operator framework that formulates PIV as a PDE-constrained inverse problem. By leveraging a Galerkin-style attention mechanism and a self-supervised fine-tuning scheme grounded in physical constraints, PIVNO achieves accurate and resolution-adaptive flow estimation across synthetic and real-world datasets. The framework demonstrates strong generalization and robustness capabilities, highlighting its potential for high-precision PIV applications.

Limitations. 2D particle velocimetry cannot fully capture the true 3D nature of fluid dynamics. Future work will focus on extending the framework to 3D flow field estimation. Additionally, the current feature extraction module lacks multi-scale feature fusion, which should be taken into consideration when processing high-resolution PIV data.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China under Grants 61871055 and 62025102.

References

[1] Jared L Callaham, Georgios Rigas, Jean-Christophe Loiseau, and Steven L Brunton. An empirical mean-field model of symmetry-breaking in a turbulent wake. *Science Advances*, 8 (19):eabm4786, 2022.

[2] Alexandre Vilquin, Julie Jagielka, Simeon Djambov, Hugo Herouard, Patrick Fischer, Charles-Henri Bruneau, Pinaki Chakraborty, Gustavo Gioia, and Hamid Kellay. Asymptotic turbulent friction in 2d rough-walled flows. *Science Advances*, 7(5):eabc6234, 2021.

[3] Yohan Sequeira, Abhradeep Maitra, Anupam Pandey, and Sunghwan Jung. Revisiting the nasa surface tension driven convection experiments. *npj Microgravity*, 8(1):5, 2022.

[4] Jiajia Ma, Shuming Chen, Peter Bellotti, Renyu Guo, Felix Schäfer, Arne Heusler, Xiaolong Zhang, Constantin Daniliuc, M Kevin Brown, Kendall N Houk, et al. Photochemical intermolecular dearomative cycloaddition of bicyclic azaarenes with alkenes. *Science*, 371(6536): 1338–1345, 2021.

[5] Solomon H Reisberg, Yang Gao, Allison S Walker, Eric JN Helfrich, Jon Clardy, and Phil S Baran. Total synthesis reveals atypical atropisomerism in a small-molecule natural product, tryptorubin A. *Science*, 367(6476):458–463, 2020.

[6] Jessica L Wilson, Amir A Pahlavan, Martin A Erinin, Camille Duprat, Luc Deike, and Howard A Stone. Aerodynamic interactions of drops on parallel fibres. *Nature Physics*, 19(11):1667–1672, 2023.

[7] Bagrat Grigoryan, Samantha J Paulsen, Daniel C Corbett, Daniel W Sazer, Chelsea L Fortin, Alexander J Zaita, Paul T Greenfield, Nicholas J Calafat, John P Gounley, Anderson H Ta, et al. Multivascular networks and functional intravascular topologies within biocompatible hydrogels. *Science*, 364(6439):458–464, 2019.

[8] Adam Shellard, András Szabó, Xavier Trepat, and Roberto Mayor. Supracellular contraction at the rear of neural crest cell groups drives collective chemotaxis. *Science*, 362(6412):339–343, 2018.

[9] Song Liu, Suraj Shankar, M Cristina Marchetti, and Yilin Wu. Viscoelastic control of spatiotemporal order in bacterial active matter. *Nature*, 590(7844):80–84, 2021.

[10] Donghwan Kim, Jisoo Shin, Yousang Son, and Sungwook Park. Characteristics of in-cylinder flow and mixture formation in a high-pressure spray-guided gasoline direct-injection optically accessible engine using piv measurements and cfd. *Energy Conversion and Management*, 248: 114819, 2021.

[11] Esther Lagemann, Steven L Brunton, Wolfgang Schröder, and Christian Lagemann. Towards extending the aircraft flight envelope by mitigating transonic airfoil buffet. *Nature Communications*, 15(1):5020, 2024.

[12] John F Zimmerman, Daniel J Drennan, James Ikeda, Qianru Jin, Herdeline Ann M Ardoña, Sean L Kim, Ryoma Ishii, and Kevin Kit Parker. Bioinspired design of a tissue-engineered ray with machine learning. *Science Robotics*, 10(99):eadr6472, 2025.

[13] Mingrui Zhang, Jianhong Wang, James B Tlhomole, and Matthew Piggott. Learning to estimate and refine fluid motion with physical dynamics. In *International Conference on Machine Learning*, pages 26575–26590. PMLR, 2022.

[14] Mehdi Dehghan. Numerical solution of the three-dimensional advection–diffusion equation. *Applied Mathematics and Computation*, 150(1):5–19, 2004.

[15] Julia Ingelmann, Sachin S Bharadwaj, Philipp Pfeffer, Katepalli R Sreenivasan, and Jörg Schumacher. Two quantum algorithms for solving the one-dimensional advection–diffusion equation. *Computers & Fluids*, 281:106369, 2024.

[16] M Thongmoon and R McKibbin. A comparison of some numerical methods for the advection-diffusion equation. 2006.

[17] Christian E Willert and Morteza Gharib. Digital particle image velocimetry. *Experiments in fluids*, 10(4):181–193, 1991.

[18] Jerry Westerweel. Fundamentals of digital particle image velocimetry. *Measurement science and technology*, 8(12):1379, 1997.

[19] Andreas Schröder and Christian E Willert. Particle image velocimetry. *Particle Image Velocimetry: New Developments and Recent Applications*, 112, 2008.

[20] Ronald J Adrian and Jerry Westerweel. *Particle image velocimetry*. Number 30. Cambridge university press, 2011.

[21] Hassan Abdulmouti. Particle imaging velocimetry (piv) technique: Principles and applications, review. *Yanbu Journal of Engineering and Science*, 6(1):35–65, 2021.

[22] Yong Lee, Hua Yang, and Zhouping Yin. Piv-dcnn: cascaded deep convolutional neural networks for particle image velocimetry. *Experiments in Fluids*, 58:1–10, 2017.

[23] Shengze Cai, Shichao Zhou, Chao Xu, and Qi Gao. Dense motion estimation of particle images via a convolutional neural network. *Experiments in Fluids*, 60:1–16, 2019.

[24] Mingrui Zhang and Matthew D Piggott. 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 35*, pages 102–115. Springer, 2020.

[25] Christian Lagemann, Kai Lagemann, Sach Mukherjee, and Wolfgang Schröder. Deep recurrent optical flow learning for particle image velocimetry data. *Nature Machine Intelligence*, 3(7): 641–651, 2021.

[26] Qi Gao, Hongtao Lin, Han Tu, Haoran Zhu, Runjie Wei, Guoping Zhang, and Xueming Shao. A robust single-pixel particle image velocimetry based on fully convolutional networks with cross-correlation embedded. *Physics of Fluids*, 33(12), 2021.

[27] Wei Zhang, Xiangyu Nie, Xue Dong, and Zhiwei Sun. Pyramidal deep-learning network for dense velocity field reconstruction in particle image velocimetry. *Experiments in Fluids*, 64(1): 12, 2023.

[28] Christian Lagemann, Kai Lagemann, Sach Mukherjee, and Wolfgang Schröder. Challenges of deep unsupervised optical flow estimation for particle-image velocimetry data. *Experiments in Fluids*, 65(3):30, 2024.

[29] Yuxuan Han and Qian Wang. An attention-mechanism incorporated deep recurrent optical flow network for particle image velocimetry. *Physics of Fluids*, 35(7), 2023.

[30] Lento Manickathan, Claudio Mucignat, and Ivan Lunati. Kinematic training of convolutional neural networks for particle image velocimetry. *Measurement Science and Technology*, 33(12): 124006, 2022.

[31] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[32] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

[33] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[34] Bogdan Raonic, Roberto Molinaro, Tobias Rohner, Siddhartha Mishra, and Emmanuel de Bezenac. Convolutional neural operators. In *ICLR 2023 Workshop on Physics for Machine Learning*, 2023.

[35] Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, pages 1–9, 2024.

[36] Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.

[37] Jean Rabault, Jostein Kolaas, and Atle Jensen. Performing particle image velocimetry using artificial neural networks: a proof-of-concept. *Measurement Science and Technology*, 28(12): 125301, 2017.

[38] P Lavoie, G Avallone, F De Gregorio, Giovanni Paolo Romano, and RA Antonia. Spatial resolution of piv for the measurement of turbulence. *Experiments in Fluids*, 43:39–51, 2007.

[39] Zhibo Wang, Xiangru Li, Luhan Liu, Xuecheng Wu, Pengfei Hao, Xiwen Zhang, and Feng He. Deep-learning-based super-resolution reconstruction of high-speed imaging in fluids. *Physics of Fluids*, 34(3), 2022.

[40] Hongping Wang, Yi Liu, and Shizhao Wang. Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network. *Physics of fluids*, 34(1), 2022.

[41] Bonan Xu, Yuanye Zhou, and Xin Bian. Self-supervised learning based on transformer for flow reconstruction and prediction. *Physics of Fluids*, 36(2), 2024.

[42] Kai Fukami, Koji Fukagata, and Kunihiko Taira. Super-resolution analysis via machine learning: a survey for fluid flows. *Theoretical and Computational Fluid Dynamics*, 37(4):421–444, 2023.

[43] Longyan Wang, Zhaohui Luo, Jian Xu, Wei Luo, and Jianping Yuan. A novel framework for cost-effectively reconstructing the global flow field by super-resolution. *Physics of Fluids*, 33 (9), 2021.

[44] Mustafa Z Yousif, Linqi Yu, and Hee-Chang Lim. Super-resolution reconstruction of turbulent flow fields at various reynolds numbers based on generative adversarial networks. *Physics of Fluids*, 34(1), 2022.

[45] Wei Zhang, Xue Dong, Zhiwei Sun, and Shuogui Xu. An unsupervised deep learning model for dense velocity field reconstruction in particle image velocimetry (piv) measurements. *Physics of Fluids*, 35(7), 2023.

[46] Gazi Hasanuzzaman, Hamidreza Eivazi, Sebastian Merbold, Christoph Egbers, and Ricardo Vinuesa. Enhancement of piv measurements via physics-informed neural networks. *Measurement Science and Technology*, 34(4):044002, 2023.

[47] Shaorong Yu, Baozhu Zhao, Jialei Song, and Yong Zhong. A hybrid unsupervised learning approach for noise removal in particle image velocimetry. *Physics of Fluids*, 36(11), 2024.

[48] C Lagemann and W Schröder. Key aspects of unsupervised optical flow models in piv applications. In *15th international symposium on particle image velocimetry*, volume 1, 2023.

[49] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024.

[50] Min Wei and Xuesong Zhang. Super-resolution neural operator. In *CVPR*, pages 18247–18256, 2023.

[51] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[52] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

[53] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? In *International Conference on Learning Representations*, 2020.

[54] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021.

[55] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[56] Zachary Teed and Jia Deng. 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*, pages 402–419. Springer, 2020.

[57] Anuj Sharma, Sukhdeep Singh, and S Ratna. Graph neural network operators: a review. *Multimedia Tools and Applications*, 83(8):23413–23436, 2024.

[58] Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, B Khaled Letaief, and Dongsheng Li. How powerful is graph convolution for recommendation? In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 1619–1629, 2021.

[59] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.

[60] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, pages 363–370. Springer, 2003.

[61] Kexin Ji, Xin Hui, and Qiang An. High-resolution velocity determination from particle images via neural networks with optical flow velocimetry regularization. *Physics of Fluids*, 36(3), 2024.

[62] M. Machado and D. Rocha. Synthetic particle image dataset (spid), May 2023.

[63] Christian J Kähler, Tommaso Astarita, Pavlos P Vlachos, Jun Sakakibara, Rainer Hain, Stefano Discetti, Roderick La Foy, and Christian Cierpka. Main results of the 4th international piv challenge. *Experiments in Fluids*, 57:1–71, 2016.

[64] Michel Stanislas, Koji Okamoto, and Christian Kähler. Main results of the first international piv challenge. *Measurement Science and Technology*, 14(10):R63, 2003.

[65] Markus Raffel, Christian E Willert, Fulvio Scarano, Christian J Kähler, Steve T Wereley, and Jürgen Kompenhans. *Particle image velocimetry: a practical guide*. springer, 2018.

[66] RJ Adrian, DFG Durao, MV Heitor, M Maeda, C Tropea, JH Whitelaw, C Fukushima, L Aanen, and J Westerweel. Investigation of the mixing process in an axisymmetric turbulent jet using piv and lif. In *Laser Techniques for Fluid Mechanics: Selected Papers from the 10th International Symposium Lisbon, Portugal July 10–13, 2000*, pages 339–356. Springer, 2002.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract accurately reflects the contributions and scope. We explicitly highlight the main contributions in the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss two key limitations of our current work in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This does not apply to our research work because our study is not theoretical in nature. Instead, it focuses on developing a neural operator framework and its application to particle image velocimetry through extensive experimental validation.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed experimental setup and implementation details in the supplementary material, and will release code and model weights upon paper acceptance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release the code and model weights upon acceptance on paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide detailed experimental setup and implementation details in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Following standard practice in the particle image velocimetry (PIV) field, we do not report error bars, as the evaluation metrics such as Average Endpoint Error (AEE) are deterministic and typically reported from a single run.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide detailed information about our experimental environment, including hardware and runtime settings, in the supplementary material to support reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We completely comply with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See the end of the introduction section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We use publicly available models and datasets, and our work does not involve the release of models or data that pose a high risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available datasets and pretrained models, each of which is properly cited in the paper. All assets are used in compliance with their respective licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We plan to release the code and pretrained models upon paper acceptance. The released assets will include documentation on training procedures, usage instructions, and license information to ensure accessibility and reproducibility.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve research with human subjects and therefore does not include an IRB.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were not used as part of the research methodology. Any LLM use was limited to minor writing and editing support and did not influence the scientific content or originality of the paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.