

Section A: RoPE Galerkin Attention

A1. Integrating Positional Embedding with Frequency Modulation

RoPE-Mixed [1] introduces a complex-valued rotation matrix to inject 2D positional information into the attention mechanism of Vision Transformers. For a token located at spatial position n with coordinates $(p_n^x, p_n^y) \in \{0, \dots, W-1\} \times \{0, \dots, H-1\}$, the positional rotation matrix is defined as:

$$R(n, t) = e^{i(\theta_t^x p_n^x + \theta_t^y p_n^y)}, \quad (1)$$

where θ_t^x and θ_t^y are learnable frequency parameters associated with the t -th dimension of the head, and $t \in \{0, \dots, \frac{d_{\text{head}}}{2} - 1\}$. This positional encoding is applied to the query and key vectors in the spatial domain by complex modulation:

$$q'_n = q_n e^{i(\theta_t^x p_n^x + \theta_t^y p_n^y)}, \quad k'_m = k_m e^{i(\theta_t^x p_m^x + \theta_t^y p_m^y)}. \quad (2)$$

The (n, m) -th component of the attention matrix is then computed as:

$$A'_{(n,m)} = \text{Re} [q'_n k_m'^*] = \text{Re} \left[q_n k_m^* \cdot e^{i(\theta_t^x (p_n^x - p_m^x) + \theta_t^y (p_n^y - p_m^y))} \right], \quad (3)$$

which encodes relative positional differences along both axes using a complex exponential form governed by frequency parameters. This formulation enables the attention mechanism to capture directional and spatial relationships through learnable frequencies.

Nevertheless, the inherent quadratic complexity of Eq. (3) and its limit to relative position encoding makes RoPE-mixed unsuitable for PIV which demands dense motion estimation on every absolute position. So we adapt the linear Galerkin-type attention (GA) [2] with absolute positional embedding and learnable frequency modulation. GA performs feature aggregation *along the channel dimension* rather than across spatial positions, which corresponds to the Petrov–Galerkin projection in the Finite Element Method [3], devised for operator learning tasks arising in partial differential equations (PDEs).

Given an input latent representation $y \in \mathbb{R}^{n \times d}$, where n denotes the number of spatial locations and d the feature dimension at each location, Galerkin attention first computes the query, key, and value representations through linear projections:

$$Q = yW_Q, \quad K = yW_K, \quad V = yW_V, \quad (4)$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ are learnable projection matrices.

Then GA is computed as:

$$\text{Attn}_{\text{Gal}}(y) = \frac{1}{n} Q(K^\top V). \quad (5)$$

From the perspective of operator learning, GA fulfills a mapping between functions, so the columns of Q , K , and V are viewed as learned basis functions evaluated at discrete spatial locations x_1, x_2, \dots, x_n .

Accordingly, the output feature at the j -th channel and spatial location x_i is given by:

$$z_j(x_i) = (\text{Attn}_{\text{Gal}}(y))_{ij} = \frac{1}{n} \sum_{l=1}^d \langle v_j, k_l \rangle \cdot q_l(x_i), \quad (6)$$

where the inner product $\langle v_j, k_l \rangle := (K^\top V)_{lj}$ represents the projection of the trial basis v_j onto the test basis k_l . Eq. (6) tells that GA performs a Petrov–Galerkin projection by testing the trial basis in V against the test basis in K , and reconstructs the output using the output basis in Q .

Now we reveal two key limitations of Eq. (6). First, it lacks explicit positional encoding. For 2D flow field prediction, where the goal is to infer the flow from two consecutive frames, absolute positional information is crucial. More importantly, both the test space K and the trial space V are obtained through linear projections of the input. This restricts their representational capability, making the model potentially inadequate for capturing complex interactions beyond linear correlations. **Attention should not be confined to linear correlations.**

To address these limitations, we adopt the positional encoding scheme of [1] to enhance the Galerkin attention mechanism. Specifically, our RoPE-GA introduces a complex-valued rotation matrix for channel-wise frequency modulation, formulated as:

$$R(n, c) = e^{i(\theta_c^x p_n^x + \theta_c^y p_n^y)}, \quad (7)$$

which modulates each basis function in K and V onto a distinct frequency defined by learnable parameters θ_c^x and θ_c^y for each channel c . This modulation enhances the expressiveness of the learned basis functions:

$$k'_{c_1} = k_{c_1} \cdot e^{i(\theta_{c_1}^x p_n^x + \theta_{c_1}^y p_n^y)}, \quad v'_{c_2} = v_{c_2} \cdot e^{i(\theta_{c_2}^x p_n^x + \theta_{c_2}^y p_n^y)}. \quad (8)$$

This modulation mechanism directly injects absolute spatial coordinates p_n^x and p_n^y into the query and key representations, while also introducing frequency-based differences across channels.

Then the attention coefficient between channels c_1 and c_2 is computed as:

$$\hat{A}(c_1, c_2) = \sum_{n=0}^{N-1} k_{c_1} v_{c_2} e^{i(p_n^x(\theta_{c_1}^x - \theta_{c_2}^x) + p_n^y(\theta_{c_1}^y - \theta_{c_2}^y))} \quad (9)$$

Eq. (9) can be interpreted as a modulation of the original attention $A(c_1, c_2)$ by a learnable frequency $(\theta_{c_1}^x - \theta_{c_2}^x, \theta_{c_1}^y - \theta_{c_2}^y)$. As a result, the final attention $\hat{A}(c_1, c_2)$ is no longer a simple linear summation over all spatial positions, but rather an enhanced synthesis with frequency modulations, contributing to more powerful expressive capability.

A2. Mapping in the function space

The objective of our proposed RoPE-GA is to achieve a mapping from particle image feature functions to flow field feature functions in their respective latent function spaces. To validate the effectiveness of this design, we visualized both the input particle image feature maps and the output flow field feature maps of RoPE-GA, as shown in Figure 1. The three images on the right in the first row represent the particle image feature maps fed into RoPE-GA, while the three images on the right in the second row correspond to the flow field feature maps generated by RoPE-GA. For the convenience of comparison, the first column additionally displays the original particle image and its corresponding flow field. We can find that the features before RoPE-GA, although undergoing the deep feature encoder’s processing, still exhibit very localized speckle-like patterns (resembling more the original particle images), while the post features apparently reveal the globally structural patterns (following the continuous nature of fluids).

A3. Ablation experiment

To **further validate** the superiority of the RoPE-GA operator in mapping particle image feature functions to flow field feature functions in the function space, we conduct an ablation study comparing our approach with a traditional Cost Volume (CV)-based method. Specifically, we construct a baseline variant, denoted as **PIVNO_{cv}**, in which the RoPE-GA operator is replaced by a standard CV construction module following the RAFT-PIV framework. As summarized in Table 1, the RoPE-GA-based PIVNO significantly outperforms the CV-based counterpart across all evaluated synthetic datasets. Our method achieves lower AEE in all scenarios, especially in more complex turbulent fields (e.g., DNS turbulence: **3.5 vs. 10.0**), demonstrating stronger expressiveness and learning capacity in function space. Notably, the RoPE-GA model requires fewer parameters (2.52M vs. 3.19M) and offers improved runtime efficiency, underscoring its advantage not only in accuracy but also in computational compactness. Even under degraded input resolutions (e.g., 64^2), our operator-based design maintains robust performance compared to **PIVNO_{cv}**. This ablation experiment complements our previous evaluations and offers additional empirical evidence that the RoPE-GA operator enables a more efficient and accurate mapping paradigm than conventional CV-based approaches.

Section B: Details of the Model and Training

B1. Model Architecture

Our feature extraction module adopts a residual block design, as illustrated in Fig. 2. The structure consists of a 1×1 convolution branch and a parallel two-stage convolutional block with 3×3 filters,

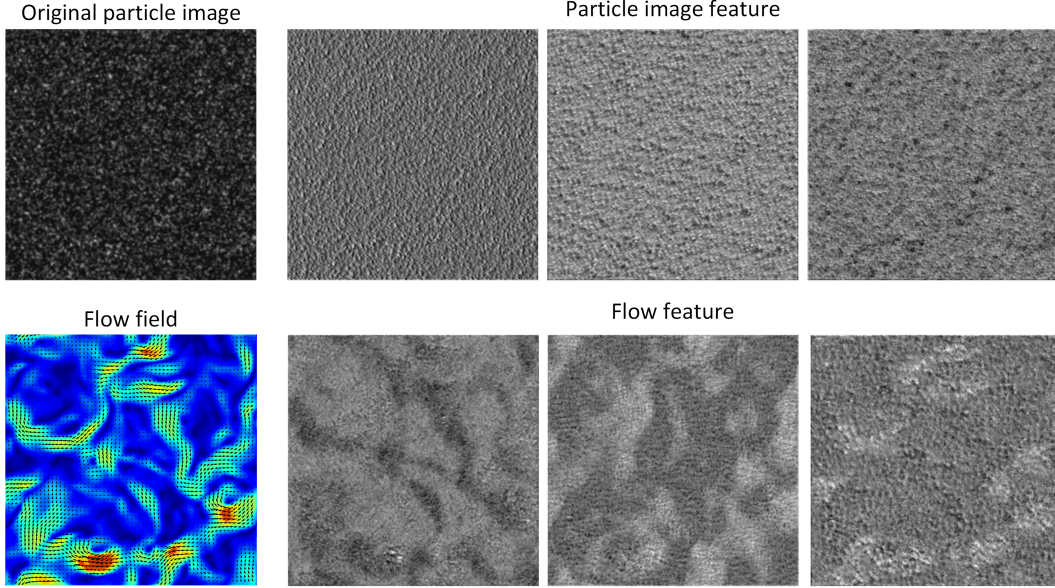


Figure 1: Visualization of the feature mapping process in RoPE-GA.

Table 1: **Further comparison of the Average Endpoint Error (AEE)** on five synthetic datasets. The error unit is pixels per 100 pixels. All models output 256^2 resolution flow fields. The top block reports performance of **PIVNO_{cv}**, which follows the RAFT-PIV architecture using Cost Volume (CV). The lower block shows our proposed RoPE-GA-based method. Input resolution variants (e.g., $64^2 \times 4$) use upsampling to match the 256^2 output. “Params” represents the number of trainable parameters (in millions); FLOPs and Runtime are measured per inference.

Methods	Backstep	Cylinder	JHTDB Channel	DNS turbulence	SQG	Params. (M)	FLOPs (G)	Runtime (ms)
PIVNO_{cv} ($256^2 \times 1$)	3.4	2.1	4.6	10.0	8.6	3.19	836	104
PIVNO_{cv} ($128^2 \times 2$)	3.5	2.6	6.1	13.7	10.1		283	27
PIVNO_{cv} ($64^2 \times 2$)	2.8	3.7	5.8	11.7	9.7		54	14
PIVNO_{cv} ($64^2 \times 4$)	6.6	7.0	17.4	29.7	28.8		145	19
PIVNO ($256^2 \times 1$)	1.9	0.8	1.7	3.5	2.5	2.52	572	73
PIVNO ($128^2 \times 2$)	0.9	1.1	2.8	4.7	4.1		217	28
PIVNO ($64^2 \times 2$)	1.0	1.4	2.9	5.0	4.2		54	14
PIVNO ($64^2 \times 4$)	2.7	3.1	11.3	18.0	18.0		128	18

each followed by Instance Normalization (IN) and ReLU activation. The outputs of both branches are combined via element-wise addition, followed by a final ReLU activation.

In the RoPE Galerkin Attention module, we adopt a lightweight iterative formulation with $t = 2$ iterations. For the GRU-based recurrent refinement module, Fig. 3 shows that performance gains saturate beyond 4 iterations, with slight degradation observed for the Uniform flow type when the count exceeds 6. Based on this analysis, we use 5 iterations in all experiments to balance accuracy and efficiency.

B2. Training Setup

We follow a two-stage training protocol, consisting of supervised pretraining and self-supervised fine-tuning. All configurations are summarized in Table 2. The experiments are conducted on two NVIDIA Tesla P40 GPUs.

B3. Sampling Strategy

To support arbitrary-scale prediction, we design a sampling strategy that explicitly incorporates random scale variation into the training process, enabling the model to generalize across a continuous

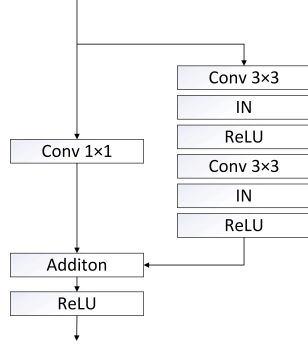


Figure 2: Residual block structure used in the feature extraction module.

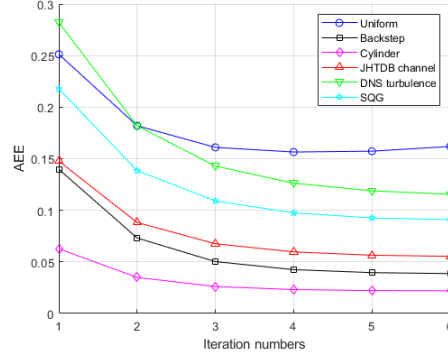


Figure 3: Recurrent flow refinement estimated by GRU on synthetic dataset 1.

Table 2: Training and fine-tuning configurations used in our study.

Config	Value
<i>Supervised Training</i>	
Datasets	Synthetic Dataset 1, 2
Loss function	L2 loss (Eq. 10)
Optimizer	Adam
Initial learning rate	1×10^{-4}
Max learning rate	5×10^{-4}
Batch size	32
Training epochs	1500
Learning rate schedule	Cosine annealing
Warm-up epochs	120
<i>Self-Supervised Fine-Tuning</i>	
Datasets	Synthetic Dataset 1 + 3 Real-world PIV tasks
Loss function	LP loss (Eq. 11)
Optimizer	Adam
Initial learning rate	5×10^{-4}
Batch size	2
Training epochs	100
Learning rate schedule	Cosine annealing
Warm-up epochs	None
Pre-trained weights	From Synthetic Dataset 1

range of resolutions. In each training batch of size B , we uniformly sample B random scale factors $\{r^{(i)}\}_{i=1}^B \sim \mathcal{U}(1, 4)$. For each scale $r^{(i)}$, we crop a high-resolution (HR) patch of size $\sqrt{n_{hc}} \cdot r^{(i)} \times \sqrt{n_{hc}} \cdot r^{(i)}$ from the training image, where $n_{hc} = 64^2$ is the fixed number of LR grid sampling points. The corresponding low-resolution (LR) patch is generated via bicubic interpolation with the same scale factor.

To supervise across varying scales consistently, we randomly sample n_{hc} pixels from each HR patch. These samples serve as supervision anchors during training. The input to the model is LR image, which is processed by the feature extraction module and the RoPE-GA operator to produce a continuous flow feature representation in function space. Given the continuous nature of this representation, the flow features at the sampled HR coordinates can be recovered via interpolation from the LR grid. This interpolation is performed by the SR module, whose detailed mechanism is described in Section 4.2 of the main paper. A visual illustration of multi-scale prediction results is provided in Fig. 4.

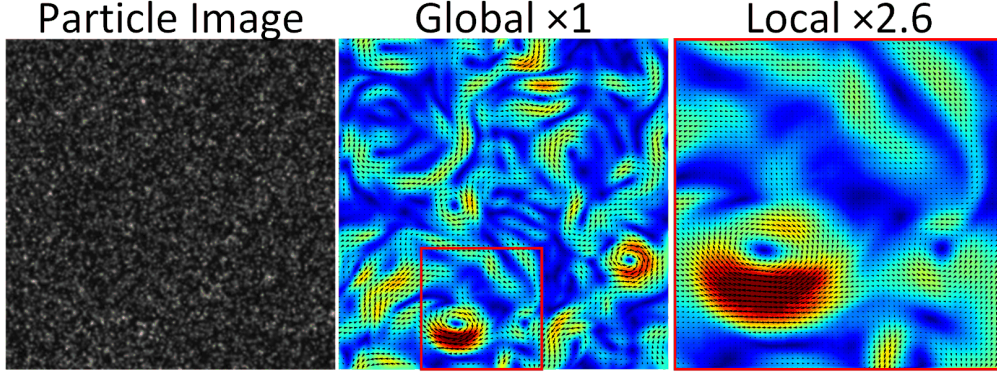


Figure 4: Visualization of predicted flow fields at different spatial scales and regions. Without retraining, our operator supports arbitrary-scale and region-specific flow prediction from a single low-resolution input.

Section C: Loss Functions and Evaluation Metrics

C1. Supervised Training Loss

For supervised training, we adopt a multi-stage regression loss that penalizes the ℓ_1 distance between the predicted and ground-truth flow fields at each iteration. The total loss is defined as:

$$L = \sum_{i=1}^N \gamma^{N-i} \|\mathbf{u}_{\text{gt}} - \mathbf{u}_{\text{es},i}\|_1 \quad (10)$$

where \mathbf{u}_{gt} is the ground-truth optical flow, $\mathbf{u}_{\text{es},i}$ is the estimated flow at iteration i , and γ is an exponential weighting factor. We set $\gamma = 0.8$ and use $N = 5$ iterations during training.

C2. Self-Supervised Fine-Tuning Loss

For domain adaptation to real-world experimental data, we employ a self-supervised loss defined as:

$$L_P(u) = L_d(u) + \lambda_s L_s(u) + \lambda_d L_{\text{div}}(u), \quad (11)$$

where:

- $L_d(u)$: appearance similarity term, computed based on normalized patch-wise correlation between I_1 and its warped version \hat{I}_1 .
- $L_s(u)$: spatial smoothness, defined via the Laplacian operator $\nabla^2 u$.
- $L_{\text{div}}(u)$: divergence regularization, enforcing incompressibility constraint.

The Charbonnier penalty function is used for all terms:

$$\sigma(z) = (z^2 + \epsilon^2)^\gamma \quad (12)$$

Hyperparameter Configuration:

- Charbonnier parameters: $\gamma = 0.45$, $\epsilon = 10^{-3}$
- Local correlation window size: 9×9
- Regularization weights: $\lambda_s = 9598$, $\lambda_d = 27.6$

The regularization weights λ_s and λ_d are set based on the relative magnitudes of the three loss terms under converged training on Synthetic Dataset 1. Since deep learning models minimize the loss via backpropagation, but $L_d(u)$, defined via patch-wise normalized correlation, cannot approach zero even in well-trained models, we calibrate the weights by using the typical magnitudes at convergence as reference. Specifically, we first pre-train the model on Synthetic Dataset 1 and compute the average values of the three loss components across all 13,650 training pairs at convergence:

- $\text{Avg}(L_d) = 0.00905$
- $\text{Avg}(L_s) = 9.43 \times 10^{-7}$
- $\text{Avg}(L_{\text{div}}) = 3.28 \times 10^{-4}$

We treat L_d as the primary reference and set $\lambda_d = 1$. The other weights are computed by aligning their magnitudes with L_d to ensure balanced gradients:

$$\lambda_s = \frac{\text{Avg}(L_d)}{\text{Avg}(L_s)} \approx 9598, \quad \lambda_d = \frac{\text{Avg}(L_d)}{\text{Avg}(L_{\text{div}})} \approx 27.6 \quad (13)$$

This normalization ensures that all three terms contribute meaningfully to the final self-supervised loss during training and reflects the ideal weighting observed in a converged regime.

C3. Evaluation Metric

We report the average endpoint error (AEE) as the evaluation metric, which measures the Euclidean distance between the final predicted optical flow and the ground truth. It is defined as:

$$\text{AEE} = \|\mathbf{u}_{\text{es},N} - \mathbf{u}_{\text{gt}}\|_1 \quad (14)$$

This value is averaged over all pixels and all test samples.

Section D: Dataset

Table 3 summarizes the details of all datasets used in our study, including both synthetic and real-world fluid dynamics datasets. We adhere to the official train/test splits provided by the dataset authors or associated benchmark guidelines.

Table 3: Overview of synthetic and real-world datasets used in the experiments.

Dataset	Resolution	Number of Samples
Synthetic Dataset 1 [4]	256×256	13,650
Synthetic Dataset 2 [5]	256×256	20,150
Synthetic Dataset 3 [6]	665×630	3240
Solid Body Rotation Flow [7]	1024×1024	2
Strong Vortex [8]	1280×1024	2
Turbulent Jet [9]	992×1004	200

D1. Solid Body Rotation Flow

In this section, we provide detailed information on Solid Body Rotation Flow.

The experimental data comes from Case F of the Main Results of the 4th International PIV Challenge [7].

The experiment was conducted using a plexiglass cylinder with a diameter of 65 mm and a depth of 10 mm, filled with a water/glycerin solution containing silver-coated hollow-glass particles (S-HGS-10, $10 \mu\text{m}$, Dantec) as tracers. To minimize surface deformation effects on optical measurements, a BK7 glass plate (Edmund Scientific) was placed on top of the cylinder. The cylinder was mounted on the turntable of a record player, which rotated at a constant angular velocity of $\omega = 33 + 1/3 \text{ rpm}$ ($10\pi/9 \text{ rad/s}$). After a short period, the liquid inside reached a steady-state solid-body rotation.

An Nd-YAG laser (EverGreen 70, $2 \times 70 \text{ mJ}$, 15 Hz, Quantel) was used to generate a horizontal laser sheet approximately 4 mm thick, illuminating a cross-section of the rotating fluid. The motion of the illuminated particles was recorded using a CMOS camera (Fastcam SA3, 1024×1024 pixels, Photron) equipped with an AF Nikkor 28–105 mm lens (Nikon). The focal length was set to 180 mm with an F-number of 22, resulting in a field of view of approximately $38 \times 38 \text{ mm}$.

To measure the velocity field, a double-pulse laser system was used, firing at times $t = 0$ and $t = \Delta t_{\text{rev}}$, where $\Delta t_{\text{rev}} = 2\pi/\omega$ corresponds to one complete revolution of the cylinder. The camera

exposure time was set to 8 ms, and a frame-straddling technique was applied to control the time interval between two laser pulses, with a standard frame interval of $\Delta t = 4$ ms. A total of 1000 particle image pairs were recorded over 100 s.

Ideally, the fluid flow in the rotating reference frame should remain stationary, but a slight convection effect introduced some measurement uncertainty in the particle displacement. The theoretical expression for particle image displacement is given by:

$$\Delta x_{\text{theory}} = -\omega \cdot \Delta t \cdot (y - y_c), \quad (15)$$

$$\Delta y_{\text{theory}} = \omega \cdot \Delta t \cdot (x - x_c), \quad (16)$$

where (x_c, y_c) represents the center of rotation, determined from experimental measurements. fig. 5 shows the particle images used for PIV evaluation had a resolution of 1024×1024 pixels, with typical particle diameters ranging from 2 to 3 pixels.

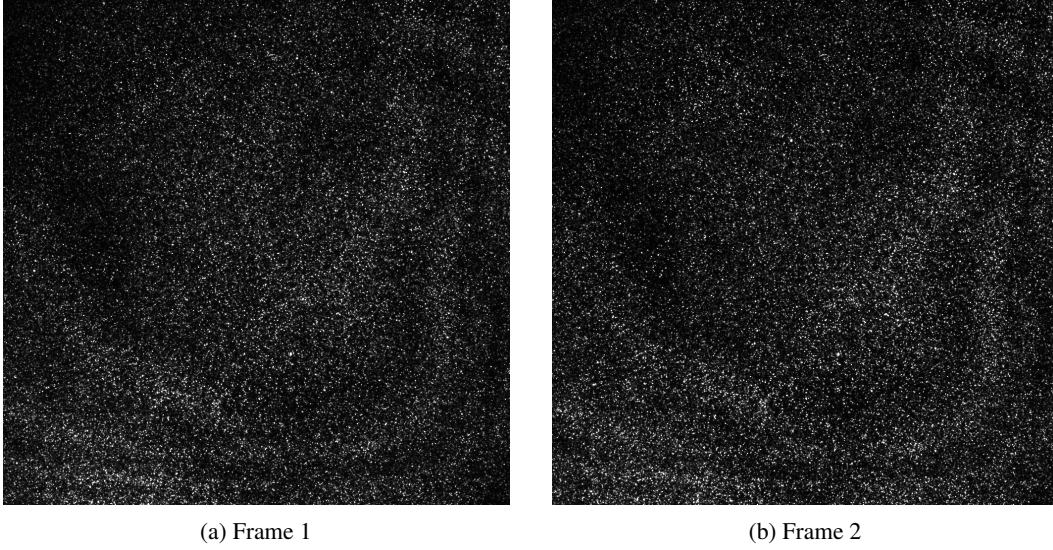


Figure 5: Particle images

Section E: Extended Experimental Analysis

E1. Statistical Robustness Analysis

Currently, in the PIV field, the Average Endpoint Error (AEE) is the mainstream evaluation metric. However, many related works have not released their source code, which makes direct comparison using alternative metrics less feasible. To provide a more comprehensive statistical perspective, we additionally report the maximum error (**Max**) and standard deviation (**Std**) of the error distribution on Synthetic Dataset 1. The quantitative results are summarized in Table 4 (unit: pixels).

It can be observed that under the $256^2 \times 1$ resolution setting, the model exhibits significantly higher maximum error and standard deviation compared to other configurations. Further inspection shows that these large deviations are mainly caused by a few extreme error points near the boundary regions, which raise the overall statistics. In contrast, the $\times 4$ and $\times 2$ resolution settings introduce inherent spatial averaging (filtering) effects during the super-resolution reconstruction process, effectively smoothing local noise and reducing both the maximum error and the error variance. These results demonstrate the statistical robustness and stability of the proposed model across different spatial resolutions.

E2. Cross-Domain Fine-Tuning Generalization

To further examine the adaptability of PIVNO across domains, we evaluated the performance of the model fine-tuned on real-world PIV datasets when tested back on synthetic datasets. The model

Table 4: Statistical robustness evaluation on Synthetic Dataset 1. Both maximum error (Max) and standard deviation (Std) are reported for different spatial resolutions.

Metric	Method	Uniform	Back-Step	Cylinder	JHTDB Channel	DNS turbulence	SQG
Max	$64^2 \times 4$	0.662	0.660	0.745	2.653	2.564	2.562
Max	$64^2 \times 2$	0.228	0.318	0.309	2.060	1.694	1.491
Max	$128^2 \times 1$	0.303	0.418	0.370	1.843	1.185	1.165
Max	$256^2 \times 1$	0.660	3.457	2.328	9.604	8.995	2.706
Std	$64^2 \times 4$	0.005	0.008	0.008	0.031	0.046	0.027
Std	$64^2 \times 2$	0.003	0.004	0.004	0.009	0.019	0.011
Std	$128^2 \times 1$	0.003	0.003	0.004	0.009	0.016	0.007
Std	$256^2 \times 1$	0.010	0.029	0.007	0.029	0.057	0.011

was initially trained on synthetic data to learn a general mapping from particle images to motion fields, and then fine-tuned in real-world scenarios to enhance task-specific performance. Although the primary objective of fine-tuning is to improve real-world accuracy, it is also important to assess how fine-tuning affects generalization to synthetic data. The quantitative results are summarized in Table 5 (unit: pixels).

Table 5: Evaluation of fine-tuned models on synthetic datasets. Results are reported as Average Endpoint Error (AEE, in pixels).

Resolution	Uniform	Back-Step	Cylinder	JHTDB Channel	DNS turbulence	SQG
$64^2 \times 4$	4.3100	2.1492	2.3437	1.1926	1.1089	0.9776
$64^2 \times 2$	6.8775	3.5106	2.5500	1.8067	1.0801	1.0307
$128^2 \times 2$	6.9424	3.0903	3.0121	1.8427	0.9936	1.0770
$256^2 \times 1$	7.4281	3.2023	3.7374	2.6118	2.2822	2.4208

Experimental results show that after fine-tuning on real-world datasets, the performance of PIVNO on synthetic datasets exhibits a moderate decline. This behavior reflects the adaptation process of fine-tuning, in which the model adjusts its feature representation to better fit the distribution of real-world data. Considering that PIVNO contains only 2.4M parameters, such trade-offs between real-world specialization and synthetic generalization are reasonable. Overall, the results confirm that PIVNO retains consistent behavior and demonstrates high adaptability, achieving strong performance across diverse domains through lightweight fine-tuning rather than relying on a single universal model.

E3. Zero-Shot Resolution Generalization

To further evaluate the generalization capability of PIVNO with respect to unseen spatial resolutions, we conducted zero-shot resolution generalization experiments. In this setting, **Var** denotes varying-resolution testing and **SR** represents super-resolution inference. Both are based on training with an input resolution of 128^2 and testing at unseen output resolutions. Specifically, Var indicates direct testing on different resolutions, while SR corresponds to super-resolution outputs at different magnification factors. For instance, $64^2 \times 4$ refers to an input resolution of 64^2 and a super-resolved output of 256^2 . **LI** denotes linear interpolation, and **Std** indicates the standard PIVNO configuration used in the main paper. The quantitative results are summarized in Table 6 (unit: pixels).

As shown in the table, the PIVNO model maintains stable performance even when tested on unseen sampling rates (e.g., $64^2 \times 1$, $256^2 \times 1$), demonstrating consistent generalization across different spatial resolutions. The results under the SR configuration indicate that the model can still perform super-resolution inference despite being trained without explicit supervision at those scales, confirming the inherent scalability of our learning framework. For comparison, the LI (linear interpolation) and Std configurations reveal that the proposed PIVNO reconstruction achieves much higher consistency

Table 6: Zero-shot resolution generalization results. Var represents varying-resolution testing, SR denotes super-resolution, LI refers to linear interpolation, and Std represents the standard PIVNO configuration.

Method	Metric	Uniform	Back-Step	Cylinder	JHTDB Channel	DNS turbulence	SQG
Var	$64^2 \times 1$	3.66	1.15	2.11	1.62	4.05	2.59
Var	$128^2 \times 1$	3.45	0.78	0.84	1.37	3.00	2.03
Var	$256^2 \times 1$	3.33	0.87	0.57	1.36	2.88	2.09
SR	$64^2 \times 4$	268.95	79.31	79.31	128.06	121.37	121.37
SR	$64^2 \times 2$	109.83	62.70	65.48	46.98	93.21	89.25
SR	$128^2 \times 2$	108.37	59.47	49.50	58.55	98.28	92.11
LI	$64^2 \times 4$	15.69	255.71	91.97	120.70	160.12	170.33
LI	$64^2 \times 2$	3.33	209.00	124.10	86.25	160.67	161.98
LI	$128^2 \times 2$	2.33	326.01	98.12	109.08	165.97	181.58
Std	$64^2 \times 4$	2.28	2.65	3.08	11.25	17.99	18.03
Std	$64^2 \times 2$	1.35	1.02	1.36	2.87	4.96	4.18
Std	$128^2 \times 2$	1.36	0.91	1.10	2.81	4.70	4.08
Std	$256^2 \times 1$	3.26	1.91	0.79	1.68	3.47	2.54

and precision than interpolation-based baselines, highlighting its robustness and adaptability under zero-shot resolution conditions.

References

- [1] 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.
- [2] Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.
- [3] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of finite elements*, volume 159. Springer, 2004.
- [4] 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.
- [5] 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.
- [6] M. Machado and D. Rocha. Synthetic particle image dataset (spid), May 2023.
- [7] 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.
- [8] Michel Stanislas, Koji Okamoto, and Christian Kähler. Main results of the first international piv challenge. *Measurement Science and Technology*, 14(10):R63, 2003.
- [9] 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.