Experiments in Fluids (2023) 64:161
https://doi.org/10.1007/s00348-023-03695-8

RESEARCH ARTICLE

A lightweight neural network designed for fluid velocimetry

Lento Manickathan[1] · Claudio Mucignat[1] · Ivan Lunati[1]

Received: 9 December 2022 / Revised: 31 July 2023 / Accepted: 12 August 2023 / Published online: 27 September 2023

Abstract
We devise a novel lightweight image matching architecture (`LIMA`), which is designed and optimized for particle image velocimetry (PIV). `LIMA` is a convolutional neural network (CNN) that performs symmetric image matching and employs an iterative residual refinement strategy, which allows us to optimize the total number of refinement steps to balance accuracy and computational efficiency. The network is trained on kinematic datasets with a loss function that penalizes larger gradients. We consider a six-level (`LIMA-6`) and a four-level (`LIMA-4`) version of the network and demonstrate that they are considerably leaner and faster than a state-of-the-art network designed for optical flow. `LIMA-6` reconstructs the velocity field from synthetic and experimental PIV images with an accuracy comparable or superior both to existing CNNs as well as to state-of-the-art cross-correlation methods (i.e., a commercial implementation of `WIDIM`). Although less accurate, `LIMA-4` allows a significant reduction of the computational costs with respect to any other method considered. All CNNs prove more robust than `WIDIM` with respect to particle loss and allow effective error reduction by increasing the particle seeding density. Thanks to reduced computational cost and memory requirement, we envision the deployment of `LIMA` on low-cost devices to provide affordable, real-time inference of the flow field during PIV measurements.

1 Introduction

Particle image velocimetry (PIV) is extensively used in experimental fluid dynamics to measure fluid velocity in a non-intrusive manner. The fluid motion is made visible by adding and illuminating tracer particles, which are close to be neutrally buoyant and are almost passively transported by the flow. The position of the illuminated particles is recorded by digital cameras at successive time intervals, and the flow field is reconstructed by comparison and analysis of these image pairs (Willert and Gharib 1991). Since the 1990s, there has been a substantial improvement in hardware and algorithms, and *window deformation iterative multigrid* (`WIDIM`) has established itself as the gold standard for PIV post-processing (Scarano 2002; Schrijer and Scarano 2008). More recently, novel machine learning (ML) techniques for optical flow, which have emerged in the field computer vision (Dosovitskiy et al. 2015), have renewed the interest of the scientific community in improving fluid velocimetry by means of non-classical techniques (Ilg et al. 2017; Chen et al. 1998; Rabault et al. 2017; Lee et al. 2017).

The use of ML has spread quickly in optical-flow problems due to their potential to provide real-time inference and reconstruct flow structures with pixel-level resolution (Hui et al. 2018d; Teed and Deng 2020). In particular, convolutional neural networks (CNNs) have shown exceptional accuracy and performance with respect to classical techniques in optical-flow applications (Dosovitskiy et al. 2015; Hui et al. 2018d; Teed and Deng 2020). However, the first attempts to transfer ML approaches to PIV processing were not successful when compared to conventional methods (Ilg et al. 2017; Chen et al. 1998; Rabault et al. 2017; Lee et al. 2017), especially due to lack of accuracy and the high computational cost.

Despite the advancement introduced by recent CNN architectures specifically designed for PIV [such as `FlowNetS` (Cai et al. 2019), `LiteFlowNet` (Cai et al. 2020), `LightPIVNet` (Yu et al. 2021; Lagemann et al. 2021), and `CC-FCN`, Gao et al. (2021)], there is not yet general consensus on the capability of CNNs to reach

✉ Claudio Mucignat
claudio.mucignat@empa.ch

Lento Manickathan
lento.manickathan@empa.ch

Ivan Lunati
ivan.lunati@empa.ch

[1] Laboratory for Multiscale Studies in Building Physics, Swiss Federal Laboratories for Materials Science and Technology, Überlandstrasse 129, 8600 Dübendorf, ZH, Switzerland

consistently higher accuracy levels than WIDIM, such that they can be routinely used as an alternative to classical methods in PIV experiments. Moreover, the reconstruction error of CNNs in case of experimental data has never been quantitatively estimated, e.g., using *a-posteriori* methods (Sciacchitano et al. 2013).

A well-designed architecture is not sufficient to guarantee a satisfactory performance of the CNN, which depends also on the data set used to train the network. Training data must be abundant and informative in order to determine the network parameter and reduce the risk of overfitting. In general, CNNs for fluid flow velocimetry have been trained on datasets consisting of synthetic image pairs that are generated from displacement fields derived from numerical solutions of the Navier–Stokes equations [e.g., from the Johns Hopkins Turbulence Database (JHTDB) (Li et al. 2008)].

The downside of this choice is that numerical simulations of flow problems are computationally demanding, and such training datasets typically remain limited in size. A strategy to overcome this issue is to generate synthetic image pairs from random displacement fields, which inform the network about the kinematic relationship between two successive positions of the particles and the corresponding displacement (or velocity). We have proven that kinematic training datasets are effective in training a state-of-the art CNN [namely pyramid warp cost-volume iterative residual refinement, PWCIRR (Hur and Roth 2019)] to reconstruct displacement fields from PIV recordings (Manickathan et al. 2022). As the generation of the displacement fields is based on an inexpensive algorithm, it is possible to obtain large datasets that enable the CNN to reconstruct the displacement field with an accuracy comparable or higher than that of WIDIM.

Despite this remarkable achievement, the size of the networks, which entail a large number of degrees of freedom, remains a limiting factor to a widespread use of CNNs as a robust and efficient replacement of conventional image processing for PIV applications. Indeed, PWCIRR has 3.6 million of trainable parameters, whereas other popular networks have up to more than hundred million of trainable parameters [RAFT (Teed and Deng 2020) has 5.3 million, Lite-FlowNet (Hui et al. 2018d) 5.37 million, FlowNetS (Dosovitskiy et al. 2015) 38 million, and FlowNet2 (Ilg et al. 2017) 162.5 million]. The large number of parameters and the complexity of the network require very large training data and large GPU memory to process high-resolution images, increasing the cost and time of the training process. To establish CNN as the standard for PIV, it is crucial to devise novel architectures that are not only highly accurate but also lean enough to handle of large images with real-time performance, hence allowing researchers to decrease turnover time and maximize the utilization of experimental facilities.

In this paper, we introduce the lightweight image matching architecture (LIMA), a novel convolutional neural network specifically optimized for PIV that enables higher accuracy than WIDIM while offering a considerable reduction of the post-processing time compared to, for example, PWCIRR or WIDIM. In Sect. 2, we give an overview of the architecture design, discuss the improvement of the loss function, and describe the training strategy.

Then, we estimate the inference time (Sect. 3.1) and assess the accuracy for a synthetic test case (Sect. 3.2), as well as for two experimental datasets consisting of flow past a cylinder and a bluff body (3.3 and 3.4, respectively). For all cases, we compare the performance of LIMA with those of PWCIRR and a commercial GPU implementation of WIDIM (Davis 10, LaVision). To assess the reconstruction error, we compare with the ground truth for the synthetic test cases, whereas for the experimental test cases we employ an *a-posteriori* uncertainty quantification (UQ) by image matching (Sciacchitano et al. 2013). This work is part of a wider effort to devise ML diagnostic tools for experimental fluid dynamics, which includes also the application of LIMA in background oriented schlieren (BOS) (Mucignat et al. 2023).

2 Convolutional neural networks for PIV

CNNs for fluid flow velocimetry estimate the displacement field d$\boldsymbol{s}$ (px) from the image intensity pair $\mathbf{I}$ (counts),

$$\mathrm{d}\boldsymbol{s} = \mathcal{N}(\mathbf{I}), \tag{1}$$

where $\mathcal{N}$ represents the network operation on $\mathbf{I} = (I_1, I_2)^{\mathrm{T}}$, which consists of particle image intensities I_1 and I_2 at time t_1 (s) and $t_2 = t_1 + \Delta t$ (s), respectively. The corresponding fluid velocity $\boldsymbol{u}$ (m s^{-1}) can be readily calculated as $\boldsymbol{u} = \mathrm{d}\boldsymbol{s}/(\Delta t \cdot M)$, where M is the magnification factor (px m^{-1}) and Δt the separation time (s) Raffael et al. (2018).

2.1 The lightweight image matching architecture (LIMA)

To reduce the number of parameters and the computational resources required by the existing CNNs for optical-flow estimation, we design a lightweight architecture that can potentially be embedded on GPU devices and achieve real-time inference with minimal loss in accuracy. The

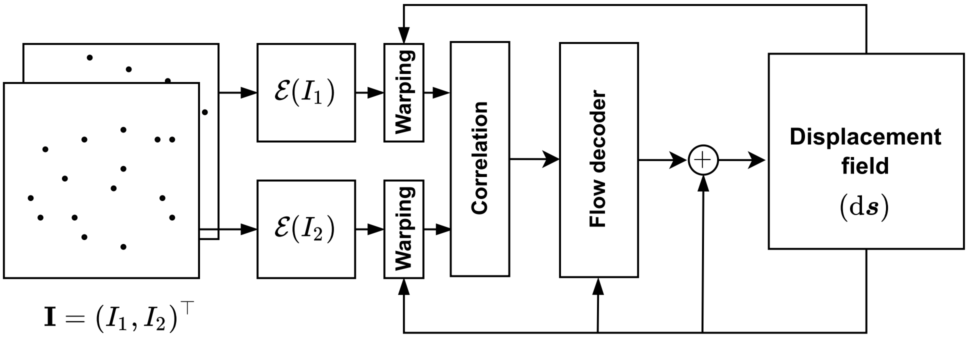

Fig. 1 LIMA: Lightweight image matching architecture using symmetric warping for second-order accurate velocity approximation

lightweight image matching architecture, LIMA, is sketched in Fig. 1. The network infers the displacement by means of an iterative residual refinement strategy (Hur and Roth 2019), which employs L refinement steps to iteratively correct and upsample the displacement. First, the encoder subnetwork, $\mathcal{E}$, constructs an L-level feature map for each image pair,

$$\mathbf{F}_1 = \mathcal{E}(I_1), \tag{2}$$

$$\mathbf{F}_2 = \mathcal{E}(I_2), \tag{3}$$

where $\mathbf{F}_1 = (F_1^1, F_1^2, ..., F_1^L)^\top$ and $\mathbf{F}_2 = (F_2^1, F_2^2, ..., F_2^L)^\top$ are the multi-level pyramid feature maps of the image I_1 and I_2, respectively. Note that the encoding operator applies a nonlinear binning with a kernel size that is halved at each level. At each refinement step l, the encoded feature from the level l is used to infer the displacement with a level-specific image resolution as it follows. First, the feature map of image i at level l is warped using the displacement estimation from level $l-1$,

$$\tilde{F}_i^l = \mathcal{W}\left(F_i^l, \hat{\mathrm{d}\boldsymbol{s}}^{l-1}\right), \tag{4}$$

where $\hat{\mathrm{d}\boldsymbol{s}}^{l-1}$ is obtained by 2× upsampling of $\mathrm{d}\boldsymbol{s}^{l-1}$ to match the resolution of F_i^l. Note that $\mathcal{W}$ is a non-trainable operator that simply warps the feature map according to the displacement map using bilinear interpolation. As the warping is symmetric for I_1 and I_2, (i.e., toward $t_1 + \Delta t/2$ and $t_2 - \Delta t/2$, respectively), we obtain a central difference image matching that is second-order accurate in time (Wereley and Meinhart 2001).

Successively, using both the warped feature map of I_1 and I_2, a cost volume is constructed to estimate the degree of matching between the two images,

$$C^l = \mathcal{C}\left(\hat{F}_1^l, \hat{F}_2^l\right), \tag{5}$$

where C^l is the cost volume of level l, and $\mathcal{C}$ is the cost volume subnetwork. Finally, with the cost volume of level l and the previous (upsampled) estimate of the displacement from level $l-1$, we can correct and refine the displacement estimate for level l,

$$\mathrm{d}\boldsymbol{s}^l = \hat{\mathrm{d}\boldsymbol{s}}^{l-1} + \mathcal{D}\left(C^l, \hat{\mathrm{d}\boldsymbol{s}}^{l-1}\right), \tag{6}$$

where $\mathcal{D}$ is a lightweight decoder subnetwork.

Several analogies between LIMA and WIDIM can be drawn. First of all, both methods employ an iterative strategy to correct and upsample the displacement. They both comprise a warping step, based on a the displacement field calculated at iteration $i-1$, to increase the accuracy of the reconstructed displacement. In the CNN, the encoding step is analogous to the fast-Fourier transform employed in most of the implementations of WIDIM, whereas the cost volume operator has a function similar to the cross-correlation analysis. The remarkable difference between the methods is that in the CNN all operators are nonlinear, which is an advantage to resolve complex patterns. Also, the flow decoder directly provides the displacement field without requiring additional vector validation or denoising, as it is the case for WIDIM (Raffael et al. 2018; Wieneke 2017).

2.2 LIMA-4 and LIMA-6

In this study, we consider two versions of LIMA that are characterized by a different number of levels: LIMA-6 and LIMA-4. LIMA-6 is a six-level version of the network (i.e., $L = 6$) that provides highest accuracy and reconstructs the displacement and velocity fields with the same resolution as the original images. LIMA-4 has two refinement steps less than LIMA-6 (hence four levels, i.e., $L = 4$) and reconstructs the displacement or the velocity fields with a resolution of one fourth of the original images (i.e., it provides a velocity vector every 4 × 4 pixels); it is designed to achieve high-throughput inference performance with minor loss in accuracy. `LIMA-4` and `LIMA-6` differ in the number of iteration levels but have the same number of trainable parameters due to the weight-sharing design of the iterative residual refinement architecture (Hur and Roth 2019). We remark that the number of trainable parameters is considerably lower than in other CNNs that have been previously introduced. Indeed, `LIMA` has 1.6 million parameters, while `PWCIRR`, for example, has 3.6 million parameters (Hur and Roth 2019). All networks are implemented using the machine learning library *PyTorch* (Paszke et al. 2019).

2.3 Multi-level loss function with penalization

The choice of the loss function (or objective function) to be used in the training optimization problem is an important component of the CNN, which influences the learning rate and may affect the output of the network, in particular when the amount of training data is limited and may potentially lead to overfitting. In general, CNNs for classic optical-flow applications employ loss functions that only contain terms that quantify the mismatch between the estimate and the ground truth. For fluid-dynamic applications, however, the fields to be reconstructed typically exhibit regularity properties that make sharp contrasts unlikely.

In order to incorporate this information into the CNN, we define a multi-level loss function that favors smoother fields by penalizing reconstructed fields with larger values of the Jacobian (hence, larger gradients),

$$\mathcal{L} = \sum_{l=1}^{L} \lambda^{l} \left(\lambda_u |\mathrm{d}\boldsymbol{s} - \mathrm{d}\boldsymbol{s}_{\mathrm{ref}}| + \lambda_J |\mathbf{J}| \right), \tag{7}$$

where $\mathrm{d}\boldsymbol{s}_{\mathrm{ref}}$ is the reference (or ground truth) displacement; $\mathrm{d}\boldsymbol{s}$ the estimated displacement; $\mathbf{J}$ the Jacobian of the estimated displacement; λ_u and λ_J are the trade-off weights of displacement and Jacobian, respectively, and are the same at all iteration levels; and λ^l is the loss weight applied to control the influence of the l-level on the final estimate (we weight higher iteration levels, with higher resolution, more than low iteration levels).

2.4 Kinematic training strategy

Typically, CNNs are trained with dataset consisting of analytical solutions and results of numerical simulations, which require to solve the Navier–Stokes equations (see, e.g., Cai et al. 2019, 2020). Due to the computationally demanding numerical simulations, the cost to obtain large training datasets, which are necessary to cover a wide set of flow configurations, may become prohibitive and, in practice, only a limited range of scales, classes of problems, and flow regimes can be represented in the training phase.

Here, we follow a different strategy that relies on the principle of kinematic training, which we have previously proposed for `PWCIRR` and demonstrated to be able to increase the robustness of the CNN with respect to image noise, particle loss between two successive frames, and particle image diameter (Manickathan et al. 2022).

The underlying idea is that, for velocimetry applications, the network must simply learn the kinematic relationship between the particle position at two successive times (separated by the time interval Δt) and the particle velocity (displacement). At typical seeding densities and separation times, the network can be trained with synthetic image pairs obtained from random displacement fields that are generated at negligible computational costs. This allows us to generate large training dataset, reducing the risk of overfitting and achieving an accuracy comparable to or higher than the state-of-the-art method `WIDIM` (Manickathan et al. 2022). To improve the generalization of the network, the training dataset is further augmented by means of random translation, scaling, rotation, reflection, brightness change, and additive Gaussian noise, which mimic the image background noise that is observed during real-world setup (we use the same data augmentation settings as in Cai et al. (2019, 2020), Hui et al. (2018d) and Ilg et al. (2017). Once the random displacement fields have been generated, the synthetic PIV images are obtained by means of a Synthetic Image Generator (SIG) that is similar to the ones proposed by Lecordier

Table 1 Particle properties and parameters of the random displacement fields used to generate the synthetic PIV images for the kinematic training dataset

Parameter	Values
Image size (px)	256 × 256
Number of examples	18,278
Filter size, σ (px)	5–100
Maximum displacement, max($\mathrm{d}\boldsymbol{s}_{\mathrm{ref}}$) (px)	0–16
Particle diameter, D (px)	1–4
Particles density, ρ_D (dpp)	0.05–0.40
Particle loss, (%)	0–2

Table 2 Summary of training parameters for `LIMA-4`, `LIMA-6`, and `PWCIRR` (Manickathan et al. 2022)

Parameter	`LIMA-4`	`LIMA-6`	`PWCIRR`
Network size	1.6M	1.6M	3.6M
Levels (L)	4	6	7
Training examples	18,278	18,278	18,278
Total epochs	200	200	200
Batch size	4	4	4
Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}
Optimizer	Adam	Adam	Adam
β_1	0.9	0.9	0.9
β_2	0.999	0.999	0.999
Scheduler	ReduceLRonPlateau	ReduceLRonPlateau	Scheduled decay
Decay rate	1/5	1/5	1/2
Loss function	l_1-loss	l_1-loss	l_2-loss
λ_u	0.91	0.91	–
λ_J	0.09	0.09	–
Loss weights λ^l	[0.01, 0.02, 0.08, 0.32]	[0.0025, 0.005, 0.01, 0.02, 0.08, 0.32]	[0.32, 0.08, 0.02, 0.01, 0.005, 0.0025, 0.00125]
Training time (h)	24 (1-node)	104 (1-node)	123 (8-node) (≈ 984 on 1-node)

and Westerweel (2004) and Armellini et al. (2012). The parameters used to generate the PIV images are listed in Table 1. We refer to Manickathan et al. (2022) for further details about the generation of the kinematic training dataset.

3 Results and discussion

To assess the performance of `LIMA`, we consider four test cases (consisting of two synthetic and two experimental datasets) and we compare the results of `LIMA-4` and `LIMA-6` with `PWCIRR` [which proved capable of reconstructing the velocity fields with low errors, Manickathan et al. (2022)] and with a state-of-the-art PIV cross-correlation method, i.e., a commercial GPU implementation of `WIDIM` (Davis10, LaVision) which includes also a final anisotropic denoising step (Wieneke 2017). All CNNs are trained on same kinematic training datasets and with the parameters reported in Table 2. The weights of the multi-level loss function, including the trade-off weights of the displacement and the Jacobian (λ^l, with $l = 1, 2, ..., L$, λ_u, and λ_J), has been chosen as they showed promising results in a preliminary sensitivity analysis.

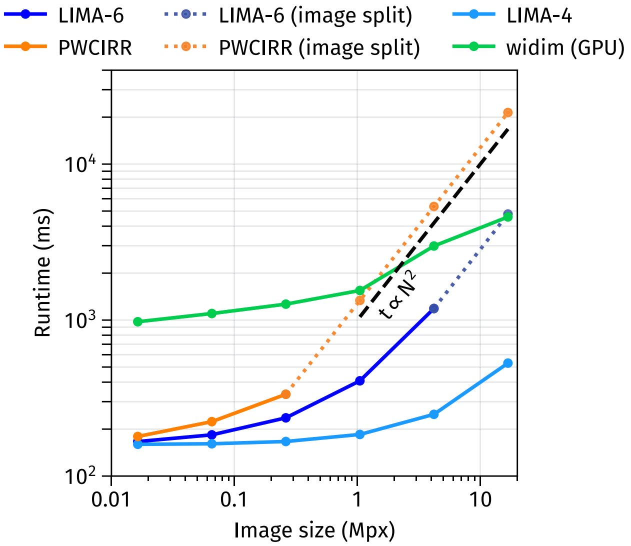

Fig. 2 Inference time (ms) as a function of the image size (Mpx) for two versions of `LIMA` (`LIMA-4` and `LIMA-6`, both with 1.6M parameters), `PWCIRR` (3.6M parameters), and `WIDIM` (*cc*). All methods are implemented on the same GPU hardware. The plot shows the mean runtime averaged over 100 runs. `WIDIM` uses a final interrogation window of size 24 × 24 px at 50% overlap

3.1 Inference-time benchmark

To demonstrate the advantages of the lightweight architecture in terms of computational cost, we first perform an inference-time benchmark employing the same GPU hardware (Nvidia RTX 2080 Ti) for all methods. We consider different image sizes, ranging from 128 × 128 to 4096 × 4096 pixels, increasing the size of the images by a factor 2 × 2 at each steps. For each size, we generate 100 image pairs and report the average inference time in Fig. 2 as a function of the image size expressed in megapixels (Mpx).

Notice that the maximum image size that can be processed by the CNN depends on the number of network parameters and on the available GPU memory. If this upper limit is exceeded, the image has to be split into sub-images, which are processed separately and stitched back together. The additional image manipulation and memory transfer that are required increase the computational cost and the processing time. On the GPU hardware used for our benchmark, the maximum image size that can be handled by PWCIRR is only 800 × 800 pixels (0.64 Mpx), while LIMA-6, resp. LIMA-4, can process images up to a resolution of 2048 × 2048 pixels (4 Mpx), resp. 8192 × 8192 pixels (64 Mpx), without the needs of splitting, which results in a significant advantage in terms of inference time. In case of images of 512 × 512 pixels that can be analyzed directly by all methods without splitting, LIMA-6 and LIMA-4 are 1.4 and 2.0 times faster than PWCIRR, respectively. However, for larger image size, i.e. 2048 × 2048 pixels, LIMA-4 and LIMA-6 have inference runtimes of only 240 ms and 1.17 s, respectively, whereas PWCIRR needs up of 5.35 s to process the data. Notice that once image splitting is required, the runtime of PWCIRR and LIMA-6 grows with N^2.

If we compare the computational performance of LIMA with that of WIDIM, we can observe that the former is always faster, but the speed up is different depending on the number of iteration levels. The runtime of the different methods for different image sizes is plotted in Fig. 2. For WIDIM, we include a denoising step and use a final grid size of 24 × 24 pixels and 50% overlap, which are typical choices in the PIV community. We observe that LIMA-4 is roughly 12 times faster than conventional PIV post-processing for 4 Mpx images. When the image size increases, the computational advantage of LIMA-6 reduces considerably. The runtime is roughly 4 times and 2.5 shorter for an image of 1024 × 1024 pixels and 2.5 times shorter for an image of 2048 × 2048 pixels, whereas for an image of 4096 × 4096 pixels LIMA-6 has the same runtime as WIDIM because the memory limit of the GPU is exceeded and the CNN requires image splitting. We observe that decreasing the grid size and increasing the overlap leads to a significant increase in processing time. For example, reducing the grid size to 16 × 16 and increasing the overlap to 75% yield a three times larger runtime.

We remark that the computational graphs of the networks have not been optimized using graph compilation and pruning, typically done during deployment of the network. Therefore, these values can be seen as the lower limit for the speedup that can be provided by LIMA. In particular, depending on the image size, GPU hardware and acquisition frequency, LIMA-4 has the potential to perform real-time processing of the image pairs.

3.2 Error estimate for a synthetic DNS dataset

We first assess the accuracy of the CNNs on a synthetic test case. The image pair is generated from a velocity field which is the solution of two-dimensional Direct Navier–Stokes (DNS) simulations and is available online (Carlier 2005). The synthetic image pair and the distribution of the magnitude of the reference displacement, $|d\boldsymbol{s}_{\text{ref}}|$ (px), are depicted in Fig. 3. We remark that the CNNs have been trained on a kinematic random dataset and have not previously seen images constructed from solutions of the Navier–Stokes equations.

In the panels a-d of Fig. 4, the magnitude of the displacements reconstructed by LIMA-4 and LIMA-6 are compared with those calculated by PWCIRR and WIDIM, which uses an initial interrogation window of size 64 × 64 pixels, whereas the size of the final (refined) window is 8 × 8 pixels with 50% overlap. We perform a vector validation based on a peak-to-peak ratio of 1.3 and apply a normalized median filter (NMF) based on universal outlier detection Westerweel and Scarano (2005) as implemented in Davis (LaVision)

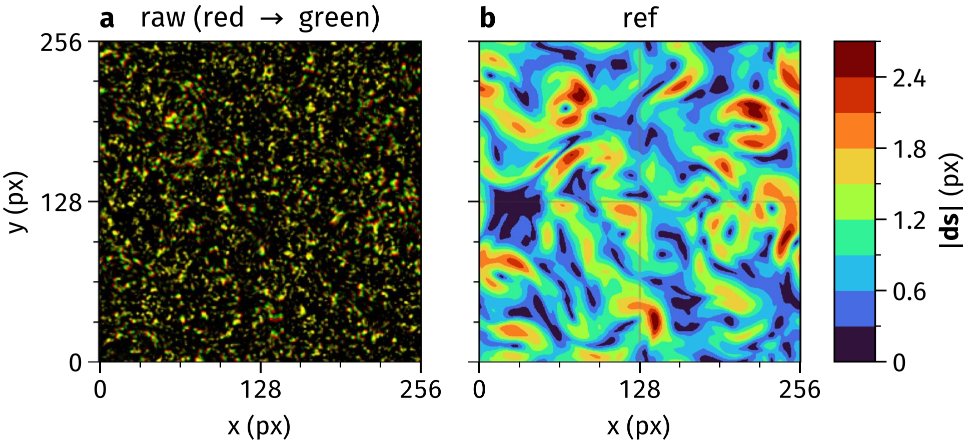

Fig. 3 **a** Representation of the pair of raw synthetic images generated from the DNS solution in Carlier (2005): the intensities of I_1 and I_2 are depicted in red and green, respectively; the regions of intensity overlap may appear in yellow. **b** Magnitude of the reference displacement field, $|d\boldsymbol{s}_{\text{ref}}|$ (px)

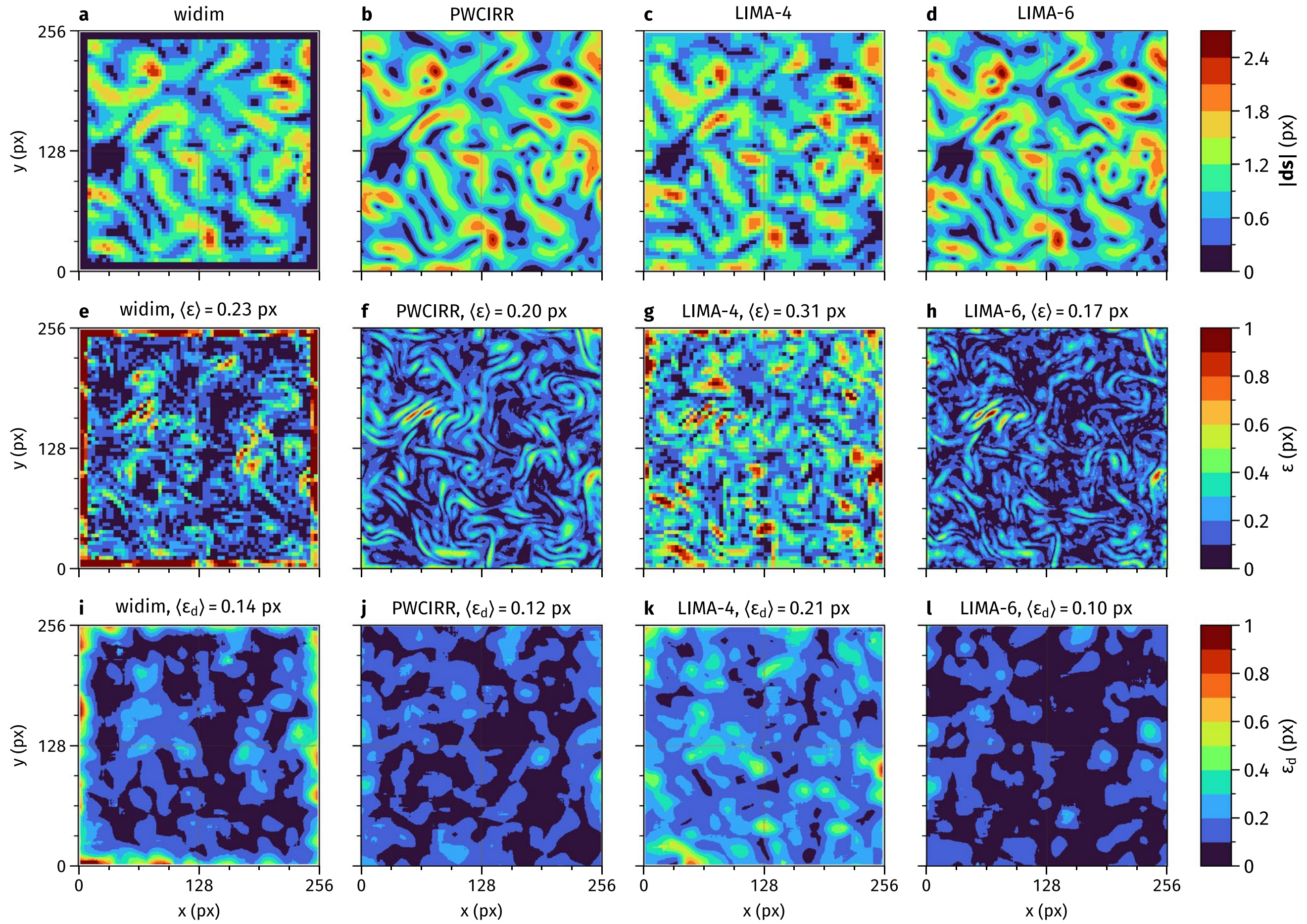

Fig. 4 Reconstructed displacement fields from the DNS dataset and uncertainty quantification of `WIDIM`, `PWCIRR`, `LIMA-4`, and `LIMA-6`. Upper row: magnitude of the reconstructed displacement fields, $|\mathrm{d}\boldsymbol{s}|$ (px); middle row: distribution of the error magnitude with respect to the ground truth, $\epsilon = |\mathrm{d}\boldsymbol{s}_{\mathrm{ref}} - \mathrm{d}\boldsymbol{s}|$ (px); bottom row: distribution of the disparity error ϵ_d (px) calculated accordingly to the *a-posteriori* uncertainty quantification by image matching (Sciacchitano et al. 2013). Also reported in the figures are the spatial average of the error magnitude and disparity error

Table 3 Spatial average of the displacement errors, $\langle\epsilon\rangle$ (px), and disparity error, $\langle\epsilon_d\rangle$ (px). The spatial average is performed excluding displacement vectors at less than 16 pixels from the edges

Method	$\langle\epsilon\rangle$ (px)	$\langle\epsilon_d\rangle$ (px)
`WIDIM`	0.23	0.14
`PWCIRR`	0.20	0.12
`LIMA-4`	0.31	0.21
`LIMA-6`	0.17	0.10

on a kernel of 5 × 5 pixels and a final anisotropic denoising step (Wieneke 2017). All reconstructed displacements show similar patterns, but `WIDIM` and `LIMA-4` estimate the displacement on a coarser grid (one displacement vector each 4 × 4 pixels), whereas `LIMA-6` and `PWCIRR` reconstruct the displacement field with pixel-level resolution.

To assess the accuracy of the reconstructed displacement, we consider two error indicators: the magnitude of the deviation from the reference (px),

$$\epsilon = |\mathrm{d}\boldsymbol{s} - \mathrm{d}\boldsymbol{s}_{\mathrm{ref}}|; \tag{8}$$

and the uncertainty of the reconstructed displacement, which is defined as the *a-posteriori* disparity error, ϵ_d (px), calculated by image matching (Sciacchitano et al. 2013).

The spatial distribution of the two error indicators are compared in the panels e-h and i-l of Fig. 4. Notice that the error magnitude is calculated pixel by pixel from the displacement components and can better detect errors is higher-gradient regions, whereas the disparity error is calculated on windows of size 8 × 8 pixels [as in Sciacchitano et al. (2013)] and has lower resolution. The spatial average of the error indicators, $\langle\epsilon\rangle$ and $\langle\epsilon_d\rangle$, is reported in Table 3 for all methods. To minimize boundary artifacts the spatial average is performed excluding velocity vectors at less than 16 pixels from the edges.

LIMA-6 provides the best accuracy both in terms of average error magnitude, $\langle\epsilon\rangle = 0.17$ px, and average disparity error, $\langle\epsilon_d\rangle = 0.10$ px. Despite the lower number of trainable parameters, LIMA-6 perform even better than PWCIRR (which already has a sensibly smaller disparity error than WIDIM, see Tab. 3). This is due to Jacobian penalization in the multi-level loss function, which leads to errors smaller than PWCIRR in regions characterized by small displacement gradient (compare top and middle rows in Fig. 4). LIMA-4 has the lowest accuracy ($\langle\epsilon\rangle = 0.31$ px and $\langle\epsilon_d\rangle = 0.21$ px), as a consequence of the fewer iteration levels which decrease the computational time and memory requirements as shown in Sect. 3.1. Optimizing the number of levels (or iterations) is crucial to balance computational costs and accuracy, adapting the CNN to the goals of the specific application.

3.3 Wind tunnel study of a cylinder wake

To assess the performance of LIMA in reconstructing displacement and velocity fields from experimental data, we consider a cylinder-wake experiment conducted in the wind tunnel of the *Swiss Federal Laboratories for Materials Science and Technology* (Empa), (Manickathan et al. 2022). The facility has a cross-section of width 1.9 m and height 1.3 m and can operate at a bulk wind speed, u_∞, in the range between 0.5 m s^{-1} and 25 m s^{-1}. The imaging setup consists of a Nd:YLF dual cavity laser with a pulse energy of 30 mJ at 1 kHz and a 4 Mpx high-speed camera that records images of size 2016×2016 pixels. By combining a 200 mm focal lens (with numerical aperture equal to $f/2.8$) with a 2× teleconverter, we obtain a magnification factor $M = 15$ px mm^{-1}. The tunnel airflow is seeded with Di-Ethyl-Hexyl-Sebacat (DEHS) tracer particles of nominal size 1 μm obtained with a lasking seeding generator (Raffael et al. 2018).

The flow past a cylinder of diameter $D = 6$ mm is measured at $u_\infty = 1.07$ m s^{-1}, which corresponds to a Reynolds number Re = 431. A standard PIV analysis is performed with WIDIM applying background subtraction, an initial interrogation window of 64×64 pixels, and a refinement step of 16×16 pixels with 75% overlap. First, a vector validation is performed based on peak-to-peak ratio $Q < 1.25$ and a NMF with 3×3 kernel based on universal outlier detection Westerweel and Scarano (2005) as implemented in Davis; then, a PIV anisotropic denoising step (Wieneke 2017) is applied to the data.

A two-dimensional (2D) section of the magnitude of the displacement, $|\mathrm{d}\boldsymbol{s}|$, reconstructed by WIDIM, PWCIRR, LIMA-4, and LIMA-6, is shown in Fig. 5 (left column; i.e., panels a, c, e, and g respectively). All methods provide similar estimates of the displacement field away from the cylinder, but differences are noticeable in the near-wake region, i.e., $x = [50, 350]$ px and $y = [-100, 100]$ px. In particular, the vector field reconstructed by WIDIM clearly exhibits artifacts in the vicinity of the cylinder where unphysically sharp displacement contrasts can be observed; this is possibly due to a low signal-to-noise ratio induced by laser reflections. In contrast, LIMA-4, LIMA-6, and PWCIRR are able to reconstruct the flow field close to the cylinder, depicting distinct physical flow features and realistic boundary layer separation from the cylinder.

As no ground truth is available for the experimental test cases, we estimate the reconstruction error by plotting the 2D distributions of the disparity error, ϵ_d (px), of WIDIM, PWCIRR, LIMA-4 and LIMA-6 in Fig. 5 (right column; i.e., panels b, d, f, and h, respectively). Note that this a-posteriori error estimate is essentially a self-consistency test that calculates the errors from the mismatch between the position of the tracer particle projected forward and backward in time. It is assumed that the processing algorithm has converged to a value that is sufficiently close to the true value [see Wieneke (2015)]. If the difference between the reconstructed and the true displacements is large, the uncertainty calculated by the disparity error may underestimate the real error. This is the case for the displacement reconstructed by WIDIM in the vicinity of the cylinder. There, the low velocity region behind the cylinder (correctly identified by the CNNs) is not captured by WIDIM, which reconstructs displacement vectors larger than 20 px and fails to converge at several locations, producing artifacts (Fig. 5(a)). As the disparity error cannot provide an accurate estimate of the error close to the cylinder, we limit our analysis to the region that is not affected by artifacts. There, we observe that the disparity patters of the different methods are rather similar. Nonetheless, LIMA-4 exhibits the largest error, and also PWCIRR is less accurate than WIDIM and LIMA-6, which in turn has the lowest disparity error. This is confirmed by comparing the spatial average of the disparity in the wake region (bounded by the dotted-black-line box in Fig. 5.b, d, f, and h): WIDIM and LIMA-6 achieve the highest accuracy with $\langle\epsilon_d\rangle = 0.35$ px and $\langle\epsilon_d\rangle = 0.36$ px, respectively, whereas we have $\langle\epsilon_d\rangle = 0.4$ px for PWCIRR and $\langle\epsilon_d\rangle = 0.44$ px for LIMA-4. Similarly, LIMA-6 has the lowest standard deviation of the error, with $\sigma_d = 0.17$ px, whereas we have $\sigma_d = 0.20$ px for WIDIM and $\sigma_d = 0.21$ px for PWCIRR and LIMA-4, thus indicating a higher precision of the reconstruction.

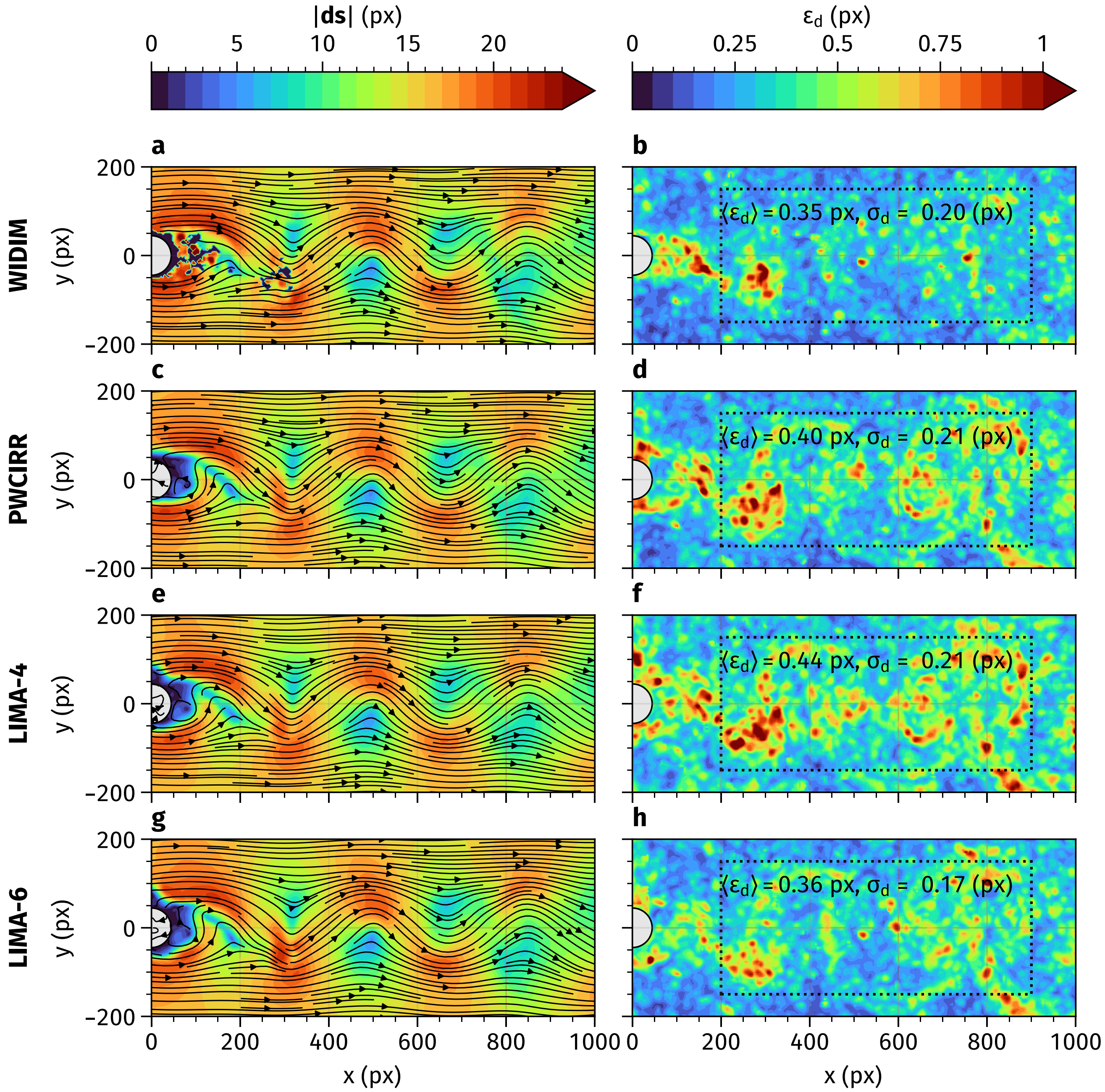

Fig. 5 Cylinder-wake experiment. Left: Two-dimensional distribution of magnitude of the displacement, $|d\mathbf{s}|$ (px), reconstructed by the different methods; right: map of the disparity error ϵ_d, (right column) calculated *a-posteriori* by image matching (Sciacchitano et al. 2013). From top to bottom: `WIDIM` with refinement step of 16×16 pixels and 75% overlap (**a**, **b**); `PWCIRR` (**c**, **d**); `LIMA-4` with $L = 4$ iteration levels (**b**, **f**); and `LIMA-6` with $L = 6$ iteration levels (**g**, **h**)

3.4 Wind tunnel study of a cubic bluff body

Finally, we investigate the performance of `LIMA` in case of a fully three-dimensional (3D) flow configuration in a tunnel experiment. The objective is to assess the robustness of the CNNs when the out of plane velocity component is important, and lead to significant particle loss between consecutive frames. In this case, `WIDIM` may suffer from the deterioration of the correlation coefficient and an increase in the uncertainty is expected (Raffael et al. 2018). At this

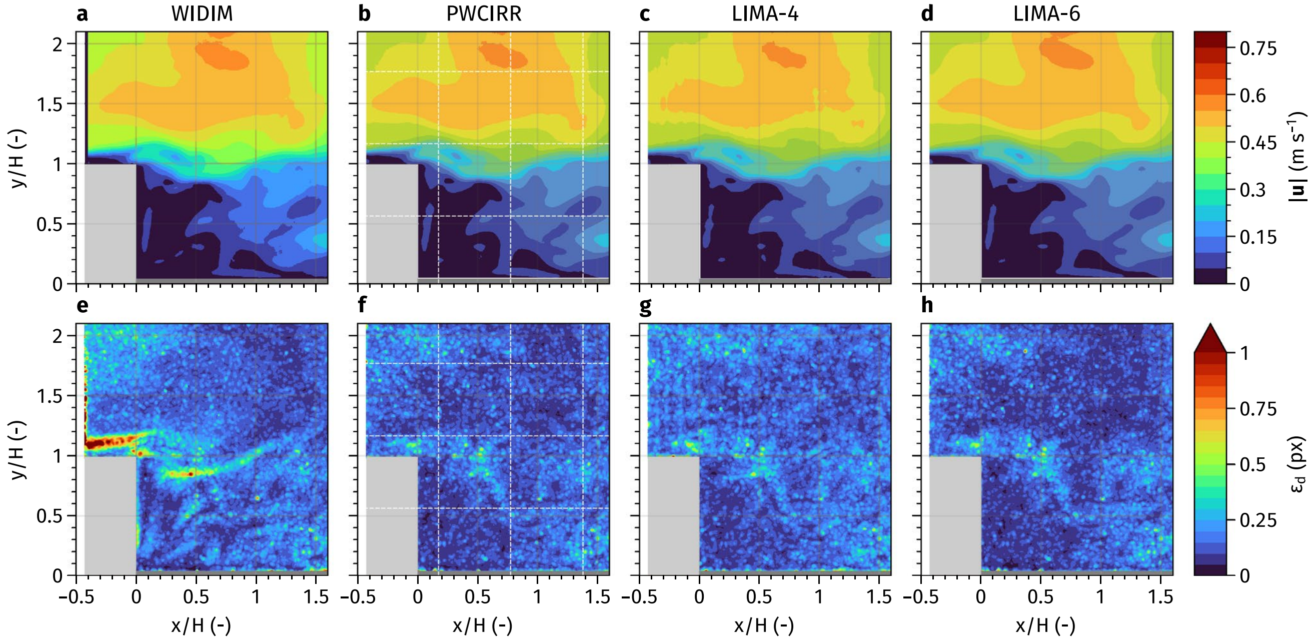

Fig. 6 Cubic-bluff-body experiment. Top: section of the distribution of the magnitude of the instantaneous velocity for a seeding density of 0.04 ppp and a separation time of $2ms$; from left to right: `WIDIM` (**a**), `PWCIRR` (**b**), `LIMA-4` (**c**), and `LIMA-6`(**d**). Bottom: distribution of the corresponding disparity error; from left to right: `WIDIM` (**e**), `PWCIRR` (**f**), `LIMA-4` (**g**), and `LIMA-6`(**h**)

end, we consider the flow over and past a bluff body in a turbulent boundary layer with the same measurement setup and hardware as described in Sect. 3.4.

The bluff body has cubical shape with a side of 50 mm, while the bulk wind speed u_∞ is 0.5 m s^{-1}. The setup corresponds to a camera magnification factor of 17 px mm^{-1}, and the images are acquired at a frequency of 1 kHz. We conduct four different recordings, increasing the seeding density from 0.03 to 0.04, 0.05 and 0.07 ppp, to evaluate the effect on the reconstructed flow fields. In this test case, `WIDIM` employs an initial interrogation window of 128 × 128 pixels, which is refined until the final grid size is 16 × 16 pixels and 75% overlap. We apply the same validation and anisotropic denoising as in Sect. 3.3.

Figure 6 shows a 2D sections of the distribution of the velocity magnitude reconstructed by the different methods from the same PIV image pair, together with the corresponding disparity maps. Overall, the velocity fields are similar and all methods are capable of consistently reconstructing the main flow features, i.e., the turbulent boundary layer approaching the obstacle, the shear region above, and the 3D flow separation beyond. A comparison of the disparity error maps shows that the velocity reconstructed by `WIDIM` exhibits the largest error among the four methods and the disparity has a non-random spatial pattern with the highest error located in the shear and in the wake regions. In these region, the limit imposed on the resolution by the size of the cross-correlation window does not allow `WIDIM` to reconstruct the small flow features that characterize high velocity gradients. On the contrary, the higher resolution of the CNNs allows a substantial reduction of the error, with `PWCIRR` and `LIMA-6` performing equally well, whereas `LIMA-4` has sensibly higher error values, but still lower than `WIDIM`.

To assess the effects of the separation time between the images of the pair, we skip an increasing number of frames from the time-resolved recording. As the flow is 3D, this increases the loss of particles between the two images of the pair, allowing us to assess the robustness of each method against this parameter. Figure 7 shows the 2D distribution of ϵ_d for the velocity fields reconstructed with a nominal seeding density of 0.4 ppp and a frame separation increasing from 1 to 5 ms (for sake of completeness, the reconstructed velocity fields are depicted in Appendix, Fig. 10). As the separation time increases, the magnitude of the true displacement field becomes much larger than the maximum displacement used to generate the training dataset (i.e., 16 px, Table 2). We remark that such large displacements are

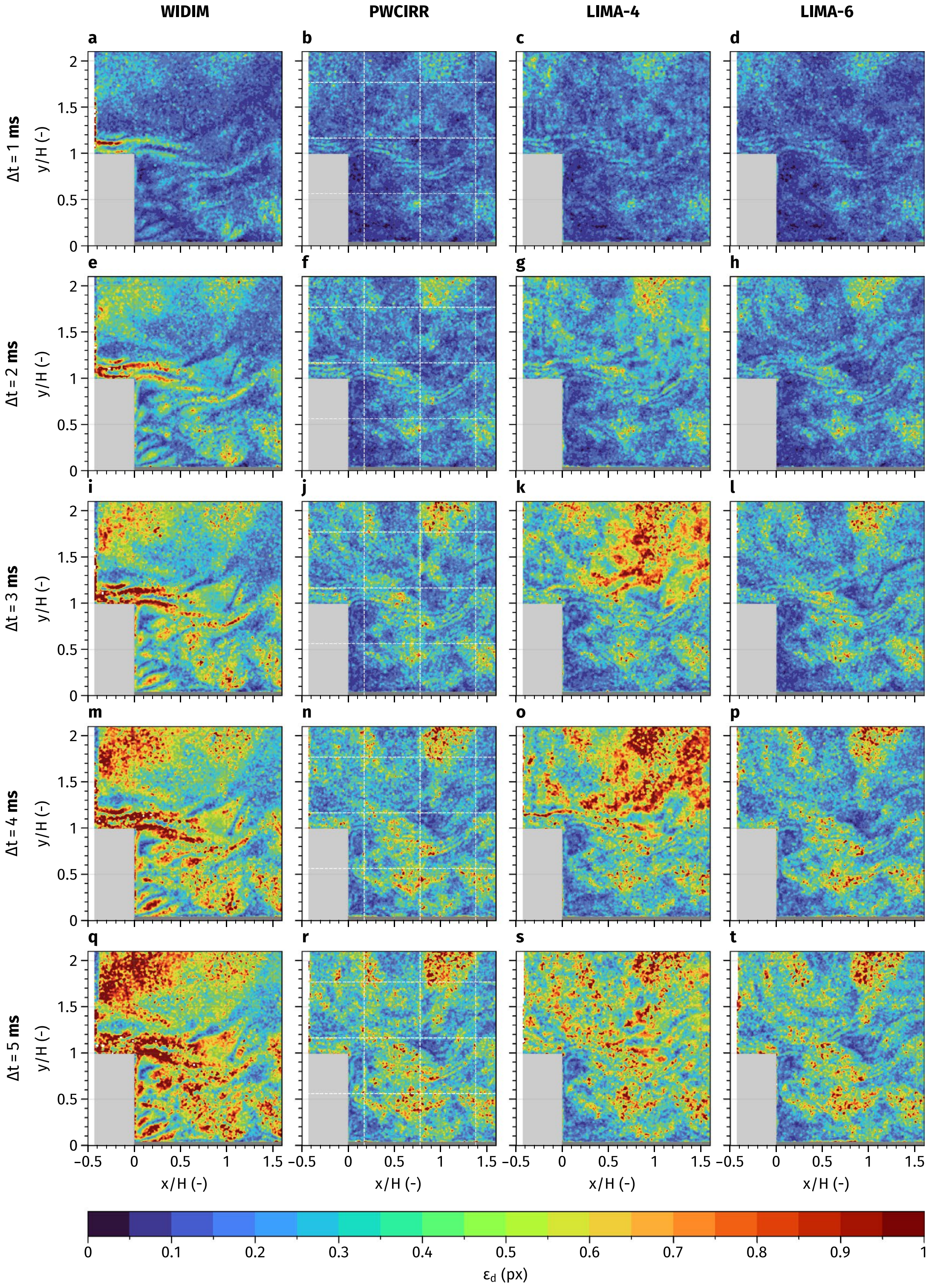

Fig. 7 Cubic-bluff-body experiment. 2D distribution of the instantaneous disparity error (px) for different methods (from left to right: WIDIM, PWCIRR, LIMA-4, and LIMA-6) and increasing separation times (from top to bottom: 1 ms, 2 ms, 3 ms, 4 ms, and 5 ms). Data are obtained with a nominal seeding density of 0.04 ppp

rather unconventional in practical applications and are commonly avoided. Here, however, we are interested in performing a sensitivity study on the reconstruction accuracy, rather than optimizing all the experimental parameters as it would be done in practice. Figure 10 shows that both `PWCIRR` and `LIMA` are still able to consistently reconstruct the fluid motion, whereas `WIDIM` suffers from an increasing number of invalid displacement vectors (the white areas in Fig. 10).

Increasing the separation time leads to a loss of the accuracy for all methods, but the deterioration is much less pronounced for `LIMA-6` and `PWCIRR` than for `LIMA-4` and `WIDIM`. Differently from Sects. 3.2 and 3.3; however, `LIMA-4` shows comparable or even lower error levels than `WIDIM`, suggesting that even the four-level network is more robust than classical methods with respect to increased separation time and particle loss. Overall, `LIMA-6` and `PWCIRR` are capable of providing reconstructed fields with the lowest disparity errors, with the former offering a big benefit in terms of computational time (Sect. 3.1) at minimal or no accuracy loss.

We also evaluate the effects of increasing the seeding density from 0.03 ppp, to 0.04 ppp, 0.05 ppp, and 0.07 ppp. The disparity error distribution is obtained by averaging over 300 samples, which allows us to reduce the fluctuations present in the error maps of a single experimental recording, which may be affected by instantaneous flow features (e.g., high out of plane motion) and lead to a biased assessment. Figure 8 shows the 2D distribution of the average disparity (The corresponding reconstructed velocity fields are depicted in Appendix, Fig. 11). For all methods, the disparity error decreases when the seeding density increases. Nevertheless, `WIDIM` shows a persistent high-error region in correspondence of the shear layer above the obstacle, and a low-error region centered at $x/H = 1.5$, $y/H = 1.75$. In contrast, `PWCIRR` and `LIMA-6` exhibit a uniform error distribution and the lowest error levels. `LIMA-4` performs in between conventional image cross-correlation and high-resolution CNNs. Remarkably, `LIMA-4` has can achieve much lower error levels than `WIDIM` if the seeding density is sufficiently increased.

Finally, we evaluate the reconstructed displacement fields near the wall of the cubic bluff body by plotting the vertical profiles of the two in-plane components of the displacement (Fig. 9). The profiles area is extracted at $X/H = 1$ from the 2D maps obtained by processing a single image pair with separation time of 1 ms and seeding density of 0.04 ppp (see the maps in Fig. 10a–d). Note that the profiles are shown only a few pixels away from the wall (a region which is affected by noise resulting from laser reflection) and that, given the magnification factor and the fact that the boundary layer is fully separated past the bluff body, the reconstructed displacements are very small (below 1 px). The profiles reconstructed by `PWCIRR` and `LIMA-6` are in good agreement down to a distance from the wall of 7–8 pixels, while closer to the wall the displacement field reconstructed by `LIMA-6` exhibits some oscillations which are more evident in the vertical component (Fig. 9b). The behavior of `LIMA-4` is similar, but the oscillations are larger and appear at a larger distance from the wall. These artifacts could be reduced by increasing the penalization of larger gradients in the loss function (see Eq. 7).

4 Conclusions

The CNNs employed for PIV may contain from a few million (e.g., `PWCIRR`) up to more than hundred million of degrees of freedom (e.g., `FlowNet2`). This large number of trainable parameters increases the computational costs (both of training and post-processing) and increases the risk of overfitting, if the training dataset is not sufficiently large and informative. To overcome these difficulties, we have implemented three different strategies. First, we have devised a novel lightweight image matching architecture (`LIMA`) that, unlike to the standard optical-flow networks, employs symmetric image matching to obtain a second-order accurate estimate of displacement or velocity fields. Second, we have introduced a multi-level loss function that penalizes reconstructed fields with larger values of the Jacobian; favoring smoother fields, this reduces the risk of overfitting, particularly in regions characterized by small gradients. Third, we have trained the CNNs with the kinematic training strategy, which allows us to generate large synthetic datasets at low computational cost, while guaranteeing state-of-the-art accuracy (Manickathan et al. 2022).

`LIMA` employs an iterative residual refinement strategy, which allows us to optimize the total number of refinement steps to balance accuracy and computational efficiency. Here, we consider six- (`LIMA-6`) and four-level (`LIMA-4`) networks that are considerably leaner and faster than a state-of-the-art network designed for optical flow (`PWCIRR`). The lower number of degrees of freedom of

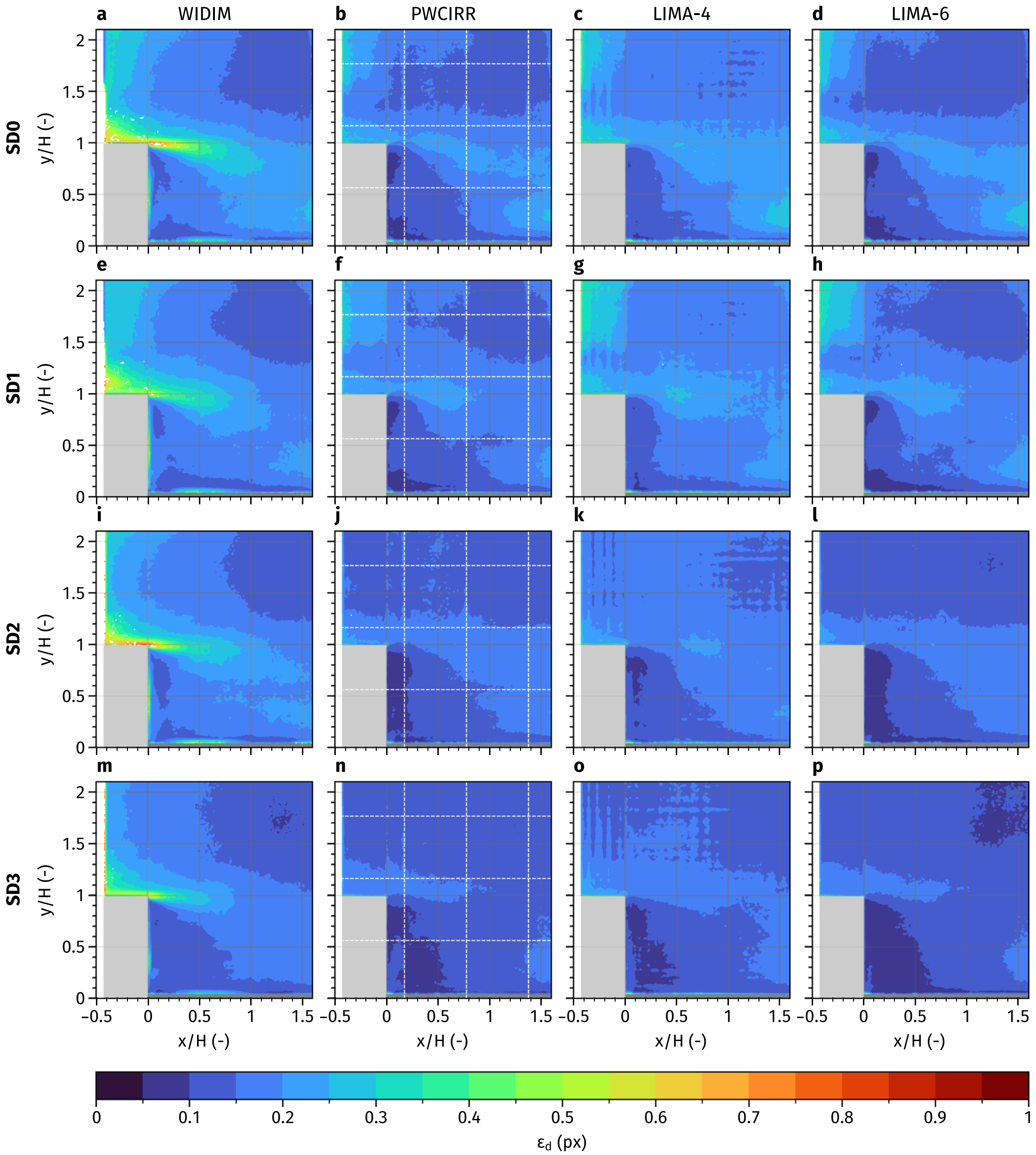

Fig. 8 Cubic-bluff-body experiment. Average over 300 samples of the 2D distribution of the disparity error (px) for the different methods (from left to right: WIDIM, PWCIRR, LIMA-4, and LIMA-6) and different seeding densities (SD) (from top to bottom: SD0, 0.03 ppp; SD1, 0.04 ppp; SD2, 0.05 ppp; and SD3 0.07 ppp. The separation time is 1ms

the lightweight architecture also reduces memory requirement, allowing us to handle larger images without resorting to splitting into sub-images, hence further reducing the runtime.

Despite the fewer degrees of freedom, LIMA-6 reconstructs the displacement or velocity fields with an accuracy comparable or superior to PWCIRR. This is demonstrated in both synthetic and experimental test cases (The accuracy is evaluated by comparing the deviation from the reference or the disparity error by image matching, when no ground truth is available). The excellent performance achieved by LIMA-6 with fewer trainable parameters can be attributed

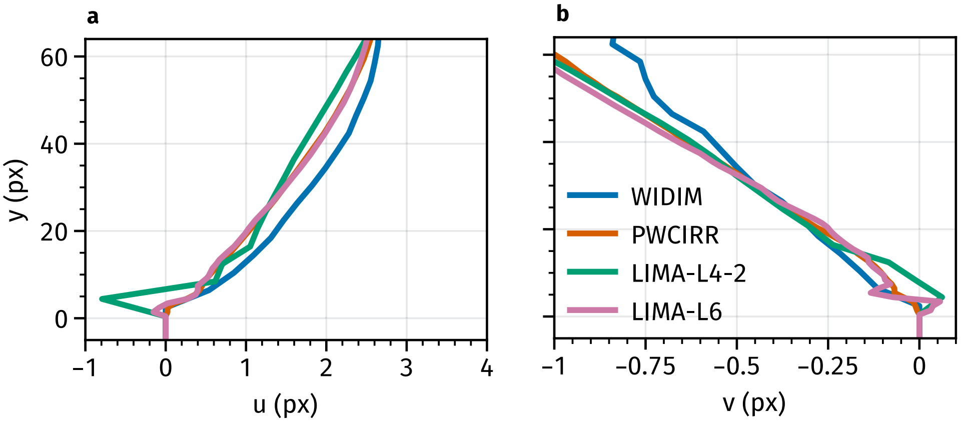

Fig. 9 Cubic-bluff-body experiment. Profiles of the horizontal (**a**) and vertical (**b**) components of the displacement near the wall for WIDIM, PWCIRR, LIMA-6, and LIMA-4. The profile is extracted at $X/H = 1$ from the 2D displacement maps obtained by processing a single image pair with separation time is 1 ms and seeding density of 0.04 ppp

to the use of a multi-level loss function that penalizes the Jacobian, preventing the network from predicting large fluctuations in low gradient regions. LIMA-4 provides the least accurate reconstruction, but allows a significant reduction of the computational costs.

The CNNs allow us to reconstruct more accurate displacement and velocity fields than the state-of-the-art PIV cross-correlation methods (i.e., a commercial implementation of WIDIM). The CNNs are in general comparable or superior to WIDIM also when applied to experimental datasets. In addition, the CNNs are more robust than WIDIM with respect to an increase in separation time and in particle loss. The error of the fields reconstructed by the CNNs can be effectively reduced by increasing the particle seeding density, whereas WIDIM exhibits persistent errors in presence of large shear flow. Remarkably, our test cases demonstrate that also LIMA-4 can achieve higher accuracy than state-of-the-art cross-correlation methods if the particle seeding density is sufficiently high or when particle loss becomes significant. In particular, LIMA-4 reconstructs more realistic velocity fields than WIDIM in high-gradient regions, avoiding the appearance of major artifacts in the wake near the cylinder and depicting realistic boundary layer separation.

In general, LIMA provides a flexible framework that allows further improvements by modifying the architecture of the iterative residual refinement (e.g., allowing the four-level network to reconstruct fields that have the same resolution as the original image), by optimizing the weights of the mutlilevel loss function, and by appropriately choosing the parameters employed to generate the kinematic training dataset. We also remark that multiple versions of LIMA, with a different number of iterative levels, could be combined into a multi-fidelity approach. For instance, LIMA-4 could be used to provide a first on-line estimate and inspect the flow during the experiment, while LIMA-6 could be then used off-line to reconstruct the flow with the highest accuracy.

Finally, we remark that the current versions of LIMA are implemented using the machine learning library *PyTorch*, and the computational performance can be further improved if the networks are compiled, drastically reducing the time needed for training and reconstruction. In this respect, we envision the deployment of the lightweight architecture on embedded GPU devices which process the recordings onboard and directly stream the reconstructed vector fields. This will considerably reduce the processing time and storage needs in situations in which the measurement matrix may become intractable. Possible future extensions include the deployment of the lightweight architecture on low-cost devices for an affordable on-the-fly inference of the flow field during wind tunnel experiments.

Appendix A: Flow field maps

Here, we show the distribution plots of the velocity magnitude corresponding to Figs. 10 and 11.

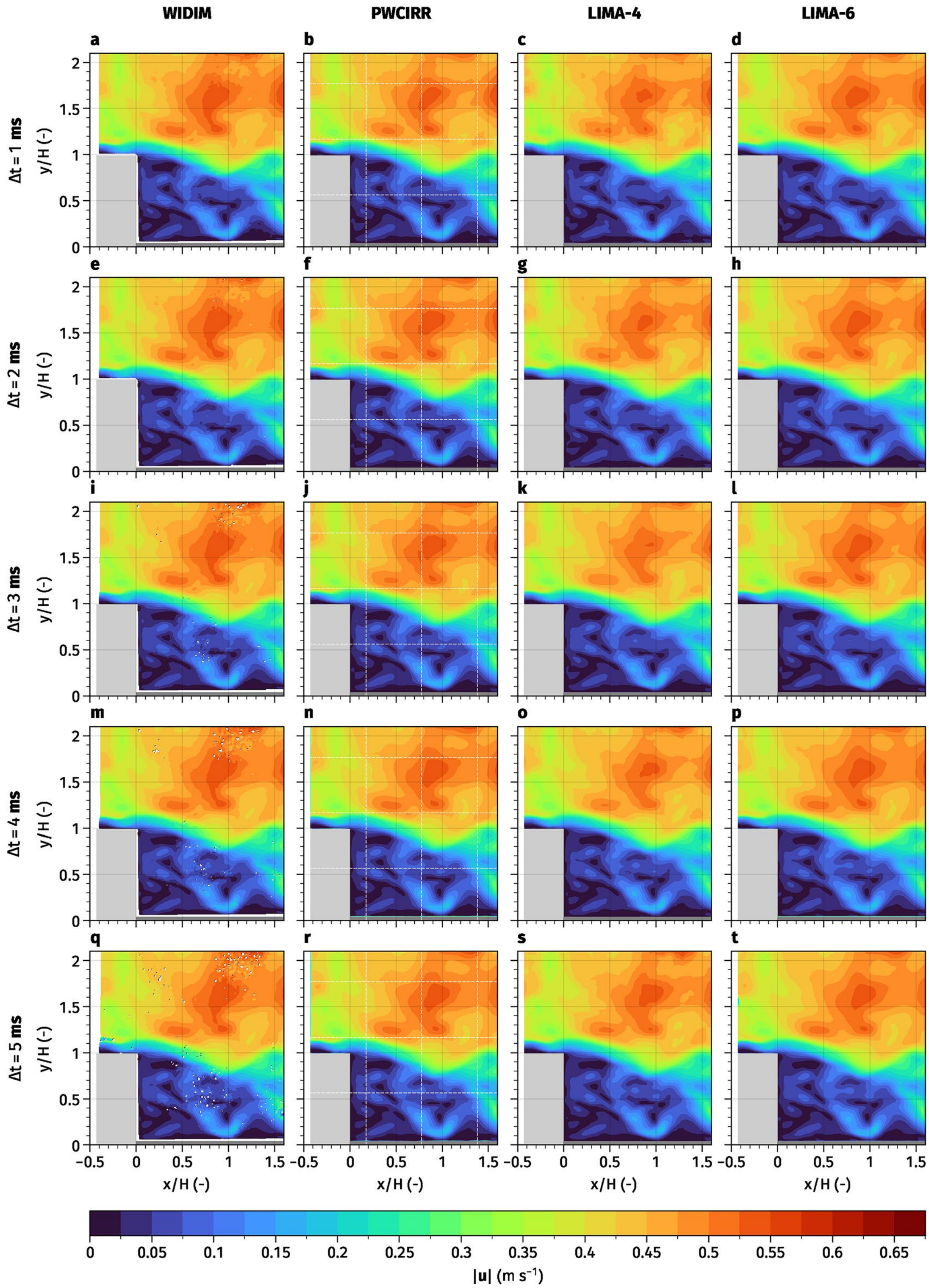

Fig. 10 2-D distribution maps of the instantaneous velocity norm associated to the disparity error shown in Fig. 7: `WIDIM` (cc) versus `PWCIRR`, `LIMA-4`, and `LIMA-6` at increasing separation times (from top to bottom: 1 ms, 2 ms, 3 ms, 4 ms, and 5 ms). Data are obtained with a nominal seeding density of 0.4 ppp. The small white regions corresponds to locations where `WIDIM` fails to converge and provide reliable estimate of the displacement

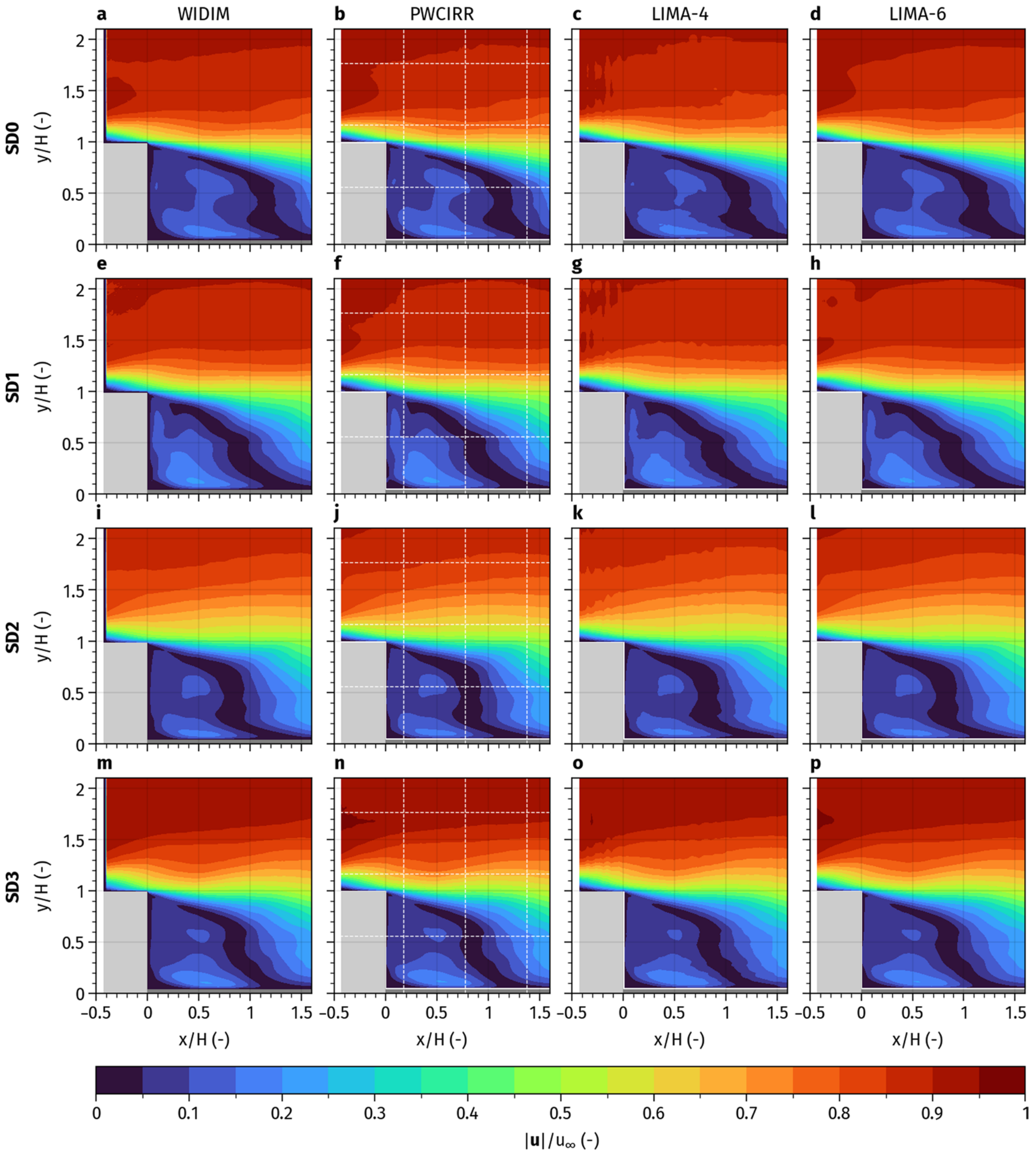

Fig. 11 2-D distribution maps of normalized velocity magnitude for the cases in Fig. 8. Comparing different seeding densities (SD), averaged over 100 samples: `WIDIM` (cc) vs. `PWCIRR`, `LIMA-4`, and `LIMA-6`. The seeding density is 0.03 (**SD0**), 0.04 (**SD1**), 0.05 (**SD2**), and 0.07 (**SD3**)

Acknowledgements We would like to thank Stephan Kunz, Roger Vonbank, and Beat Margelisch for their support at Empa for the experimental setup. We acknowledge access to Swiss National Supercomputing Centre (CSCS), Switzerland, under Empa's share with the project ID: *em13*. We thank Hossein Gorji for fruitful discussions.

Author Contributions LM performed the experiments, derived and implemented the models and analyzed the data. CM designed the experiments, analyzed the data, and wrote the original draft. IL supervised the project, analyzed the data, reviewed, and edited the manuscript. All authors conceptualized the study and approved the final manuscript.

Funding Open Access funding provided by Lib4RI – Library for the Research Institutes within the ETH Domain: Eawag, Empa, PSI & WSL.

Data availability All data and scripts are available upon reasonable request to the authors.

Declarations

Conflict of interest The authors have no competing interests

Ethical approval Not applicable

References

Armellini A, Mucignat C, Casarsa L, Giannattasio P (2012) Flow field investigations in rotating facilities by means of stationary PIV systems. Measur Sci Technol. https://doi.org/10.1088/0957-0233/23/2/025302

Cai S, Liang J, Gao Q, Xu C, Wei R (2020) Particle image velocimetry based on a deep learning motion estimator. IEEE Trans Instrum Meas 69(6):3538–3554

Cai S, Zhou S, Xu C, Gao Q (2019) Dense motion estimation of particle images via a convolutional neural network. Exp Fluids 60(4):73

Carlier J (2005) Second set of fluid mechanics image sequences. European Project fluid image analysis and description (FLUID). http://www.fluid.irisa.fr, pp. 0018–9456

Chen P-H, Yen J-Y, Chen J-L (1998) An artificial neural network for double exposure PIV image analysis. Exp Fluids 24(5–6):373–374. https://doi.org/10.1007/s003480050185

Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, Smagt PVD, Cremers D, Brox T (2015) FlowNet: learning optical flow with convolutional networks. In: 2015 IEEE international conference on computer vision (ICCV), vol. 2015 Inter, pp 2758–2766. IEEE. https://doi.org/10.1109/ICCV.2015.316. https://ieeexplore.ieee.org/document/7410673/

Gao Q, Lin H, Tu H, Zhu H, Wei R, Zhang G, Shao X (2021) A robust single-pixel particle image velocimetry based on fully convolutional networks with cross-correlation embedded A robust single-pixel particle image velocimetry based on fully convolutional networks with cross-correlation embedded. Phys Fluids 33:127125. https://doi.org/10.1063/5.0077146

Hui TW, Tang X, Loy CC (2018)LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. Proceedings of the IEEE computer society conference on computer vision and pattern recognition, 8981–8989 arXiv:1805.07036

Hur J, Roth S (2019) Iterative residual refinement for joint optical flow and occlusion estimation, pp 5747–5756 https://doi.org/10.1109/CVPR.2019.00590

Ilg E, Mayer N, Saikia T, Keuper M, Dosovitskiy A, Brox T (2017) FlowNet 2.0: evolution of optical flow estimation with deep networks, pp 1647–1655 https://doi.org/10.1109/CVPR.2017.179

Lagemann C, Lagemann K, Mukherjee S, Schröder W (2021) Deep recurrent optical flow learning for particle image velocimetry data. Nat Mach Intell 3(7):641–651. https://doi.org/10.1038/s42256-021-00369-0

Lecordier B, Westerweel J (2004) The EUROPIV Synthetic Image Generator (S.I.G.). In: Stanislas M, Westerweel J, Kompenhans J (eds) Particle image velocimetry: recent improvements. Springer, Berlin, pp 145–161

Lee Y, Yang H, Yin Z (2017) PIV-DCNN: cascaded deep convolutional neural networks for particle image velocimetry. Exp Fluids 58(12):171. https://doi.org/10.1007/s00348-017-2456-1

Li Y, Perlman E, Wan M, Yang Y, Meneveau C, Burns R, Chen S, Szalay A, Eyink G (2008) A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. J Turbul 9(31), 1–29 arXiv:0804.1703. https://doi.org/10.1080/14685240802376389

Manickathan L, Mucignat C, Lunati I (2022) Kinematic training of convolutional neural networks for particle image velocimetry. Meas Sci Technol. https://doi.org/10.1088/1361-6501/ac8fae

Mucignat C, Manickathan L, Shah T, Rösgen I Lunati (2023) A lightweight convolutional neural network to reconstruct deformation in BOS recordings. Exp Fluids 64:72. https://doi.org/10.1007/s00348-023-03618-7

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) Pytorch: An imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d' Alché-Buc F, Fox E, Garnett R (eds) Advances in neural information processing systems 32. Curran Associates, Inc., New York, pp 8024–8035

Rabault J, Kolaas J, Jensen A (2017) Performing particle image velocimetry using artificial neural networks : a proof-of-concept. Measur Sci Technol 28:125301. https://doi.org/10.1088/1361-6501/aa8b87

Raffael M, Willert C, Wereley ST, Kompenhans J (2018) Particle image velocimetry, 3rd edn. Springer, Berlin, p 680. https://doi.org/10.1007/978-3-540-72308-0

Scarano F (2002) Iterative image deformation methods in PIV. Meas Sci Technol 13(1):1–19. https://doi.org/10.1088/0957-0233/13/1/201

Schrijer FFJ, Scarano F (2008) Effect of predictor-corrector filtering on the stability and spatial resolution of iterative PIV interrogation. Exp Fluids 45(5):927–941. https://doi.org/10.1007/s00348-008-0511-7

Sciacchitano A, Wieneke B, Scarano F (2013) PIV uncertainty quantification by image matching. Measur Sci Technol. https://doi.org/10.1088/0957-0233/24/4/045302

Teed Z, Deng J (2020) RAFT: recurrent all-pairs field transforms for optical flow. arXiv https://doi.org/10.48550/ARXIV.2003.12039. https://arxiv.org/abs/2003.12039

Wereley ST, Meinhart CD (2001) Second-order accurate particle image velocimetry. Exp Fluids 31(3):258–268. https://doi.org/10.1007/s003480100281

Westerweel J, Scarano F (2005) Universal outlier detection for PIV data. Exp Fluids 39(6):1096–1100. https://doi.org/10.1007/s00348-005-0016-6

Wieneke B (2015) PIV uncertainty quantification from correlation statistics. Measur Sci Technol. https://doi.org/10.1088/0957-0233/26/7/074002

Wieneke B (2017) PIV anisotropic denoising using uncertainty quantification. Exp Fluids 58(8):1–10. https://doi.org/10.1007/s00348-017-2376-0

Willert CE, Gharib M (1991) Digital particle image velocimetry. Exp Fluids 10(4):181–193. https://doi.org/10.1007/BF00190388

Yu C, Bi X, Fan Y, Han Y, Kuai Y (2021) LightPIVNet: an effective convolutional neural network for particle image velocimetry. IEEE Trans Instrum Meas 70:1–15. https://doi.org/10.1109/TIM.2021.3082313