Kinematic training of convolutional neural networks for particle image velocimetry

OPEN ACCESS
IOP Publishing
Measurement Science and Technology
Meas. Sci. Technol. **33** (2022) 124006 (16pp)
https://doi.org/10.1088/1361-6501/ac8fae

Kinematic training of convolutional neural networks for particle image velocimetry

Lento Manickathan, Claudio Mucignat and Ivan Lunati*

Empa, Swiss Federal Laboratories for Materials Science and Technology, Laboratory of Multiscale Studies in Building Physics, Überlandstrasse 129, CH-8600 Dübendorf, Switzerland

E-mail: ivan.lunati@empa.ch

Received 29 April 2022, revised 26 August 2022
Accepted for publication 6 September 2022
Published 16 September 2022

Abstract

Convolutional neural networks (CNNs) offer an alternative to the image cross-correlation methods used in particle image velocimetry (PIV) to reconstruct the fluid velocity field from the experimental recording. Despite the flexibility of CNNs, the accuracy and robustness of the standard image processing remains unsurpassed for general PIV data. As CNNs are non-linear and typically entail up to millions of trainable parameters, they require large and carefully designed training datasets to avoid over-fitting and to obtain results that are accurate for a wide range of flow conditions and length scales. Most training datasets consist of PIV-like data that are generated from displacement fields resulting from numerical flow simulations, which, in addition of being computationally expensive, may be able to inform the network only about relatively few classes of flow problems. To overcome this issue and improve the accuracy of the velocity reconstructed by CNNs, we propose to train the networks with synthetic PIV-like data generated from random displacement fields. The underlying idea is that the training dataset simply needs to teach the network about the kinematic relationship between position and velocity. These kinematic training datasets are computationally inexpensive and may allow a much richer variability in terms of length scales by varying the generation parameters. By training a state-of-the-art CNN, we investigate the accuracy of the reconstructed displacement and velocity with synthetic and experimental test cases, such as a sinusoidal flow and wind-tunnel data from a turbulent-boundary-layer and a cylinder-wake experiment. We demonstrate that kinematic training can drastically improve the accuracy of the CNN and allows the network to outperform conventional cross-correlation methods, being more robust with respect to data noise and providing reconstructed velocity fields that have considerably higher spatial resolution (at pixel level).

Keywords: particle image velocimetry, machine learning, convolutional neural network, kinematic training

(Some figures may appear in colour only in the online journal)

* Author to whom any correspondence should be addressed.

Nomenclature

$\boldsymbol{u}$	velocity vector field ($\mathrm{m\,s^{-1}}$)
$\mathbf{I}$	image pair matrix (counts)
$\mathrm{d}\boldsymbol{s}$	displacement vector field (px)
$\mathrm{d}t$	separation time (s)
A_p	particle peak intensity (counts)
C	constant (—)
D	cylinder diameter (mm)
E	energy spectra (px^2)
I	image intensity (counts)
L	number of pyramid levels (—)
M	magnification factor ($\mathrm{px\,mm^{-1}}$)
N	number of images (—)
N_p	number of particles (—)
N_L	percentage of lost particles between two consecutive images (%)
Q	cross-correlation peak-to-peak ratio (—)
Re	Reynolds number (—)
St	Strouhal number (—)
a	scaling factor (—)
d_p	particle image diameter (px)
f	acquisition frequency (Hz)
k	wavenumber ($\mathrm{px^{-1}}$)
l	pyramid level (—)
m_batch	training mini-batch size (—)
n	number of *numerical* training examples (—)
r	number of *random* training examples (—)
x	horizontal image coordinate (px)
y	vertical image coordinate (px)

Greek symbols

$\boldsymbol{\xi}$	random vector field (—)
β_1	Adam optimization 1st-moment decay rate (—)
β_2	Adam optimization 2nd-moment decay rate (—)
ε	pointwise instantaneous error (px)
ϵ_rms	root-mean-square error (px)
ϵ_d	disparity error (px)
θ	neural network trainable parameters
λ	Adam optimization learning rate (—)
μ	kinematic viscosity ($\mathrm{m^2\,s^{-1}}$)
ν	dynamic viscosity (Pa s)
ρ_p	particle density (particle per pixel) (ppp)
ρ_air	air density ($\mathrm{kg\,m^{-3}}$)
σ	Gaussian filter size (—)
τ_w	wall shear-stress (Pa)
ω	vorticity ($\mathrm{s^{-1}}$)

Operators

$\bar{\phi}$	area-averaged ϕ
$\hat{\phi}$	interpolated ϕ
$\mathcal{C}$	cost volume subnetwork
$\mathcal{D}$	decoder subnetwork
$\mathcal{E}$	encoder subnetwork
$\mathcal{F}$	mapping function
$\mathcal{G}_\sigma$	Gaussian filter function
$\mathcal{N}$	neural network
$\mathcal{U}$	uniform random distribution
$\mathcal{W}$	warping function

Subscripts and superscripts

$+$	wall-dimensionless
1	frame 1
2	frame 2
∞	bulk
cc	cross-correlation (i.e. `WIDIM` method)
cmb	*combined* dataset
nn	neural network
num	*numerical* dataset
ref	reference
rnd1	*random* dataset 1
rnd2	*random* dataset 2
rnd3	*random* dataset 3

1. Introduction

Particle image velocimetry (PIV) (Willert and Gharib 1991, Adrian 2005, Raffel *et al* 2018) is a well-established optical technique that measures fluid velocity by recording the position of fluid tracers using digital cameras and pulsed light sources. The displacement of the tracer particles that occurs during the time interval separating two consecutive realizations, $\mathrm{d}t$, is reconstructed by means of an advanced cross-correlation scheme from the two images recording the tracer positions. This process, which is performed iteratively and dividing each image into multiple interrogation windows, allows us to obtain the displacement vector that best describes the motion of each particle included in each sub-window. Once the displacement $\mathrm{d}\boldsymbol{s}$ is known, the velocity is readily calculated as $\boldsymbol{u} = \mathrm{d}\boldsymbol{s}/\mathrm{d}t$. During the last decades, the method has been continuously improved (Schrijer and Scarano 2008, Raffel *et al* 2018) to obtain the average displacement of the cluster of particles in each interrogation window with sub-pixel accuracy.

So far, machine learning methods have not been extensively used in PIV due to the lack of the necessary computational power and the limited accuracy reported by early studies (Hassan and Philip 1997, Chen *et al* 1998, Labonté 2000). However, thanks to the considerable increase graphics processing units (GPUs) performance, machine learning has improved and several new techniques have emerged in the field of computer vision that are also suitable for PIV. Convolutional neural networks (CNNs) (LeCun *et al* 1989), for example, have been used for optical flow detection in computer vision (Dosovitskiy *et al* 2015, Ilg *et al* 2017, Sun *et al* 2017b, Hui *et al* 2018, Hur and Roth 2019, 2020, Liu *et al* 2019, Teed and Deng 2020) and, recently, to reconstruct two-dimensional (2D) fluid velocity fields in a plane (Lee *et al* 2017, Rabault *et al* 2017, Cai *et al* 2019a, 2019b, 2020, Lagemann *et al* 2021, Yu *et al* 2021). In principle, CNNs are used to learn the function that maps the intensity distributions of a pair of images to the displacement vector field, from which the velocity field can be readily calculated. The success in image processing and image recognition, where they can outperform handcrafted algorithms at reduced computational cost (Krizhevsky *et al* 2012, Szegedy *et al* 2015, He *et al* 2016, Alom *et al* 2018), makes CNNs appealing also for PIV.

Several approaches have been proposed to reconstruct the velocity field from the intensity images. Rabault *et al* (2017) specifically designed two CNNs to reconstruct both the velocity and the Jacobian deformation matrix at the center of subwindows with fixed size (i.e. 32×32 pixels). Their method closely mimics the conventional PIV output but takes advantage of the characteristics of CNN to identify nonlinear dependency. Instead, Cai *et al* (2019a, 2019b, 2020) and Lee *et al* (2017) directly applied existing CNNs, originally designed for optical flow applications, to determine the velocity field with pixel-level resolution. More recently, Gao *et al* (2021) devised a mixed architecture that combines a conventional cross-correlation operator with a neural network. Validation against synthetic data showed a decrease of the reconstruction error, but the architecture is rather complex, which may reduce the advantages offered by CNNs in terms of computational cost.

The success of CNN-based methods depends on an appropriate choice of the training sets, which must be informative and very large in order to reduce the risk of overfitting. Indeed, due to the complexity and the size of the network parameter space, CNNs may contain up to millions of trainable parameters. In most cases, the networks are trained on synthetic datasets consisting of image pairs that are generated from displacement fields obtained by means of numerical solutions of the Navier–Stokes equations (e.g. from the Johns Hopkins Turbulence Database (JHTDB) (Li *et al* 2008)). As numerical simulations of flow problems are computationally expensive, the training datasets are limited in size (typically in the range between 10^4 and 2×10^4 examples (Lee *et al* 2017, Cai *et al* 2019a, 2019b, 2020, Gao *et al* 2021)). This can lead to overfitting and prediction biases: a network trained only on laminar flow examples may, for instance, perform poorly for highly turbulent flows. To overcome this problem without performing costly flow simulations, Rabault *et al* (2017) trained their networks with intensity image pairs obtained from motion fields that are second-order polynomials with coefficients randomly sampled from independent uniform distributions. These coefficients describe the Jacobian and the Hessian of the motion field used to generate the training image and are constant in the processing window (of size 32×32 pixels). This data generation procedure is computationally inexpensive and allows them to drastically enlarge the size of the training set (which comprises about six hundred millions of image pairs). However, the generated displacement fields differ from real PIV images, in which the Jacobian and the Hessian can vary within an interrogation window.

In this work, we propose to train neural networks on datasets generated by random displacement of the tracer particles. In its essence, this kinematic training informs the network about the relationship between two successive positions of the particles and the corresponding displacement. The approach may be regarded as a simplified version of the random flow generation (RFG) technique for inflow boundary condition generation in large-eddy simulations (LESs) (Smirnov *et al* 2001, Klein *et al* 2003). As the particle displacement is obtained through a simple algebraic technique, the generation of a large number of training examples is straightforward and computationally inexpensive. Furthermore, as our approach is independent of specific solutions to the governing equations (it is kinematic-based rather than dynamic-based) it has a potentially larger domain of application. In the following, we investigate the accuracy of displacements and velocity fields obtained by processing synthetic or experimental image pairs using neural networks trained on kinematic random datasets. We also compare the reconstructed velocity fields with those obtained with the same network architecture trained on datasets generated from numerical flow simulations and with state-of-the-art PIV methods (i.e. window deformation iterative multigrid (WIDIM), (Scarano 2001)).

2. CNN for fluid flow velocimetry

A CNN for fluid flow velocimetry is designed to estimate the displacement vector field from images of particle positions, i.e.

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

where $\mathcal{N}$ denotes the operation performed by the neural network, $\mathbf{I} = (I_1, I_2)^\top$ is the pair of particle image intensities (with I_1 and I_2 the image intensities at the old and new times, t_1 and t_2, respectively), and θ the set of trainable parameters of the network. The training process aims at determining θ such that $\mathrm{d}\boldsymbol{s}_{\mathrm{nn}}$ is the best estimate of the true displacement, $\mathrm{d}\boldsymbol{s}_{\mathrm{ref}}$.

After the introduction of the pioneering `FlowNet` (Dosovitskiy *et al* 2015), various network architectures have been proposed to reconstruct the motion of objects from pairs of recorded images. Recently, the focus has been on lightweight architectures such as `LiteFlowNet` (Hui *et al* 2018, 2021) and `PWCNet` (Sun *et al* 2017b). A new class of networks based on iterative residual refinement (IRR) has also been introduced to allow the reduction of the number of trainable parameters and improve the accuracy by iteratively feeding the output back into the network (Hur and Roth 2019, 2020). Here, we employ the IRR version of `PWCNet`, shown in figure 1 and referred to as `PWCIRR`, which combines a pyramid-feature encoder $\mathcal{E}$, warping $\mathcal{W}$, cost volume $\mathcal{C}$, and a decoder $\mathcal{D}$ to estimate the flow. In the architecture, the encoder, cost volume, and the decoder subnetworks are trainable with the specific subsets of parameters $\theta_{\mathcal{E}}$, $\theta_{\mathcal{C}}$, and $\theta_{\mathcal{D}}$, respectively.

The first step is to encode the pair of particle image intensities $\mathbf{I} = (I_1, I_2)^\top$ with the pyramid-feature encoder using the network trainable parameters $\theta_{\mathcal{E}}$. More precisely, the Siamese-style subnetwork consists of L levels and generates a pyramid-feature output such that the level l that has twice the resolution of the previous level $l-1$. Then, the encoded feature I_2 of at level l is dewarped according to the displacement estimate at the previous level $l-1$,

$$\mathcal{W}\left(\mathcal{E}^l(I_2;\theta_{\mathcal{E}}), \hat{\mathrm{d}\boldsymbol{s}}_{\mathrm{nn}}^{l-1}\right), \tag{2}$$

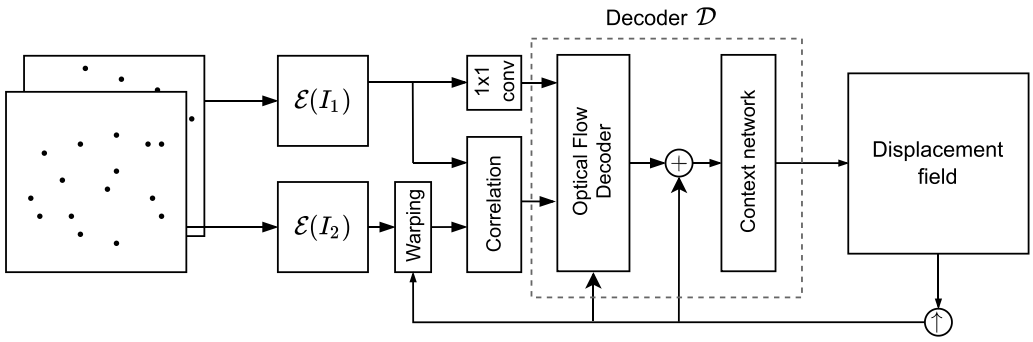

Figure 1. Schematic representation of `PWCIRR` (Hur and Roth 2019): a Pyramid-Warp-Cost Volume Network (`PWCNet`) (Sun *et al* 2017b) based on iterative residual refinement (IRR).

where $\hat{\mathrm{d}}\boldsymbol{s}_{\mathrm{nn}}^{l-1}$ is obtained from the displacement estimate at the level $l-1$, and is two-time upsampled by bilinear interpolation to match the resolution of $\mathcal{E}^l(I_2;\theta_\mathcal{E})$ for image warping.

Thereafter, the cost volume $\mathcal{C}$ operator assesses the goodness of the previous displacement estimate by cross-correlating the features of the encoded image at the old time with the warped features of the image at the new time, i.e.

$$\mathcal{C}\left(\mathcal{E}^l(I_1;\theta_\mathcal{E}),\mathcal{W}\left(\mathcal{E}^l(I_2;\theta_\mathcal{E}),\hat{\mathrm{d}}\boldsymbol{s}_{\mathrm{nn}}^{l-1}\right);\theta_\mathcal{C}\right). \tag{3}$$

Once the cost volume is available, the optical flow decoder $\mathcal{D}$ estimates the displacement field at level l. The decoder requires the feature of the first image, $\mathcal{E}^l(I_1;\theta_\mathcal{E})$, the output of the cost volume $\mathcal{C}$, and the previous (two-time upsampled) estimate of the displacement field $\hat{\mathrm{d}}\boldsymbol{s}_{\mathrm{nn}}^{l-1}$, i.e.

$$\mathcal{D}\left(\mathcal{E}^l(I_1;\theta_\mathcal{E}),\mathcal{C}\left(\mathcal{E}^l(I_1;\theta_\mathcal{E}),\mathcal{W}\left(\mathcal{E}^l(I_2;\theta_\mathcal{E}),\hat{\mathrm{d}}\boldsymbol{s}_{\mathrm{nn}}^{l-1}\right);\theta_\mathcal{C}\right),\hat{\mathrm{d}}\boldsymbol{s}_{\mathrm{nn}}^{l-1};\theta_\mathcal{D}\right). \tag{4}$$

`PWCIRR` differs from `PWC` in that the weights of each level in the optical flow decoder are shared (Hur and Roth 2019), which reduces the total number of parameters. Thus, the displacement field is iteratively estimated as,

$$\mathrm{d}\boldsymbol{s}_{\mathrm{nn}}^{l}=\mathcal{D}\left(\mathcal{E}^l(I_1;\theta_\mathcal{E}),\mathcal{C}\left(\mathcal{E}^l(I_1;\theta_\mathcal{E}),\mathcal{W}\left(\mathcal{E}^l(I_2;\theta_\mathcal{E}),\hat{\mathrm{d}}\boldsymbol{s}_{\mathrm{nn}}^{l-1}\right);\theta_\mathcal{C}\right),\hat{\mathrm{d}}\boldsymbol{s}_{\mathrm{nn}}^{l-1};\theta_\mathcal{D}\right)+\hat{\mathrm{d}}\boldsymbol{s}_{\mathrm{nn}}^{l-1}. \tag{5}$$

In order to enable the use of the same decoder at each level, an additional 1×1 convolution layer is inserted in between $\mathcal{E}(I_1;\theta_\mathcal{E})$ and the optical flow decoder $\mathcal{D}$ to ensure that the dimension of the input features matches. Furthermore, a context network is included in the network design. This additional subnetwork employs dilated convolutional kernels to increase the spatial-range dependency in the displacement reconstruction. A more detailed description of $\mathcal{E}$, D and the context network can be found in Sun *et al* (2017). The pixel level estimate of the displacement $\mathrm{d}\boldsymbol{s}_{\mathrm{nn}}$ is obtained after the final iteration (see figure 2). We note that the image intensities are always normalized before performing the inference with the CNN, spanning between 0 and 1, as done with the training data.

2.1. Overview of the training datasets

The network parameters are determined by a supervised learning procedure that trains the network on datasets consisting of pairs of particle image intensities ($\mathbf{I}=(I_1,I_2)^\top$) and the corresponding displacement vector fields ($\mathrm{d}\boldsymbol{s}_{\mathrm{ref}}$), from which the velocity field is readily calculated dividing by the separation time. To train the network for accurate reconstruction of the velocity field, large training datasets are required.

In general, the training datasets are created starting from a numerically generated velocity field, from which pairs of synthetic particle image intensities are obtained. We consider various training datasets, which are summarized in table 1, and compare their performance in training the network to estimate the velocity field for both synthetic and experimental test cases. In previous studies, the training displacement fields were obtained from analytical flow solutions (Li *et al* 2008, Cai *et al* 2019b). Similarly, we consider a numerical training dataset (`num`) that consists of analytical flow solutions such as uniform flow, shear flow, and Lamb–Oseen vortex, and data from the JHTDB (see table 2).

Here, we propose to train the network with datasets generated from random displacement fields. These datasets can train the network to identify the kinematic relationship between two successive tracer particle positions and the corresponding displacement and velocity fields, regardless of the underlying dynamics of the physical process. These datasets have the advantage that they can be generated at a relatively inexpensive computational cost and that their size can be arbitrarily increased. We consider three random datasets for kinematic training of the network containing an increasing number of training examples, i.e. `rnd1`, `rnd2`, and `rnd3`, each of which has twice the number of examples as the previous one (see table 1). Finally, we also consider a dataset (`cmb`) that combines the data of `rnd1` and `num`, hence having the same size as `rnd2`.

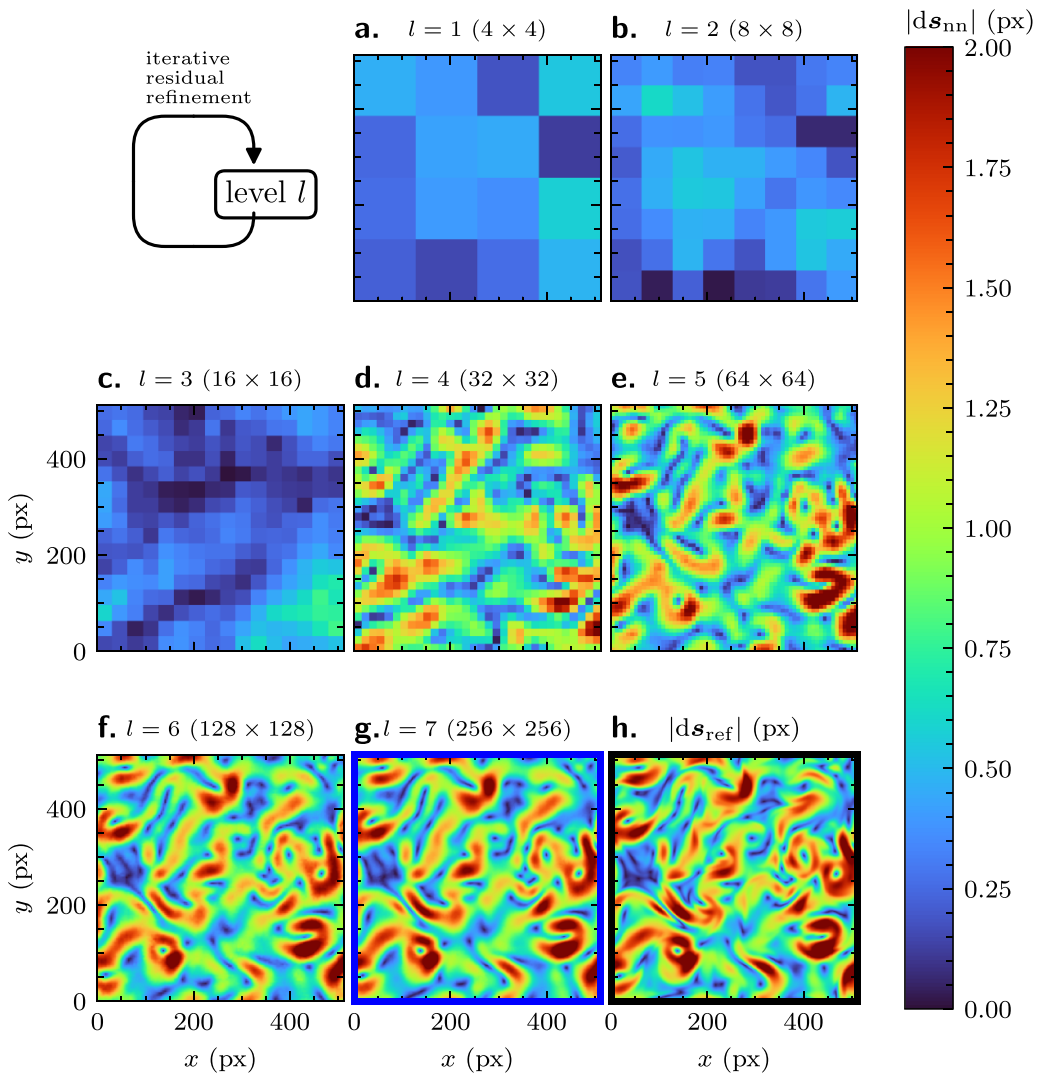

Figure 2. Illustration of the reconstruction of the displacement by iterative residual refinement (IRR) (equation (5)): (a)–(g) the magnitude of the displacement, $|\mathrm{d}\boldsymbol{s}_{\mathrm{nn}}|$ (px), at level (l) is predicted from the output of level $l-1$; (h) reference displacement magnitude $|\mathrm{d}\boldsymbol{s}_{\mathrm{ref}}|$ (px).

Table 1. Composition of the datasets used to train the network. The numerical dataset, `num`, comprises image pairs generated from analytical solutions and numerical solutions of the Johns Hopkins Turbulence Database (JHTDB) (see table 2); `rnd1`, `rnd2`, and `rnd3` contains only image pairs generated from random displacement fields; and `cmb` includes combines `rnd1` and `num`. The last column gives number of random (r) and numerical (n) pairs in each dataset.

Name	Dataset	Number of samples (—)
`num`	Numerical dataset: analytic + JHTDB (see table 2)	$9139n$
`rnd1`	Random dataset	$9139r$
`rnd2`	Large random dataset ($2\times$)	$18278r$
`rnd3`	Very large random dataset ($4\times$)	$36556r$
`cmb`	Combined random (`rnd1`) + numerical (`num`)	$9139\,r + 9139n$

In section 2.2, we describe in detail the generation of the random-displacement datasets for kinematic training (`rnd1`, `rnd2`, and `rnd3`), which consists of the generation of the displacement field and the generation of the particle image pairs from the displacement. To create the realizations of the `num` dataset, we employ the same image generation process, but the displacement fields are calculated from numerical or analytical velocity fields that are solutions to flow problems.

Table 2. The numerical dataset, `num`, is similar to the training dataset used by Cai *et al* (2019b) and contains image pairs generated from analytical solutions of flow problems and from the numerical solutions of the Johns Hopkins Turbulence datasets (JHTDB; (Li *et al* 2008)).

Dataset type	Number of samples (—)
Uniform	1000
Shear	1000
Lamb–Oseen vortex	1000
JHTDB (isotropic turbulence)	5000
JHTDB (turb. channel)	2000
JHTDB (MHD)	1024

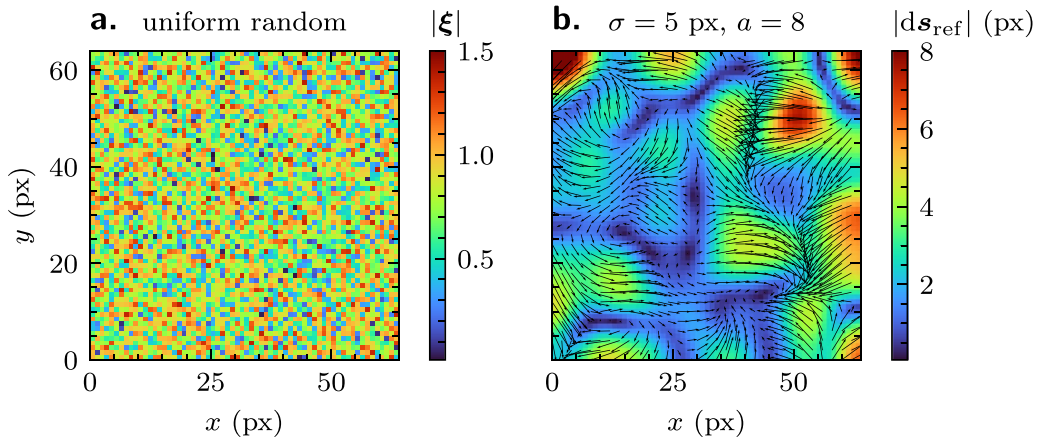

Figure 3. Generation of the training data from a random displacement field: (a) magnitude of the uniform random displacement, $|\boldsymbol{\xi}|$; (b) generated displacement field, $\mathrm{d}\boldsymbol{s}_{\mathrm{ref}}$ (vectors), and its magnitude, $|\mathrm{d}\boldsymbol{s}_{\mathrm{ref}}|$ (color scale), obtained after applying a Gaussian filter of size $\sigma = 5$ px and a linear transformation scale factor $a = 8$.

2.2. Generation of the kinematic training datasets

To train the network for identifying the kinematic relationships between the successive particle positions and the corresponding velocity field, we developed an RFG that produces PIV-like images from kinematic motion data. The algorithm is similar to the techniques that are used to generate inflow boundary conditions in LES (Smirnov *et al* 2001, Klein *et al* 2003). First, we generate a 2D random vector field by independently sampling each random velocity component, $\xi_{i\in\{x,y\}}$, from a uniform distribution $\mathcal{U}(-1,1)$; then, each component is passed through a Gaussian filter of size σ, $\bar{\xi}_i = \mathcal{G}_\sigma * \xi_i$, to obtain a vector field that has finite correlation length; and finally, the velocity is scaled by a factor a to obtain the reference vector field, $\mathrm{d}\boldsymbol{s}_{\mathrm{ref}} = a\,\bar{\xi}$.

To increase the variability of the displacement field and enrich the information contained in the training set, for each realization the filter size and the scaling factor are sampled from a uniform distribution, i.e. $\sigma \sim \mathcal{U}[\sigma_{\min}, \sigma_{\max}]$ and $a \sim \mathcal{U}[a_{\min}, a_{\max}]$, respectively. The parameters of the uniform distributions are reported in table 3, whereas figure 3 shows an example of the random kinematic displacement fields that are provided by the algorithm starting from an initial random distribution.

Once a displacement field is obtained, we generate the pairs of particle images using a synthetic image generator that is similar to the ones proposed by Lecordier and Westerweel (2004), Armellini *et al* (2012). First, we create the image intensity at the old time, I_1, by seeding the image domain with N_{p} tracer particles such that we achieve the desired particle density ρ_{p} (ppp) (i.e. particle per pixel). Then, we model each particle intensity as a 2D Gaussian distribution (see, e.g. Raffel *et al* 2018),

$$I_{\mathrm{p}}(x,y) = A_{\mathrm{p}} \exp\left(-\frac{(x-x_{\mathrm{p}})^2 + (y-y_{\mathrm{p}})^2}{(1/8)d_{\mathrm{p}}^2}\right), \tag{6}$$

where A_{p} is the maximum particle intensity, $(x_{\mathrm{p}}, y_{\mathrm{p}})$ is the particle center expressed in pixel, and d_{p} (px) is the particle image diameter. The image intensity at the old time is obtained by adding the contribution of each particle,

$$I_1 = \sum_{p\in N_{\mathrm{p}}} I_{\mathrm{p}}. \tag{7}$$

To obtain the image intensity at the new time, I_2, we first employ a multistep approach to move the particles according to the local value of the generated displacement field that they encounter along their trajectories. During the motion, a fraction of the tracer particles is randomly removed and replaced to mimic the particle loss, N_{L} (%), between the two image pairs, which is unavoidable in real experiments. After the particle positions at the new time have been determined, the new image intensity field, I_2, can be obtained in the same manner as done for I_1.

Table 3. The range of particle properties and random displacement field settings used for synthetic particle image generation.

Parameter	Min	Max
Filter size σ (px)	5	100
Scale factor a (—)	0	16
Particle diameter d_p (px)	1	4
Seeding density ρ_p (ppp)	0.05	0.40
Particle loss N_L (%)	0	2

For each training dataset, we varied the particle seeding properties d_p, ρ_p, and N_L according to table 3. We remark that the percentage of lost particles, the flow field scale factor and the filter size have a uniform random distribution, while for a given image particle diameter there is a realizable range of particle density. Indeed, for a given particle size d_p, there is an upper limit for the particle density $\rho_\mathrm{p,eff}$ above which individual particles cannot be distinguished.

2.3. Training strategy and data augmentation

To improve the generalization of the network, the training datasets are augmented during the training process by means of random translations, scaling, rotation, reflection, brightness change, and additive Gaussian noise, which represents image background noise observed during real-world setup. In this study, we use the same data augmentation settings as in previous studies (Ilg *et al* 2017, Hui *et al* 2018, Cai *et al* 2019b, 2020). The network is trained on 8 Cray XC50 compute nodes at CSCS (Swiss National Supercomputing Center), each equipped with one Nvidia Tesla P100. The training time for 200 epochs with 18 278 images is approximately 123 h (5 days, 2 h, 50 min) with the eight-node setup. The objective function training optimization is performed by means of the Adam optimization method with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a mini-batch size $m_\mathrm{batch} = 4$ and an initial learning rate $\lambda = 1 \times 10^{-4}$. The learning rate is halved at epochs [60, 80, 100, 120 and 140].

3. Results and discussions

We consider three test cases to assess the performance of the trained CNN: (a) a spatial resolution study aiming at retrieving an analytical sinusoidal displacement field; (b) an experimental study of a turbulent boundary layer (TBL), which includes the uncertainty quantification of the estimated velocity field; and (c) an experimental study of a cylinder wake with uncertainty quantification. For all three test cases, we compare the displacement and velocity fields estimated by the CNN with those obtained by convectional PIV cross-correlation using the `WIDIM` scheme (Scarano 2001, Schrijer and Scarano 2008) as implemented in Davis 10 (LaVision). Hereafter, we use the subscripts `num`, `cmb`, `rnd1`, `rnd2`, and `rnd3` to indicate the dataset (see table 1) that has been used to train the CNN for estimating the displacement (or velocity) field, i.e. we write $\mathrm{d}\boldsymbol{s}_\mathrm{num}$, $\mathrm{d}\boldsymbol{s}_\mathrm{cmb}$, $\mathrm{d}\boldsymbol{s}_\mathrm{rnd1}$, $\mathrm{d}\boldsymbol{s}_\mathrm{rnd2}$, and $\mathrm{d}\boldsymbol{s}_\mathrm{rnd3}$, respectively. Similarly, the subscript `cc` indicates the displacement field obtained with `WIDIM`, i.e. $\mathrm{d}\boldsymbol{s}_\mathrm{cc}$.

3.1. Spatial resolution study: sinusoidal flow

The ability to resolve coherent velocity structures at different scales from the image pairs is a key feature of PIV. To assess the accuracy of the estimated displacement fields, we consider a multi-wavelength vortical displacement field defined by the sinusoidal function,

$$\mathrm{d}\boldsymbol{s}_\mathrm{ref} = \begin{pmatrix} 3\cos(\varphi)\sin(\gamma) \\ 3\sin(\varphi)\cos(\gamma) \end{pmatrix}, \tag{8}$$

where we define

$$\varphi(x,y) = 22\exp\left(-\frac{\sqrt{2}\sqrt{(x-256)^2}}{128}\right), \tag{9}$$

$$\gamma(x,y) = 22\exp\left(-\frac{\sqrt{2}\sqrt{(y-256)^2}}{128}\right). \tag{10}$$

The distribution plot of $\mathrm{d}\boldsymbol{s}_\mathrm{ref}$ as defined by equation (8) is shown in figure 4(a). The maximum magnitude of the displacement is 3 px, and the structures with the shortest spatial wavelengths are placed in the center of the domain to ensure that boundary artifacts are not affecting the detection of the smallest flow features. Figure 4 plots the maps of the displacement estimated by $\mathcal{N}_\mathrm{rnd1}$, $\mathcal{N}_\mathrm{num}$, and `WIDIM` (i.e. $\mathrm{d}\boldsymbol{s}_\mathrm{rnd1}$, $\mathrm{d}\boldsymbol{s}_\mathrm{num}$, and $\mathrm{d}\boldsymbol{s}_\mathrm{cc}$, respectively) with the reference displacement, $\mathrm{d}\boldsymbol{s}_\mathrm{ref}$. For `WIDIM` we employ an initial interrogation window of 64×64 px and a refinement step of 8×8 px; the vector validation is based on the peak-to-peak ratio criterion and the normalized median filter (NMF, (Westerweel and Scarano 2005)) with a 5×5 kernel. In this specific case, the image pairs had a particle density $\rho_\mathrm{p} = 0.1$ ppp, particle diameter of $d_\mathrm{p} = 3$ px, and no Gaussian noise is added.

By comparing the plots in figure 4 we observe that $\mathcal{N}_\mathrm{num}$ is unable to correctly reconstruct small-scale vortical structures and has lower accuracy than `WIDIM`; in contrast, $\mathcal{N}_\mathrm{rnd1}$ allows reconstructing a significantly higher level of small-scale details of the displacement field.

3.1.1. Error assessment. To quantitatively assess the displacement error, we generate $N = 100$ pairs of image intensities with the same seeding properties and use each pair to estimate the displacement. For each method, we calculate the pointwise root-mean-square error ϵ_rms (px) of the estimated displacement field defined,

$$\epsilon_\mathrm{rms} = \sqrt{\frac{1}{N}\sum_{i\in N} |\mathrm{d}\boldsymbol{s}_\mathrm{ref} - \mathrm{d}\boldsymbol{s}_\mathrm{method}|_i^2}, \tag{11}$$

and the average error, $\bar{\epsilon}_\mathrm{rms}$, as the spacial average of the pointwise error. (In equation (11), method $\in$ {num, cmb, rnd1, rnd2, rnd3, cc} denotes the method used to estimate the displacement field.) The distribution plots of ϵ_rms corresponding to the displacement fields obtained with the CNN trained on different datasets and with `WIDIM` are shown in figure 5, whereas the average errors are reported in table 4.

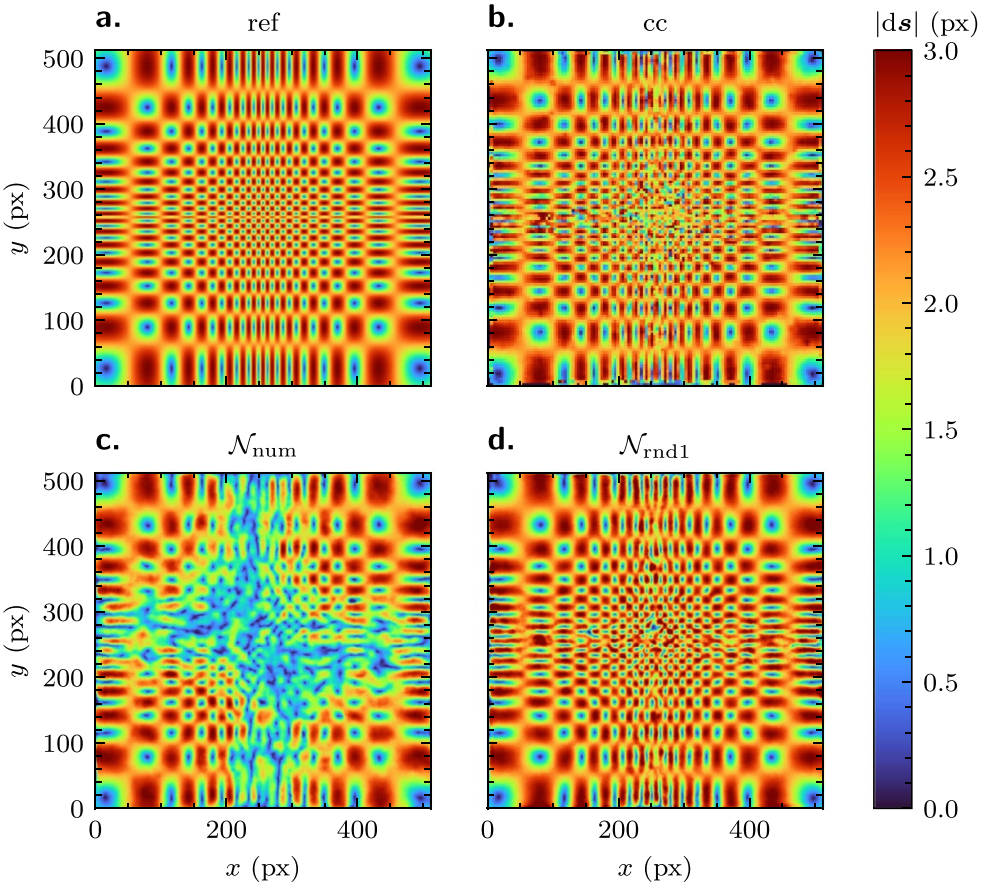

Figure 4. Spatial distribution of magnitudes of the displacement fields: (a) reference, $|\mathrm{d}\boldsymbol{s}_{\text{ref}}|$ (px); (b) reconstruction by `WIDIM`,$|\mathrm{d}\boldsymbol{s}_{\text{cc}}|$ (px); (c) reconstruction by $\mathcal{N}_{\text{num}}$, $|\mathrm{d}\boldsymbol{s}_{\text{num}}|$ (px); (d) reconstruction by $\mathcal{N}_{\text{rnd1}}$, $|\mathrm{d}\boldsymbol{s}_{\text{rnd1}}|$ (px). The reference particle settings used for assessment are $\rho_{\text{p}} = 0.1$ ppp, $d_{\text{p}} = 3$ px, and $\sigma = 0\%$ (i.e. no Gaussian noise).

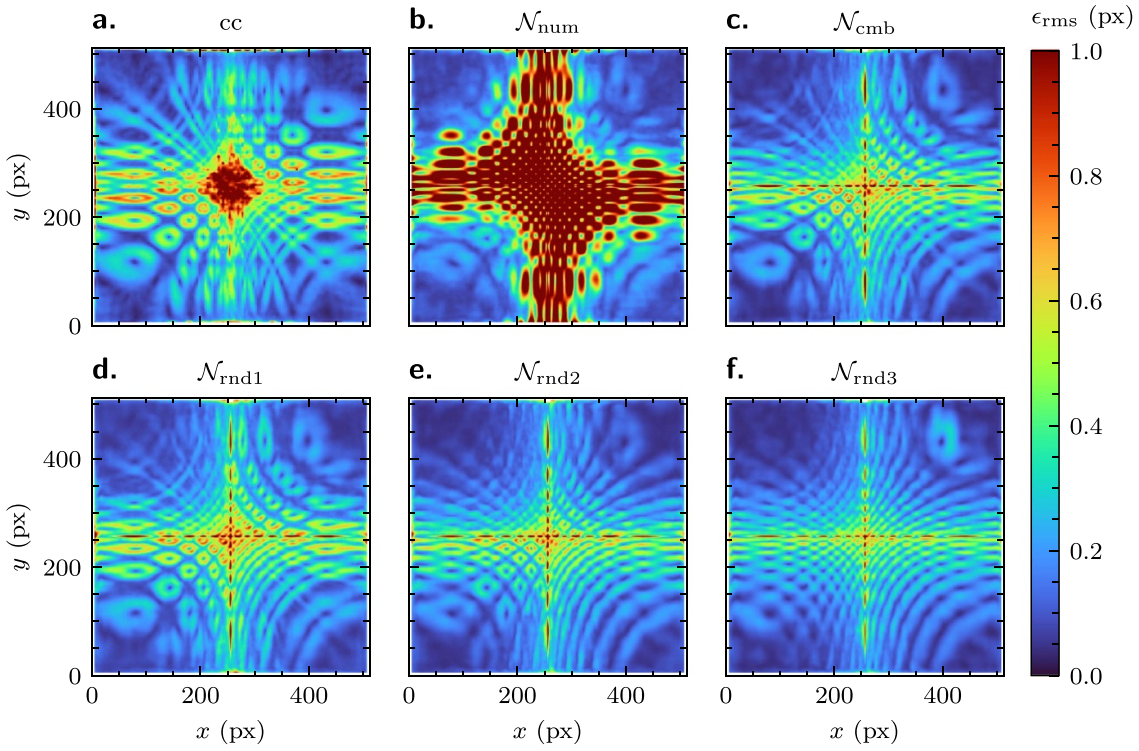

Figure 5. Spatial distribution of point-wise root-mean-square error, ϵ_{rms}, for the displacement estimated (a) by `WIDIM`; (b)–(f) by the CNN, $\mathcal{N}$, trained on different datasets (see table 1). The point-wise error is calculated as the average of the estimates obtained from 100 image pair samples with reference seeding properties $\rho_{\text{p}} = 0.1$ ppp, $d_{\text{p}} = 3$ px, and no Gaussian noise.

Table 4. Spatial average of the root-mean-square error, $\bar{\epsilon}_{\text{rms}}$, calculated from a set of 100 samples of displacement fields reconstructed by `WIDIM` or by $\mathcal{N}$ (trained on different datasets, see table 1) with the following seeding properties: $\rho_{\text{p}} = 0.1$ ppp, $d_{\text{p}} = 3$ px, and $\sigma = 0\%$.

Method	Average RMSE $\bar{\epsilon}_{\text{rms}}$ (px)
`WIDIM`	0.29
$\mathcal{N}_{\text{num}}$	0.60
$\mathcal{N}_{\text{cmb}}$	0.21
$\mathcal{N}_{\text{rnd1}}$	0.23
$\mathcal{N}_{\text{rnd2}}$	0.20
$\mathcal{N}_{\text{rnd3}}$	0.18

All the estimated displacements show the largest error (above 0.5 px) in regions characterized by the smallest vortex cores (i.e. mostly in the central region of the domain). The displacement obtained by training the CNN with the datasets consisting of flow solutions, $\mathrm{d}\boldsymbol{s}_{\text{num}}$, is the least accurate, with the largest average error, $\bar{\epsilon}_{\text{rms}} = 0.6$ px, and a large portion of the domain exhibiting large errors. The errors are sensibly reduced if the CNN is trained with datasets that include examples generated from random-displacement fields. In particular, $\mathcal{N}_{\text{rnd1}}$ allows us to estimate the displacement with much greater accuracy than $\mathcal{N}_{\text{num}}$ even if it employs a dataset that contains the same number of samples. This suggests that kinematic training is more effective in training the CNN for velocity estimation than the dataset obtained from flow-problem solutions.

If we supplement the `num` dataset with data generated from random-displacement fields (i.e. we use the `cmb` dataset), the large-error region is reduced and $\bar{\epsilon}_{\text{rms}}$ drops to 0.21 px. Notice, however, that a better accuracy can be achieved by increasing the size of the random datasets, i.e. training the network with the `rnd2` and `rnd3` datasets, which reduce the average displacement error to 0.20 px and 0.18 px, respectively. We observe that the accuracy gain obtained by enlarging the random dataset tends to decrease with the dataset size. Indeed, doubling the dataset size from 9139 (`rnd1`) to 18 278 (`rnd2`) samples yields a reduction of the average error from 0.23 px to 0.20 px, whereas doubling the size further to 36 556 samples (`rnd3`) reduces the error to only 0.18 px, despite a significant increase in the training cost. This suggests that the amount of information that the CNN is able to extract from the kinematic training datasets tends to saturate with the size, at least for the range of parameters that we have used for the dataset generation (see table 3). A more effective improvement in accuracy may require, for instance, reducing the Gaussian filter size, σ, which may allow the CNN to better estimate the smallest displacement structures.

Finally, we observe that all CNNs that have been trained with datasets generated by random kinematic motion provide a better displacement estimate than `WIDIM` (see figure 5 and table 4). Employing larger training datasets (i.e. `rnd2`, `rnd3`, or `cmb`), the CNN yields a reduction of the average displacement error, $\bar{\epsilon}_{\text{rms}}$, in the range between 28% and 38% with respect to `WIDIM`. This demonstrates the effectiveness of the kinematic training and the potential to achieve higher accuracy than conventional PIV post-processing in the case of real experimental data.

3.1.2. Spatial dynamic range. In order to assess the capability of the different methods to retrieve small-scale coherent structures, we calculate the spatial energy spectra, $E(k)$, as a function of the spatial wave number, k, according to the definition given by Pope (2000). Figure 6 shows the plots of $E(k)$ in image coordinates, calculated from the data in figure 5.

The spectra obtained using the $\mathcal{N}_{\text{num}}$ dataset significantly deviates from the true spectra for $k > 7\,\text{px}^{-1}$. If compared with the spatial dynamic range of `WIDIM`, $\mathcal{N}_{\text{num}}$ displays larger errors that bias the measurement also for coherent structures of fairly large sizes. On the contrary, employing datasets that include samples generated for kinematic training (i.e. `cmb`, `rnd1`, `rnd2`, or `rnd3`) allows the CNN, $\mathcal{N}$, to detect small-scale features with much higher accuracy than `WIDIM`. In general, increasing the number of random training samples enhances the ability of the network to detect coherent structures of higher wavenumbers, mimicking the true spectrum up to wave numbers in the order of $k = 45\,\text{px}^{-1}$ with `rnd2` and `rnd3` (cf E_{rnd2} and E_{rnd3} with E_{ref} in figure 6(b)). The spectrum obtained by `WIDIM` deviates from the true spectrum at wave numbers in the order of $30\,\text{px}^{-1}$. This is due to the fact that `WIDIM` provides estimates of the displacement that are averaged over each interrogation window. Therefore, if the interrogation windows contain larger gradients, the average displacement is not a good estimate of the true displacement. To mitigate this problem, several non-uniform weighting functions (e.g. Gaussian) have been implemented (Astarita 2007); however, the results presented here indicate that the non-linearity of the CNN is more capable of recovering strong spatial gradients, self-adapting to the local characteristics of the displacement field.

3.1.3. Sensitivity analysis. To perform an analysis of the sensitivity of the estimated displacement to the image intensity parameters (i.e. particle density, particle diameter, and additive image Gaussian noise), we vary one parameter at a time while keeping the others constant (the reference seeding values are $\rho_{\text{p}} = 0.1$ ppp for the particle density, $d_{\text{p}} = 3$ px for the particle diameter, and no Gaussian noise). For each set of parameters, we generate and process a set of 100 images and calculate the average displacement error $\bar{\epsilon}_{\text{rms}}$.

Figure 7(a) shows $\bar{\epsilon}_{\text{rms}}$ as a function of the particle seeding density ρ_{p} (ppp). The error increases at low seeding densities for all methods and training datasets employed. As they rely on the presence of tracers to retrieve flow motion, small particle densities do not allow sufficient coverage of the domain. The displacement estimated by the CNN trained on the `num` dataset displays an average error that is roughly twice the error of the standard method. However, if the training datasets include samples for kinematic training, the CNN achieve a better accuracy than `WIDIM`. Similar observations can be made about the dependency of the error on the particle

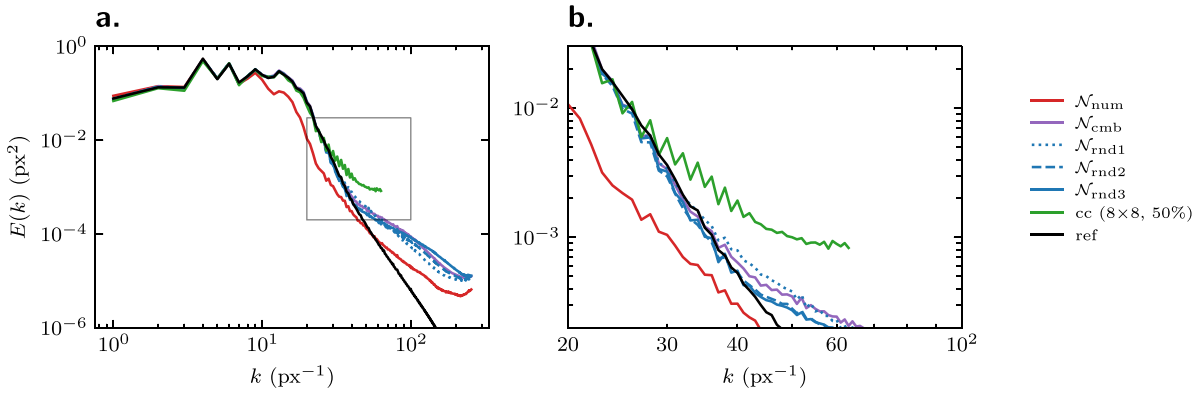

Figure 6. Radial kinetic energy spectra E (px^2) of the sinusoidal flow: reference (black), reconstruction by $\mathcal{N}_{\text{num}}$ (red), $\mathcal{N}_{\text{cmb}}$ (violet), $\mathcal{N}_{\text{rnd1}}$ (blue dotted), $\mathcal{N}_{\text{rnd2}}$ (blue dashed), $\mathcal{N}_{\text{rnd3}}$ (blue solid), and `WIDIM` (green). (a) Full-range of the energy spectra; (b) detail of the region indicated by gray box in (a).

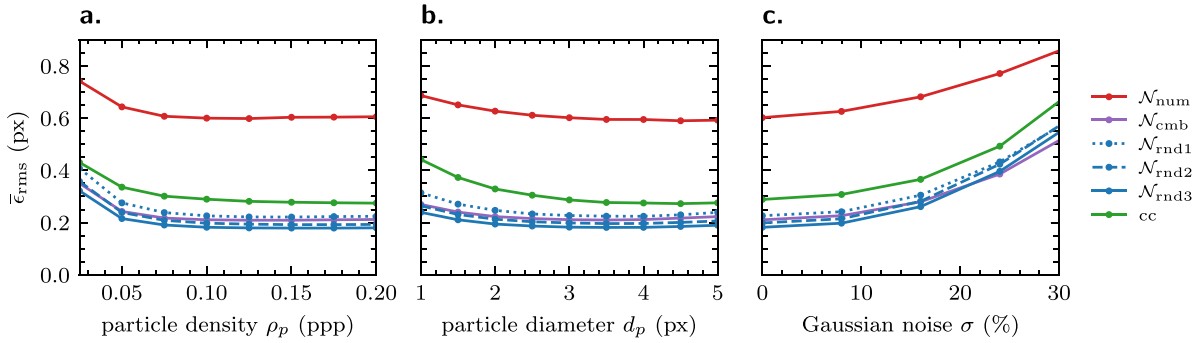

Figure 7. Average root-mean-square error $\bar{\epsilon}_{\text{rms}}$ (px), calculated from a set of 100 samples of displacement fields and reconstructed by `WIDIM` or by $\mathcal{N}$ trained on different datasets (see table 1): (a) sensitivity to the particle density, ρ_{p} (ppp); (b) sensitivity to the particle diameter, d_{p} (px); (c) sensitivity to the image Gaussian noise. Baseline properties are $\rho_{\text{p}} = 0.1$ ppp, $d_{\text{p}} = 3$ px, and $\sigma = 0\%$.

diameter (figure 7(b)), for which the error increase at small values is more pronounced for `WIDIM` than for the CNN. Again, large kinematic training datasets allows the highest accuracy, even compared with classic cross-correlation image processing.

Finally, we investigate the effect of adding noise to the input images. As expected, the error increases with the standard deviation, σ, of the Gaussian noise (figure 7(c)). The accuracy achieved by `WIDIM` is always lower than the accuracy of the CNN, except if the `num` dataset is used. Notice that the average error of `WIDIM` with $\sigma = 3\%$–4% is approximately as large as the error of the displacement estimated by the CNN with σ in the range between 16% and 19% When the training datasets include samples for kinematic training (i.e. `cmb`, `rnd1`, `rnd2`, or `rnd3`). This suggests that the CNN has superior performance in the case of difficult experimental conditions such as in the presence of laser reflections, low laser power, or cameras with reduced sensitivity. It is worth noticing that in case of high-noise data ($\sigma > 20\%$) the CNN trained with the `cmb` dataset provides the most accurate estimate of the displacement. This indicates that, in the case of high-noise data, combining the numerical dataset with kinematic training datasets helps prevent the CNN from misinterpreting noise as displacement.

3.2. TBL

In this section, we investigate the accuracy achieved by the CNN when applied to estimate the velocity from real experimental recordings of a wall-bounded TBL flow. The experiment is performed in the wind tunnel of the Swiss Federal Laboratories of Material Science and Technology. The facility has a cross-section of 1.9 m × 1.3 m (width and height, respectively) and can operate with a bulk wind speed u_∞ ranging from 0.5 to 25 m s^{-1}. The imaging setup comprises diethyl-hexyl-sebacat tracer particles of size 1 µm obtained with a Laskin seeding generator, illuminated by Nd:YLF dual cavity laser with a pulse energy of 30 mJ at 1 kHz, and imaged by a 4 Mpx high-speed camera. The camera sensor consists of 2016 px × 2016 px, with a pixel size of 11 µm. A magnification factor M equal to 40 px mm^{-1} is achieved using a 200 mm focal lens with a numerical aperture equal to $f/2.8$ in combination with a 2 × teleconverter. The separation time is chosen to obtain a maximum displacement of 16 px at a wind speed of 2.5 m s^{-1}. To compute the average velocity field, a set of 10 000 images is recorded. An *a-posteriori* analysis shows the velocity fields displays convergence of the second-order statistics already after 3000 samples. Classical PIV analysis is performed by means of the `WIDIM` scheme with background

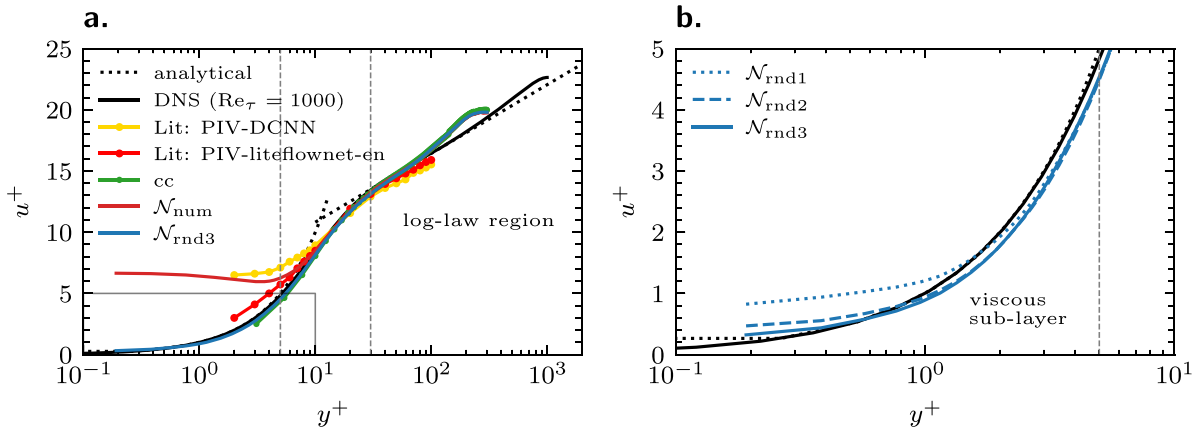

Figure 8. Dimensionless mean velocity profiles in the TBL, u^+ vs. y^+, as a function of the dimensionless distance from the wall, estimated by `WIDIM` (green with dots) and by the CNN, $\mathcal{N}$, after training on `num` (red solid), `rnd1` (blue dotted), `rnd2` (blue dashed), and `rnd3` (blue solid) datasets. The profiles are compared with the reference profiles obtained from the analytical solution (equation (12)) (black dotted line) and from DNS (Graham *et al* 2016) (black solid). For comparison, we also plot the velocity profiles reconstructed by means of CNN architectures used in previous studies, i.e. `PIV-DCNN` (Lee *et al* 2017) (yellow with dots), `PIV-liteflownet-en` (Cai *et al* 2020) (red with dots). Figure (a) illustrates the full-range profiles; figure (b) shows the detail of the region close to the wall and indicated by a gray box in (a).

subtraction, an initial interrogation grid of 96 px × 96 px, and a refinement step of 24 px × 24 px with four passes. Vector validation was performed based on peak-to-peak ratio $Q < 1.5$ and NMF with a 3×3 kernel.

In the TBL the only non-zero component of the time-averaged velocity is the component parallel to wall, $\langle u \rangle$, which has a universal profile (Landau and Lifshitz 1959). In the immediate vicinity of the wall we have the viscous sublayer in which the mean velocity increases linearly with the distance from the wall, y; whereas at a sufficiently large distance the mean velocity has a logarithmic profile (law-of-the-wall region or log-law region). If we introduce the friction velocity, $u_\tau = \sqrt{\tau_w/\rho_f}$ (where τ_w is the shear-stress at the wall and ρ_f the density of the fluid), we can define the dimensionless mean velocity, $u^+ = \langle u \rangle / u_\tau$, and the dimensionless, wall-normal coordinate, $y^+ = u_\tau y/\nu$ (where ν is the kinematic viscosity), and write (Pope 2000)

$$u^+ = \begin{cases} y^+ & y^+ < 5, \\ \frac{1}{\kappa} \log y^+ + C^+ & 30 < y^+ < 10^2 - 10^3, \end{cases} \tag{12}$$

where κ is the von Kármán constant and C^+ another constant of integration. Equation (12) gives the mean velocity profile in the viscous sublayer and in the log-law region, which are relied by the buffer sublayer. Notice that, even if no analytical solution is available for the viscous sublayer, the mean velocity profile is universal also in this region (Landau and Lifshitz 1959, Pope 2000) and can be obtained, for instance, by direct numerical simulation (DNS).

The analytical solutions of the dimensionless mean velocity profile in the viscous sublayer and log-law region (equation (12)) are plotted in figure 8(a) together with the profile obtained from a DNS solution of the TBL (Graham *et al* 2016) that serve as reference solution in the buffer sublayer. These reference profiles are compared with the velocity profile estimated by `WIDIM` and by the CNN trained with the `num` and the `rnd3` datasets. Notice that DNS solution has a Reynolds number which differs from the one of our TBL experiments. This explains the differences between the DNS solution and the reconstructed velocity profile: due to the lower Reynolds number, in the DNS, the log-low region extends further into the outer layer than in the experiments, in which the departure from the logarithmic profile occurs closer to the wall. Therefore, all methods provide fairly accurate velocity profiles at large distance from the wall (i.e. in the log-law region and beyond).

Close to the wall, instead, the velocity reconstructed by $\mathcal{N}_{num}$ is largely affected by bias errors and deviates from the reference already in the buffer sublayer, at a distance of about $y^+ = 10$. Notice that, for the current imaging parameters and flow conditions, $y^+ = 1$ unit corresponds to $y = 5$ px. On the contrary, the CNN trained on kinematic datasets performs better than `WIDIM` and estimates very well the velocity profile far into the viscous sublayer. This is demonstrated in figure 8(b) that shows compares the velocity reconstructed by $\mathcal{N}_{rnd1}$, $\mathcal{N}_{rnd2}$, and $\mathcal{N}_{rnd3}$ with the analytical and DNS solutions in the viscous sublayer. Even with the smallest training dataset (`rnd1`), the higher spatial dynamic range allows the CNN to reconstruct the velocity profile down to $y^+ = 1$. Increasing the size of the kinematic training dataset allows the reconstruction of an accurate velocity profile down to the wall. Indeed, $\mathcal{N}_{rnd3}$ is able to satisfactorily match the reference data at $y^+ = 0.2$, which corresponds to the pixel adjacent to the wall. As cross-correlation methods require a minimum interrogation window (with a typical size of 24 px or 32 px), `WIDIM` is capable of resolving the velocity profile only down to $y^+ = 3$.

For comparison, the TBL profiles reconstructed in previous studies that employed CNN architectures (i.e. `PIV-DCNN` in (Lee *et al* 2017) and `PIV-liteflownet-en` in (Cai *et al* 2020)) are also plotted in figure 8(a). They are able to match the analytical and DNS solutions only down to a dimensionless distance from the wall $y^+ = 10$, while they significantly deviate from the reference at smaller distances. Although these velocity estimates have been obtained from different

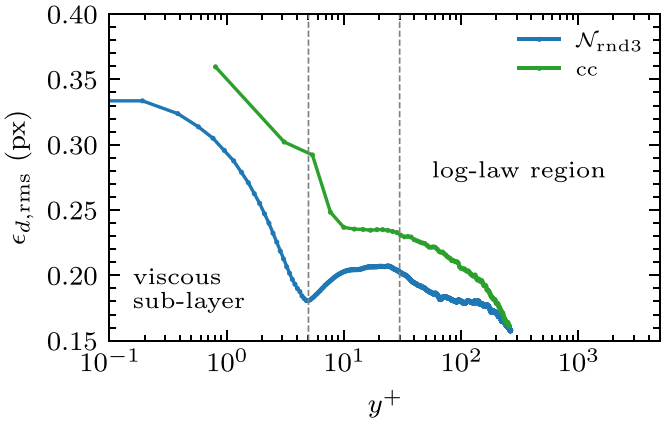

Figure 9. Profiles of the RMS disparity error, $\epsilon_{\mathrm{d,rms}}$ (calculated by equation (6) in (Sciacchitano *et al* 2013)): $\mathcal{N}_{\mathrm{rnd3}}$ (`rnd3`) (blue) vs. `WIDIM` (green).

experimental data (Lee *et al* 2017, Cai *et al* 2019b, 2020), which may affect the accuracy, the comparison with our results suggests that CNN trained on kinematic dataset provides unprecedented and superior performance in estimating the mean velocity profile in the TBL.

For the velocity vector fields reconstructed by `WIDIM` and by the CNN trained on `rnd3` (i.e. $\mathcal{N}_{\mathrm{rnd3}}$), we perform uncertainty quantification through image matching (Sciacchitano *et al* 2013), which allows an *a-posteriori* assessment of the accuracy by matching (i.e. warping) the particle images using the estimated velocity field. The method obtain two predictions of the particle-position image at the mid-separation time, $\mathrm{d}t/2$, by symmetrically transforming forward the image at the old time and backward the image at the new time. Then, the particle centers are detected in both transformed images and the residual disparity error due to an imperfect superimposition of the particle-image positions is calculated. Although this requires image manipulations (such as warping, peak detection, and peak fitting) that may bias the assessment, this uncertainty quantification method allows us to isolate the errors arising from the post-processing of the images. We refer to the original paper Sciacchitano *et al* (2013) for more details about the method. The profiles of the root-mean-squared disparity error, $\epsilon_{\mathrm{d,rms}}$ (Sciacchitano *et al* 2013, equation (6)), are plotted in figure 9. `WIDIM` is affected by higher error levels in all regions. Especially near the wall, $\mathcal{N}_{\mathrm{rnd3}}$ allows greater accuracy, with a minimum error at the beginning of the viscous sublayer. This demonstrates that velocimetry techniques based on CNNs are capable of achieving better accuracy than classic cross-correlation methods.

3.3. Cylinder wake

In the TBL, the time-averaged flow is primarily one-dimensional and oriented along the streamwise direction. In order to assess the performance of different velocity-reconstruction methods and investigate the effects of the different training datasets in the presence of shear flows, we perform a cylinder wake experiment with the same imaging hardware and tracers as described in section 3.2. Two different datasets were recorded at wind speeds $u_\infty = 0.47\,\mathrm{m\,s^{-1}}$ and $u_\infty = 1.07\,\mathrm{m\,s^{-1}}$ for a cylinder with a diameter $D = 6\,\mathrm{mm}$. With a magnification factor of $15\,\mathrm{px\,mm^{-1}}$ and an acquisition frequency $f = 1\,\mathrm{kHz}$ we obtain a mean pixel displacement of 7 and 14 px. Conventional PIV analysis is performed by means of `WIDIM` including background subtraction, an initial interrogation grid of $64\,\mathrm{px} \times 64\,\mathrm{px}$, and a refinement step of $32\,\mathrm{px} \times 32\,\mathrm{px}$ with 50% overlap. Vector validation was performed based on peak-to-peak ratio $Q < 1.25$ and a NMF with 3×3 kernel.

To compare the velocity fields reconstructed by $\mathcal{N}_{\mathrm{num}}$, $\mathcal{N}_{\mathrm{rnd2}}$ and `WIDIM` we consider the dimensionless vorticity, $\omega D/u_\infty$, which is shown in figure 10 for the two bulk wind velocities, $u_\infty = 0.47\,\mathrm{m\,s^{-1}}$ and $u_\infty = 1.07\,\mathrm{m\,s^{-1}}$, which correspond to the Reynolds numbers $\mathrm{Re} = 189$ and $\mathrm{Re} = 431$, respectively. All three velocity-reconstruction methods correctly reproduce the von Kármán vortex street that dominates these flow regimes. However, the vorticity field obtained by $\mathcal{N}_{\mathrm{num}}$ exhibits significant noise and artifacts, especially at $u_\infty = 1.07\,\mathrm{m\,s^{-1}}$ (figure 10(b)). By comparing the vorticity obtained with $\mathcal{N}_{\mathrm{rnd3}}$ and `WIDIM` we observe a better agreement between the two scalar fields, but the CNN trained on the kinematic dataset achieves pixel-level resolution and proves much more capable of reconstructing the flow in the presence of high vorticity gradients, such as in the regions 'A' and 'B' highlighted in figures 10(b), (d) and (f). Notice that `WIDIM` fails in characterizing the velocity and vorticity fields in the vicinity of the cylinder (the white regions in figures 10(e) and (f)).

In figure 11, we plot the longitudinal profiles of the instantaneous normalized vorticity extracted at $y/D = 0$. We observe that $\mathcal{N}_{\mathrm{num}}$ is affected by noise at $u_\infty = 0.47\,\mathrm{m\,s^{-1}}$ and, at $u_\infty = 1.07\,\mathrm{m\,s^{-1}}$, significantly underestimates the vorticity peaks reconstructed at $x/D = 2$, 3.5 and 5.75 by $\mathcal{N}_{\mathrm{rnd3}}$ and `WIDIM`. This indicates the limited capability of $\mathcal{N}_{\mathrm{num}}$ to reconstruct strong flow gradients. In contrast, the kinematic training dataset `rnd3` allows the CNN to be as reliable as `WIDIM` away from the cylinder, but offers the possibility of reconstructing information also close to the cylinder (at $x/D < 2$) where shear effects are the largest and the vorticity estimated by `WIDIM` is unreliable.

As for the TBL case, we also quantify the PIV uncertainty by calculating the disparity error through image matching (Sciacchitano *et al* 2013). Figure 12 shows the spatial distribution of the norm of disparity error, $|\epsilon_{\mathrm{d}}|$ (Sciacchitano *et al* 2013, equation (4)), as well as the norm of disparity error normalized by the local displacement, i.e. $|\epsilon_{\mathrm{d}}/\mathrm{d}\boldsymbol{s}|$. The error is calculated on an image pair using the output velocity field reconstructed by $\mathcal{N}_{\mathrm{num}}$, $\mathcal{N}_{\mathrm{rnd3}}$ and `WIDIM` at $u_\infty = 1.07\,\mathrm{m\,s^{-1}}$. While $\mathcal{N}_{\mathrm{num}}$ yields considerably higher disparity errors than $\mathcal{N}_{\mathrm{rnd3}}$ and `WIDIM`, the latter two show comparable error distributions except in the vicinity of the near-wake of the cylinder where `WIDIM` shows much larger errors than $\mathcal{N}_{\mathrm{rnd3}}$. In far-wake regions, the majority of the errors lie below 0.4 px for both methods (figures 12(c) and (e)). We remark that we are considering an instantaneous reconstruction of the flow field, and that, if a disparity error of 0.4 px may seem fairly large, the relative error of $\mathcal{N}_{\mathrm{rnd3}}$ and `WIDIM` is mostly below 3%, except

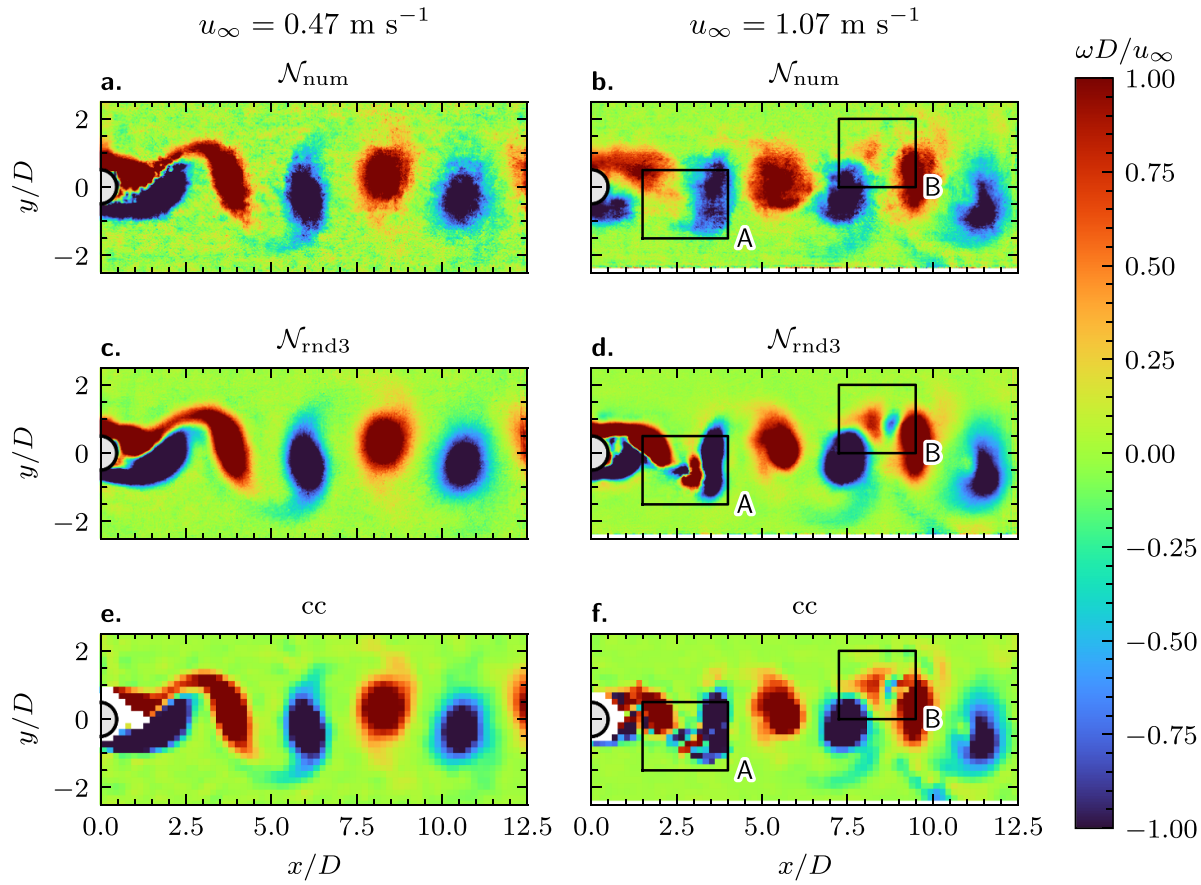

Figure 10. Normalized vorticity, $\omega D/u_\infty$, of the flow past a cylinder of diameter $D = 6$ mm for $u_\infty = 0.47\,\text{m s}^{-1}$ (left) and $u_\infty = 1.07\,\text{m s}^{-1}$ (right). From top to bottom the results obtained by (a), (b) $\mathcal{N}_{\text{num}}$; (c), (d) $\mathcal{N}_{\text{rnd3}}$; (e), (f) `WIDIM`.

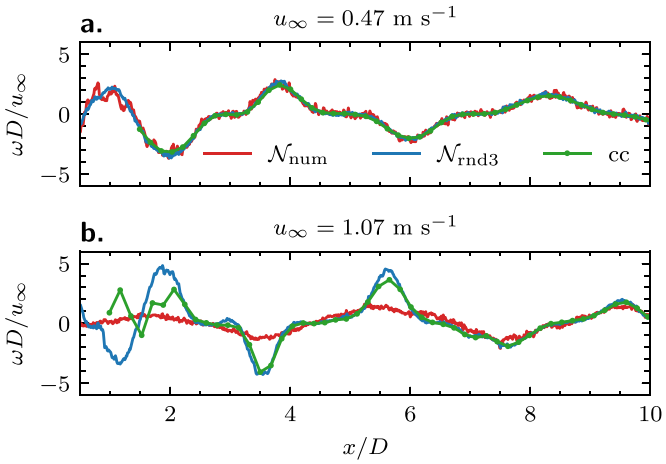

Figure 11. Longitudinal profile of the normalized vorticity, $\omega D/u_\infty$, at $y = 0$ for the velocity fields obtained by $\mathcal{N}_{\text{num}}$, $\mathcal{N}_{\text{rnd3}}$, and `WIDIM`: (a) $u_\infty = 0.47\,\text{m s}^{-1}$ and (b) $u_\infty = 1.07\,\text{m s}^{-1}$.

in high-shear and near-wake regions with strong flow circulation (figures 12(d) and (f)).

Finally, we analyze the temporal characteristics of the velocity fields to assess whether they are consistent or affected by artifacts. We recall that the output resolution of the CNN is 15 vectors mm^{-1}, whereas it is only about 1 vector mm^{-1} for `WIDIM`. To understand whether the higher spatial discretization comes with increased measurement noise on the time axis, in figure 13 we plot the pointwise spectra of the velocity magnitude at $(x, y) = (5D, 0)$ for $u_\infty = 1.07\,\text{m s}^{-1}$. Shown are the spectra corresponding to the velocity estimated by $\mathcal{N}_{\text{num}}$, $\mathcal{N}_{\text{rnd3}}$ and `WIDIM`, presented as a function of both the frequency, f and the Strouhal number $\text{St} = fD/u_\infty$. The plot shows that $\mathcal{N}_{\text{num}}$ is unable to detect the same amplitude values of the

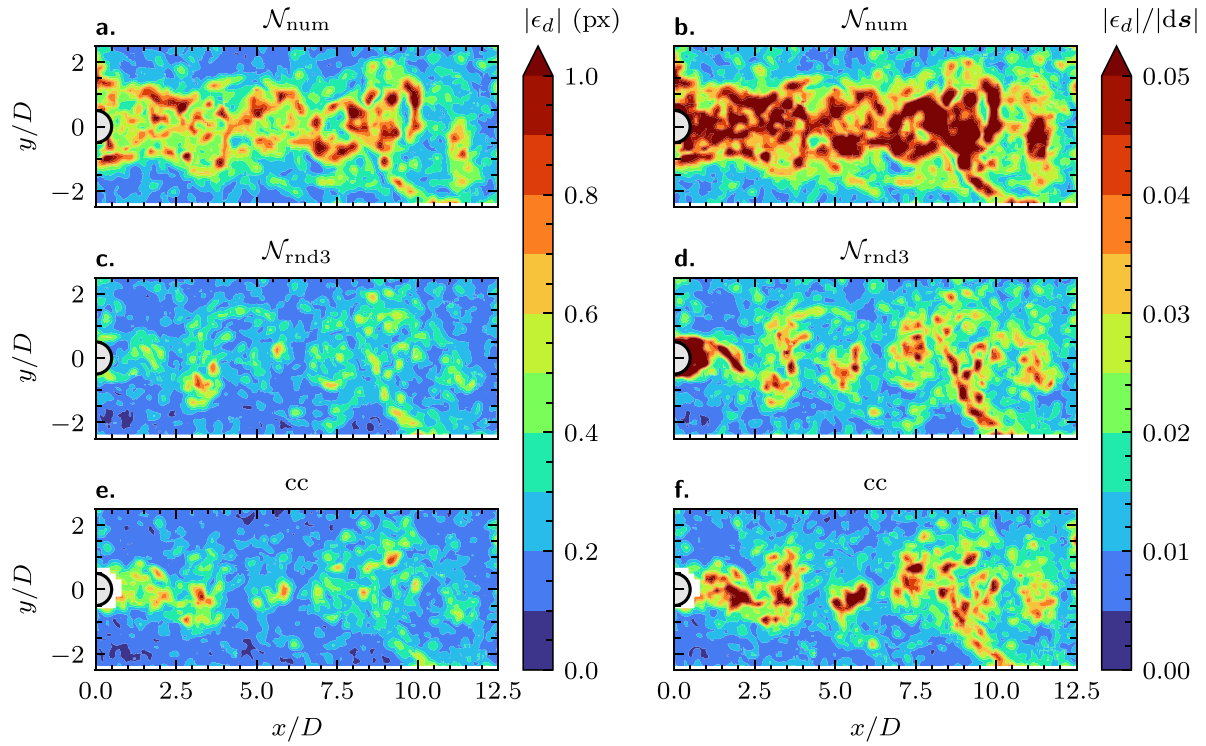

Figure 12. Disparity maps of the cylinder wake at $u_\infty = 1.07\,\mathrm{m\,s^{-1}}$ for the velocity estimated by $\mathcal{N}_{\text{num}}$ (top), $\mathcal{N}_{\text{rnd3}}$ (middle), and `WIDIM` (bottom). (a), (c), (e) The magnitude of the absolute disparity error, $|\epsilon_{\text{d}}|$ (px) (Sciacchitano *et al* 2013, equation (4)); (b), (d), (f) the normalized magnitude of the disparity error $|\epsilon_{\text{d}}/\mathrm{d}\boldsymbol{s}|$.

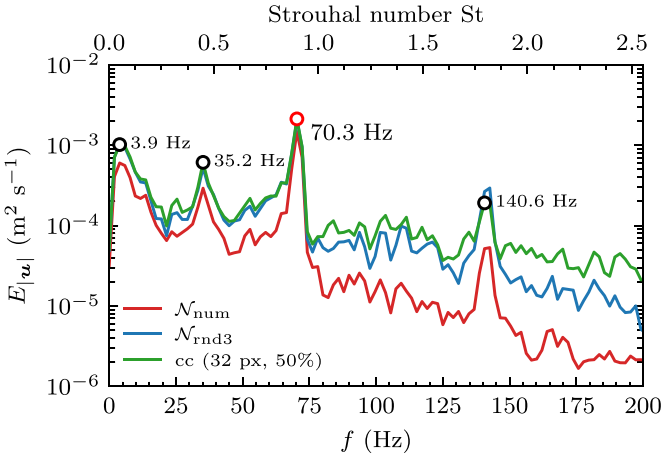

Figure 13. The pointwise energy spectra of velocity norm $E_{|\boldsymbol{u}|}$ (m^2 s^{-2}) at $(x,y) = (5D, 0)$ for $u_\infty = 1.07\,\mathrm{m\,s^{-1}}$. Shown are the spectra corresponding to the velocities estimated by $\mathcal{N}_{\text{num}}$ (red), $\mathcal{N}_{\text{rnd3}}$ (blue) and `WIDIM` (green), presented as a function of both the frequency, f, and the Strouhal number $\mathrm{St} = fD/u_\infty$.

peaks at 3.9, 35.2 and 140.6 Hz and displays a strong decay of the signal after 75 Hz with respect to $\mathcal{N}_{\text{rnd3}}$ and `WIDIM`. On the contrary, $\mathcal{N}_{\text{rnd3}}$ and `WIDIM` show a consistent behavior and clearly identify four distinct peaks. This demonstrates that the temporal characteristics of the flow are consistently reconstructed by the CNN if the training is done on sufficiently large kinematic datasets.

4. Conclusions

To improve the reconstruction of the velocity from PIV data, we train a state-of-the-art CNN on datasets that include synthetic PIV-like images generated by means of random displacement fields. Datasets of this kind allow us to teach the CNN about the kinematic relationship that links position and velocity (or displacement) and is at the core of PIV techniques.

These kinematic training datasets offer the advantage that their size can be easily increased at a much lower computational cost than training datasets composed of numerical solutions to flow problems, reducing the risk that a lack of sufficient training data may lead to overfitting. Most importantly, our performance assessment demonstrates that kinematic training is necessary to improve the accuracy of the velocity fields estimated by the CNN and allows a faithful reconstruction of small-scale flow features that cannot be recovered by CNN trained on datasets generated only from DNS of the flow problem. The latter may require an impracticable number of realizations to allow the network to recover small-scale features. Although they may teach the network to recognize large-scale dynamic structures, PIV data are typically acquired at a sufficiently small separation time and large tracer-particle density to make kinematic information sufficient to reconstruct accurate velocity fields.

Our synthetic and experimental test cases (including TBL and cylinder-wake experiments) demonstrate that kinematic

training allows the CNN to estimate displacement and velocity fields more accurately than state-of-the-art post-processing techniques based on cross-correlation analysis (such as `WIDIM`). The better performance of the CNN in processing experimental data is confirmed by the disparity error analysis and by the observation that `WIDIM` is unable to reconstruct the velocity in regions characterized by large shear (e.g. close to the cylinder in the near-wake). These results are unprecedented because previous studies, employing machine learning architectures trained only on synthetic images obtained from DNS solutions, were unable to match the accuracy of iterative 2D cross-correlation with window deformation.

It is worth noting that CNNs offer two additional advantages. First, they allow the reconstruction of velocity fields with pixel-level resolution, which is not attainable by cross-correlation methods as they rely on the use of relatively large interrogation windows. Second, they are more robust to variations of the experimental parameters (e.g. tracer-particle density and size, and image noise), paving the way for applications to experimental data with challenging imaging conditions or to a relaxation of the technical requirements on the imaging hardware (e.g. on camera resolution).

The results of the kinematic training are very promising and suggest that the impact of the parameters used to generate the dataset (e.g. filter size, scale factor, particle diameter, seeding density, and particle loss) should be further investigated to optimize the training strategy. We envision that training CNN on well-designed combinations of kinematic and dynamic datasets can push PIV methods far beyond the current limits set in terms of separation time, seeding density, and hardware technical requirements. For instance, kinematic training may be used to pre-train highly complex networks to efficiently reconstruct small-scale features; then the network may be fine-tuned on dynamic flow datasets (generated by DNS) in order to learn to recognize complex dynamic features. At the same time, kinematic training can be applied to other network architectures that are specifically designed for PIV data and optimized to further reduce the number of network parameters. Such lightweight networks may be integrated onto experimental hardware to enable, for instance, on-board processing on the cameras and real-time 3D PIV or 3D particle tracking velocimetry (PTV) analysis.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

We would like to thank Beat Margelisch, Roger Vonbank, and Stephan Kunz for their support at Empa for the experimental setup. We acknowledge access to Piz Daint at the Swiss National Supercomputing Centre, Switzerland, under Empa's share with the Project ID `em13`. Furthermore, we thank Dr Hossein Gorji for fruitful discussions.

ORCID iDs

Lento Manickathan https://orcid.org/0000-0002-0425-4679
Claudio Mucignat https://orcid.org/0000-0003-2676-2815
Ivan Lunati https://orcid.org/0000-0002-3205-7429

References

Adrian R J 2005 Twenty years of particle image velocimetry *Exp. Fluids* **39** 159–69
Alom M Z, Taha T M, Yakopcic C, Westberg S, Sidike P, Nasrin M S, Esesn B C V, Awwal A A S and Asari V K 2018 The history began from alexnet: a comprehensive survey on deep learning approaches (arXiv:1803.01164)
Armellini A, Mucignat C, Casarsa L and Giannattasio P 2012 Flow field investigations in rotating facilities by means of stationary PIV systems *Meas. Sci. Technol.* **23** 025302
Astarita T 2007 Analysis of weighting windows for image deformation methods in PIV *Exp. Fluids* **43** 859–72
Cai S, Liang J, Gao Q, Xu C and Wei R 2020 Particle image velocimetry based on a deep learning motion estimator *IEEE Trans. Instrum. Meas.* **69** 3538–54
Cai S, Liang J, Zhou S, Gao Q, Xu C, Wei R, Wereley S and Kwon J-S 2019a Deep-PIV : a new framework of PIV using deep learning techniques *ISPIV 2019 (MüNich, Germany)*
Cai S, Zhou S, Xu C and Gao Q 2019b Dense motion estimation of particle images via a convolutional neural network *Exp. Fluids* **60** 73
Chen P-H, Yen J-Y and Chen J-L 1998 An artificial neural network for double exposure PIV image analysis *Exp. Fluids* **24** 373–4
Dosovitskiy A, Fischer P, Ilg E, Häusser P, Hazirbas C, Golkov V, Smagt P V D, Cremers D and Brox T 2015 FlowNet: learning optical flow with convolutional networks *2015 IEEE Int. Conf. on Computer Vision (ICCV)* pp 2758–66
Gao Q, Lin H, Tu H, Zhu H, Wei R, Zhang G and Shao X 2021 A robust single-pixel particle image velocimetry based on fully convolutional networks with cross-correlation embedded *Phys. Fluids* **33** 127125
Graham J *et al* 2016 A web services accessible database of turbulent channel flow and its use for testing a new integral wall model for LES *J. Turbul.* **17** 181–215
Hassan Y A and Philip O G 1997 A new artificial neural network tracking technique for particle image velocimetry *Exp. Fluids* **23** 145–54
He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*
Hui T-W, Tang X and Loy C C 2018 LiteFlowNet: a lightweight convolutional neural network for optical flow estimation *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* pp 8981–9
Hui T-W, Tang X and Loy C C 2021 A lightweight optical flow CNN—revisiting data fidelity and regularization *IEEE Trans. Pattern Anal. Mach. Intell.* **43** 2555–69
Hur J and Roth S 2019 Iterative residual refinement for joint optical flow and occlusion estimation *2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* vol 2019 pp 5747–56
Hur J and Roth S 2020 Optical flow estimation in the deep learning age *Modelling Human Motion* (Cham: Springer International Publishing) pp 119–40
Ilg E, Mayer N, Saikia T, Keuper M, Dosovitskiy A and Brox T 2017 FlowNet 2.0: evolution of optical flow estimation with

deep networks *2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* vol 2017 (IEEE) pp 1647–55
Klein M, Sadiki A and Janicka J 2003 A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations *J. Comput. Phys.* **186** 652–65
Krizhevsky A, Sutskever I and Hinton G E 2012 ImageNet classification with deep convolutional neural networks *Advances in Neural Information Processing Systems* vol 25, ed F Pereira, C Burges, L Bottou and K Weinberger (Curran Associates, Inc.) (available at: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf)
Labonté G 2000 On a neural network that performs an enhanced nearest-neighbour matching *Pattern Anal. Appl.* **3** 267–78
Lagemann C, Lagemann K, Mukherjee S and Schröder W 2021 Deep recurrent optical flow learning for particle image velocimetry data *Nat. Mach. Intell.* **3** 641–51
Landau L and Lifshitz E 1959 *Fluid Mechanics* (Oxford: Pergamon)
Lecordier B and Westerweel J 2004 The EUROPIV synthetic image generator (S.I.G.) *Particle Image Velocimetry: Recent Improvements* ed M Stanislas, J Westerweel and J Kompenhans (Berlin: Springer) pp 145–61
LeCun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W and Jackel L 1989 Handwritten digital recognition with a back-propagation network *Advances in Neural Information Processing Systems* vol 2, ed D Touretzky (Morgan-Kaufmann)
Lee Y, Yang H and Yin Z 2017 PIV-DCNN: cascaded deep convolutional neural networks for particle image velocimetry *Exp. Fluids* **58** 171
Li Y, Perlman E, Wan M, Yang Y, Meneveau C, Burns R, Chen S, Szalay A and Eyink G 2008 A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence *J. Turbul.* **9** 1–29
Liu P, Lyu M, King I and Xu J 2019 SelFlow: self-supervised learning of optical flow *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* vol 2019 pp 4566–75
Pope S B 2000 *Turbulent Flows* (Cambridge: Cambridge University Press)
Rabault J, Kolaas J and Jensen A 2017 Performing particle image velocimetry using artificial neural networks: a proof-of-concept *Meas. Sci. Technol.* **28** 125301
Raffel M, Willert C E, Scarano F, Kähler C J, Wereley S T and Kompenhans J 2018 *Particle Image Velocimetry: A Practical Guide* (Berlin: Springer)
Scarano F 2001 Iterative image deformation methods in PIV *Meas. Sci. Technol.* **13** R1–R19
Schrijer F F J and Scarano F 2008 Effect of predictor–corrector filtering on the stability and spatial resolution of iterative PIV interrogation *Exp. Fluids* **45** 927–41
Sciacchitano A, Wieneke B and Scarano F 2013 PIV uncertainty quantification by image matching *Meas. Sci. Technol.* **24** 045302
Smirnov A, Shi S and Celik I 2001 Random flow generation technique for large eddy simulations and particle-dynamics modeling *J. Fluids Eng. Trans. ASME* **123** 359–71
Sun D, Yang X, Liu M-Y and Kautz J 2017a PWC-Net: CNNs for optical flow using pyramid, warping and cost volume *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 8934–43
Sun D, Yang X, Liu M-Y and Kautz J 2017 PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume (arXiv:1709.02371)
Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*
Teed Z and Deng J 2020 RAFT: recurrent all-pairs field transforms for optical flow *European Conf. on Computer Vision (Lecture Notes in Computer Science)* vol 12347 pp 402–19
Westerweel J and Scarano F 2005 Universal outlier detection for PIV data *Exp. Fluids* **39** 1096–100
Willert C E and Gharib M 1991 Digital particle image velocimetry *Exp. Fluids* **10** 181–93
Yu C, Bi X, Fan Y, Han Y and Kuai Y 2021 LightPIVNet: an effective convolutional neural network for particle image velocimetry *IEEE Trans. Instrum. Meas.* **70** 1–15