

機械学習に基づくデータ拡張による PIV の精度向上

Improvement of PIV by data augmentation based on machine learning

- 森本 将生, 慶大, 横浜市港北区日吉 3-14-1
- 深見 開, 慶大院, 横浜市港北区日吉 3-14-1
- 長谷川 一登, 慶大院, 横浜市港北区日吉 3-14-1
- 村田 高彬, 慶大院, 横浜市港北区日吉 3-14-1
- 村上 光, 慶大, 横浜市港北区日吉 3-14-1
- 深淵 康二, 慶大, 横浜市港北区日吉 3-14-1, E-mail : fukagata@mech.keio.ac.jp

Masaki MORIMOTO, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522

Kai FUKAMI, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522

Kazuto HASEGAWA, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522

Takaaki MURATA, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522

Hikaru MURAKAMI, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522

Koji FUKAGATA, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522

We use a supervised machine learning to augment velocity fields around a square cylinder from particle images obtained by the PIV measurement which has some regions of deteriorated particle images. We utilize artificial particle images (API) based on the histogram of experimental particle magnitudes as the input data. The solution of models is obtained from a direct numerical simulation (DNS) at $Re_D = 300$. Two cases are considered as the input data types: API without and with lacked portions, respectively. The lacked area is given following the exact experimental situation. The estimated velocity fields from both full and lacked APIs are in great agreement with the reference DNS data in terms of statistical assessments. At last, we also apply the machine-learned model to experimental images and show its applicability to various situations in experimental fluid dynamics.

1. 背景及び目的

流体の計測技術として、熱線流速計⁽¹⁾やシュリーレン法⁽²⁾など様々な手法が提案されているが、その中でも PIV 計測⁽³⁾は流れ場を可視化することで直感的に物理現象を理解できることから、近年一般的な計測方法として幅広く利用されている。PIV 計測の問題点の一つに、流れ場に追従するトレーサー粒子を撮像しにくい領域が存在することが挙げられる。模型の表面近傍、もしくは模型の陰になる部分においてレーザー光の乱反射や、レーザー光の強度不足などから十分な輝度が得られない場合がある。また、模型の形状によっては、レーザー光が局所的に集中し白くぼやけてしまうハレーションや、逆に影となる部分などが生じる⁽⁴⁾。このため、これらの領域において正しくトレーサー粒子を撮像できず、変位ベクトルの推定が困難になる。このような問題を解決・緩和するために幾つかの手法が提案されてきた。蛍光粒子を用いた Lazer Induced Fluorescence (LIF) 法はその代表的な手法の一つである⁽⁵⁾。特殊な蛍光塗料をトレーサー粒子に着色することで、照射するレーザー光と異なる波長の光を励起させ、励起光のみを観察することで照射光の乱反射等の影響を大幅に抑制することができる。しかし、一般的に LIF 法において使用されるローダミン系の染料が発ガン性物質を含んでいることなどから、LIF 法は気相への適用が困難であることなどのデメリットがある。他の手法として、模型への黒色塗料もしくは蛍光塗料の塗装や、反射防止コート (MgF_2) を施したガラスの使用などが提案されている⁽⁴⁾が、いずれもレーザー光の乱反射やハレーションを完全に防止することは困難であるとされている。このように、PIV 計測におけるレーザー光の乱反射やハレーションによる測定精度への影響を完全に抑制することは現状困難であり、新たな計測手法もしくはデータの補完手法の開発が期待されている。一方近年、大量のデータから特徴を抽出することを得意とする機械学習と流体力学の親和性に大きな注目が集まっている⁽⁶⁾。Ling *et al.*⁽⁷⁾はレイノルズ平均ナビエストークス計算のための機械学習を用いた乱流モデリング

を提案し、不変量が埋め込まれた学習モデルを用いることで、従来の多層パーセプトロンよりも予測精度を高められることを報告した。また、乱流場時系列予測の入口乱流生成器への応用⁽⁸⁾や、低解像度の流れ場のデータから高解像度のデータを生成する超解像手法⁽⁹⁾、さらには流れ場の非線形モードの可視化および抽出⁽¹⁰⁾など、機械学習は流体力学分野の様々な需要に対し有効利用されている。PIV においても例外ではなく、Chen *et al.*⁽¹¹⁾が実験によって得られた一様流中における粒子画像から速度場を推定する、全結合型のニューラルネットワークを構築し、推定された平均速度が実験の結果とよく一致することを示した。その後 Rebault *et al.*⁽¹²⁾がランダムに設定された速度場において流れに追従する粒子画像を擬似的に作成し、粒子画像から速度場を推定する Convolutional neural network を構築した。さらに Cai *et al.*⁽¹³⁾が、円柱周り流れおよび一様等方性乱流に対して同様に粒子画像を作成し、FlowNetS architecture⁽¹⁴⁾を用いて擬似的に作成された粒子画像から速度場を再構築する手法を提案している。また同文献において実験データにも提案手法を適用しており、その応用可能性を示唆している。このように流体力学分野において機械学習は、PIV を含む多くの問題への適用が提案されており、PIV の領域においては精度、計算コストの両面から従来の測定方法を上回ることが期待されている。

本研究では、実験における角柱周り流れの粒子画像に対してオートエンコーダ型 Convolutional neural network を適用し、粒子を撮像することが困難な領域を補完する機械学習モデルを構築することで、欠損のない速度場を得ることを目的とする。角柱周りでは、模型表面におけるレーザー光の乱反射や、角部によるハレーション等が顕著に表れるため、角柱周りの流れ場を対象とした。3 節において擬似粒子画像を用いたモデルの訓練手法を示し、4 節において擬似粒子画像と実験の粒子画像を入力した際の機械学習モデルの推定結果について報告する。

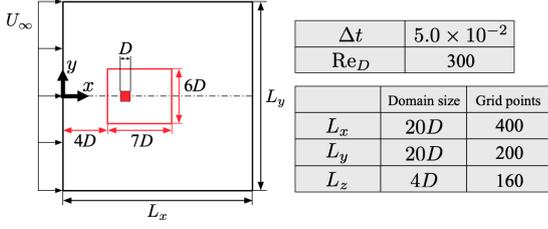


Fig. 1: Computational domain and condition. Note that three-dimensional DNS is considered in the present study.

2. 理論

2.1 訓練データ

本研究では、角柱周りの三次元流れを対象とする。訓練データは、直接数値シミュレーション (DNS) を用いて作成する。支配方程式は、連続の式およびナビエ・ストークス方程式

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \frac{1}{Re_D} \nabla^2 \mathbf{u} \quad (2)$$

である。支配方程式中の全ての物理量は動粘性係数 ν^* 、主流速度 U_∞^* 、円柱直径 D^* によって無次元化されている。なお、 $(\cdot)^*$ は有次元数を表す。本研究ではレイノルズ数 Re_D を

$$Re_D = \frac{U_\infty^* D^*}{\nu^*} = 300 \quad (3)$$

としてシミュレーションを行う。Figure 1 に計算領域および計算条件を示す。計算領域は無次元化された代表長さ D を用いて $(L_x \times L_y \times L_z) = (20D \times 20D \times 4D)$ 、時間刻み幅 Δt は、 $\Delta t = 5.0 \times 10^{-2}$ とする。本研究では、計算領域中の赤枠で囲われた領域 $(7D \times 6D \times 0.5D)$ を訓練データとして抽出する。抽出した流れ場の格子数は $(N_x^{\#} \times N_y^{\#} \times N_z^{\#}) = (140 \times 120 \times 20)$ である。 $Re_D = 300$ において角柱周りでは、三次元的な渦構造が形成されるため、 z 方向に 20 枚の $x-y$ 断面を使用する。各 $x-y$ 断面における主流方向および主流に垂直な方向の速度場 500 タイムステップ分を訓練データとする。以上より訓練データに使用するスナップショットは 10000 枚とする。

2.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN)⁽¹⁵⁾ は、画像認識において用いられる機械学習手法であり、主に畳み込み層とプーリング層から構成される。畳み込み層では、画像のイメージに任意のサイズのフィルターを重ねあわせ、重なる要素同士の積の和を出力する畳み込み演算

$$u_{ijm} = \sum_{k=0}^{K-1} \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p, j+q, k} h_{pqkm} + b_m, \quad (4)$$

を行う。畳み込み層に入力される $(W \times W \times K)$ のサイズのデータにおけるインデックスを (i, j, m) と表し、点 (i, j, m) における入力画像の値を x_{ijm} と表す。 h_{pqkm} は畳み込み層における m 個の $(H \times H \times K)$ のフィルターであり、 b_m はフィルタごとに存在するバイアスを表す。 m 種類のフィルターを用いて畳み込み演算を行うことで画像の特徴パターンを抽出する。 u_{ijm} は畳み込み演算により出力される二次元データの (i, j, m) 成分である。畳み込み層の出力 z_{ijm} は、活性化関数 $f(\cdot)$ を用いて、

$$z_{ijm} = f(u_{ijm}) \quad (5)$$

と表される。通常、活性化関数として単調増加の非線形関数を用いられる。また、プーリング層において、入力データ中の任意の領域内に対して平均値や最大値などの代表値を取り出す操作を行う。これにより、モデルの位置に対する感度を低下させ、入力に対するモデルの汎化性能を高めるほか、入力画像のノイズを除去する効果も見込まれる。

また、CNN を用いて入力のデータの次元を低次元化し、再び入力の次元に復元する構造を持つネットワークを CNN オートエンコーダ⁽¹⁶⁾ と呼ぶ。CNN オートエンコーダは入力と出力が等しくなるように訓練される。本研究では、CNN オートエンコーダのネットワークを参考に、畳み込み層とプーリング層を用いて入力イメージを低次元化した後、畳み込み層とアップサンプリング層を用いて出力データの次元が入力データの次元と一致するまで復元する、オートエンコーダ型 CNN を構築する。Figure 2 に本研究で使用したオートエンコーダ型 CNN の概略図を示す。本研究で使用するモデルは入力と出力のデータが異なるため、CNN オートエンコーダとは区別される、オートエンコーダと類似した形状を持つ CNN であることに注意されたい。

3. 訓練手法及びモデル構成

3.1 フローチャート

本研究のフローチャートを Figure 3 に示す。まず第一段階として、粒子がランダムに配置された擬似的な粒子画像 ζ を作成する。擬似粒子画像の作成については 3.2 節で説明する。ランダムに配置された各粒子の次のタイムステップにおける位置を DNS データを用いて計算し、擬似粒子画像 \mathbf{q} を得る。また、 \mathbf{q} に対して、実験においてデータを取ることが困難な領域の輝度を 0 に設定し、データを欠損させた擬似粒子画像 $\hat{\mathbf{q}}$ を作成する。

第二段階において、 \mathbf{q} と ζ を足し合わせた画像 $\mathbf{q} + \zeta$ を入力、 \mathbf{q} を得る際に使用した DNS データを出力として、完全な擬似粒子画像から速度場を出力する機械学習モデルを作成する。実験によって得られる粒子画像は、カメラの露光時間が有限であることから、理想的な瞬時場ではなく、粒子が露光時間中に微量移動した軌跡が撮像された画像となる。このため、実験の粒子画像に対応した擬似粒子画像を作成する際、連続する二時刻の擬似粒子画像を重ね合わせて粒子の軌跡を模擬する必要がある。本研究では、 \mathbf{q} と ζ を足し合わせることで、対応するタイムステップにおける速度場の情報をモデルへの入力画像に持たせている。このモデル中における重みの \mathbf{w} は、正解となる DNS データの速度場 $\{u, v\}_{\text{DNS}}$ と、機械学習モデルの出力 $\mathcal{F}(\mathbf{q} + \zeta)$ の差が最小となるように訓練される。つまり、

$$\mathbf{w} = \operatorname{argmin}_{\mathbf{w}} \| \{u, v\}_{\text{DNS}} - \mathcal{F}(\mathbf{q} + \zeta); \mathbf{w} \|_2^2 \quad (6)$$

と定式化される。式 (6) に示す通り、本研究では L_2 誤差ノルムを誤差関数として使用する。

第三段階では、第二段階において使用した機械学習モデルの形状を参考に、新たなモデルを作成する。入力に、データを欠損させた擬似粒子画像 $\hat{\mathbf{q}}$ と ζ を足し合わせた $\hat{\mathbf{q}} + \zeta$ を用い、出力には第二段階のモデルと同様に DNS データを使用する。このモデルによって、データが欠損した粒子画像から欠損のない速度場を出力する機械学習モデル $\hat{\mathcal{F}}$ を構築する。つまり、

$$\hat{\mathbf{w}} = \operatorname{argmin}_{\hat{\mathbf{w}}} \| \{u, v\}_{\text{DNS}} - \hat{\mathcal{F}}(\hat{\mathbf{q}} + \zeta); \hat{\mathbf{w}} \|_2^2 \quad (7)$$

と定式化される。

最終段階において、通常の PIV 計測と同様に実験によって得られた粒子画像を、第三段階において作成した機械学習モデル $\hat{\mathcal{F}}$ に入力し、実験への応用性を検証する。最終段階については、4.2 節で詳細を述べる。

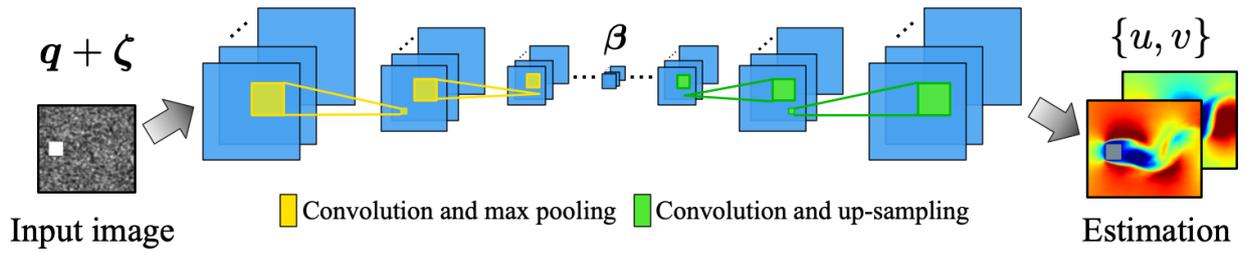


Fig. 2: Illustration of an autoencoder-based convolutional neural network in the present study. The model is trained to output velocity fields $\{u, v\}$ from an artificial particle image $(q + \zeta)$ such that $\{u, v\} = \mathcal{F}\{(q + \zeta); w\}$. In the present model, a low-dimensional latent space β is included inbetween input and output images so that the model can possess the robustness against noise and spatial sensitivity.

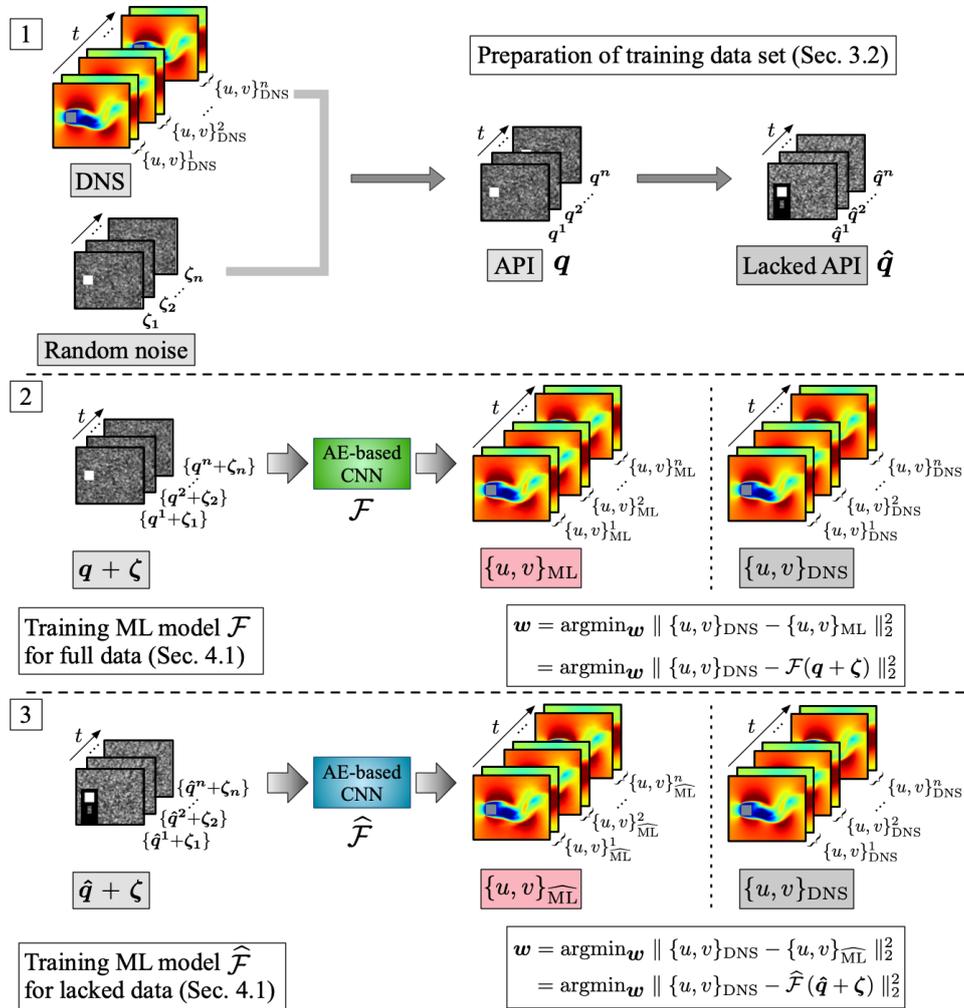


Fig. 3: An illustration of the flow chart of the present study. Step 1: The API (q) is generated by the combination of DNS data u, v with the Gaussian random noise (ζ). Here, the lacked API (\hat{q}) is also prepared. Step2: The machine learning model \mathcal{F} for the full API (q) is trained with the training data set, $q + \zeta$, prepared in the step 1. Step3: The machine learning model $\hat{\mathcal{F}}$ for the lacked API \hat{q} is trained with the training data set, $\hat{q} + \zeta$, prepared in the step 1.

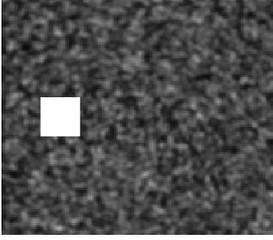


Fig. 4: An example of artificial particle image (API) $\mathbf{q} + \zeta$.

3.2 擬似粒子画像

実験によって得られた粒子画像から速度場を出力するモデルを作成するためには、入力粒子画像に対応する速度場が必要である。PIV 計測によって得られる速度場は前述の通り正しく速度を計測できない領域があることや、様々な誤差が含まれるので、機械学習モデルを作成するためには、正しいと考えられる完全な速度場とそれに対応した粒子画像が必要となる。そのため、本研究では DNS によって計算された速度場を用いて擬似的に実験の粒子画像と同等の輝度分布を持つ粒子画像を作成し、これらをデータセットとして学習を行うことでモデルを訓練する。これによって機械学習モデルは粒子画像から速度場を出力する関数を学習することになる。

Figure 4 に、ランダムに配置された擬似粒子画像 ζ と \mathbf{q} を足し合わせた画像 $\mathbf{q} + \zeta$ の例を示す。これらの擬似粒子画像は、Okamoto *et al.*⁽¹⁷⁾ を参考に以下の式 (8) および (9) を用いて作成する。二次元イメージ内におけるピクセル位置を (x, y) で表し、粒子の位置を (x_p, y_p) 、径を d_p と表す。この時ピクセル位置 (x, y) における輝度 $I(x, y)$ を、

$$I(x, y) = I_0 \exp\left(-\frac{(x - x_p)^2 + (y - y_p)^2}{(d_p/2)^2}\right) \quad (8)$$

と定義する。粒子の位置 (x_p, y_p) を中心にガウス分布を用いて粒子像を再現している。実験によって実際に撮像される粒子像は、粒子自体とその周りのエアリディスクと呼ばれる光の回折によって生じる同心円状の明暗のパターンから構成される。そのため撮像される粒子像の径は、粒子自体の径とエアリディスクの径の和となる。ガウス分布を用いて粒子像を再現することで、実際の粒子像と類似した輪郭のはっきりとしない擬似粒子を作成することができる。また式 (8) 中の係数 I_0 は粒子の z 方向 (壁面垂直方向) 位置依存性による輝度の変化を表す項であり、

$$I_0 = 240 \exp\left(-\frac{z_p^2}{\sigma_l^2}\right) \quad (9)$$

と定義される。ここで z_p は粒子の z 方向位置、 σ_l はレーザーシートの厚みを表す。実験において粒子はレーザーシートを用いて照射されるが、レーザーシートは一般的に数 mm 程度の厚みを持っている。このため粒子はレーザーシートの厚み方向 (壁面垂直方向) の位置に依存して輝度変化することが知られている。これを擬似粒子画像において再現するために I_0 によって、 z 方向の位置依存性をガウス分布を用いて再現している。本研究では気相における角柱周り流れの粒子画像に対してアプローチするため、PIV 計測のための適切な画像処理を行うと粒子像ははっきりとした円ではなく、輪郭がぼやけた形状となる。擬似粒子画像の輪郭を同様に曖昧にするために本研究では式 (9) における係数を 240 の代わりに 0.06 とした。

Tab. 1: Network structure of auto encoder based CNN.

Layer	Data size	Activation
Input	(120,140,1)	
1st Conv2D(5,5,32)	(120,140,32)	ReLU
2nd Conv2D(5,5,32)	(120,140,32)	ReLU
1st Max Pooling	(24,28,32)	
3rd Conv2D(5,5,32)	(24,28,32)	ReLU
4th Conv2D(5,5,32)	(24,28,32)	ReLU
2nd Max Pooling	(12,14,32)	
5th Conv2D(5,5,32)	(12,14,32)	ReLU
6th Conv2D(5,5,32)	(12,14,32)	ReLU
3rd Max Pooling	(6,7,32)	
7th Conv2D(3,3,16)	(6,7,16)	ReLU
8th Conv2D(3,3,16)	(6,7,16)	ReLU
1st Up-sampling	(12,14,16)	
9th Conv2D(5,5,32)	(12,14,32)	ReLU
10th Conv2D(5,5,32)	(12,14,32)	ReLU
2nd Up-sampling	(24,28,32)	
11th Conv2D(5,5,32)	(24,28,32)	ReLU
12th Conv2D(5,5,32)	(24,28,32)	ReLU
3rd Up-sampling	(120,140,32)	
13th Conv2D(5,5,32)	(120,140,32)	ReLU
14th Conv2D(5,5,2)	(120,140,2)	Linear

3.3 ネットワーク構成

Table 1 に本研究で使用したオートエンコーダ型 Convolutional neural network の構造を示す。Conv2D 層 ($h \times h \times m$) では、 $h \times h$ のサイズのフィルタによる畳み込み演算を m 種類のフィルタを用いて行う。14th Conv2D 層 (出力層) 以外の全ての Conv2D 層において活性化演算を行う。本研究のモデルでは、活性化関数に Rectified linear units (ReLU)⁽¹⁸⁾ を使用する。ReLU は勾配消失問題が発生しにくいことなどの利点から多くの機械学習モデルで採用されている。また、Max pooling 層において、 5×5 もしくは 2×2 の小領域に対して最大値のみを抽出することでデータの低次元化を行い、先述の通りモデルの位置に対する感度を低下させることで汎化性能を高め、その後 Up-sampling 層において一点のデータを 5×5 もしくは 2×2 領域に拡大することで低次元場を高次元場へ写像する。

本研究のモデルでは、欠損のない擬似粒子画像 ($\mathbf{q} + \zeta$) を例にとると、入力 ($\mathbf{q} + \zeta$) $\in \mathbb{R}^{120 \times 140 \times 1}$ を低次元場 $\beta \in \mathbb{R}^{6 \times 7 \times 16}$ を介し、出力 $\{u, v\} \in \mathbb{R}^{120 \times 140 \times 2}$ を得る。ここで、出力イメージが入力イメージの 2 倍となっているのは、主流方向速度 u と主流に垂直な方向の速度 v の 2 種類が出力されるためである。

4. 結果および結論

4.1 擬似粒子画像を入力した際の速度場の予測

擬似粒子画像 \mathbf{q} および $\hat{\mathbf{q}}$ から推定された速度場の例を Figure 5 に示す。上側の青い領域で囲われている図が、データ欠損のない擬似粒子画像 \mathbf{q} から速度場を予測した結果であり、下の赤い領域で囲われている図はデータ欠損を持たせた擬似粒子画像 $\hat{\mathbf{q}}$ から予測した結果である。

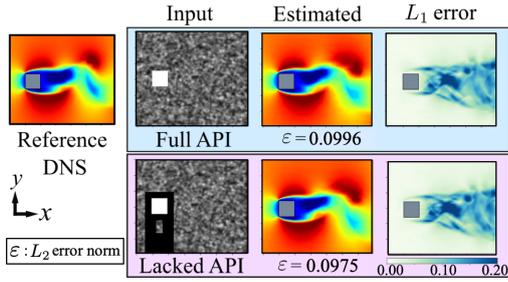


Fig. 5: Estimation of streamwise velocity field from full and lacked APIs.

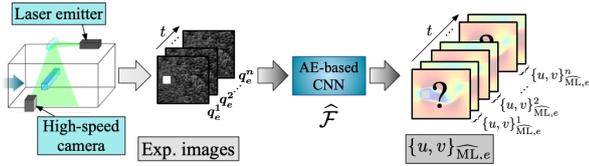


Fig. 6: Application to experimental data (Sec. 4.2). Experimental particle images are fed into the machine-learned model $\hat{\mathcal{F}}$ trained by the lacked data.

それぞれ左から入力された擬似粒子画像、予測された主流方向速度分布の例、時間平均された L_1 error (DNS との差の絶対値) の分布である。図中の ε は L_2 誤差ノルムであり、

$$\varepsilon = \frac{\| \{u, v\}_{\text{DNS}} - \mathcal{F}(\mathbf{q} + \zeta) \|_2^2}{\| \{u, v\}_{\text{DNS}} \|_2^2} \quad (10)$$

または、

$$\varepsilon = \frac{\| \{u, v\}_{\text{DNS}} - \hat{\mathcal{F}}(\hat{\mathbf{q}} + \zeta) \|_2^2}{\| \{u, v\}_{\text{DNS}} \|_2^2} \quad (11)$$

と定義される。データ欠損の有無に関係なく、予測された速度場の L_2 誤差ノルムは 10% 程度であり、速度場が高い精度で再構築されていることが確認できる。また、時間平均された L_1 error の分布を見ると、角柱周辺のデータを欠損させた部分においてデータの欠損がある場合もない場合もほぼ変化がないことが分かる。一方後流の渦放出部分に L_1 error の大きい領域が集中しており、その分布も二つの入力画像に対して大きな変化は見られない。これらのことから、予測の精度は欠損部分の有無に依らず、また欠損が与えられた場合においても後流の渦放出の領域における予測の方が欠損領域よりも困難であることが分かる。

4.2 実験による粒子画像を入力した際の速度場の予測

Figure 6 に示すように、前章までに得た機械学習モデル $\hat{\mathcal{F}}$ を用いて、実験から得た粒子画像への応用可能性を

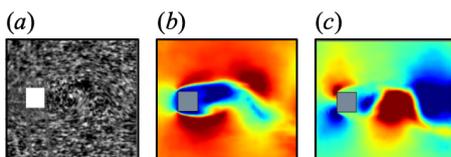


Fig. 7: (a) Interpolated experimental particle image and estimated (b) streamwise and (c) wall-normal velocity field.

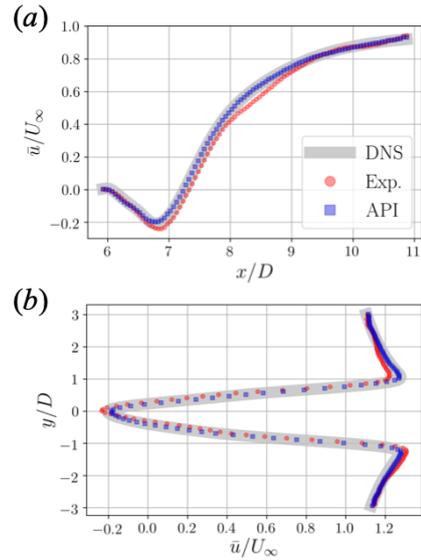


Fig. 8: Mean streamwise velocity at (a) $y/D = 0$ and (b) $x/D = 7$.

探る。本研究では、実験の粒子画像を機械学習モデルに入力する際、格子数を擬似粒子画像と等しくするために、内挿によって $(N_x^{\sharp} \times N_y^{\sharp}) = (480 \times 560)$ から (120×140) とする。実験の粒子画像から予測した速度場を Figure 7 に示す。渦放出の様子など、DNS データからも確認できる大域的な構造が確認された。しかしながら DNS を用いた Sec. 4.1 とは異なり、実験への応用では粒子画像の予測結果に対応する答えとなる速度場は得られないため、予測結果の評価指標として統計量を算出し、DNS のそれと比較した。Figure 8 に $y/D = 0$ および $x/D = 7$ における主流方向平均速度分布を示す。実線が DNS、青い四角が欠損のある擬似粒子画像に対する予測結果、赤い丸が実験の粒子画像に対する予測結果を表している。まず、擬似粒子画像の予測結果に注目すると DNS の結果とほぼ一致しており、速度場が高い精度で再構築されていることが分かる。実験データに対する予測結果に関して、Figure 8(a) に着目すると、 $\bar{u}/U_{\infty} > 0.8$ の領域においては DNS の結果とほぼ一致しており、それ以外の領域においても DNS の結果より若干速度が小さく見積もられているものの、大域的な傾向は推定できていることが確認できる。(b) の図に着目すると、実験データに対する予測結果は \bar{u}/U_{∞} の最大値もしくは最小値付近の領域において DNS データとの差が現れているものの、それ以外の領域においては DNS データとほぼ一致する傾向を示している。このように、実験の粒子画像に対する予測結果は DNS の結果と比較して誤差が生じる領域があるものの、非常に似た分布を推定できていることが確認できる。

また、Figure 9 に DNS と実験の粒子画像の予測結果からそれぞれ計算されたレイノルズ応力 $\bar{u}_i' u_j'$ を示す。レイノルズ応力は、各計算点における速度の変動成分を時間平均した値であり、機械学習モデルの精度を変動成分に着目して統計的に DNS の結果と比較することが可能である為、評価指標の一つとして採用した。それぞれのレイノルズ応力において機械学習モデルの予測結果は絶対値の大きさが DNS と比べて小さいことが確認できるが、平均流速分布と同様に大域的な構造はよく一致していることが確認できる。

実験の粒子画像に対する予測結果の平均流速分布において DNS の分布と異なる部分が観察されること、また変動成分が小さく見積もられていることの最も大きな要因は、実験の粒子画像を入力する際に内挿によって 1/4 に画像がリスケールされていることが考えられる。リス

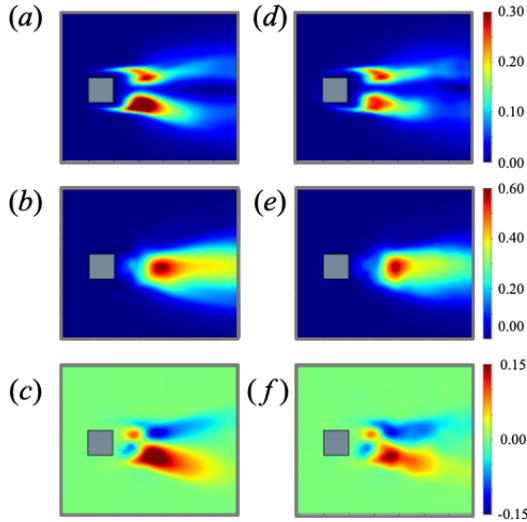


Fig. 9: Summary of Reynolds stress $u'_i u'_j$: (a) $\overline{u'u'}$, (b) $\overline{v'v'}$, (c) $\overline{u'v'}$ of the DNS data. (d) $\overline{u'u'}$, (e) $\overline{v'v'}$, (f) $\overline{u'v'}$ obtained from the experimental data.

ケールの際に、粒子画像が流れ場の細かなスケールの構造を表現しにくくなり、結果として変動が小さく推定されるのではないかと考えられる。

5. 結論

実験の粒子画像から欠損のない速度場を得るための機械学習モデルを構築した。訓練データにはDNSデータを用いて作成した擬似粒子画像 $\mathbf{q} + \zeta$ と、欠損を与えた擬似粒子画像 $\hat{\mathbf{q}} + \zeta$ の二種類を用いた。それぞれのモデルにおいて擬似粒子画像から速度場を高い精度で再構築できることが確かめられた。また、欠損を与えた擬似粒子画像で訓練を行った機械学習モデルに、実験によって得られた粒子画像を入力し、予測された速度場の統計量をDNSと比較した。 $y/D = 0$, $x/D = 7$ における平均流速分布を比較したところ、実験の粒子画像に対する予測結果はDNSの結果と比較して差があるものの、高い精度で速度場を推定できていることが確認できた。また、レイノルズ応力をDNSと予測された速度場からそれぞれ算出し比較したところ、全体的な傾向は類似しているものの、機械学習モデルの予測結果の変動成分がDNSの結果よりも全体的に小さく見積もられている傾向にあることが確認された。これらの結果から、今後は入力として位置情報を追加することなどにより、モデルに物理的な情報を組み込むことで精度が向上する可能性が見込まれる。

参考文献

- (1) Hinze, J. O., *Turbulence*, McGraw-Hill College (1959).
- (2) Settles, G. S., *Schlieren and Shadowgraph Techniques*, Springer (2001).
- (3) Adrian, R. J., “Twenty years of particle image velocimetry,” *Exp. Fluids*, 39 (2005), pp. 159–169.
- (4) 橋本拓郎, 加藤裕之, “ステレオ PIV 計測における模型表面ハレーション防止法,” 可視化情報学会誌, 25 (2005), pp. 219–222.
- (5) 水谷夏樹, 鈴木武, “波動場計測における可視化技術について,” 国土技術政策総合研究所資料 No.259, (2005).

- (6) Brunton, S. L., Noack, B. R., and Koumoutsakos, P., “Machine learning for fluid mechanics,” *arXiv preprint*, (2019), 1905.11075.
- (7) Ling, J., Kurzawski, A., and Templeton, J., “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance,” *J. Fluid Mech.*, 807 (2016), pp. 155–166.
- (8) Fukami, K., Nabae, Y., Kawai, K., and Fukagata, K., “Synthetic turbulent inflow generator using machine learning,” *Phys. Rev. Fluids*, 4 (2019), 064603.
- (9) Fukami, K., Fukagata, K., and Taira, K., “Super-resolution reconstruction of turbulent flows with machine learning,” *J. Fluid Mech.*, 870 (2019), pp. 106–120.
- (10) Murata, T., Fukami, K., and Fukagata, K., “Non-linear mode decomposition with convolutional neural networks for fluid dynamics,” *J. Fluid Mech.*, accepted (2019).
- (11) Chen, P. H., Yen, J. Y., and Chen, J. L., “An artificial neural network for double exposure PIV image analysis,” *Exp. Fluids*, 24 (1998), pp. 373–374.
- (12) Rebaud, J., Kolaas, J., and Jensen, A., “Performing particle image velocimetry using artificial neural networks: a proof-of-concept,” *Meas. Sci. Technol.*, 28 (2017), 125301.
- (13) Cai, S., Zhou, S., Xu, C., and Gao, Q., “Dense motion estimation of particle images via a convolutional neural network,” *Exp. Fluids*, 60 (2019), pp. 60–73.
- (14) Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazrbas, C., Golkov, V., Smagt, P., Cremers, D., and Brox, T., “FlowNet: Learning optical flow with convolutional networks,” *Proc. IEEE Int. Conf. on Computer Vision* (2015).
- (15) LeCun, Y. A., Bottou, L., Bengio, Y., and Haffner, P., “Gradient-based learning applied to document recognition,” *Proc. IEEE*, 86 (1998), pp. 2278–2324.
- (16) Hinton, G. E., and Salakhutdinov, R. R., “Reducing the dimensionality of data with neural networks,” *Science*, 313 (2006), pp. 504–507.
- (17) Okamoto, K., Nishio, S., Saga, T., and Kobayashi, T., “Standard images for particle-image velocimetry,” *Meas. Sci. Technol.*, 11 (2000), pp. 685–691.
- (18) Nair, V., and Hinton, E. G., “Rectified linear units improve restricted Boltzmann machines,” *Proc. Int. Conf. Mach. Learn.* (2010), pp. 807–814.