

# Deep learning fluid flow reconstruction around arbitrary two-dimensional objects from sparse sensors using conformal mappings

Cite as: AIP Advances 12, 045126 (2022); <https://doi.org/10.1063/5.0087488>

Submitted: 06 February 2022 • Accepted: 04 April 2022 • Published Online: 26 April 2022

 Ali Girayhan Özbay and  Sylvain Laizet



View Online



Export Citation



CrossMark

## ARTICLES YOU MAY BE INTERESTED IN

[The scales of the leading-edge separation bubble](#)

Physics of Fluids **33**, 045101 (2021); <https://doi.org/10.1063/5.0045204>

[Direct numerical simulations of incompressible Rayleigh–Taylor instabilities at low and medium Atwood numbers](#)

Physics of Fluids **33**, 054114 (2021); <https://doi.org/10.1063/5.0049867>

[Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations](#)

Physics of Fluids **33**, 107101 (2021); <https://doi.org/10.1063/5.0062546>



Read Now!

AIP Advances

Biophysics & Bioengineering Collection

# Deep learning fluid flow reconstruction around arbitrary two-dimensional objects from sparse sensors using conformal mappings

Cite as: AIP Advances 12, 045126 (2022); doi: 10.1063/5.0087488

Submitted: 6 February 2022 • Accepted: 4 April 2022 •

Published Online: 26 April 2022



Ali Girayhan Özbay<sup>a)</sup> and Sylvain Lazet

## AFFILIATIONS

Turbulence Simulation Group, Department of Aeronautics, Imperial College London, SW7 2AZ London, United Kingdom

<sup>a)</sup> Author to whom correspondence should be addressed: [aligirayhan.ozbay14@imperial.ac.uk](mailto:aligirayhan.ozbay14@imperial.ac.uk)

## ABSTRACT

The usage of neural networks (NNs) for flow reconstruction (FR) tasks from a limited number of sensors is attracting strong research interest owing to NNs' ability to replicate high-dimensional relationships. Trained on a single flow case for a given Reynolds number or over a reduced range of Reynolds numbers, these models are unfortunately not able to handle flows around different objects without re-training. We propose a new framework called Spatial Multi-Geometry FR (SMGFR) task, capable of reconstructing fluid flows around different two-dimensional objects without re-training, mapping the computational domain as an annulus. Different NNs for different sensor setups (where information about the flow is collected) are trained with high-fidelity simulation data for a Reynolds number equal to  $\sim 300$  for 64 objects randomly generated using Bezier curves. The performance of the models and sensor setups is then assessed for the flow around 16 unseen objects. It is shown that our mapping approach improves percentage errors by up to 15% in SMGFR when compared to a more conventional approach where the models are trained on a Cartesian grid and achieves errors under 3%, 10%, and 30% for predictions of pressure, velocity, and vorticity fields, respectively. Finally, SMGFR is extended to predictions of snapshots in the future, introducing the Spatiotemporal MGFR (STMGFR) task. A novel approach is developed for STMGFR involving splitting deep neural networks into a spatial and a temporal component. We demonstrate that this approach is able to reproduce, in time and in space, the main features of flows around arbitrary objects.

© 2022 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0087488>

## I. INTRODUCTION

Most fluid dynamics experiments have access to only sparse measurements due to the intrusive (i.e., flow-altering) nature of pitot tubes and pressure probes used for measurements. Although non-invasive methods to obtain full flow fields in experiments such as particle imaging velocimetry<sup>1</sup> (PIV), magnetic resonance velocimetry<sup>2</sup> (MRV), and laser Doppler flowmetry<sup>3</sup> (LDF) exist, their usage can be limited by practicality, cost, or safety constraints; for instance, PIV systems often require “class IV” lasers that can gravely harm human eyes and cost thousands of dollars. Despite these practical limitations in experiments, knowledge of the full flow fields is often critical to understanding the dynamics of many complex fluid flows. Flow reconstruction (FR) methodologies can offer reliable estimation of a full flow field from only sparse measurements.

Erichson *et al.*<sup>4</sup> described the FR task in terms of a high- and a low-dimensional state vector  $x \in \mathbb{R}^m$ ,  $s \in \mathbb{R}^p$ ,  $m \gg p$ , where

$x$  represents the “full” flow field and  $s$  are sparse sensor measurements. The two state vectors are linked through measurement and reconstruction operators  $H : \mathbb{R}^m \rightarrow \mathbb{R}^p$  and  $P : \mathbb{R}^p \rightarrow \mathbb{R}^m$  such that

$$s = H(x), \quad (1)$$

$$x = P(s). \quad (2)$$

The goal of the FR task is to find an approximation mapping  $R : \mathbb{R}^p \rightarrow \mathbb{R}^m$  such that some measure of error, typically  $L_2$ , between  $\hat{x} = R(s)$  and  $x = P(s)$  is minimized. In practice,  $F$  is a statistical or deep learning algorithm with a parameter set  $w$  optimized to fit some dataset, i.e.,  $R(s) = R(s, w)$ . Deviating from the formulation of Erichson *et al.*<sup>4</sup> to use some generic objective function  $L$  instead of the  $L_2$  norm, the flow reconstruction task can be expressed as a minimization problem of the following form:

$$\arg \min_w L(x, R(s, w)). \quad (3)$$

Historically, methods such as Gappy Principal Orthogonal Decompositions (PODs)<sup>5,6</sup> and Linear Stochastic Estimation (LSE)<sup>7</sup> are some of the main methodologies investigated for FR, but they are often unsuitable for multi-geometry FR (the reconstruction of the flow field generated past an arbitrary geometry), as detailed in Sec. II. Research in this field has recently intensified, and a new wave of studies—the vast majority focused on using neural networks (NNs)—have been published; see Refs. 4 and 8–10 to name a few. As a starting point for neural network based flow reconstruction, these publications largely focus on obtaining models that work on a single fluid flow case, typically predicting vorticity fields for incompressible flow past a circular cylinder at a single (or over a narrow range) of Reynolds numbers. As a result, such approaches do not have the potential yet to be used in wind tunnel testing driven shape optimization, as training such NNs first requires collecting large datasets of the flow past specific objects.

Training NNs for multi-geometry FR from sparse sensors is not straightforward. Within the aforementioned setting of vorticity field reconstruction on Cartesian grids for two-dimensional (2D) incompressible flows past arbitrary objects, naively augmenting the dataset with multiple objects results in models that fail to reproduce key flow features; concentrations of vorticity in boundary layers and near stagnation points often disappear, and the objects themselves are engulfed by amorphous blobs of non-physical vorticity concentrations. The root cause of these issues is that, within these settings, the models lack information regarding the shape of the object they are making predictions on, as the need for such information is obviated by the single-shape nature of the datasets.

Overcoming these issues requires a representation of the flow field in a way that removes the necessity for the model to predict the shape of the object immersed in the fluid. An effective tool for this is a mapping, whereby all possible geometries are mapped to a single shape. For 2D cases, this can be achieved via the Schwarz–Christoffel (S–C) conformal mapping, which can be used to map any  $l$ -connected domain to a disk with  $l$  holes.<sup>11</sup> Thus, the fluid domain in any bluff body flow with a single object can be mapped to an annulus.

In this work, the Spatial Multi-Geometry Flow Reconstruction (SMGFR) task is introduced, with the objective of reconstructing pressure, velocity, and vorticity fields surrounding randomly generated objects immersed in a fluid flow from sparse sensor measurements. S–C mappings are utilized to map the fluid domains surrounding the said randomly generated bluff bodies to annuli, and a dense field sampling approach based on grids uniformly spaced in angular and radial directions in the annular domains is developed. Over a comprehensive set of 24 experiments (encompassing different models and sensor arrangements), the mapping approach is compared to reconstruction based on uniformly spaced Cartesian grids, and the performance of different sensor setups and NN architectures [feed forward, U-Net,<sup>12</sup> and Fourier Neural Operator (FNO)<sup>13</sup>] in SMGFR are investigated.

As a further step toward spatiotemporal reconstruction, a modified version of the SMGFR is proposed whereby the model is expected to construct snapshots at future times, given sensor readings at a present time. In this task, dubbed Spatiotemporal Multi-Geometry Flow Reconstruction (STMGFR), a model obtained from the spatial-only SMGFR task (called the spatial model), is coupled with a second neural network model. This second model, called the

temporal model, accepts the reconstructed dense field as its input and predicts the state of the full flow field  $k$  time steps in the future. The resulting system, composed of the spatial and the temporal models, is thus able to reconstruct the full flow fields  $k$  time steps in the future given current sensor measurements.

This work is organized as follows: first, a brief overview of recent and historical approaches to the FR task is provided in Sec. II. Subsequently, an introduction to the S–C mapping and details of its novel application to the FR task are presented in Sec. III. The dataset constructed to take advantage of the S–C mapping is described in Sec. IV, and the models chosen to fit this dataset and their training procedures are described in Sec. V. Finally, the results are shown in Sec. VI, and a summary of this work and plans for future investigations are provided in Sec. VII.

## II. RELATED WORK

The FR task, as introduced in Sec. I, falls under the broad umbrella of inverse problems.<sup>14</sup> Commonly encountered in a wide range of scientific and engineering fields, including fluid dynamics, inverse problems often lack well-defined, unique solutions and rely on minimization of some objective such as the  $L_2$  norm, as encountered, e.g., in the Moore–Penrose pseudo-inverse for linear least-squares problems. Dubois *et al.*<sup>8</sup> divided the types of approaches to FR into three categories:

1. **Direct reconstruction:** A set of parameters is optimized to learn an approximator  $R$  to the inverse operator  $G$ , as summarized in Eq. (3).
2. **Regressive reconstruction:** The unsupervised learning counterpart of direct reconstruction methods; these methods attempt to fit a series of modes<sup>15</sup> to the available flow data. Methods relying on the principal orthogonal decomposition (POD) fall within this category.
3. **Data assimilation:** Dynamical systems based approaches such as Kalman filtering<sup>16</sup> are used to evolve high-dimensional systems in time based on sensor measurements.

Historically, some of the earliest works in FR originate from meteorology, dating back to the 1980s. Falling mostly within the third category of this classification, Gustafsson *et al.*<sup>17</sup> provided a comprehensive survey of these methods. Framing the FR task in a variational setting,<sup>18–20</sup> they aimed to predict the time evolution of weather systems, given past states and point observations from satellites and ground stations. Despite widespread adoption in weather prediction throughout the world, such approaches are unsuitable for the specific setting explored in this work (reconstruction of current and future vorticity fields from a single instantaneous measurement) due to the necessity of supplying high-quality initial guesses and the time evolution of sensor measurements over a long period of time, which are assumed to be unavailable.

Interest in FR outside the meteorological community emerged in the late 1980s, with methods that fall within the second category of the above classification. Linear stochastic estimation (LSE)<sup>7</sup> is a prominent tool with roots in this era, which reconstructs flow fields based on the computation of a correlation matrix between the sensor inputs and the full flow fields. Recently, it has been applied to reconstruction of PIV-obtained fields in flows over flat plates from microphone measurements,<sup>21</sup> reconstruction of velocity

fields in internal combustion engines from sensor measurements to identify locations of vortical structures,<sup>22</sup> or estimation of all components of the dense velocity field in wall turbulence from hot-wire measurements of the horizontal velocity only.<sup>23</sup>

A further early statistical technique for estimating dense flow fields given sensor measurements is the Gappy POD<sup>5</sup> method, originally developed for reconstructing images of human faces. Examples of the application of the Gappy POD method to FR have included predicting missing information in Direct Numerical Simulation (DNS) data,<sup>6</sup> estimation of the lift coefficient of a NACA0012 airfoil under plunging motion in a Mach 0.6 flow by reconstructing the simulated dense velocity and pressure fields from pressure sensors on its surface,<sup>24</sup> and filling in missing data points in PIV snapshots from gas turbine combustors.<sup>25</sup>

The spectrum of statistical methods applied to FR contains several further less-investigated avenues. Notably, the sparse representation<sup>26</sup> technique, with origins in facial recognition, has been recently applied for FR of free-shear and mixing layer flows.<sup>27</sup> New methods, such as the Sparse Fourier Divergence Free method<sup>28</sup> dedicated specifically to the investigation of incompressible flows, are being developed. Unfortunately, the linear unsupervised nature of the methods in this category lends itself to making predictions for flow datasets incorporating a single geometry only, and we are not aware of any previous works using such methods for multi-geometry datasets. The universal approximation capabilities of deep neural networks (DNNs) with nonlinear activation functions,<sup>29</sup> belonging to the first category in the above classification, are, therefore, a better fit for multi-geometry FR as evidenced by their state-of-the-art performance in many generative Machine Learning (ML) tasks, such as text<sup>30</sup> and image<sup>31</sup> generation. The strength of NNs is evident in FR research as well, with a rapid rise in the number of works investigating the application of neural networks to FR.

Still, most NN-based FR techniques focus on training models that work for a single flow configuration only (without retraining), some notable examples of which are provided: Erichson *et al.*<sup>4</sup> reconstructed vorticity fields in the wake of a circular cylinder from measurements on its surface using a “shallow” NN (i.e., feedforward NN with few layers and units), while Kumar *et al.*<sup>32</sup> improved upon this technique with a recurrent autoencoder based architecture, which can reduce the error by over an order of magnitude when extremely sparse sensor setups involving only a single sensor are used. Dubois *et al.*<sup>8</sup> investigated linear and nonlinear autoencoder NNs with and without variational training and found that non-linearities in NNs enable identification of dominant flow modes, while variational training leads to higher robustness to noise at the cost of higher error. Fukami *et al.*<sup>9</sup> compared the performance of feedforward NNs with three non-neural ML techniques on reconstructing the wake behind a circular cylinder and a flapped airfoil based on noisy sparse measurements from the object surfaces, finding that although neural models are not necessarily the best performing at low noise, they constitute the most robust option under high noise situations. Sun and Wang<sup>10</sup> investigated the usage of physics-constrained Bayesian NNs for flow reconstruction from sparse sensors in stenotic vessels and T-shaped geometries, which permit significantly higher noise robustness compared to standard NNs. Usage of NNs in this context has also been extended to experimental as opposed to simulated data, for example, by Carter *et al.*<sup>33</sup> who

used feedforward NNs for the reconstruction of experimental PIV-obtained velocity fields above the suction surface of a NACA0012 airfoil, achieving superior results compared to non-neural methods.

In comparison, the body of works investigating deep learning based FR for the reconstruction of the flow past arbitrary objects without re-training is small. One notable recent work in this area is by Chen *et al.*,<sup>34</sup> using graph convolutional neural networks (GCNNs) for reconstructing steady flow fields around random objects. Training a GCNN to predict the velocity and pressure fields around 1600 random objects generated via Bezier curves, they applied the model to predict the pressure and velocity fields around 400 test geometries, which permitted for the estimation of drag and lift coefficients with very small mean percentage error levels. The present work differs from Chen *et al.*<sup>34</sup> as it focuses on the usage of a novel mapping approach to achieve the geometry invariance as opposed to graph convolutions, which permits the usage of traditional NN architectures. Here, the aim is to reconstruct instantaneous snapshots as opposed to steady fields and to explore predicting future instantaneous snapshots from current measurements. Additionally, this work is conducted at a substantially higher  $Re = 300$  as opposed to  $Re = 10$  in the work of Chen *et al.*,<sup>34</sup> which leads to the emergence of unsteady rotational flows. To assess the performance of the models, the focus is on reconstructing the vorticity fields (which are difficult to predict from pressure and velocity sensors, as shown in Sec. VI A). Reconstructed data for the pressure and velocity fields are also briefly presented for completeness.

### III. SCHWARZ-CHRISTOFFEL MAPPINGS

Conformal transformations have been used extensively in fluid dynamics, especially the well-known Joukowski and Kármán–Trefftz<sup>35</sup> (K–T) transformations, which map the unit circle to airfoil shapes. However, the usefulness of these two transformations can be limited due to their inability to generalize to arbitrary shapes. A more flexible alternative is the Schwarz–Christoffel (S–C) mapping, which is a conformal transformation historically used to map polygonal simply connected domains to the unit disk. The S–C mapping has been extended in recent decades to multiply connected domains. Although the *existence* of a conformal mapping between any given two  $l$ -connected domains is guaranteed,<sup>11</sup> the practical computation of such an S–C mapping typically requires the use of numerical methods to determine a number of parameters in the mapping expression, referred to as the Schwarz–Christoffel parameter problem. Numerically implementing the S–C mapping for doubly (or higher) connected domains is not a trivial undertaking. In this work, the DSCPACK code,<sup>36</sup> which is a Fortran package aimed at computing S–C mappings between doubly connected domains bounded by polygons to annuli, was utilized to solve the parameter problem.

A rigorous treatment of the methodology used in this package lies beyond the scope of this work, for which the reader is directed to established works in S–C mapping literature,<sup>11,37,38</sup> although the general strategy can be summarized as follows: denoting  $z$  as the complex coordinates in the original domain and  $w$  as the complex coordinates in the annulus domain, DSCPACK uses an expression of

the form

$$z = g(w) = g(w_c) + C \int_{w_c}^w W(w) dw, \quad (4)$$

$$W(w) = \prod_{q=1}^M \left[ \sigma \left( \mu, \frac{-w}{\mu w_{0q}} \right) \right]^{\alpha_{0q}-1} \prod_{r=1}^N \left[ \sigma \left( \mu, \frac{-\mu w}{w_{1r}} \right) \right]^{\alpha_{1r}-1}, \quad (5)$$

$$\sigma(\mu, w) = 1 + \sum_{b=1}^{\infty} \mu^{b^2} (w^b + w^{-b}), \quad (6)$$

as the mapping, where  $C$  is some complex valued constant;  $\mu$  is the radius of the inner ring of the annulus in the  $w$  domain;  $M$ ,  $\alpha_{0q}$ ,  $w_{0q}$  and  $N$ ,  $\alpha_{1r}$ ,  $w_{1r}$  are the number of vertices, the turning angles, and prevertices<sup>39</sup> of the outer and the inner polygon, respectively. Of these variables,  $C$ ,  $\mu$ ,  $w_{0q}$ , and  $w_{1r}$  are unknowns (“accessory parameters” of the mapping) and must be computed by solving a series of nonlinear integral equations,

$$z_{01} - z_{0M} = C \int_{w_{0M}}^{w_{01}} W(w) dw, \quad (7)$$

$$|z_{0,q+1} - z_{0q}| = \left| C \int_{w_{0q}}^{w_{0,q+1}} W(w) dw \right|, \quad q \in [1, M), \quad (8)$$

$$z_{11} - z_{1N} = C \int_{w_{1N}}^{w_{11}} W(w) dw, \quad (9)$$

$$|z_{1,r+1} - z_{1r}| = \left| C \int_{w_{1r}}^{w_{1,r+1}} W(w) dw \right|, \quad r \in [1, N), \quad (10)$$

$$z_{1N} - z_{0M} = C \int_{w_{0M}}^{w_{1N}} W(w) dw, \quad (11)$$

where  $z_{0q}$  and  $z_{1r}$  denote the complex coordinates of the polygon vertices in the original domain. The DSCPACK code solves this nonlinear system using a Newton iteration scheme. The resulting expression maps the outer ring (with unity radius) of an annulus to the outer polygonal boundary, while the inner ring is mapped to the inner polygon.

Once the forward mapping  $g$  is known, the inverse mapping  $f$  can be approximated for any (fixed)  $z = \bar{z}$  using the Newton iteration,

$$\bar{z} - g(w) = 0, \quad (12)$$

$$w_{n+1} = w_n + \frac{\bar{z} - g(w_n)}{g'(w_n)}, \quad (13)$$

$$g'(w_n) = CW(w_n), \quad (14)$$

where Eq. (14) follows from the application of the fundamental theorem of calculus to Eq. (4).

To ensure smooth interoperability of DSCPACK with modern machine learning packages, a set of Python bindings to a modified form of the code were developed, dubbed pydscpack. Furthermore, a number of enhancements to the original code were made to parallelize performance-critical sections with OpenMP. Figure 1 depicts an S-C mapping computed using pydscpack for a geometry used in this study.

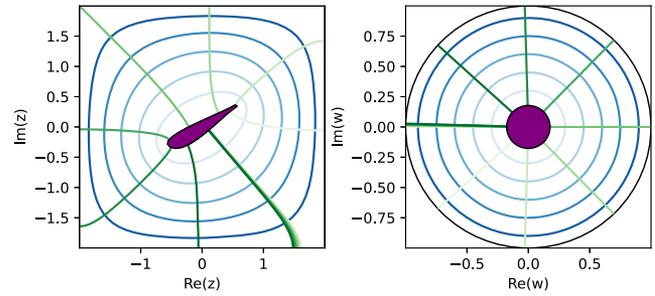


FIG. 1. A random geometry (left) and its annular preimage (right). Blue and green contours depict the norm and argument in the  $w$  domain, respectively. The outer boundary of the domain is smaller than the ones used in the actual study for illustrative purposes.

#### IV. DATASET

##### A. Geometries

A collection of 80 geometries  $G_i, i \in [0, 79]$  were generated using random Bezier curves via the bezier\_shapes package by Viquerat *et al.*<sup>40</sup> The control points for the Bezier curves were chosen randomly in a square domain with characteristic length  $L_m$ . Each geometry was placed in the center of a  $40L_m/3 \times 40L_m/3$  square domain. The fluid domains were meshed with gmsh using a combination of triangular and tetrahedral elements with  $\sim 20\,000$  elements per geometry. Figure 2 shows 12 of the geometries generated using this method.

##### B. Ground truth values

Using uniform Dirichlet velocity boundary conditions  $(u, v) = (1.0, 0.0)$  along the external edges of the domain, the flow around each object was computed for  $Re = uL_m/v = 300$  (where  $v$  is the kinematic viscosity) using the PyFR solver,<sup>41</sup> which is a flux reconstruction<sup>42</sup> based advection-diffusion equation solver using the artificial compressibility approach to solve the incompressible Navier-Stokes equations. It was chosen for its Python interface and graphics processing unit (GPU) acceleration capabilities. The simulations were

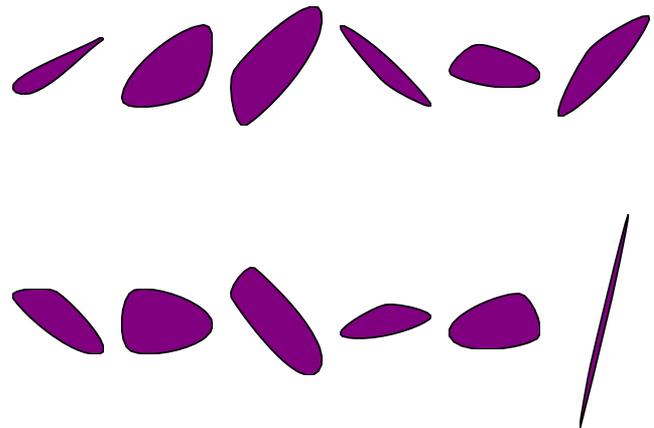


FIG. 2. Twelve geometries used in this study among a total of 80.

performed on two Nvidia V100 GPUs. Normalizing the physical time  $\tau$  by the large eddy turnover time to obtain  $\tau^* = u\tau/L_m$ ,  $t \in [0.600]$  snapshots (containing the pressure and velocity field components  $\mathbf{p}_{t,i}$ ,  $\mathbf{u}_{t,i}$ , and  $\mathbf{v}_{t,i}$ ) were recorded per geometry for a total of 48 080 snapshots between  $\tau^* = 3.333$  and  $\tau^* = 23.333$ .

Following the simulations, referring to the fluid domain around each  $G_i$  as  $F_i$ , the forward and inverse mappings  $g_i$  and  $f_i$  between  $F_i$  and the corresponding annuli  $A_i$  were computed using `pydscpack`.  $64 \times 256$  grids uniformly spaced in the radial and angular directions with coordinates  $\mathbf{w}_{A,i}$  were generated for each  $A_i$ . Subsequently,  $\mathbf{w}_{A,i}$  were mapped back to the original domains  $F_i$  using the computed mappings  $g_i$  to obtain the annular grid coordinates in the original domain  $\mathbf{z}_{A,i} = g_i(\mathbf{w}_{A,i})$ . The velocity, pressure, and vorticity fields  $\mathbf{u}_{t,i}$ ,  $\mathbf{v}_{t,i}$ ,  $\mathbf{p}_{t,i}$ , and  $\boldsymbol{\omega}_{t,i}$  from the high-fidelity simulation data were interpolated to  $\mathbf{z}_{A,i}$  to obtain the interpolated fields  $\hat{\mathbf{u}}_{t,i}$ ,  $\hat{\mathbf{v}}_{t,i}$ ,  $\hat{\mathbf{p}}_{t,i}$ , and  $\hat{\boldsymbol{\omega}}_{t,i}$ , which form the ground truth values of the annular dataset.

This sampling strategy and grid resolution provide a high grid density near the object, scaling with a factor of  $1/r$  based on distance to the center of the annulus. It ensures that the regions of the flow with high vorticity concentrations have enough grid points for a correct representation of the vortical structures. Additionally, as a baseline case, a further collection of ground truth values sampled naively on a  $128 \times 128$  uniformly spaced Cartesian grid was also produced with the same number of grid point as for the mapping approach. Note that these grids are used solely for the interpolation of flow variables, not to perform the fluid simulations.

### C. Inputs

The inputs of the dataset are vectors of pressure and/or velocity values  $s_{t,i}$  at a sparse number of sensor locations, obtained via interpolation of the PyFR solution to the sensor locations. The sensor setup to build the inputs of the dataset is split into two sensor types, chosen to represent a setup that can be practically implemented in a laboratory environment.

- **Pressure:** placed on the surface of each  $G_i$ , with equal angular spacing along the inner ring of each  $A_i$ .
- **Velocity:** positioned on a rectangular grid spanning a  $2L_m/3 \times 4L_m/3$  region, the left edge of which is  $L_m/6$  units behind the rearmost point of each  $G_i$  and the centroid of which is vertically level with each  $G_i$ .

Based on this general template, three setups with varying sensor quantities were considered and summarized in Table I. Figure 3 depicts the medium sensor setup for a sample geometry.

TABLE I. Number of sensors by type and configuration.

	Number of pressure sensors	Number of velocity sensors
Small	12	4
Medium	25	9
Large	50	25

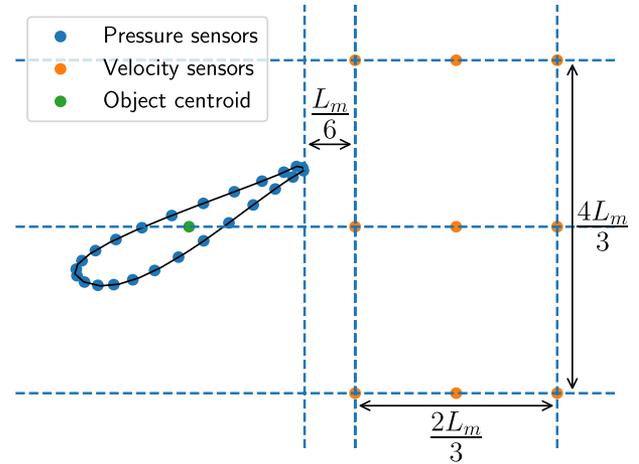


FIG. 3. Illustration of the medium sensor setup for one of the geometries used in the study.

### D. Normalization

Normalizing inputs and/or outputs plays an important role in obtaining good performance from deep learning algorithms, as it permits for better conditioning of the gradients within the optimization landscape during training, by keeping the per-layer statistical distribution of the gradients similar.<sup>43</sup> A variety of data normalization methodologies, including mean centering, standardization, and min–max scaling, were tried in a preliminary study. Denoting  $X$  as the dataset inputs and  $T$  as the target values,  $X^+$  and  $X^-$  as the maximum and minimum values of  $X$ , respectively, and  $\mu$  and  $\sigma$  as the mean and standard deviation, respectively, the three normalization methods can be summarized as follows:

$$\text{Mean centering: } \hat{X} = X - \mu, \hat{T} = T - \mu, \quad (15)$$

$$\text{Standardization: } \hat{X} = (X - \mu)/\sigma, \hat{T} = (T - \mu)/\sigma, \quad (16)$$

$$\text{Min – max scaling: } \hat{X} = \frac{X - X^-}{X^+ - X^-} + X^-, \hat{T} = \frac{T - X^-}{X^+ - X^-} + X^-. \quad (17)$$

Mean centering both inputs and ground truth values based on the ground truth mean values was chosen as the data normalization method, as it provides the results with the lowest validation loss levels for models trained using either the Cartesian or Annulus datasets.

### V. EXPERIMENTAL SETUP

Two related tasks have been investigated, with the dataset detailed in Sec. IV. The first is identical to spatial flow reconstruction tasks from sparse sensors in previous literature,<sup>4,8,33,44</sup> but with the inclusion of snapshots taken from a multitude of geometries in the training and validation datasets. As a reminder, the name Spatial Multi-Geometry Flow Reconstruction (SMGFR) is used to

describe the FR task in this specific configuration. The second task is a generalization of SMGFR where target snapshots are in the future relative to the sensor measurements by a fixed amount of time  $\Delta\tau^*$ , as opposed to SMGFR where the target snapshots are contemporaneous with the measurements, dubbed Spatiotemporal Multi-Geometry Flow Reconstruction (STMGFR).

For both tasks, the dataset from Sec. IV was split by randomly choosing the data associated with 64 geometries as the training set; the remaining 16 geometries constituted the validation set. Training in all experiments was conducted using the Adam<sup>45</sup> algorithm using an initial learning rate (LR) of  $10^{-3}$ , reduced by 90% each time the loss values plateaued.

### A. Spatial multi-geometry flow reconstruction (SMGFR)

Using the notation in Sec. IV, the SMGFR task can be summarized as predicting  $\hat{p}_{t,i}$ ,  $\hat{u}_{t,i}$ ,  $\hat{v}_{t,i}$ , or  $\hat{\omega}_{t,i}$ , given  $s_{t,i}$ . The experiments investigate the performance of four different models, all implemented using Tensorflow<sup>46</sup> v2.5.1, with the parameter counts available in Table II. The latter two models are described schematically in detail in Fig. 4. Below is an overview of the models investigated, with some arguments to justify their use in the present study.

1. **Shallow Decoder (SD)**<sup>4</sup>: a three layer feedforward neural network with 40 units and ReLU activations in the intermediate layers, identical to the setup in the work of Erichson *et al.*<sup>4</sup>
2. **SD-Large**: a larger SD with four layers and 2048 units in each layer, included to assess whether the SD is sufficiently parameterized. Additionally, this model incorporates leaky ReLU activations and batch normalization,<sup>47</sup> motivated by the well-known positive impact of rectifier non-linearities<sup>48</sup> and activation normalization<sup>43</sup> on DNN performance.
3. **SD-UNet**: an SD model with 512 and 2048 units in the intermediate layers, followed by a reshape operation to a 2D grid and a four-level U-Net<sup>12</sup> model with 64 channels in the base level, leaky ReLU activations, batch normalization, and  $p = 0.25$  dropout. This model is included to assess the performance of a well-studied convolutional image-to-image translation model, as opposed to the fully connected SD architecture.
4. **SD-FNO**: an SD model identical to the one present in the SD-UNet, followed by four Fourier Neural Operator (FNO)<sup>13</sup> layers, included due to the remarkable performance of the FNO architecture in previous studies related to fluid flows.

### B. Spatiotemporal multi-geometry flow reconstruction (STMGFR)

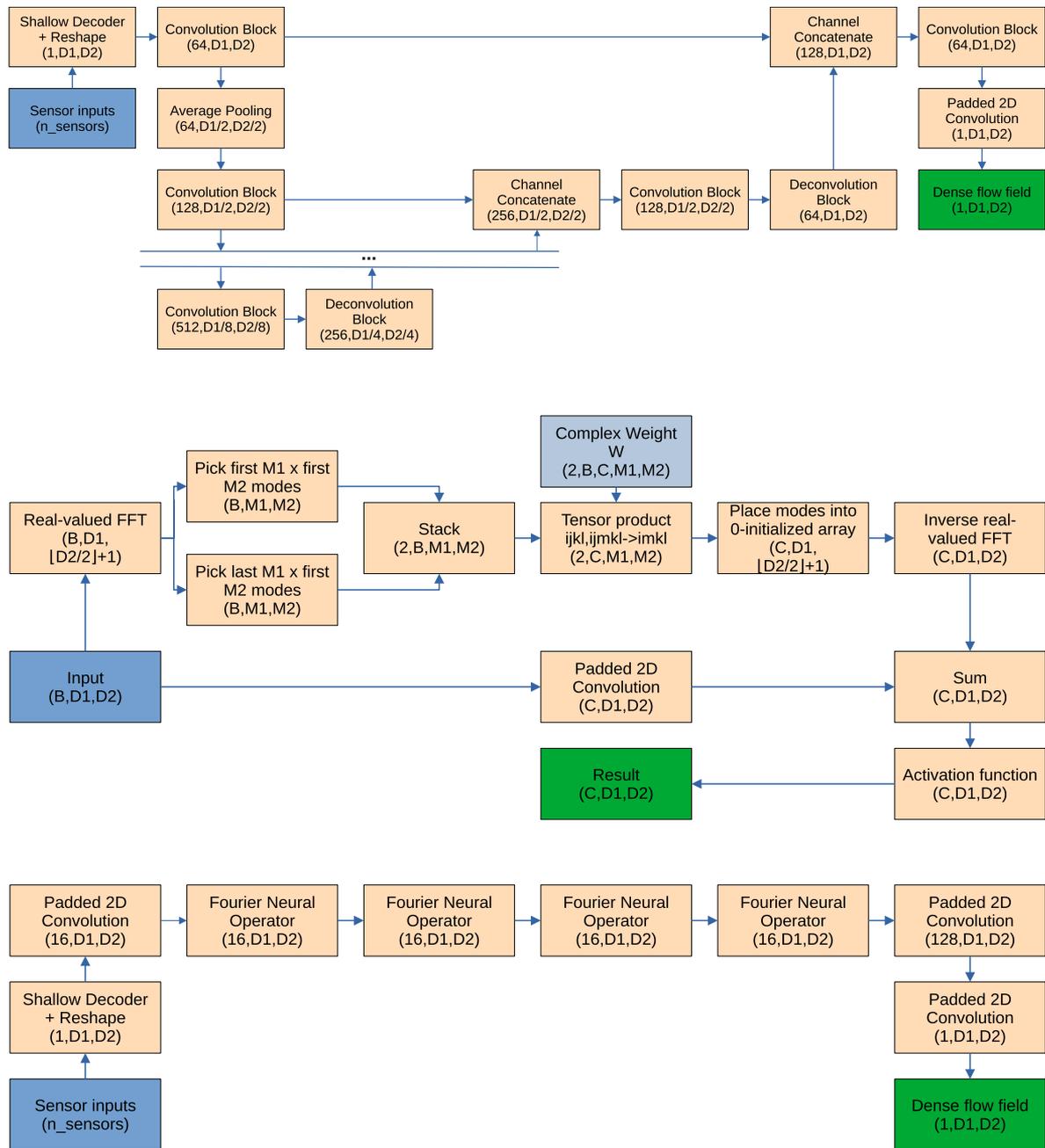
The spatiotemporal multi-geometry flow reconstruction (STMGFR) task extends the purely spatial SMGFR task presented under Sec. V A. Whereas SMGFR focuses on obtaining a reconstruction  $\hat{x}_{t,i}$  of the (interpolated) target field  $\tilde{x}_{t,i}$  (e.g.,  $\tilde{x}_{t,i} = \tilde{\omega}_{t,i}$ ), given the measurements  $s_{t,i}$ , STMGFR is a generalization to flow fields at  $k$  time steps in the future, given the current measurements, i.e., obtaining a reconstruction  $\hat{x}_{t+k,i}$  of  $\tilde{x}_{t+k,i}$ , given  $s_{t,i}$ . In this work, two values for the temporal interval are investigated: a short interval where  $k = 20$  (equal to  $\Delta\tau^* = 0.667$  or 3.33% of the total simulation time) and a longer interval where  $k = 80$  (equal to  $\Delta\tau^* = 2.667$  or  $\sim 13\%$  of the total simulation time). This way, it is possible to investigate temporal gaps both shorter and longer than the large eddy turnover time  $\tau^* = 1.0$ .

An effective way to tackle this FR problem is first obtaining  $\hat{x}_{t,i}$  from  $s_{t,i}$  using one of the models detailed in Sec. V A, dubbed the spatial model. Subsequently, a second model (the temporal model) is used to obtain  $\hat{x}_{t+k,i}$  from  $\hat{x}_{t,i}$ , as summarized in Fig. 5. A stack of six FNO layers was used as the temporal model, which have been demonstrated to be accurate when used to time-march the two-dimensional Navier–Stokes equations.<sup>13</sup> Every FNO layer has 64 channels and 32 modes per spatial dimension, while the convolutions use  $[1, 1]$  kernels, translating to 50 635 313 parameters. The spatial model was chosen as the SD-UNet based on its high performance in the spatial task (as detailed in Sec. VI B, with identical weights), with a configuration identical to the one in Sec. V A. Likewise, the chosen sampling strategy is annular due to its superior performance in Sec. VI B.

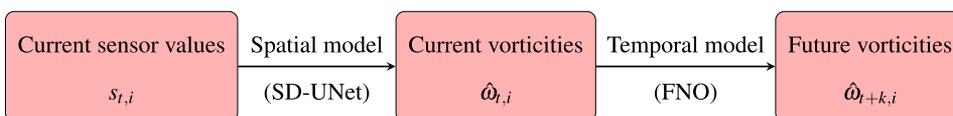
Training the temporal model is done in a supervised manner. Since the temporal model would be expected to perform well, given reconstructed inputs  $\hat{x}_{t,i}$  from the spatial model in an inference scenario, in each epoch, the input associated with every sample is randomly chosen to be either a reconstructed vorticity field  $\hat{x}_{t,i}$  or a ground truth field  $\tilde{x}_{t,i}$  with a 50% chance for each. Using separate spatial and temporal models with parameter counts  $P_s$  and  $P_t$  (vs a larger single model with parameter count  $P_s + P_t$  directly predicting  $\hat{x}_{t+k,i}$  from  $s_{t,i}$ ) is highly computationally efficient as it permits for easy re-training of multiple temporal models for different values of  $k$ . The results from the spatial model can be easily cached and re-used when training a new temporal model, which translates to computational speedups and model accuracy benefits owing to the possibility of using larger batch sizes, given a fixed pool of memory.

**TABLE II.** Number of parameters per model and sensor setup, broken down by layer type, for SMGFR. Only the first feedforward layer's parameter numbers are affected by the sensor setup.

	Large sensor setup			Medium sensor setup			Small sensor setup		
	Dense	Conv/FNO	Total	Dense	Conv/FNO	Total	Dense	Conv/FNO	Total
SD	677 424	...	677 424	675 144	...	675 144	674 224	...	674 224
SD-large	46 399 488	...	46 399 488	46 282 752	...	46 282 752	46 235 648	...	46 235 648
SD-UNet	34 683 392	7 711 301	42 394 693	34 654 208	7 711 301	42 365 509	34 642 432	7 711 301	42 353 733
SD-FNO	34 683 392	2 105 857	36 789 249	34 654 208	2 105 857	36 760 065	34 642 432	2 105 857	36 748 289



**FIG. 4.** Diagrams of the SD-UNet (top), an FNO layer (middle), and the SD-FNO (bottom). The values in the parentheses indicate the shape of each block's output tensor;  $D1 = 64$  and  $D2 = 256$  for the Annulus dataset, and  $D1 = D2 = 128$  for the Cartesian dataset. In the SD-UNet, each convolution block is formed of two convolution layers, each preceded by a batch normalization layer and followed by a dropout layer; each deconvolution block is formed of a batch normalization layer followed by a deconvolution with stride 2.



**FIG. 5.** Summary of the two-step spatiotemporal reconstruction approach.

## VI. RESULTS

The results of a series of SMGFR and STMGFR experiments are detailed in this section, the setups of which are detailed in the previous chapters. First, to compare the accuracy and quality of our reconstruction methodology to the previous work by Chen *et al.*,<sup>34</sup> we briefly present results on reconstructing pressure and velocity fields in Sec. VI A. Additionally, we numerically demonstrate that the reconstruction of vorticity presents a greater challenge than the reconstruction of pressure and velocity from pressure and velocity sensors.

Subsequently, we proceed to detailed comparisons of vorticity reconstruction performance, where the differences between the various setups are visible more acutely. Section VI B details the results from the spatial reconstruction task with four models and three sensor setups, as detailed in Sec. IV and V A. With the best performing configuration for the spatial task identified in this section, Sec. VI C presents the efforts in combining the spatial-only configuration with a further time-marching model to predict future vorticity fields from current sensor measurements, as explained in Sec. V B. Section VI D compares the wall-clock runtimes of all models considered.

To focus on the most rotational regions of the flows, the analyses are constrained to a region of the computational domain immediately surrounding and downstream of the objects investigated, encompassing a rectangular region with vertices  $[-L_m, L_m]$ ,  $[-L_m, -L_m]$ ,  $[4L_m, L_m]$  and  $[4L_m, -L_m]$ .

### A. Spatial multi-geometry pressure and velocity reconstruction

In order to compare the quality of our reconstruction methodology with the previous multi-geometry reconstruction work by Chen *et al.*,<sup>34</sup> we briefly present the results of training the  $k = 0$  SD-UNet+FNO combination from Sec. V B on pressure and velocity data in Table III using the large sensor setup.

Additionally, to demonstrate that a fairly complicated and non-linear reconstruction relationship  $P$  is present between the sensor measurements and the vorticity fields investigated in Secs. V A and V B, we include two difficulty measures  $\mathcal{D}$  and  $\mathcal{M}$  in Table III. These

measures are based on the Frobenius norms of the Spearman rank correlation coefficient<sup>49</sup> (SRCC) matrix and Mutual Information<sup>50</sup> (MI). Defining  $\psi_{t,i} = [s_{t,i} \ x_{t,i}] \in \mathbb{R}^{p+m}$  as a vector concatenating the sensor measurements  $s \in \mathbb{R}^p$  and the full field  $x \in \mathbb{R}^m$  (pressure, vorticity, etc.) for a particular snapshot  $i$  at a particular time  $t$ , we construct a large matrix  $\Psi$  containing the entirety of the data in our dataset,

$$\Psi = \begin{bmatrix} \psi_{0,0} \\ \psi_{0,1} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} s_{0,0} & x_{0,0} \\ s_{0,1} & x_{0,1} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix}. \quad (18)$$

Subsequently, we compute the SRCC and MI matrices  $\mathbf{D}, \mathbf{M} \in \mathbb{R}^{(p+m) \times (p+m)}$  based on the columns of  $\Psi$ . Both are composed of four sub-matrices  $\mathbf{D}_{s,s}, \mathbf{M}_{s,s} \in \mathbb{R}^{p \times p}$ ,  $\mathbf{D}_{x,x}, \mathbf{M}_{x,x} \in \mathbb{R}^{m \times m}$ ,  $\mathbf{D}_{s,x}, \mathbf{M}_{s,x} \in \mathbb{R}^{p \times m}$ , and  $\mathbf{D}_{x,s}, \mathbf{M}_{x,s} \in \mathbb{R}^{m \times p}$ . The first two sub-matrices contain the SRCC values of the sensor measurements and full field values among themselves, while the latter two contain the SRCC values between the sensor measurements and the full field. The difficulty measures  $\mathcal{D}$  and  $\mathcal{M}$  are defined as the Frobenius norms of  $\mathbf{D}_{s,x}$  and  $\mathbf{M}_{s,x}$ , respectively,

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_{s,s} & \mathbf{D}_{s,x} \\ \mathbf{D}_{x,s} & \mathbf{D}_{x,x} \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_{s,s} & \mathbf{M}_{s,x} \\ \mathbf{M}_{x,s} & \mathbf{M}_{x,x} \end{bmatrix}, \quad (19)$$

$$\mathcal{D} = \|\mathbf{D}_{s,x}\|_2 = \|\mathbf{D}_{x,s}\|_2, \quad \mathcal{M} = \|\mathbf{M}_{s,x}\|_2 = \|\mathbf{M}_{x,s}\|_2. \quad (20)$$

The difficulty metrics clearly show the reason for lower performance when predicting  $\omega$  with all setups. The  $\mathcal{D}$  scores display greater correlation between the sensor inputs and pressure/velocity fields compared to the vorticity field. This translates to, on average, greater monotonicity in the relations mapping the sensor data to the full field data for the pressure and velocity fields compared to the vorticity field.  $\mathcal{M}$ , meanwhile, demonstrates that the probability

**TABLE III.** Mean absolute error (MAE) and mean absolute percentage error (MAPE) levels from the pressure and velocity reconstruction experiments using the  $k = 0$  SD-UNet+FNO combination from Sec. V B. MI scores were computed using  $50 \times 50$  bins via sklearn's<sup>51</sup> `adjusted_mutual_info_score`.  $\omega$  results are identical to  $k = 0$  results in Sec. VI C—note that MAE is lower with Cartesian sampling despite higher MAPE, as Annulus sampling concentrates grid points near the boundary of the object where larger vorticity concentrations are typically observed.

		p	u	v	$\omega$
$\mathcal{D}$					
(Higher is easier)		306.22	298.99	301.85	242.82
$\mathcal{M}$					
(Higher is easier)		302.72	296.37	269.81	207.75
Annular	MAE	$1.18 \times 10^{-2}$	$2.64 \times 10^{-2}$	$1.22 \times 10^{-2}$	$3.10 \times 10^{-1}$
	MAPE	2.43%	8.26%	9.40%	28.89%
Cartesian	MAE	$1.33 \times 10^{-2}$	$3.32 \times 10^{-2}$	$1.64 \times 10^{-2}$	$4.95 \times 10^{-2}$
	MAPE	3.32%	11.56%	15.61%	33.56%

distributions of the sensor inputs and full field values are substantially more like (i.e., have lower relative entropy). Both of these have profound effects on the accuracy of the neural networks, which manifests in the difference in the MAPE scores when predicting different target fields.

Since the higher difficulty associated with predicting the vorticity field is illustrated, we move forward to comparing the quality of our pressure and velocity results with previous works. Chen *et al.*<sup>34</sup> reported reconstruction errors amounting to  $7.70 \times 10^{-3}$  in a similar setup, but at a substantially lower Re, predicting on flow cases within steady, laminar flow regimes only. Thus, considering the substantially higher Re in this work, which results in the creation of unsteady vortical structures, the differences in data generation methodologies, and the different objective involving the prediction of instantaneous as opposed to steady fields, the error levels exhibited are in line with previous works. The MAPE levels, under 3% and 10%, respectively, for pressure and the velocity components, clearly demonstrate that our work is a clear step forward for SMGFR. A gallery of sample velocity and pressure predictions is provided in [Appendix A](#).

## B. Spatial multi-geometry vorticity reconstruction

Considering the higher difficulty in predicting the vorticity field from pressure/velocity sensors and pushing the boundaries of the neural networks for flow reconstruction tasks, a comprehensive set of 24 experiments for the vorticity SMGFR have been conducted to highlight the differences between the combinations of sensor setups, model architectures, and sampling strategies. [Table IV](#) summarizes the performance of all combinations.

Two different metrics of percentage error, as seen in [Table IV](#), are used in this section: the standard MAPE and the High Vorticity MAPE (HV-MAPE) computed from points for which the vorticity magnitude exceeds 1% of the maximum absolute vorticity in a snapshot. Comparing the MAPE and HV-MAPE, it can be seen that HV-MAPEs are consistently lower than the MAPEs, which is not surprising, given the choice of the MAE as the loss function, which assigns a greater penalty to the regions of higher target field magnitude, given constant percentage errors. As a consequence, the models implicitly prioritize lowering the percentage errors for areas with high vorticity, which is desirable since these are

the most important features when analyzing the dynamics of fluid flows. Overall, MAPEs are between 40% and 47% for the annular sampling, while they are between 46% and 60% for the Cartesian sampling. HV-MAPEs can go as low as 31% for the annular sampling and 39% for the Cartesian sampling.

Among the three major variables differentiating the experiments—the sampling method, model architecture, and sensor setup—the factor most consistently leading to superior results is the sampling method. The annular sampling method enables substantially lower MAPE and HV-MAPE with all models and sensor setups tested, lowering MAPE and HV-MAPE by up to 15% and 21% points, respectively (translating to 26% and 40%), in the case of the SD model with the large sensor setup. The error-reducing effect of the annular sampling is especially prominent in the high vorticity regions of the flow, as evidenced by comparing the differences in HV-MAPE vs differences in MAPE when comparing the two sampling strategies; out of the 12 combinations of models and sensor setups, HV-MAPE has a larger absolute gap than MAPE between the two sampling methodologies in all cases except for the SD model using the Large sensor setup, with the differences in HV-MAPE being larger by about 3% points on average.

Next in the order of importance is the model architecture. The two architectures, which more explicitly isolate the various spatial length scales, by the way of pooling-convolution-upsampling branches for the SD-UNet or convolution in Fourier space in the case of the SD-FNO, exhibit superior percentage error metrics for all sensor and sampling setups. This is not surprising, given the established superiority of convolutional architectures in various computer vision tasks over fully connected networks.<sup>52</sup> However, the importance of the architecture varies by the sampling method. While MAPEs and HV-MAPEs are at most within a 5% and 3% range of each other (respectively) between models for a particular sensor setup in the case of annular sampling, this gap increases to as much as 14% with Cartesian sampling. Controlling the number of parameters (as seen in [Table II](#)), the impact of the architecture declines even further, as the SD-large model is consistently ahead of the standard SD model.

The final variable to discuss is the sensor setup, which does not have a great impact on the accuracy of the models. The largest setup

**TABLE IV.** % errors of the model predictions, averaged over the validation dataset, per sensor setup, and sampling strategy. Two different values for the % errors are presented; the standard MAPE and the High Vorticity MAPE (HV-MAPE) filtering out the data points with vorticity magnitude below 1% of the max ground truth vorticity. In both cases, sampling points with percentage errors above 200% were excluded from the MAPE calculation due to error values approaching infinity near the ground truth zero vorticity field contours. The best performing sampling strategy-model architecture combinations for each sensor setup are highlighted with italics and underscores for the MAPE and HV-MAPE, respectively. Bold values denote the lowest MAPE or HV-MAPE sampling strategy for a sensor setup-model configuration; italic values denote the model achieving the lowest MAPE or HV-MAPE for a particular sensor setup.

Sensor setup	Sampling	SD		SD-large		SD-UNet		SD-FNO	
		MAPE	HV-MAPE	MAPE	HV-MAPE	MAPE	HV-MAPE	MAPE	HV-MAPE
Large	Annular	<b>44.29%</b>	<b>34.28%</b>	<b>43.80%</b>	<b>31.85%</b>	<b>39.92%</b>	31.37%	<b>40.83%</b>	<b>31.78%</b>
	Cartesian	59.88%	46.14%	57.52%	53.36%	47.64%	39.88%	46.56%	39.34%
Medium	Annular	<b>46.86%</b>	<b>34.35%</b>	<b>45.45%</b>	<b>33.77%</b>	<b>42.61%</b>	32.73%	<b>43.66%</b>	<b>32.93%</b>
	Cartesian	55.41%	48.99%	58.65%	48.68%	48.49%	42.99%	48.17%	42.12%
Small	Annular	<b>47.35%</b>	<b>34.72%</b>	<b>47.20%</b>	<b>34.06%</b>	<b>45.03%</b>	33.17%	<b>44.04%</b>	<b>33.77%</b>
	Cartesian	57.40%	50.89%	59.77%	48.59%	51.56%	44.47%	51.81%	45.81%

provides five times more information as the smallest, only leading to modest improvements in terms of the MAPE, which do not exceed 5% for any model or sampling strategy. However, especially in the case of the SD and SD-large models, taking advantage of the extra information is possible only in conjunction with the usage of annular sampling. The merit of having a large sensor setup is that, from a purely computational perspective, increasing the number of sensors is very cost-effective, as it leads only to an increase in the number of parameters and computational cost in the first layer of the model, opening up an avenue for modest accuracy improvements almost “for free.” In an experimental setting, however, this may be counteracted by the burden of a significantly more labor-intensive physical sensor setup.

To provide a deeper insight into the performance of the models with the different sensor and sampling setups beyond the overall error values, a gallery of predictions is provided. Figures 6–8 depict the predictions for a bluff body-like shape, dubbed “shape A,” using the large, medium, and small sensor setups, respectively. Furthermore, Figs. 9 and 10 show the predictions for “shape B,” an oval-like object set at a high incidence angle relative to the incident flow displaying separation over its upper surface, and “shape C,” a thick flat plate-like object with characteristic counter-rotating vortices immediately downstream of the object for the large sensor setup. Juxtaposing the models’ performance over a range of flows showing a diverse range of dynamics, the three snapshots are chosen to illustrate different levels of relative performance between the sampling strategies and models; shape A’s snapshot is chosen to display the relative strengths of the top performing combination (SD-UNet with annular sampling), shape B displays a case where Cartesian sampling wins in terms of MAPE (albeit losing by a far greater margin in terms of HV-MAPE), and in shape C’s case, the top performers are closely matched.

Starting with shape A in Figs. 6–8, the focus is on the differences between the Cartesian and annular sampling methods. The boundary layers stand out as the areas of greatest difference between the sampling methods. With Cartesian sampling, none of the models can accurately predict the existence of the high vorticity concentrations near the stagnation point, the highly positive concentration of vorticity along the upper surface of the object, or the location of the separation on the lower surface. In contrast, all three of these features are correctly predicted with the annular sampling, even by the relatively simple feedforward SD and SD-large models, regardless of the sensor setup. The two sampling strategies are closer in terms of performance in the wake, as evidenced by the greater similarity in the error maps, but the Cartesian models have higher error downstream of the object.

Additionally, comparing the predictions from different sensor setups for shape A, the above conclusion regarding the limited impact of the setup on model performance is reinforced. Error levels do not show a strong tendency to decline with larger sensor setups, exemplified by the lowest errors for the SD-UNet being displayed for the medium sensor setup in Fig. 7. Randomness in model training, caused by model initialization and stochastic model architecture features, such as dropout, has a substantially greater impact on performance, in particular snapshots. This effect disappears when averaged over the whole dataset and many epochs of training, resulting in the paradoxical looking situations, such as predictions from models, using fewer

sensors being more accurate for specific samples despite higher overall error.

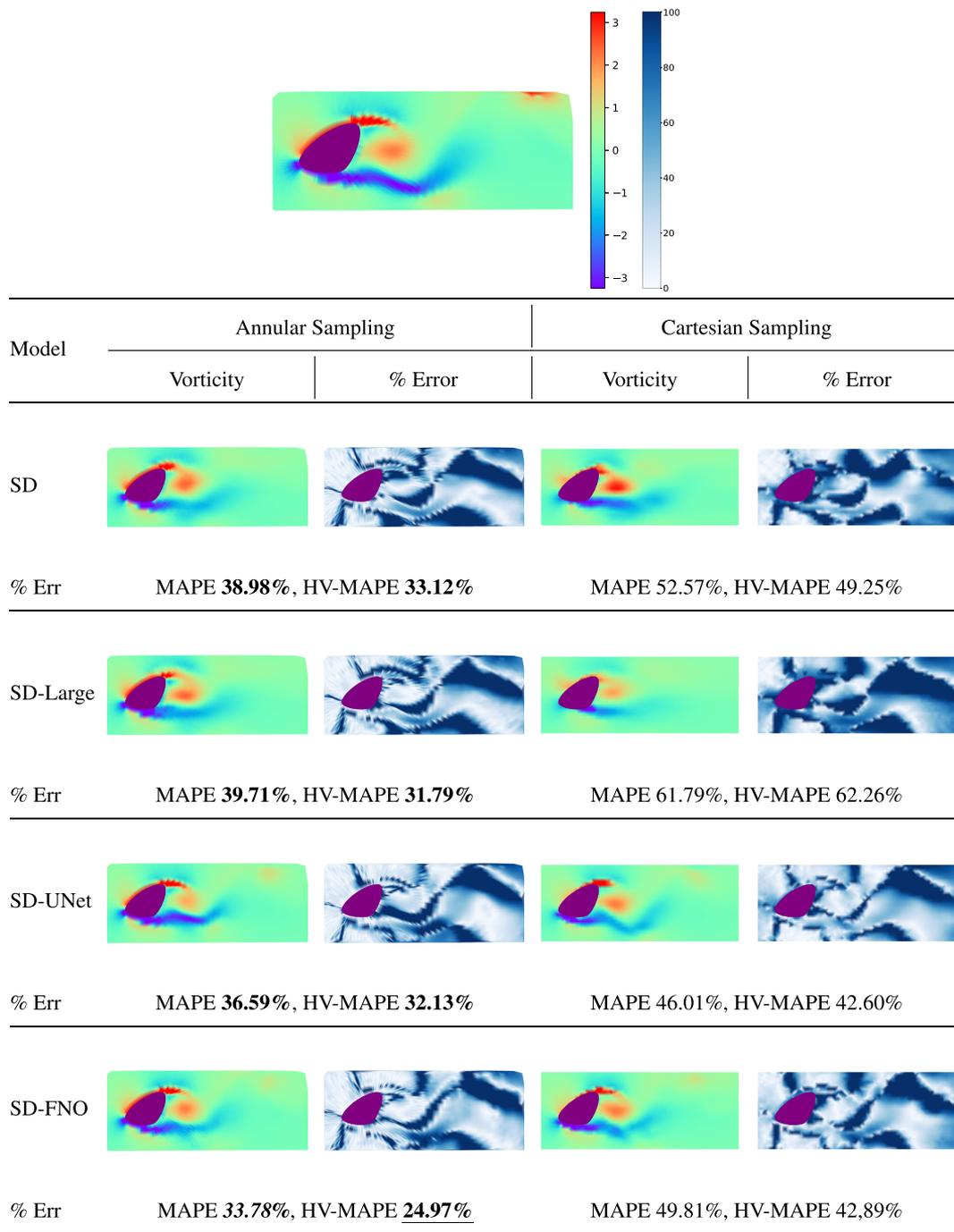
Moving forward to shape B in Fig. 9, the difference is most striking for the SD and SD-large, where the Cartesian versions of the models predict a high concentration of negative vorticity entirely engulfing the object, which is highly non-physical. Additionally, the location of the high positive vorticity blob near the “leading edge” of the object is predicted as *detached* from the object surface. In contrast, the same models with annular sampling correctly predict these key flow features, appropriately placing the concentrations of vorticity near the stagnation point, the separation near the leading edge, and the starting vortex-like formation near the rear of the object, although also performing poorly with respect to the vortex shed further downstream of the object. These issues are not as prevalent for the more complicated SD-UNet model, with the Cartesian sampling eking out a rare quantitative win in terms of the MAPE (by 5%) against annular for the SD-UNet owing to the high error with annular sampling far downstream of the object despite more accurate predictions near the object. Lower performance further away from the object in the case of annular sampling is caused by the same radial and angular spacing, which translates to a lower grid point density further away from the object (see Sec. IV B). As a consequence, as with the other snapshots, annular sampling displays greater performance nearer the object and achieves a decisive win vs Cartesian sampling once the low vorticity regions far downstream of the object are filtered out, achieving a 15% lower HV-MAPE.

The trends for the other shapes continue in the case of shape C in Fig. 10. Similar to both feedforward models for shape B, the SD-large model with Cartesian sampling spuriously predicts a large concentration of very high vorticity surrounding the object while also substantially under-estimating the intensity of the downstream vortices. The SD-UNet with annular sampling performs the best, but with Cartesian sampling, the error is much higher also due to an underestimation of the intensity of the two counter-rotating vortices behind the object. Finally, the SD-FNO is the trend-breaker, with Cartesian sampling managing a narrow quantitative win thanks to a better estimation of the vortex intensity. However, from a qualitative perspective, both SD-FNO predictions are noisy and do not accurately reproduce the smoothness of the vorticity field unlike the SD-UNet.

As a final remark, we draw attention to the presence of high error along the same contours for different sampling strategies and models among the images for each snapshot. This is due to the presence of very high percentage error (despite low absolute error) near the zero contours of the target field due to very small denominators. Visible in low vorticity areas across all geometries, it is ultimately caused by the objective function as explained above. It is also the main reason why the MAPE and HV-MAPE may appear high with values consistently between 20% and 60%.

### C. Spatiotemporal multi-geometry vorticity reconstruction

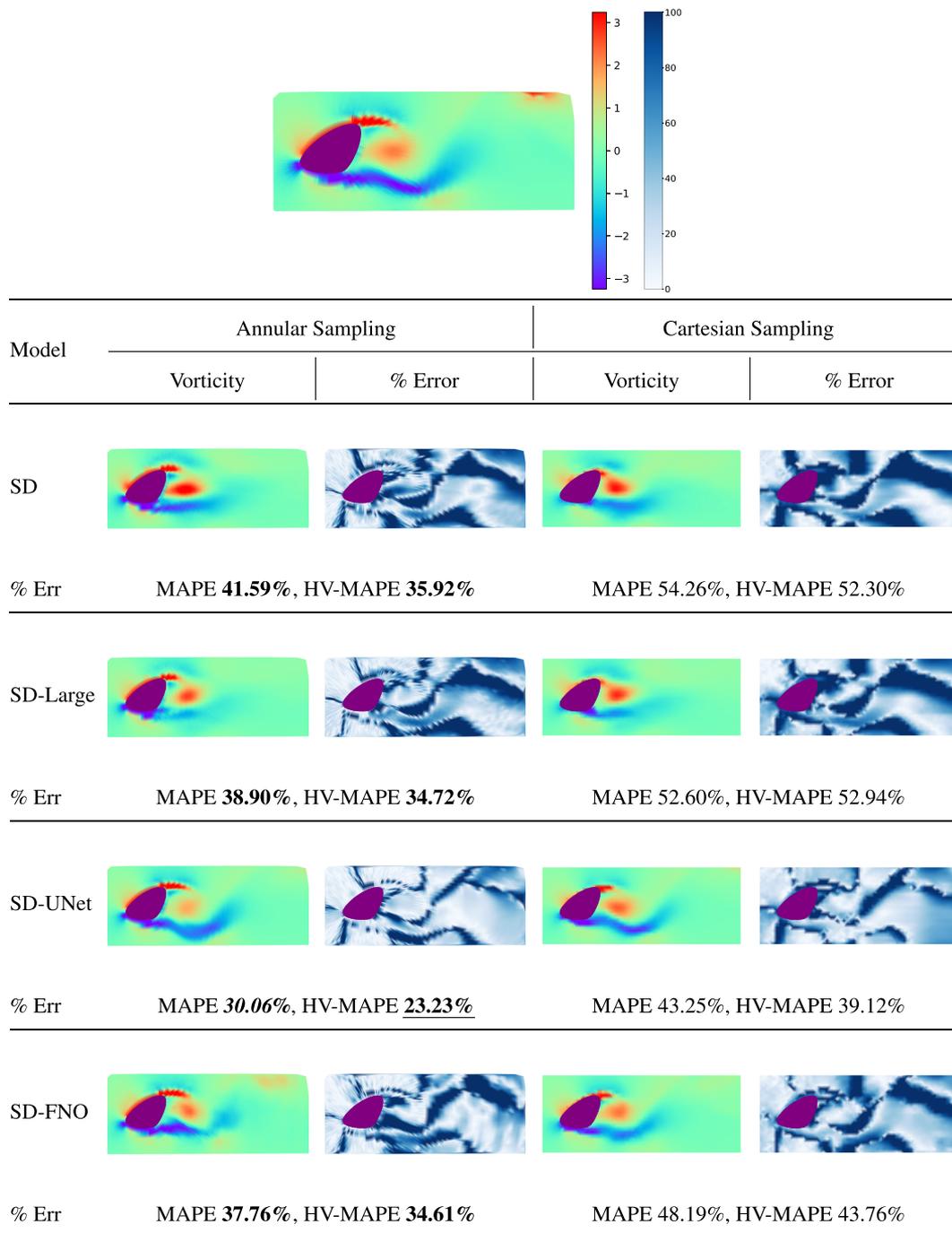
Since the SD-UNet model, used alongside annular sampling and the large sensor setup, was identified as the best performing combination in Sec. VI B, this combination was chosen for the spatial model in this work’s approach to the STMGRF task.



**FIG. 6.** Ground truth (top) and predicted (bottom) vorticity fields and percentage error maps for a snapshot from shape A using the large sensor setup. Vorticity colormap constrained to 7.5% of  $\max(|\bar{\omega}_{t_i}|)$ .

The performance of the temporal model is summarized in Table V for different values of the temporal gap  $k$  (refer to Table IV for the spatial model’s performance). A set of results with  $k = 0$ , at which point the temporal model (FNO) effectively acts like an

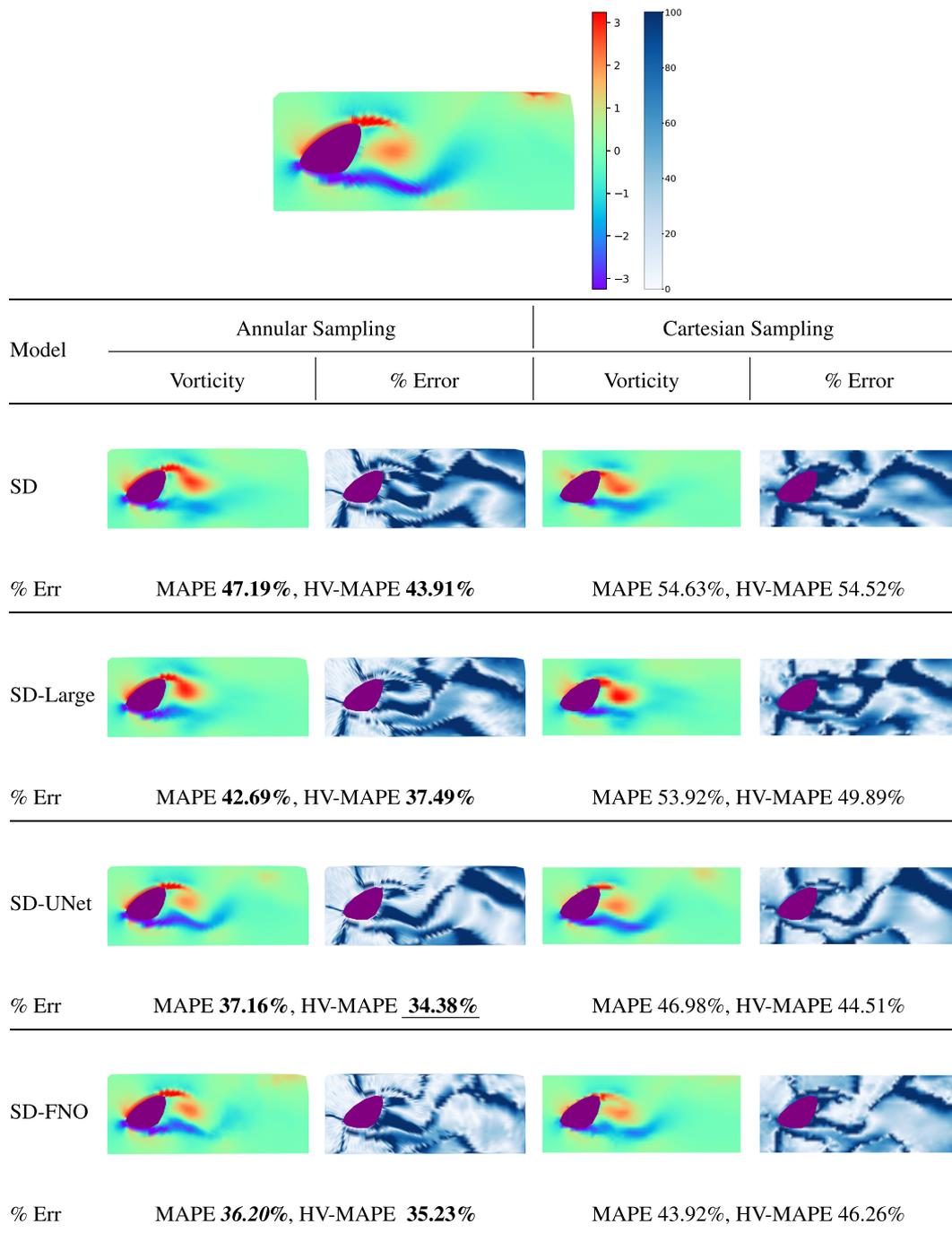
autoencoder since ground truth snapshots are provided as inputs, are included in addition to the  $k = 20$  and  $k = 80$  values mentioned in Sec. V B to provide a greater insight into the temporal model’s behavior.



**FIG. 7.** Ground truth (top) and predicted (bottom) vorticity fields and percentage error maps for a snapshot from shape A using the medium sensor setup. Vorticity colormap constrained to 7.5% of  $\max(|\hat{\omega}_{t_i}|)$ .

As expected, the error declines as the temporal gap  $k$  is reduced and the model performs better when ground truth snapshots are provided as the inputs as opposed to reconstructed inputs. Surprisingly, however, MAPE levels for this task are substantially

lower than results for the purely spatial task in Table IV despite its greater difficulty. The reason for this is clearly illustrated by the  $k = 0$  results: the additional FNO model placed “at the end” of the SD-UNet model, with a parameter count exceeding

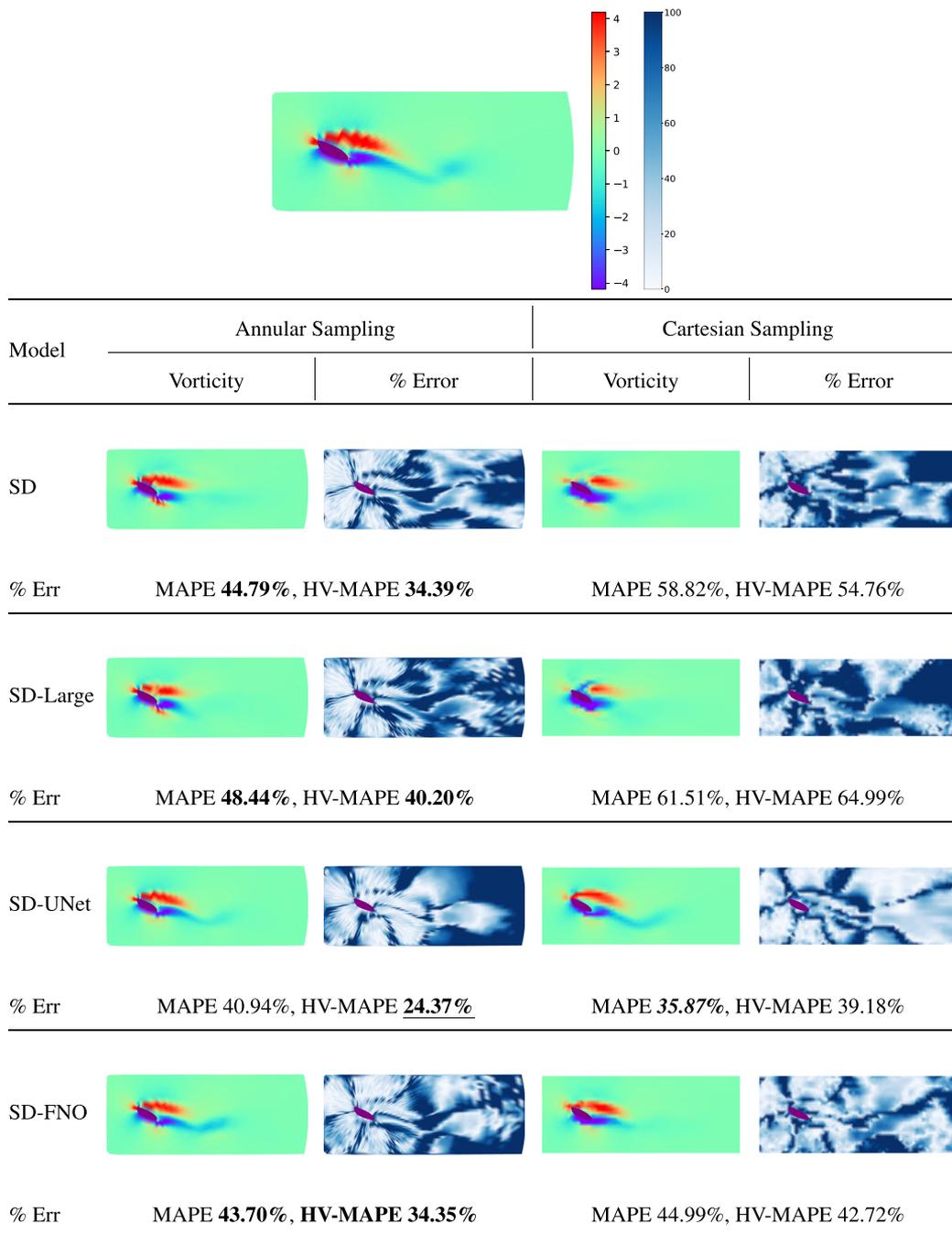


**FIG. 8.** Ground truth (top) and predicted (bottom) vorticity fields and percentage error maps for a snapshot from shape A using the small sensor setup. Vorticity colormap constrained to 7.5% of  $\max(|\hat{\omega}_{t,i}|)$ .

the entirety of the SD-UNet, serves as a denoising autoencoder,<sup>53</sup> which substantially reduces the error of the SD-UNet output. Effectively acting like a model employed in SMGFR as opposed to STMGFR in this setting, examples of reconstructed snapshots

from this  $k = 0$  “SD-UNet+FNO” configuration are presented in [Appendix B](#).

As  $k$  increases, the denoising effect declines monotonically as the temporal model must simultaneously correct for the errors in

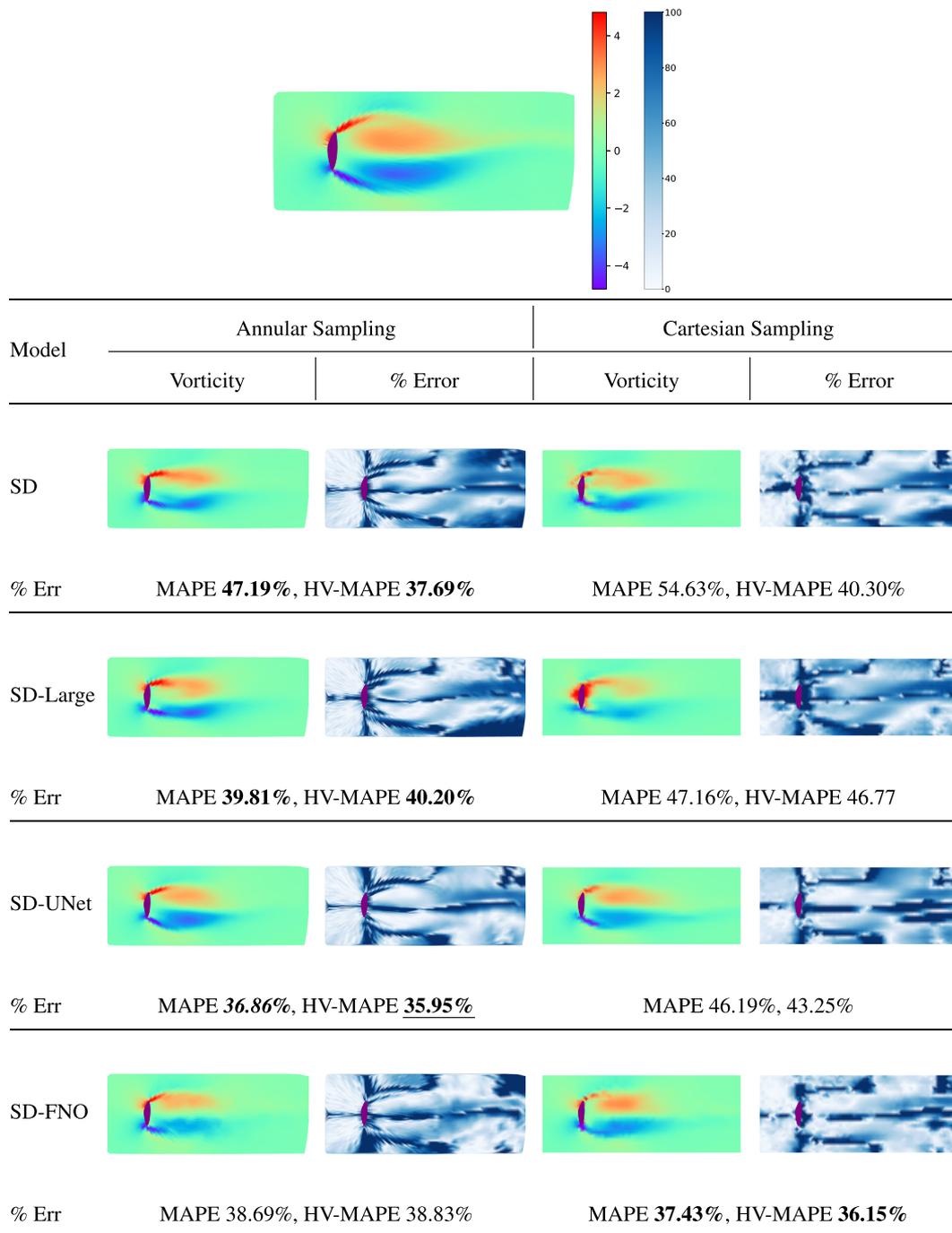


**FIG. 9.** Ground truth (top) and predicted (bottom) vorticity fields and percentage error maps for a snapshot from shape B using the large sensor setup. Vorticity colormap constrained to 3% of  $\max(|\hat{\omega}_{t,i}|)$ .

its input and predict the time evolution of the input snapshot. Hence, to dissect the sources of error, we focus on results with  $k = 80$  over a number of example snapshots; similar to Sec. V A, Figs. 11–13 provide example ground truth and predicted snapshots from three further shapes—dubbed shapes D, E, and F—chosen

from the validation dataset for displaying a diverse range of flow dynamics.

Shape D, in Fig. 11, is a thin flat plate-like object set at a high angle of attack relative to the incoming flow. The temporal model predictions replicate key flow features successfully,



**FIG. 10.** Ground truth (top) and predicted (bottom) vorticity fields and percentage error maps for a snapshot from shape C using the large sensor setup. Vorticity colormap constrained to 5% of  $\max(|\hat{\omega}_{t,i}|)$ .

correctly predicting the location of the two shed vortices behind the object, demonstrating that the model is effective at capturing the mechanisms of advection in this flow. Some notable sources of error are the under-estimation of the intensity of these

vortices and the missing concentration of high positive vorticity on the lower surface of the object, which is uncharacteristic, given the good performance of SD-FNO predictions in Sec. V A in this regard.

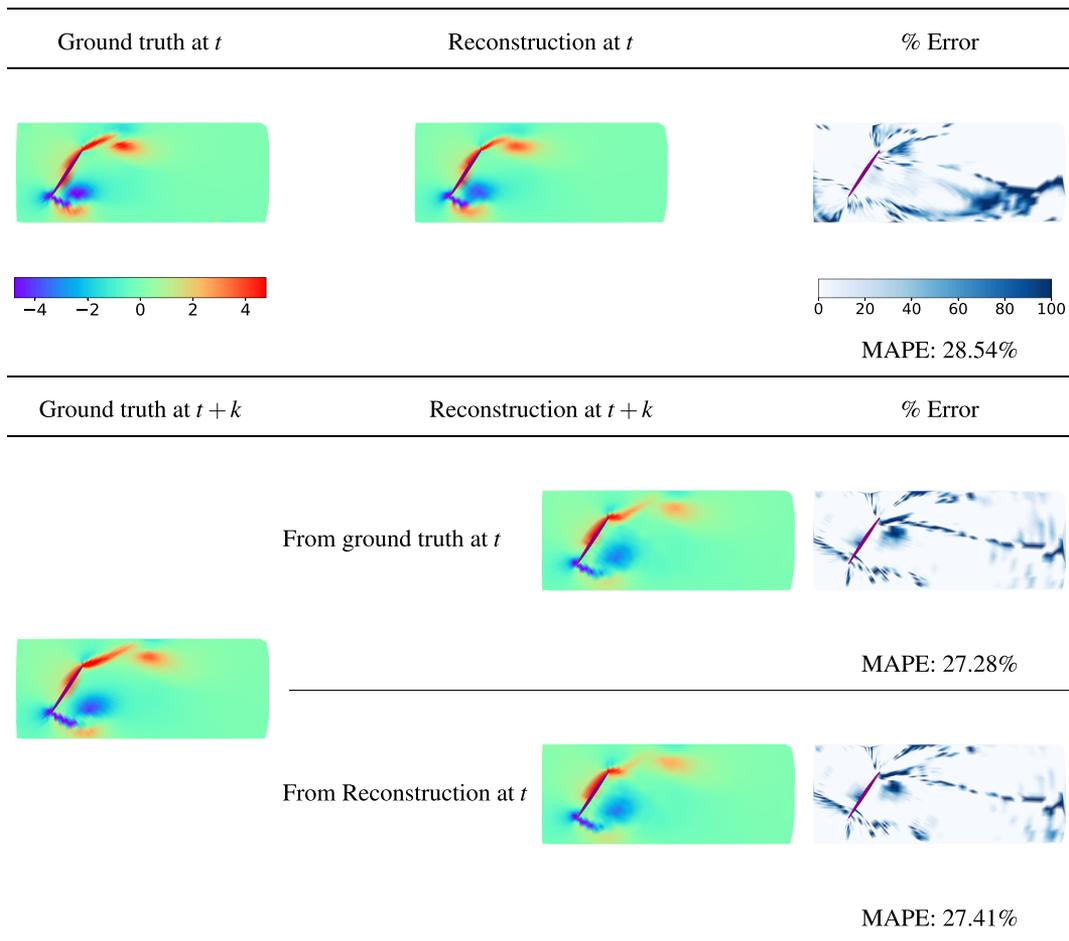
**TABLE V.** % errors of the temporal model predictions, averaged over the validation set, given ground truth snapshots and reconstructed snapshots (i.e., spatial model outputs) as inputs. Outliers filtered identically to Table IV. Only the temporal model was re-trained for different values of  $k$ .

$k(\Delta\tau^*)$	0 (0.0)		20 (0.667)		80 (2.667)	
	MAPE	HV-MAPE	MAPE	HV-MAPE	MAPE	HV-MAPE
From ground truth	19.76%	10.75%	23.40%	11.75%	29.58%	19.53%
From reconstruction	28.89%	17.86%	31.02%	17.86%	31.88%	21.97%

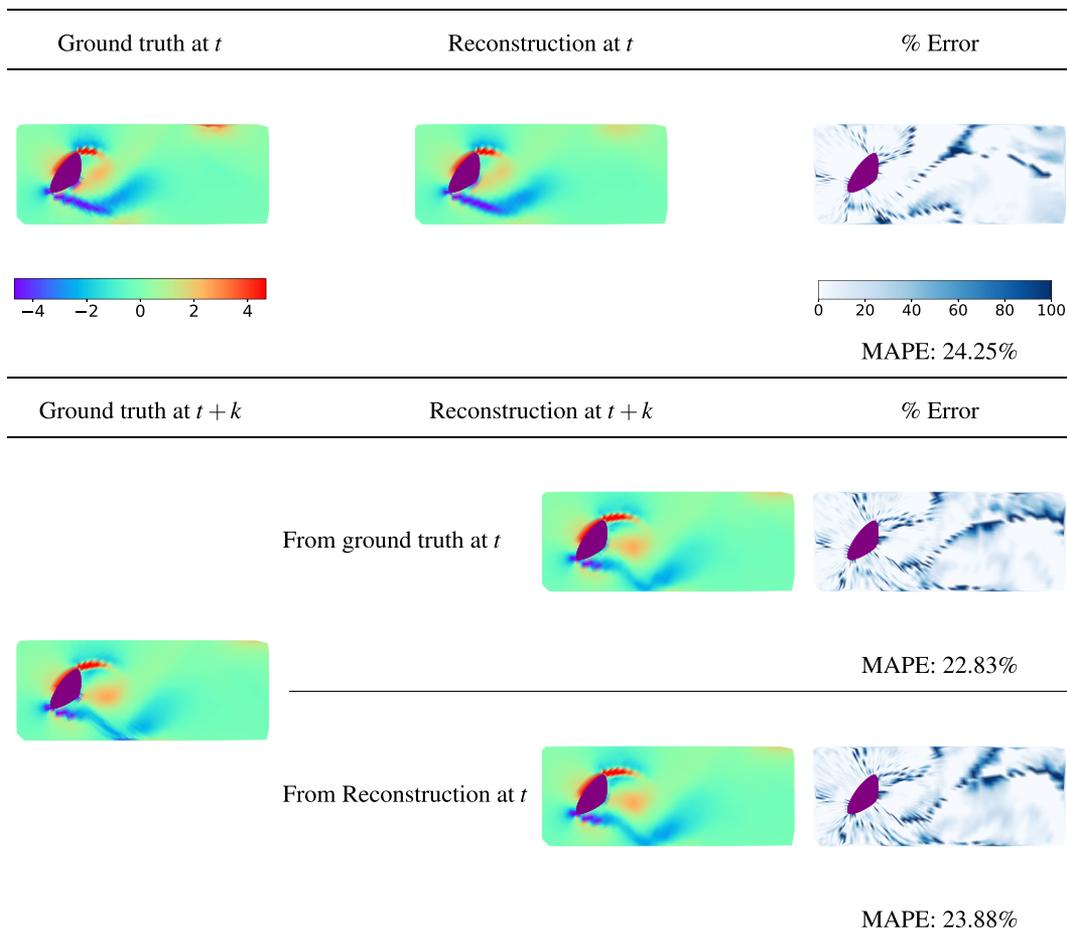
Shape E (Fig. 12) is a bluff body-like object similar to shape A (Fig. 6). The temporal reconstruction in Fig. 12 has the lowest MAPE among the three snapshots presented in Figs. 11–13. The phenomenon of note in this example is that, unlike the previous example where the most challenging aspect of the problem was predicting the advection of vortices, the model has to predict the *formation*

of a vortex at time  $t + k$  from the snapshot at time  $t$ , in this instance immediately behind the object. This is executed very well by the model, as evidenced by the error map, which shows very low error (below 10% throughout) for the region corresponding to the vortex.

Finally, shape F in Fig. 13 is a thick airfoil-like shape, also set at a high incidence angle relative to the incoming flow. The



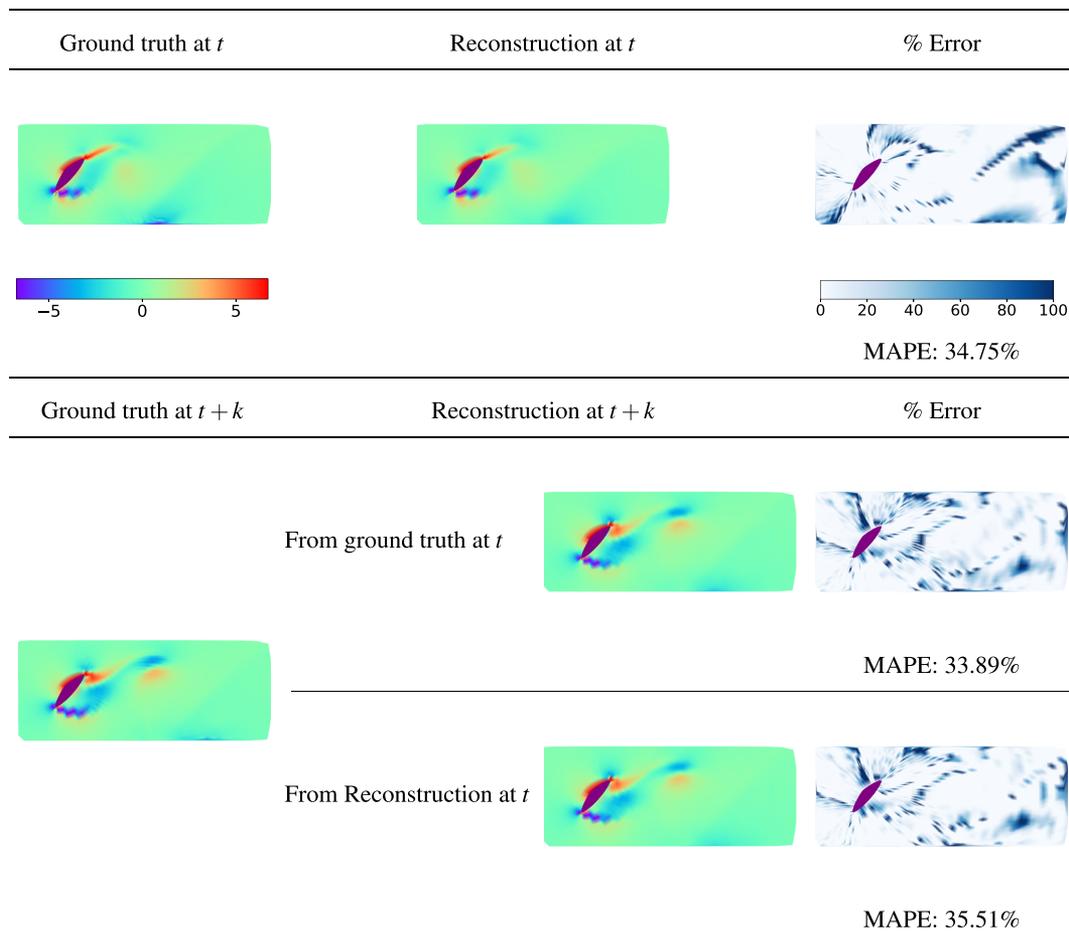
**FIG. 11.** STMGFR as applied to a snapshot from shape D. The left column shows the ground truth vorticity fields at time  $t$  (top) and  $t + k$  (bottom). The middle column depicts the reconstruction of the snapshot at  $t$  from sensors by the spatial model (top); the reconstruction of the snapshot at  $t + k$ , given the ground truth snapshot by the temporal model (middle); and the reconstruction of the snapshot at  $t + k$ , given the spatial model reconstruction (bottom). The right column displays the corresponding error maps. Vorticity colormap constrained to 5% of  $\max(|\hat{\omega}_{t+k,j}|)$ .



**FIG. 12.** STMGFR as applied to a snapshot from shape E. The left column shows the ground truth vorticity fields at time  $t$  (top) and  $t + k$  (bottom). The middle column depicts the reconstruction of the snapshot at  $t$  from sensors by the spatial model (top); the reconstruction of the snapshot at  $t + k$ , given the ground truth snapshot by the temporal model (middle); and the reconstruction of the snapshot at  $t + k$ , given the spatial model reconstruction (bottom). The right column displays the corresponding error maps. Vorticity colormap constrained to 5% of  $\max(|\hat{\omega}_{t+k,j}|)$ .

challenge present in this example is essentially the combination of those in Figs. 11 and 12; there is no vortical structure present at time  $t$ , but by time  $t + k$ , vortical structures have already formed and advected downstream from the object. This more complicated challenge translates to relatively higher quantitative error metrics for the snapshot depicted in Fig. 13 compared to the previous examples, but critical flow features are very well replicated at time  $t + k$ . The two shed counter-rotating vortices are predicted at the correct location with correct intensity, the shape of the interface between high and low vorticity regions immediately downstream of the object is correct, and the highly rotational intense separation regions past the leading and trailing edges of the object are present. Overall, despite the substantially more difficult challenge in this example, the model runs well and offers a good understanding of key fluid dynamics phenomena in this flow.

The good performance of the temporal model when given ground truth snapshots as inputs is largely consistent with the previous literature on the FNO,<sup>13</sup> where the capability of the FNO to time-march the vorticity field for turbulence-in-a-box settings was demonstrated with low error levels. The additional insight, however, is that these results demonstrate that the FNO model coupled with our training methodology (whereby 50% of the inputs are randomly replaced with spatial model predictions) is robust to noisy inputs, with a loss of accuracy of the order of 2% (as shown in Table V) despite average errors in the input approaching 40%. In fact, in two of the three cases displayed in Figs. 11–13, the MAPE of the temporal model prediction from the spatial model reconstruction relative to the ground truth at time  $t + k$  is lower than the MAPE of the spatial model reconstruction relative to the ground truth at time  $t$ . In a physical experimental



**FIG. 13.** STMGFR as applied to a snapshot from shape F. The left column shows the ground truth vorticity fields at time  $t$  (top) and  $t + k$  (bottom). The middle column depicts the reconstruction of the snapshot at  $t$  from sensors by the spatial model (top); the reconstruction of the snapshot at  $t + k$ , given the ground truth snapshot by the temporal model (middle); and the reconstruction of the snapshot at  $t + k$ , given the spatial model reconstruction (bottom). The right column displays the corresponding error maps. Vorticity colormap constrained to 3% of  $\max(|\hat{\omega}_{t+k,j}|)$ .

setting, where measurement noise is a real concern, this is a key capability as the impact of the measurement errors on the spatial reconstruction will not catastrophically degrade the accuracy of the temporal reconstruction.

#### D. Training and inference time

The final topic of discussion for comparing the relative merits of the different model architectures in Sec. V is the computational cost of training and using each model. Table VI outlines the wall-clock runtimes for running training [conducted on an IBM AC922 system with two 20-core POWER9 central processing units (CPUs) and two Nvidia V100 GPUs] and inference (conducted with an AMD EPYC 7443 CPU and a single Nvidia A100 GPU) using each model using the single-precision floating point format.

While the SD-UNet and SD-FNO consistently displayed better performance in terms of numerical error metrics and qualitative factors, such as correct predictions of locations and intensities of vortical structures in Sec. VI B, this improvement comes at a substantial cost in terms of training and model runtime. This can substantially hinder the relative usefulness of the SD-UNet and SD-FNO in a laboratory setting where real-time reconstruction of flow on lower-power devices is desired.

The runtime costs for the (temporal) FNO model used as the temporal component of the approach to STMGFR are even higher than the spatial models, although it is likely that an implementation-specific issue was present as GPU power draw was observed to be substantially lower for this model during inference than the four spatial architectures despite high utilization statistics.

**TABLE VI.** Wall-clock runtimes for training and running the models in Secs. VI B and VI C for the large sensor setup. Training runtime statistics are compiled for the training dataset (consisting of  $601 \times 64 = 38\,464$  snapshots); inference statistics are for the validation dataset (consisting of  $601 \times 16 = 9\,616$  snapshots). The training time column shows the *total* training time (equal to  $[\text{Time/epoch}] \times [\text{No. of epochs}]$ ).

	Model	Batch size	Batches/epoch	Time/epoch (s)	No. of epochs	Training time (s)	Inference time (s)
Spatial	SD	500	77	5	813	4 065	0.76
	SD-large	500	77	7	560	3 920	1.52
	SD-UNet	48	802	157	183	28 731	8.98
	SD-FNO	50	770	147	196	28 812	6.91
Temporal	FNO ( $k = 0$ )	100	372	187	248	46 376	44

## VII. CONCLUSION AND FUTURE WORK

This work introduced the Spatial and Spatiotemporal Multi-Geometry Flow Reconstruction (SMGFR, STMGFR) tasks for reconstructing dense contemporaneous or future vorticity fields of flows past arbitrary objects from current sparse sensor measurements, respectively, without geometry-specific training. To achieve optimal performance in these tasks, the use of Schwarz–Christoffel mappings to choose the sampling points of the dense fields was explored.

The performance of four different models was investigated on the SMGFR task using datasets generated via both the novel mapping aided sampling strategy and a more traditional Cartesian sampling strategy with three different sensor setups. The results showed that the mapping aided approach provides a substantial boost in accuracy for all model and sensor setup configurations, enabling percentage errors under 3%, 10%, and 30% for reconstructions of pressure, velocity, and vorticity fields, respectively. Improvements in terms of mean absolute percentage error exceed 15% points in select cases for the challenging vorticity reconstruction tasks, while the impact of the size of the sensor setup is modest. The best performing model architecture was a convolutional architecture based on the U-Net.<sup>12</sup> Comparisons of snapshots from the different configurations revealed that the usage of the mapping approach substantially boosted the accuracy of the predictions in the immediate vicinity of the objects.

For the novel STMGFR task involving the prediction of future snapshots, given current sensor measurements, an innovative approach separating the task into spatial and temporal components was developed, whereby a spatial model first reconstructs the current snapshot, given the current measurements (equivalent to the SMGFR task), and subsequently, a temporal model predicts the future snapshot, given the predicted current snapshot. A stack of Fourier neural operator<sup>13</sup> layers acting as the temporal model was coupled with the best performing configuration from the SMGFR experiments used as the spatial model. The temporal model was trained to be robust to input noise caused by inaccuracies in spatial model predictions, by randomly providing it spatial model predictions or ground truth snapshots for each sample during training. Experimentation indicated that this approach is capable of accurately reconstructing future vorticity snapshots with mean absolute percentage error levels on the order of 30%. Furthermore, using a temporal gap of zero, the same two model setup can be used to

improve accuracy of models used in SMGFR, also bringing their MAPE levels below 30%.

We hope to expand the investigations in this work in the future by the following steps:

- Developing models that perform well over a range of Reynolds numbers. The present work focused on predictions for flows at  $\text{Re} \approx 300$ ; the models presented are not expected to perform well for other  $\text{Re}$  and likely require re-training to adjust to different Reynolds numbers. Two potential ways of overcoming this are using a dataset containing snapshots from a range of Reynolds numbers and using “physics-informed” loss functions.
- Experimentation with more advanced neural network architectures. While this work focused on the relatively simple case of supervised training of feedforward and convolutional architectures to focus on investigating the mapping approach presented, techniques such as generative adversarial networks<sup>34</sup> and variational autoencoders<sup>44</sup> are gaining traction in FR literature. Adoption of such techniques may lead to higher accuracy in SMGFR and STMGFR tasks.
- Extending the methodology to 3D fluid flows. The mapping approach in this work relies on the Schwarz–Christoffel mapping, which is defined for the complex plane  $\mathbb{C}$  only. A version of this work for  $\mathbb{R}^3$  will require an alternative mapping approach.
- Prediction of lift and drag coefficients. Although this work focused on the reconstruction of vorticity fields for the ease of comparison with previous works, changing the target fields to velocity and pressure can permit for the prediction of the lift and drag coefficients. The mapping approach is especially conducive to this as, unlike Cartesian sampling, it eliminates the need to further process the vorticity and pressure fields to obtain values at the object boundary.

## ACKNOWLEDGMENTS

This work was supported by a Ph.D. studentship funded by the Department of Aeronautics, Imperial College London, and an Academic Hardware Grant provided by Nvidia. The authors would like to thank Sean Chai, Neil Ashton, and Thomas Delillo for fruitful discussions at the start of this study.

**AUTHOR DECLARATIONS**

**Conflict of Interest**

The authors have no conflicts to disclose.

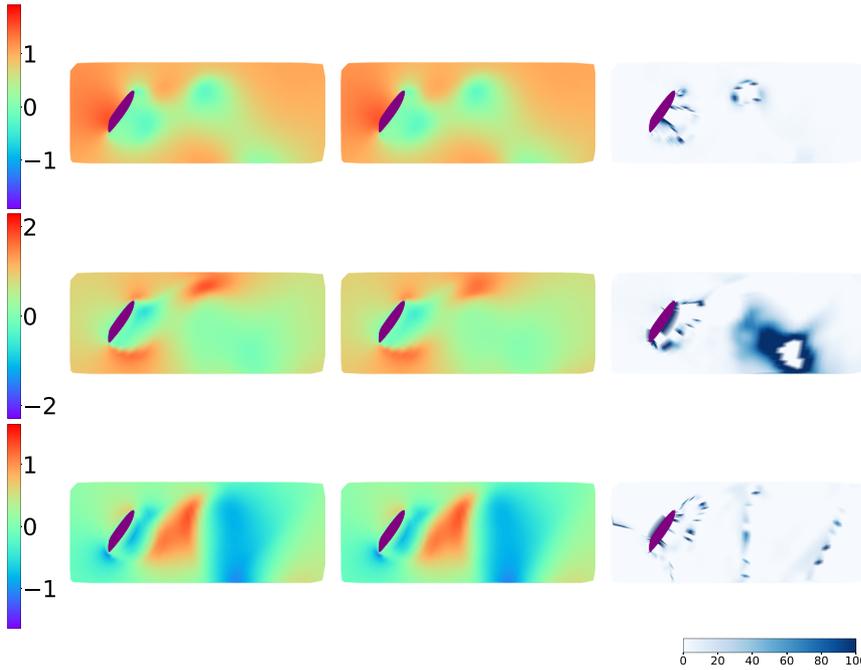
**DATA AVAILABILITY**

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

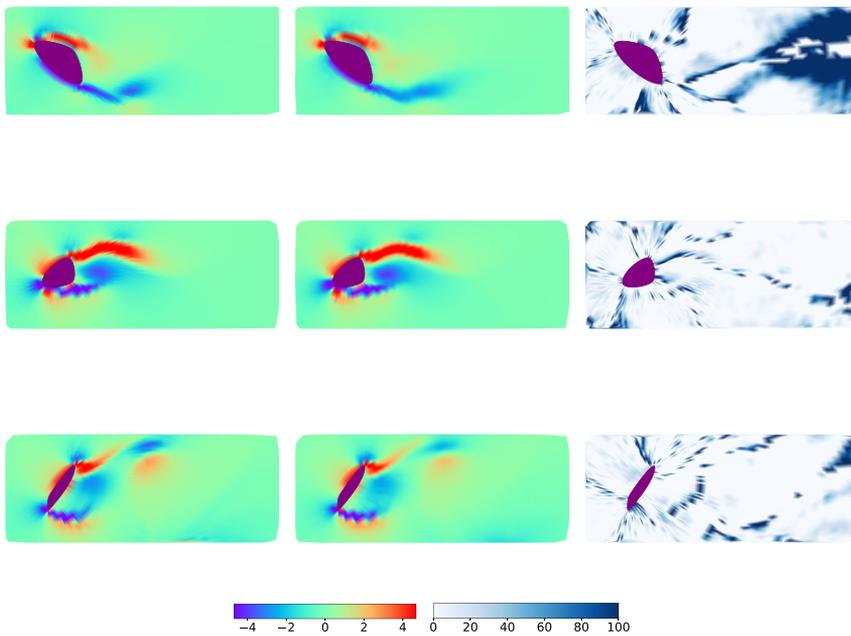
pydscpack library for computing Schwarz–Christoffel mappings and the code for replicating the results in this work can be found on GitHub.

**APPENDIX A: VELOCITY AND PRESSURE PREDICTIONS USING THE SPATIOTEMPORAL MODEL WITH A ZERO TEMPORAL GAP ( $k = 0$ )**

Ground truth (left), predicted (middle), and percentage error (right) fields for pressure (top),  $u$ -velocity (middle), and  $v$ -velocity (bottom) for a validation snapshot.



**FIG. 14.** Ground truth (left), predicted (middle), and percentage error (right) fields for pressure (top),  $u$ -velocity (middle), and  $v$ -velocity (bottom) for a validation snapshot.



**FIG. 15.** Examples using the spatial (SD-UNet) plus temporal (FNO) model architecture with  $k = 0$  from Sec. V B. The left column contains the ground truth snapshots, the middle column contains the predictions, and the right column contains the percentage error maps.

$v$ -velocity (bottom) for a validation snapshot are shown in Fig. 14.

## APPENDIX B: VORTICITY PREDICTIONS USING THE SPATIOTEMPORAL MODEL WITH A ZERO TEMPORAL GAP ( $k = 0$ )

Examples using the spatial (SD-UNet) plus temporal (FNO) model architecture with  $k = 0$  from Sec. V B are given in Fig. 15.

## REFERENCES

- <sup>1</sup>I. Grant, "Particle image velocimetry: A review," *Proc. Inst. Mech. Eng., Part C* **211**, 55–76 (1997).
- <sup>2</sup>C. J. Elkins and M. T. Alley, "Magnetic resonance velocimetry: Applications of magnetic resonance imaging in the measurement of fluid motion," *Exp. Fluids* **43**, 823–858 (2007).
- <sup>3</sup>V. Rajan, B. Varghese, T. G. van Leeuwen, and W. Steenbergen, "Review of methodological developments in laser Doppler flowmetry," *Lasers Med. Sci.* **24**, 269–283 (2009).
- <sup>4</sup>N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz, "Shallow neural networks for fluid flow reconstruction with limited sensors," *Proc. R. Soc. London, Ser. A* **476**, 20200097 (2020).
- <sup>5</sup>R. Everson and L. Sirovich, "Karhunen–Loève procedure for gappy data," *J. Opt. Soc. Am. A* **12**, 1657–1664 (1995).
- <sup>6</sup>D. Venturi and G. E. Karniadakis, "Gappy data and reconstruction procedures for flow past a cylinder," *J. Fluid Mech.* **519**, 315–336 (2004).
- <sup>7</sup>R. J. Adrian and P. Moin, "Stochastic estimation of organized turbulent structure: Homogeneous shear flow," *J. Fluid Mech.* **190**, 531–559 (1988).
- <sup>8</sup>P. Dubois, T. Gomez, L. Planckaert, and L. Perret, "Machine learning for fluid flow reconstruction from limited measurements," *J. Comput. Phys.* **448**, 110733 (2022).
- <sup>9</sup>K. Fukami, K. Fukagata, and K. Taira, "Assessment of supervised machine learning methods for fluid flows," *Theor. Comput. Fluid Dyn.* **34**, 497–519 (2020).
- <sup>10</sup>L. Sun and J.-X. Wang, "Physics-constrained Bayesian neural network for fluid flow reconstruction with sparse and noisy data," *Theor. Appl. Mech. Lett.* **10**, 161–169 (2020).
- <sup>11</sup>D. Crowdy, *Solving Problems in Multiply Connected Domains* (SIAM, 2020).
- <sup>12</sup>O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2015), pp. 234–241.
- <sup>13</sup>Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," in *ICLR 2021* (2021), available at <https://openreview.net/pdf?id=c8P9NQVtmnO>.
- <sup>14</sup>E. De Vito, L. Rosasco, A. Caponnetto, U. De Giovannini, F. Odone, and P. Bartlett, "Learning from examples as an inverse problem," *J. Mach. Learn. Res.* **6**, 883 (2005).
- <sup>15</sup>K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. M. Dawson, and C.-A. Yeh, "Modal analysis of fluid flows: Applications and outlook," *AIAA J.* **58**, 998–1022 (2020).
- <sup>16</sup>C. H. Colburn, J. B. Cessna, and T. R. Bewley, "State estimation in wall-bounded flow systems. Part 3. The ensemble Kalman filter," *J. Fluid Mech.* **682**, 289–303 (2011).
- <sup>17</sup>N. Gustafsson, T. Janjić, C. Schraff, D. Leuenberger, M. Weissmann, H. Reich, P. Brousseau, T. Montmerle, E. Wattlelot, A. Bučánek, M. Mile, R. Hamdi, M. Lindskog, J. Barkmeijer, M. Dahlbom, B. Macpherson, S. Ballard, G. Inverarity, J. Carley, C. Alexander, D. Dowell, S. Liu, Y. Ikuta, and T. Fujita, "Survey of data assimilation methods for convective-scale numerical weather prediction at operational centres," *Q. J. R. Meteorol. Soc.* **144**, 1218–1256 (2018).
- <sup>18</sup>F.-X. Le Dimet and O. Talagrand, "Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects," *Tellus* **38**, 97–110 (1986).
- <sup>19</sup>P. Courtier, "Dual formulation of four-dimensional variational assimilation," *Q. J. R. Meteorol. Soc.* **123**, 2449–2461 (1997).
- <sup>20</sup>D. P. G. Foures, N. Dovetta, D. Sipp, and P. J. Schmid, "A data-assimilation method for Reynolds-averaged Navier–Stokes-driven mean flow reconstruction," *J. Fluid Mech.* **759**, 404–431 (2014).
- <sup>21</sup>C. He and Y. Liu, "Time-resolved reconstruction of turbulent flows using linear stochastic estimation and sequential data assimilation," *Phys. Fluids* **32**, 075106 (2020).
- <sup>22</sup>D. Butcher and A. Spencer, "Linear stochastic estimation of the coherent structures in internal combustion engine flow," *Int. J. Engine Res.* **21**, 1738–1749 (2020).
- <sup>23</sup>B. Podvin, S. Ngumatsia, J.-M. Foucaut, C. Cuvier, and Y. Fraigneau, "On combining linear stochastic estimation and proper orthogonal decomposition for flow reconstruction," *Exp. Fluids* **59**, 58 (2018).
- <sup>24</sup>K. Willcox, "Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition," *Comput. Fluids* **35**, 208–226 (2006).
- <sup>25</sup>P. Saini, C. M. Arndt, and A. M. Steinberg, "Development and evaluation of gappy-POD as a data reconstruction technique for noisy PIV measurements in gas turbine combustors," *Exp. Fluids* **57**, 122 (2016).
- <sup>26</sup>J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**, 210–227 (2009).
- <sup>27</sup>J. L. Callahan, K. Maeda, and S. L. Brunton, "Robust flow reconstruction from limited measurements via sparse representation," *Phys. Rev. Fluids* **4**, 103907 (2019).
- <sup>28</sup>L. Espath, D. Kabanov, J. Kiessling, and R. Tempone, "Statistical learning for fluid flows: Sparse Fourier divergence-free approximations," *Phys. Fluids* **33**, 097108 (2021).
- <sup>29</sup>S. Sonoda and N. Murata, "Neural network with unbounded activation functions is universal approximator," *Appl. Comput. Harmonic Anal.* **43**, 233–268 (2017).
- <sup>30</sup>T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," in *NeurIPS Proceedings* (2020), available at <https://papers.nips.cc/paper/2020/hash/1457c0d6bfb4967418bfb8ac142f64a-Abstract.html>.
- <sup>31</sup>T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," [arXiv:1912.04958](https://arxiv.org/abs/1912.04958) [cs.CV] (2020).
- <sup>32</sup>Y. Kumar, P. Bahl, and S. Chakraborty, "State estimation with limited sensors—A deep learning based approach," *J. Comput. Phys.* **457**, 111081 (2020).
- <sup>33</sup>D. W. Carter, F. De Voogt, R. Soares, and B. Ganapathisubramani, "Data-driven sparse reconstruction of flow over a stalled aerofoil using experimental data," *Data-Centric Eng.* **2**, e5 (2021).
- <sup>34</sup>J. Chen, E. Hachem, and J. Viquerat, "Graph neural networks for laminar flow prediction around random two-dimensional shapes," *Phys. Fluids* **33**, 123607 (2021).
- <sup>35</sup>J. Blom, "Some characteristic quantities of Karman–Trefftz profiles," Technical Report No. NASA-TM-77013, 1983.
- <sup>36</sup>C. Hu, "Algorithm 785: A software package for computing Schwarz–Christoffel conformal transformation for doubly connected polygonal regions," *ACM Trans. Math. Software* **24**, 317–333 (1998).
- <sup>37</sup>T. A. Driscoll and L. N. Trefethen, *Schwarz–Christoffel Mapping*, Cambridge Monographs on Applied and Computational Mathematics (Cambridge University Press, 2002).
- <sup>38</sup>L. N. Trefethen, "Numerical computation of the Schwarz–Christoffel transformation," *SIAM J. Sci. Stat. Comput.* **1**, 82–102 (1980).
- <sup>39</sup>Vertex coordinates in the  $w$  domain.
- <sup>40</sup>J. Viquerat, J. Rabault, A. Kuhnle, H. Ghraieb, A. Larcher, and E. Hachem, "Direct shape optimization through deep reinforcement learning," *J. Comput. Phys.* **428**, 110080 (2021).
- <sup>41</sup>F. D. Witherden, A. M. Farrington, and P. E. Vincent, "PyFR: An open source framework for solving advection–diffusion type problems on streaming architectures using the flux reconstruction approach," *Comput. Phys. Commun.* **185**, 3028–3040 (2014).
- <sup>42</sup>H. T. Huynh, "A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods," in *18th AIAA Computational Fluid Dynamics Conference* (AIAA, 2007), p. 4079.

- <sup>43</sup>L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training DNNs: Methodology, analysis and application," [arXiv:2009.12836](https://arxiv.org/abs/2009.12836) [cs.LG] (2020).
- <sup>44</sup>K. Gundersen, A. Oleynik, N. Blaser, and G. Alendal, "Semi-conditional variational auto-encoder for flow reconstruction and uncertainty quantification from limited observations," *Phys. Fluids* **33**, 017119 (2021).
- <sup>45</sup>D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG] (2017).
- <sup>46</sup>M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, "TensorFlow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (Advanced Computing Systems Association, 2016), pp. 265–283.
- <sup>47</sup>S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning* (PMLR, 2015), pp. 448–456.
- <sup>48</sup>A. F. Agarap, "Deep learning using rectified linear units (ReLU)," [arXiv:1803.08375](https://arxiv.org/abs/1803.08375) [cs.NE] (2019).
- <sup>49</sup>Y. Dodge, "Spearman rank correlation coefficient," in *The Concise Encyclopedia of Statistics* (Springer, New York, 2008), pp. 502–505.
- <sup>50</sup>T. M. Cover, J. A. Thomas *et al.*, "Entropy, relative entropy and mutual information," *Elem. Inf. Theory* **2**, 12–49 (1991).
- <sup>51</sup>F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
- <sup>52</sup>L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.* **40**, 834–848 (2018).
- <sup>53</sup>I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016), pp. 504–505, <http://www.deeplearningbook.org>.
- <sup>54</sup>H. Kim, J. Kim, S. Won, and C. Lee, "Unsupervised deep learning for super-resolution reconstruction of turbulence," *J. Fluid Mech.* **910**, A29 (2021).