Check for updates

Received: 14 February 2024 | Revised: 27 May 2024 | Accepted: 20 July 2024

DOI: 10.1002/cnm.3858

APPLIED RESEARCH

WILEY

A comparison of machine learning methods for recovering noisy and missing 4D flow MRI data

Hunor Csala[1,2] | **Omid Amili**[3] | **Roshan M. D'Souza**[4] | **Amirhossein Arzani**[1,2]

[1]Department of Mechanical Engineering, University of Utah, Salt Lake City, Utah, USA

[2]Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, Utah, USA

[3]Department of Mechanical, Industrial and Manufacturing Engineering, University of Toledo, Toledo, Ohio, USA

[4]Department of Mechanical Engineering, University of Wisconsin-Milwaukee, Milwaukee, Wisconsin, USA

Correspondence
Amirhossein Arzani, Department of Mechanical Engineering, University of Utah, Salt Lake City, Utah, USA.
Email: amir.arzani@sci.utah.edu

Abstract

Experimental blood flow measurement techniques are invaluable for a better understanding of cardiovascular disease formation, progression, and treatment. One of the emerging methods is time-resolved three-dimensional phase-contrast magnetic resonance imaging (4D flow MRI), which enables noninvasive time-dependent velocity measurements within large vessels. However, several limitations hinder the usability of 4D flow MRI and other experimental methods for quantitative hemodynamics analysis. These mainly include measurement noise, corrupt or missing data, low spatiotemporal resolution, and other artifacts. Traditional filtering is routinely applied for denoising experimental blood flow data without any detailed discussion on why it is preferred over other methods. In this study, filtering is compared to different singular value decomposition (SVD)-based machine learning and autoencoder-type deep learning methods for denoising and filling in missing data (imputation). An artificially corrupted and voxelized computational fluid dynamics (CFD) simulation as well as in vitro 4D flow MRI data are used to test the methods. SVD-based algorithms achieve excellent results for the idealized case but severely struggle when applied to in vitro data. The autoencoders are shown to be versatile and applicable to all investigated cases. For denoising, the in vitro 4D flow MRI data, the denoising autoencoder (DAE), and the Noise2Noise (N2N) autoencoder produced better reconstructions than filtering both qualitatively and quantitatively. Deep learning methods such as N2N can result in noise-free velocity fields even though they did not use clean data during training. This work presents one of the first comprehensive assessments and comparisons of various classical and modern machine-learning methods for enhancing corrupt cardiovascular flow data in diseased arteries for both synthetic and experimental test cases.

KEYWORDS

data imputation, deep learning, denoising, hemodynamics, sparse data-driven modeling

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.
© 2024 The Author(s). *International Journal for Numerical Methods in Biomedical Engineering* published by John Wiley & Sons Ltd.

1 | INTRODUCTION

Understanding complex hemodynamics phenomena is crucial for better cardiovascular disease diagnosis and treatment. Both experimental and computational hemodynamics modeling methods have improved significantly in the past decades. Nonetheless, obtaining clean, high-resolution blood flow velocity measurements experimentally inside relatively smaller arteries such as the cerebral vasculature is challenging, as the existing experimental techniques have several limitations. Time-resolved three-dimensional phase-contrast magnetic resonance imaging (4D flow MRI) is a popular approach in research settings,[1] but it is also becoming available in multiple medical centers for clinical use. It can provide noninvasive 3D time-resolved velocity vector field measurements in both in in vivo and in vitro settings. However, it is constrained by low spatiotemporal resolution, noise, and other artifacts,[1] like velocity aliasing or phase offset errors associated with eddy currents. The typical spatial resolution of 4D flow MRI in in vitro research settings is around 0.5–2 mm, with 15–25 snapshots per cardiac cycle.[2] While submillimeter resolutions can be achieved in ultra-high-field MRI in in vivo settings,[3] the typical resolution is around 2–3 mm in common clinical settings[1] for large arteries and whole heart scans. Noise and other corruption are usually handled by postprocessing software.[1] With all its limitations, 4D flow MRI is rapidly growing as patient-specific hemodynamic analysis is becoming more important for identifying abnormalities associated with various cardiovascular disorders.

Computational fluid dynamics (CFD) simulations provide an alternative approach. CFD can produce velocity and pressure fields at orders of magnitude higher spatiotemporal resolutions than experimental techniques, and high-resolution CFD is practically noise-free. Nonetheless, experimental data are still indispensable for validation and patient-specific boundary condition information. Other measurement techniques like particle image velocimetry (PIV), often considered the gold standard in experimental fluid dynamics, also suffer from limitations. For example, PIV can also produce corrupt data,[4,5] for example, background noise, pixel corruptions, or missing data.

Handling corrupt experimental blood flow data is a key challenge that needs to be addressed for more accurate and robust cardiovascular flow modeling. There are well-developed algorithms in the machine learning community that can tackle similar issues, such as data imputation, denoising, outlier detection, super-resolution, or inpainting.[6–9] Data imputation, also known as matrix completion, undertakes the challenge of filling in missing entries in a partially observed matrix. Candès et al.[10] presented near-optimal matrix completion for data missing completely at random. However, in most real-world scenarios, data is missing in certain locations that are not entirely random, even if there is some underlying element of randomness. Sun et al.[6] reviewed and compared statistical, conventional machine learning, and recent deep learning approaches for filling in point-wise missing data. They showed that different imputation techniques can lead to distinct results depending on the pattern of missing data. When large connected chunks of the data are missing, gappy proper orthogonal decomposition (gappy POD)[11] or gappy spectral proper orthogonal decomposition (gappy SPOD)[12] can be leveraged, introduced by the fluid dynamics community. Lee et al.[13] provided a unified framework on the relationship between gappy POD and probabilistic principal component analysis (PPCA), and applied the method for filling in missing values in fluid dynamics data. Generative deep learning models also have a strong potential for data imputation and enhancement. Diffusion models have been used for super-resolution of fluid flows.[14] Theoretically, they could also be applied for filling in large missing parts of the data, similar to the image inpainting problem.[8] Generative adversarial networks (GANs) have also shown promising results for super-resolving fluid flows.[15,16] Fluid dynamics and data imputation share an intriguing history, as video inpainting methods were developed based on an analogy between the vorticity-streamfunction formulation of Navier–Stokes and image pixel intensity and smoothness values.[17] Imputation methods have been occasionally investigated for blood flow problems. Arzani and Dawson[18] demonstrated the feasibility of low-rank matrix completion for hemodynamics in 2D and 3D diseased arteries. Gunn et al.[19] filled in missing data using trace norm minimization in cerebral blood flow velocity signals. Similar techniques based on low-rank structures and sparse representations were used for accelerated[20] and multi-modality reconstruction of 4D flow MRI.[21]

Denoising is another interesting problem related to improving corrupt data. Denoising is the task of separating a signal from the noise and it has been extensively studied in the field of image processing.[22] A brief review of the most widely used image denoising algorithms is presented in Reference 7, where methods such as filtering, nonlocal means (NLM), block-matching and 3D filtering (BM3D), low-rank minimization and sparse representation-based approaches, and deep learning techniques are discussed. The most well-known denoising neural network (NN) architectures are known as denoising autoencoders (DAE).[23,24] They reduce the dimensionality of the data into a latent space and are trained with noisy and clean data pairs. Dimensionality reduction is a powerful tool that has seen increasing popularity in fluid flow modeling. Different use cases include development of reduced-order models (ROMs),[25,26] data reduction,[27,28] visualization,[29,30] and also denoising.[23,31–33] DAEs are the cornerstone of supervised denoising and have been widely used in many applications, such as image processing[34] or seismology.[35] Often, these networks were

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

employed in scenarios where large amounts of clean data were easily accessible, such as noise-free images. However, obtaining clean data might be difficult or even impossible to acquire in some applications. Therefore, semi-supervised, low-supervised, and unsupervised techniques are needed for such cases. The denoising revolution without clean data started with the introduction of Noise2Noise (N2N)[36] and Noise2Void[37] techniques. These methods leveraged a second noisy sample or the input noisy image itself and could achieve similar performance to conventional training with clean data. State-of-the-art deep learning denoising algorithms achieve remarkable performance for real-world camera and microscopy images, both in supervised,[36] unsupervised,[38,39] and self-supervised[40] scenarios. Recently, even zero-shot learning algorithms have been proposed.[41,42] Most methods rely on the main assumption that noise is pixel-wise independent,[37] while the underlying signal is not. A more conventional approach for unsupervised denoising is robust principal component analysis (RPCA),[43] which assumes a low-rank structure in the signal and sparsity in the noise. RPCA was previously used for benchmark fluid flow problems and PIV data,[33] and for cardiovascular flows[18,44] with promising results. A deep learning version of RPCA (robust autoencoder) was also proposed, mainly for outlier detection in the computer vision community.[45] There have been various previous works on denoising fluid flow fields based on filtering,[46,47] some form of PCA,[32,48] and deep learning techniques.[49,50] Nevertheless, most of these use cases were either 2D[32,33,44,50] or 3D CFD results with synthetic noise.[18]

In the context of 4D flow MRI, two general machine learning approaches are taken in the literature to handle noise. The first one relies on data assimilation, using either physics-informed neural networks (PINNs),[51–53] or conventional machine learning algorithms without NN[54,55] to denoise or super-resolve the velocity field. The second approach is to train deep NN with synthetic 4D flow MRI data in a supervised data-driven manner.[56–59]

In this work, we compare the performance of different machine learning methods in denoising and data imputation (filling in missing data) of 4D flow MRI data. We particularly focus on two broad classes of methods: singular value decomposition (SVD)-based models and deep learning. For each of the two broad classes, we investigate different approaches toward handling noisy and corrupt data. Given the complexity of offline training based on different complex patient-specific geometries,[60] we adopt a deep learning strategy different from most other studies (e.g.[59]). That is, we train the deep learning model specific to the patient by fixing the geometry. While this approach does not offer real-time predictions and needs to be retrained for each patient, as later discussed, it significantly reduces the need for large datasets and reduces the risk of input data being outside the training landscape.[60] It also significantly simplifies the training of the NN. For example, the training of 4DFlowNet in[56] took 10 days on a GPU. Additionally, we provide a comparison of these machine learning techniques with the basic filtering techniques[61] that are commonly used for 4D flow MRI data[62–67] and their performance compared to machine learning is not investigated in detail. In our comparison, we consider pulsatile blood flow in a synthetically corrupted and voxelized 3D internal carotid artery (ICA) aneurysm model and an in vitro 4D flow MRI aneurysm model.

The rest of the paper is structured as follows. In Section 2, we present the machine learning methods used for denoising and filling in missing data and the two test cases investigated are introduced. In Section 3, the denoising and imputation results are presented and different techniques are compared. In Section 4, the results are discussed and in Section 5 conclusions are summarized.

2 | METHODS

Below we explain the different machine learning methods used in this study to address denoising and data imputation (filling in missing data). For each category, we present different supervised and unsupervised machine learning approaches. Subsequently, we describe the test cases used to evaluate these methods.

2.1 | Denoising

Noise reduction, or denoising, is the process of removing noise from a signal. Usually, when real-world quantities are measured, the obtained data is the sum of the underlying signal and some noise embedded in it.

$$x_\zeta = x + \boldsymbol{\zeta}, \tag{1}$$

where x_ζ is the collected data (e.g., a velocity snapshot), x is the true data, and $\boldsymbol{\zeta}$ is noise. In denoising, the goal is to clean the data by trying to remove the noise $\boldsymbol{\zeta}$. Highly corrupted measurement data can be challenging to interpret and

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

may lead to nonphysical and incorrect conclusions. A key challenge during the denoising process is the possibility that the underlying signal can be altered. In training algorithms, it is common to generate noisy data from clean samples by artificially adding noise, as illustrated in Figure 1A for a snapshot of CFD data. The different denoising methods used in this study are presented below and an overview is shown in Figure 1.

2.1.1 | Filtering

The simplest and most commonly used traditional denoising algorithms use a form of filtering. They are fully unsupervised, computationally cheap, and easy to use. A common filtering technique for fluid flow data is the universal outlier detection filter,[61] which has been frequently used in post-processing PIV[5,68] and 4D Flow-MRI[62,67] data. Often this is combined with a median and Gaussian filter.[69] This approach is used here as well, where the outlier filter is followed by a median filter with a $3 \times 3 \times 3$ kernel and a Gaussian filter with a standard deviation of 0.5.

2.1.2 | Robust PCA

Principle component analysis (PCA) and SVD are at the heart of many machine-learning algorithms.[9] RPCA was introduced by Candes et al.[43] as an expansion of PCA to data with corrupted entries. RPCA is motivated by the observation that the data reduction offered by PCA is highly sensitive to noise[18,33] and offers an approach to remove the noise. First, it is assumed that the data matrix X is generated as the sum of two matrices:

$$X = L + S, \tag{2}$$

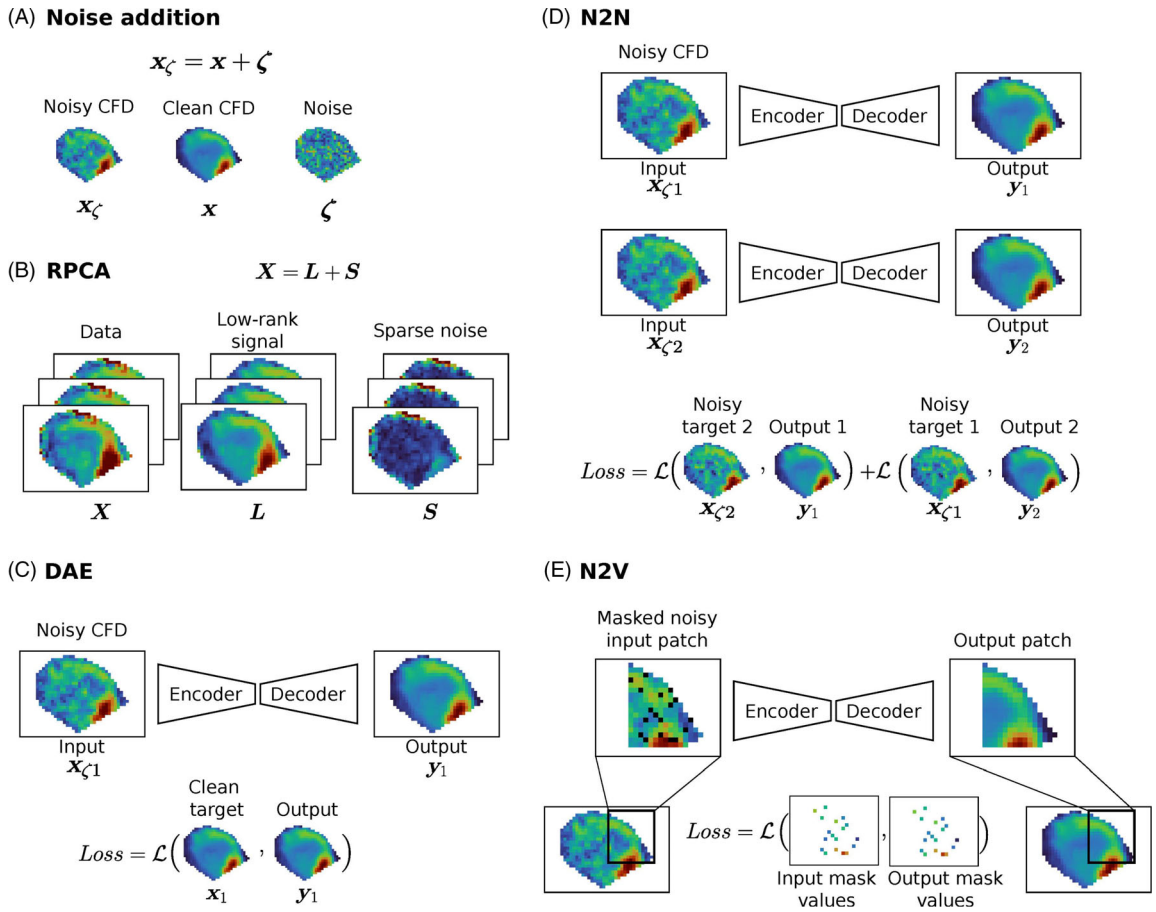

FIGURE 1 (A) Noise addition to CFD data for creating training data needed by some of the algorithms. (B) Robust principal component analysis (RPCA). (C) Denoising autoencoder (DAE) framework. (D) Noise2Noise (N2N) framework. (E) Noise2Void (N2V) framework.

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where L is a low-rank matrix containing the signal, and S is a sparse matrix containing the noise. In RPCA, the data matrix X is provided and the goal is to find the low-rank matrix L and the sparse matrix S that satisfy Equation 2. This can be cast as the following optimization problem:

$$\min_{L,S} \text{rank}(L) + \lambda \| S\|_0 \text{such that} X = L + S, \tag{3}$$

where $\| S\|_0$ is the number of nonzero entries in S and λ is a tuning parameter. If λ is too small, then L will tend to zero, while if λ is too large, then S will tend to zero. Equation 3 is a nonconvex problem that is in general NP-hard,[31] therefore its convex relaxation is used:

$$\min_{L,S} \| L\|_* + \lambda \| S\|_1 \text{such that} X = L + S, \tag{4}$$

where $\| \cdot \|_*$ is the nuclear norm (i.e., the sum of singular values) and $\| \cdot \|_1$ is the l_1-norm. The RPCA setup is illustrated in Figure 1B. For practical usage, fluid flow snapshots are stacked in a matrix $X \in \mathbb{R}^{(n \times m)}$, where n is the number of spatial locations (i.e., mesh points) and m is the number of temporal snapshots. For most real-world applications $m \ll n$, therefore X is tall-and-skinny. The velocity components are also stacked column-wise, therefore, the final size of the matrix in 3D is $X \in \mathbb{R}^{(3n \times m)}$.

A myriad of different versions and implementations of the RPCA algorithm exist; for example, principal component pursuit, augmented Lagrange multipliers, trimmed Grassmann averages, and alternating direction method. A comprehensive list of different RPCA algorithms and computational time comparisons was presented by Sobral et al.[70] In this study, an implementation based on the principal component pursuit from Candes et al.[43] is used. The hyperparameter λ was also set according to the suggestions from Candes et al.,[43] resulting in $\lambda = 0.006$ and $\lambda = 0.0026$ for the CFD and 4D flow MRI test cases, respectively.

2.1.3 | Supervised DAE

The simplest deep learning method for denoising is to train a fully connected or convolutional autoencoder (AE) in a supervised manner.[23,24] The input to the network is a noisy snapshot, while the output is compared with the ground-truth clean snapshot in the loss function, as shown in Figure 1C. Obviously, this requires access to clean data, which might not be available in many real-world scenarios, such as denoising PIV or 4D flow MRI data. Here, artificially corrupted CFD data is leveraged and used to train the AE in a supervised fashion and then tested on 4D flow MRI data. Data were rescaled to [0,1] for all NN cases. The specific NN architectures, hyperparameters, and computational resources will be described below for each test case separately.

2.1.4 | Noise2Noise

Lehtinen et al.[36] presented a NN-based denoising formulation called N2N that does not require any clean data. It uses two different samples from the same noise distribution, one as the input and one as the target. It was shown that N2N can achieve similar performance compared to supervised denoising with clean targets. Its capabilities were demonstrated for Gaussian, Poisson, and Bernoulli noise, text removal, and random-valued impulse noise.[36] Different loss functions can be used, such as l_0, l_1, and l_2, each working well in different scenarios. The N2N formulation is independent of the network architecture and can be used with any type of NN if a second set of noisy samples is available. The ground truth values are changed from clean to another noisy sample of the same snapshot in the loss function. In this context, the DAE approach explained above could also be called Noise2Clean.

For all test cases, the architecture of the N2N and DAE networks will be exactly the same. A symmetric loss[41] (also known as twin-loss) is used in the N2N scenario. A noisy sample $X_{\zeta 1}$ is passed into the network to obtain the output y_1, which is compared to the second noisy sample $X_{\zeta 2}$ in the loss function. Similarly, the second noisy sample $X_{\zeta 2}$ is fed

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

through the network to get the output y_2, which is compared to $X_{\zeta 1}$ in the loss functions. The two losses can be weighted, but here both weights are equal to 1/2. Therefore, the N2N symmetric loss function can be written as:

$$\mathscr{L}_{tot} = \mathscr{L}_{\zeta 1} + \mathscr{L}_{\zeta 2} = \frac{1}{2}\mathscr{L}(X_{\zeta 1}, y_2) + \frac{1}{2}\mathscr{L}(X_{\zeta 2}, y_1). \tag{5}$$

Note that the loss functions $\mathscr{L}$ could be either l_1 or l_2 (mean squared error [MSE]). The chosen loss functions will be specified for each test case. The N2N method is illustrated in Figure 1D. After training the network in the N2N fashion, inference mode operation is the same as for the DAE. The network simply takes as input a noisy flow snapshot and produces a denoised snapshot as output.

2.1.5 | Noise2Void

Acquiring a second set of noisy samples in N2N can still be challenging or even impossible in some scenarios. Krull et al.[37] developed a novel technique called Noise2Void (N2V), which does not require clean data nor a second noisy sample. N2V is widely used in the image processing community with great results.[37,38,71] The main underlying assumption is that the noise is pixel-wise independent, that is, from one pixel's noise it is impossible to infer the neighboring pixel's noise. On the other hand, it should be possible to estimate the signal from its neighboring pixels. The original idea of the N2V technique was to use a convolutional neural network (CNN) where the receptive field does not include the center pixel. That is, the network has to leverage the information from the neighbors to estimate the value of the center pixel. The simplest way to do this is to replace the value of the center pixel with a random pixel from the snapshot. The implementation uses a masking procedure where a certain percentage of the pixels are masked, then the loss is only calculated on these masked pixels. In the loss function, the original pixels that were masked are compared with the prediction. This creates a similar effect to the N2N scenario but overcomes the need for a second noisy sample. The masking is usually done in a patch-wise fashion, where smaller subsets of the input snapshot (i.e., patches) are fed into the network with some pixels randomly masked out. This process is shown in Figure 1E. An l_1 loss was set for the masked pixels. The deep learning architecture used is a specific type of convolutional AE with skip connections called U-Net[72] and the implementation used here is similar to Krull et al.[73]

2.2 | Data imputation

Data imputation, also known as matrix completion, is the process of filling in missing entries in the data. The simplest approach is to just assign constant values (e.g., row or column averages) to all missing values. More sophisticated methods try to leverage the assumed low-rank structure of the data to complete the gaps. The general task is to find a matrix $X_{n\times m}$ while observing only a small portion of its entries. Most real-world data lie on a low-dimensional manifold, therefore it can be assumed that X is low-rank, that is, it can be written as:

$$X = V_{n\times k} G_{k\times m}, \tag{6}$$

with $k \ll \min(m,n)$, where V and G are two matrices of the noted size and are defined based on the decomposition algorithm. This task became famous as the "Netflix prize" problem, where rows and columns represent movies and viewers, respectively, and the goal is to fill in the matrix to provide movie recommendations for the viewers. Classical machine learning methods for matrix completion include $k-$ nearest neighbors or random forests.[6] Some of these algorithms are also based on PCA, referred to as PCA with robustness to missing entries.[31] Generally, they work well when the data is missing at completely random locations. It has been shown theoretically that under certain assumptions the missing entries can be filled in with high accuracy.[10] In this work, SVD and deep learning-based algorithms will be compared for imputing blood flow data. All methods in this section will be tested on synthetically voxelized data. Defining missing data for real 4D flow MRI can be ambiguous, therefore here we focus on synthetic data, which enables experimenting with different missing data patterns and provides ground-truth values for error calculation. In this section, only the velocity magnitude will be used, therefore, all predictions will be for a scalar field. Below, we briefly describe the different data imputation techniques used.

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2.2.1 | Iterative SVD

Most conventional matrix completion algorithms are based on SVD. The first one used here is called Iterative SVD (itSVD) or SVD-Impute.[74] The idea is to do an SVD decomposition first, then use the k most significant eigenvectors for reconstructing the missing entries. The next step is to update the matrix and recompute the SVD. This procedure is repeated until the change in the matrix is smaller than an empirically defined threshold, in this case 10^{-5}. SVD can only be performed on matrices without missing entries, therefore all missing values are initialized with row averages. In the first iteration, the rank is set to $k = 2$, and then it is gradually increased to $k = 10$.

2.2.2 | SoftImpute

The second algorithm used here is SoftImpute, introduced by Mazumder et al.,[75] where the imputation problem is written as a convex optimization:

$$\min_{Z} \frac{1}{2} \sum_{(i,j) \in \Omega} \left(X_{ij} - Z_{ij}\right)^2 + \lambda \| Z \|_*, \tag{7}$$

where Ω is the set of the observed indices, X is the full data matrix, but X_{ij} with $(i,j) \in \Omega$ is the only part observed. λ is a trade-off parameter, and Z is the unknown, that is, the approximation of X. The nuclear norm is used as a surrogate for rank[10] similar to RPCA. The solution to the optimization problem in Equation 7 is given by:

$$Z = D_\lambda(X) = US_\lambda(\Sigma)V^T, \tag{8}$$

where the columns of U contain the left-singular vectors, the columns of V contain the right-singular vectors, $\mathcal{D}_\lambda$ is the singular value thresholding operator and $\mathcal{S}_\lambda$ is the soft-thresholding operator.[31,75] This is iterated until the change in the solution Z is smaller than the predefined value of 10^{-3}. The method is initialized by setting $Z = 0$. The singular value shrinkage parameter is set to be equal to the maximum singular value of the observed matrix (setting all missing values to zero) divided by 50, set empirically. For the itSVD and SoftImpute algorithms, the *fancyimpute* Python package[76] was used.

2.2.3 | Probabilistic PCA

A probabilistic reformulation of PCA can be obtained from a Gaussian latent variable model.[77] This enables the handling of missing data by using an expectation–maximization (EM) algorithm.[78] PPCA assumes that each latent variable $z \in \mathbb{R}^k$ is normally distributed:

$$p(z) = \mathcal{N}(x|0,1). \tag{9}$$

The conditional distribution of the observed variable $x \in \mathbb{R}^n$ also has a Gaussian form

$$p(x|z) = \mathcal{N}\left(x|Wz + \boldsymbol{\mu}, \sigma^2 I\right), \tag{10}$$

where $W \in \mathbb{R}^{(n \times k)}$ is the principal subspace, σ^2 is the variance, and $\boldsymbol{\mu} \in \mathbb{R}^k$ permits the model to have a nonzero mean. Usually, the dimension of the latent space is much smaller than the dimension of the observed variable ($k \ll n$). The observed variable x can be defined as:

$$x = Wz + \boldsymbol{\mu} + \varepsilon, \tag{11}$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ is a Gaussian-distributed noise variable. The marginal distribution of the observed variable will be the following:

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$p(x) = \mathcal{N}(x|\boldsymbol{\mu}, C), \tag{12}$$

where the covariance matrix is defined as $C = WW^T + \sigma^2 I$. The EM algorithm is used to find the model parameters. In the E-step, the latent variables z and the missing parts of x are estimated, while in the M-step, W and σ^2 are updated. Classical PCA is a special case of PPCA when $\sigma^2 \to 0$. For a more detailed description of the PPCA and EM algorithms, readers are referred to References 77–79. In this work, the dimension of the latent space was empirically set to $k = 20$. A Python implementation was used here, taken from Reference 80 and based on Reference 81.

2.2.4 | Supervised AE

Similar to the denoising scenario, an AE can be trained with corrupt and clean snapshot pairs in a supervised fashion, where all missing entries are replaced with zeros in a pre-processing step. All data sets used for training are scaled to [0,1]. A fully connected NN is utilized with an AE architecture. Both the encoder and decoder had six layers, each followed by a ReLU activation function. The encoder layers had the following sizes 27000-8192-1024-256-64-16-8 and the decoder was the symmetric counterpart of this. The optimizer chosen was Adamax with an initial learning rate of 10^{-3}, decreased by a factor of 10 after 200 epochs. The total number of epochs was set to 300 and an MSE loss was formed between the network output and the clean data. A batch of 128 snapshots was fed into the network at once.

A summary of the algorithms presented above is listed in Table 1.

2.3 | Test cases

To investigate and compare the different denoising and imputation methods two test cases were used as shown in Figure 2. The first case is an artificially corrupted and voxelized CFD data of flow inside a brain aneurysm motivated by 4D flow MRI. Second is an in vitro 4D flow MRI patient-specific data of blood flow in a brain aneurysm from a previous study.[82] All patient data used were anonymous and de-identified.

2.3.1 | Test case 1: Corrupt voxelized ICA vneurysm

Voxel-based data mimicking some of the 4D flow MRI data features was created from CFD simulations. Pulsatile blood flow inside an ICA aneurysm was simulated using SimVascular.[83] The geometry shown in Figure 2A is from the Aneurisk dataset[84] (Aneurisk model ID: C0035) and the inlet flow rate Q waveform is taken from,[85] scaled according to Reference 86 (shown in Figure 2A). The Reynolds number was $\mathrm{Re} = 555$ based on the maximum systolic inlet bulk velocity and inlet diameter. The vessel walls were considered rigid with a no-slip boundary condition. A split-resistance boundary condition was applied to the outlets with a reference pressure of $p_{ref} = 100\mathrm{mmHg}$, and the flow was split according to Murray's law. Newtonian viscosity model was used with a density of $\rho = 1060\mathrm{kg/m^3}$ and dynamic viscosity

TABLE 1 A summary of denoising and imputing algorithms.

Methods	Type	Case	Algorithm	Hyperparameters	Use clean data
Filtering	Unsupervised	Denoising	Snapshot-by-snapshot	2	N
RPCA	Unsupervised	Denoising	All data at once	2	N
DAE	Supervised	Denoising	Snapshot-by-snapshot	Many	Y
N2N	Supervised	Denoising	Snapshot-by-snapshot	Many	N
N2V	Self-supervised	Denoising	Snapshot-by-snapshot	Many	N
itSVD	Unsupervised	Imputing	All data at once	2	N
softImpute	Unsupervised	Imputing	All data at once	2	N
PPCA	Unsupervised	Imputing	All data at once	2	N
AE	Supervised	Imputing	Snapshot-by-snapshot	Many	Y

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

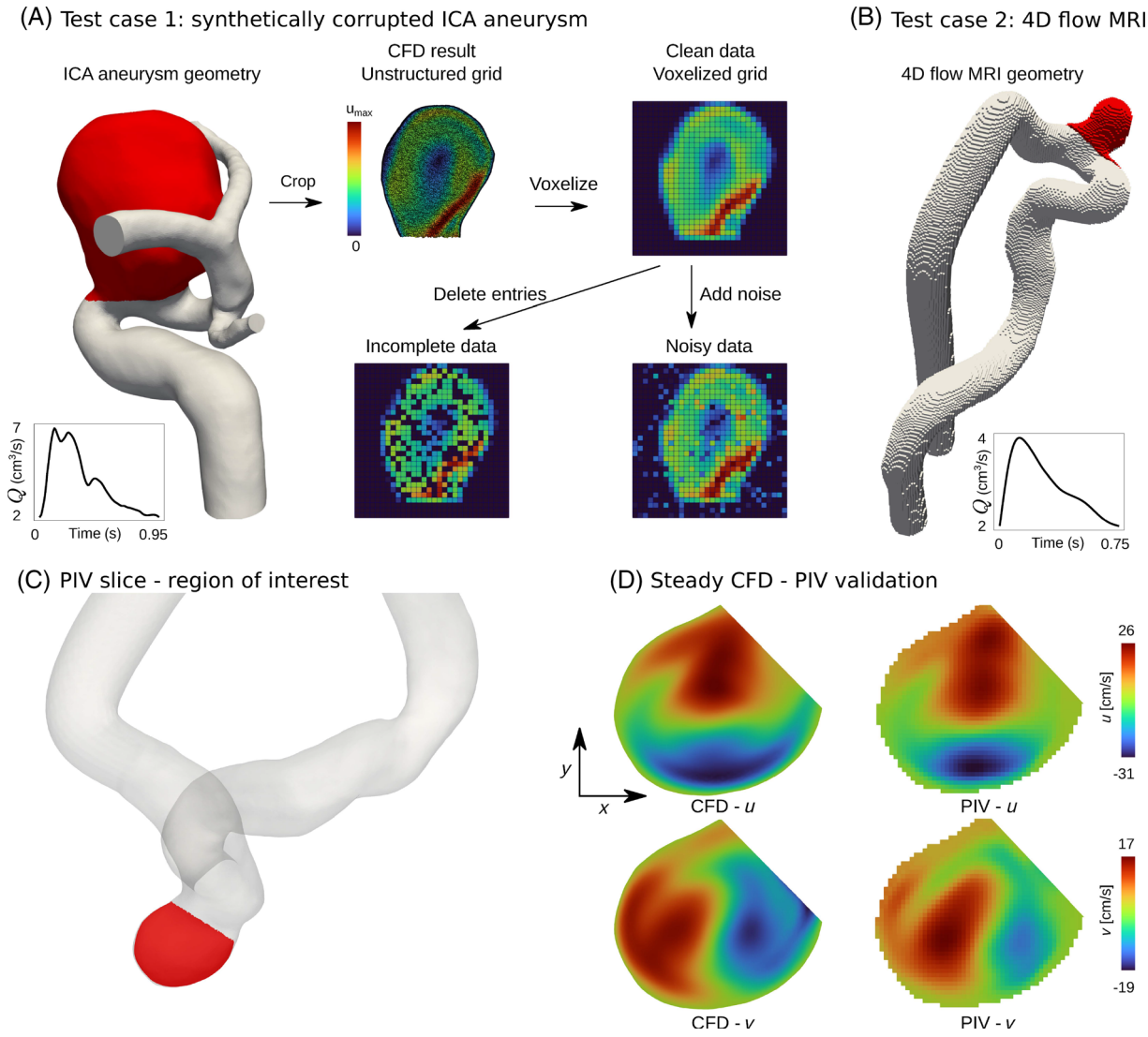

FIGURE 2 (A) The process of creating synthetically corrupt voxel-based flow data. The left panel shows the full ICA aneurysm geometry. The aneurysmal region used for further study is highlighted in red. The middle panel shows the voxelization and artificial corruption steps. The in vitro 4D flow MRI geometry is shown in (B), and the aneurysmal region is highlighted in red. The time-dependent inlet flow rate waveforms are shown at the bottom for both cases. The same geometry from (B) is shown from a different view in opaque gray in (C), where the PIV plane is highlighted in red. Panel (D) shows results from PIV experiments and the corresponding steady CFD results inside the aneurysm, where u and v velocity components are shown for validation.

of $\mu = 0.004\text{kg/ms}$. The simulation was run for 3 cardiac cycles and 1000 uniformly spaced snapshots were saved in the last cycle. The unstructured mesh consisted of 6.6 M elements, which was resampled to a voxelized grid with a uniform isotropic spatial resolution of 0.5 mm. For further analysis, the model was cropped to contain only the region of interest around the aneurysm as highlighted in red in Figure 2A. After resampling, the structured data was filled up with zeros to create a 3D box shape, facilitating the denoising and imputation algorithms. This is necessary since some of the techniques (e.g., CNNs or filtering) can only be applied to data lying on structured rectangular grids occupying a full 3D box. An overview of the process is shown in Figure 2A.

Two types of artificial data corruption were investigated here: missing data and Gaussian noise. These were artificially added to the voxelized data. Voxels were randomly removed in space and time for the missing data case. The percentage of data missing will be noted with α. The entire box domain is taken into account when calculating α, including the parts filled up with zeros outside the flow region. The process of creating corrupt synthetic voxel-based data is illustrated in Figure 2A. For the spatiotemporally random missing data case, the fraction of missing data was varied between 10% and 90%, that is, $\alpha \in \{0.1,0.3,0.5,0.7,0.9\}$. Two additional scenarios were created for investigating

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

imputation techniques of missing data. Data were removed at random locations in space, but fixed locations in time. This could correspond to a faulty sensor location for example. Another case is where all data is considered missing below a certain velocity magnitude threshold. This is motivated by the fact that low-speed flow regions in 4D flow MRI commonly experience low signal-to-noise ratios, and data is less trustworthy. In the vicinity of the walls, this effect is even further amplified due to the partial-volume phenomenon. Three separate networks have been trained for the three different cases of missing data (random in space and time, random in space but fixed in time, and missing below a threshold). In all cases, the training data had the same missing data distribution as the test case. For data missing completely at random in space and time, the training and testing data were generated in a similar fashion, by assuming a random subset of the indices in the data matrix to correspond to missing entries. For data missing at random locations in space but fixed in time, the training data had missing entries at the same spatial location for all snapshots. For data missing below a velocity magnitude threshold, the threshold was set to be 15% of the max velocity in all training and test cases.

For the denoising scenario, Gaussian noise with zero mean and standard deviation σ was added to α fraction of the data at random spatiotemporal locations. The fraction of data tagged as noisy was varied between 10% and 70%, that is, $\alpha \in \{0.1,0.3,0.5,0.7\}$. The standard deviation of the added noise was 10% of the maximum velocity value, $\sigma = 0.1$.

For training the supervised DAE and the AE for filling in missing data, seven simulations at different Reynolds numbers were run, while keeping all other parameters unchanged. The Re was varied by scaling the inlet velocity waveforms, keeping the geometry, viscosity, and waveform shape the same. The training set was $\mathrm{Re}_{train} \in \{349,383,418,488,523,616,650\}$, while the testing set was $\mathrm{Re}_{test} = 555$, which was the case directly used for the unsupervised methods. The small training dataset is due to the fixed geometry. Therefore, one can think of the supervised learning methods as an algorithm (rather than a real-time prediction) that needs to be retrained for each new geometry.

A 3D convolution neural network (3DCNN) AE architecture was utilized for the DAE and N2N cases. The velocity components were treated as a 3-channeled input. The encoder had three convolutional layers with a kernel size of 3, stride 1, and padding 1. A ReLU activation function and a 3D max pooling layer with kernel size 2 and stride 2 followed each convolutional layer in the encoder. In the decoder, only the activation functions were present after the convolutions. The input to the network was a data cube of $30 \times 30 \times 30$ voxels with 3 channels. The Adamax optimizer was used for updating the weights and biases with an initial learning rate of 10^{-3}. The learning rate was decreased by a factor of 10 every 200 epochs, the total number of epochs was set to 600, while the batch size was 128. The hyperparameters of the network were identical in the DAE and N2N cases. MSE loss was used for both DAE and N2N, while l_1 loss for N2V, decided empirically. N2V was used in a slice-by-slice manner, taking as input a 30×30 one velocity component slice in the $x-y$ plane. It trained all 3 velocity components at the same time with a batch size of 8. The number of epochs was set to 200, the initial learning rate was 10^{-3}, the patch size was set to 16×16, with the number of masked pixels being 16 in each patch. For N2V, the U-net depth was chosen to be 5. DAE, N2N, and N2V were all trained on an NVIDIA RTX A4500 GPU, taking around 5 h at maximum.

2.3.2 | Test case 2: In vitro 4D flow MRI data

In vitro 4D flow MRI scans were performed for an intracranial aneurysm model at a 3 T Siemens scanner. The lumen geometry was segmented from in vivo computed tomography angiography scans, and inlets and outlets were added to the native vessel to be easily accommodated in the experimental setup later. The full geometry is shown in Figure 2B. The inlet flow rate waveform Q comes from in vivo 4D flow MRI of the same patient at 15 cardiac phases, shown in the bottom of Figure 2. The in vitro model was scaled up by a factor of three, and its working fluid kinematic viscosity was 2 times smaller than that of blood. As such, the in vitro Reynolds and Womersley numbers were matched to the physiological in vivo values to ensure dynamic similarity. The in vitro 4D flow MRI voxel size was 0.6 mm isotropic, and the total domain size was $239 \times 192 \times 128$ voxels, while the temporal resolution was 15 snapshots per cardiac cycle. The corresponding CFD simulation was carried out at the in vivo scale and viscosity, therefore for a comparison of the results, the in vitro 4D flow MRI results were scaled up by a factor of six. The Reynolds number was $\mathrm{Re} = 415$ based on the peak systolic inlet bulk velocity and the inlet vessel diameter. Besides noise, 4D flow MRI scans can contain artifacts such as phase offset errors and velocity aliasing.[1] These are outside of the scope of this article and the sole focus here will be denoising. In all scans, the velocity encoding (VENC) was always higher than the maximum velocity, and

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

therefore the measured velocity field was free of any aliasing. The 4D flow MRI procedure and experimental setup are described in more detail in Reference 82.

A CFD simulation was performed on the same geometry and boundary conditions as the in vitro 4D flow MRI data on the original in vivo scale assuming viscosity of $\mu = 0.004$ kg/ms and density of $\rho = 1060\text{kg}/\text{m}^3$, giving $\text{Re} = 415$. The simulation was set up similarly to the ICA aneurysm case presented in Section 2.3.1 and the inlet boundary condition was similar to the in vitro experiment. This geometry had only one inlet and one outlet, and the outlet boundary condition was a resistance with a reference pressure of $p_{\text{ref}} = 100$ mmHg. The unstructured mesh had 12.5 M elements with a boundary layer mesh. The results were later resampled to the voxel grid and time-spacing of the 4D flow MRI setup for comparison. Similar to the previous case, the aneurysm region was extended to a 3D box and filled up with zeros outside the flow region. This data will be referred to as "ground-truth" CFD data and will not be used in the training of any of the deep learning models. The sole purpose of this high-fidelity CFD simulation is to provide a base of comparison for the denoised 4D flow MRI fields.

For a partial validation of the CFD methodology, a planar refractive-index-matched PIV measurement was performed at a slice cutting through the aneurysm sac as shown in Figure 2C. The model has the same size as the in vitro 4D flow MRI model (scaled-up with a factor of 3 with respect to the in vivo setting). A high-resolution stereolithography model with a wall thickness of 1 mm fabricated at 25 μm layers of Somos WaterShed XC 11122 guaranteed hydro-dynamically smooth surfaces. A mix of anise seed oil and mineral seal oil was used as the working fluid that matched the refractive index of the 3D-printed material. The kinematic viscosity was approximately 2.7cSt. The steady-state inflow has the Reynolds number of approximately 440 based on the inflow bulk velocity and the inlet vessel diameter. A two-cavity Nd:YAG laser at 532 nm and 120 mJ/pulse was used to generate a light sheet with a thickness of approximately 1 mm. The flow seeded with 10 μm hollow glass sphere particles was imaged using a 12-bit CCD camera at a resolution of 1376 × 1040 pixels. A 105 mm Micro-Nikkor lens at a F-number of 5.6 provided an imaging resolution of approximately 25 μm/px. Two thousand image pairs (with a time separation of 2 ms) were acquired at rate of 5 Hz. A standard 3-point Gaussian fit in each direction was used for the particle peak detection. The interrogation window with the final pass of 48 × 48 pixels with 75% overlap was used for the fast Fourier transform-based cross correlation. The phantoms for MRI and PIV modalities had the exact same lumen, 3D-printing fabrication method and material, layer resolution, and printing direction. However, the PIV phantom had a thin 1 mm wall thickness compared with a 3 mm thickness in the MRI phantom, to ensure minimal optical distortions.

To generate the required flow for 4D flow MRI and PIV experiments, a flow circuit capable of producing both steady-state and physiologically realistic flow waveforms was used. The major components include an in-house-built cardiac pump, a centrifugal pump with a flow regulator, a high-resolution ultrasonic flow sensor, a highly sensitive pressure transducer, and multiple types of valves. The impedance of the flow loop mimics the arterial bed through standard Windkessel boundary conditions and is controlled using two resistors and a capacitance module.

For training the deep learning models, four other CFD simulations were created at different Reynolds numbers of $\text{Re} \in \{378,472,566,660\}$ by scaling the inlet flow rate waveform. $\text{Re} = 566$ was used for validation, while others for training. The trained model was tested on the in vitro 4D flow MRI data. The cases were run with a smaller mesh of 1.2 M elements and all other parameters were set identical to the previous case. One hundered snapshots were saved from each simulation at uniform time spacing, leading to a total of 300 snapshots for training and 100 for validation. The region of interest around the aneurysm, highlighted in red in Figure 2B was used for the denoising setup.

A fully-connected NN with an AE architecture was used for both DAE and N2N cases. All three velocity components were fed into the network at the same time. The network had six layers in the encoder section with sizes of 151515-1024-512-256-64-32 and a ReLU activation function between them. The decoder section also had six layers, symmetric to the encoder. The Adamax optimizer was used with an initial learning rate of 10^{-3}, decreased after every 200 epochs by a factor of 10. The total number of epochs was 600 and the batch size was 16. The network was saved every time the validation loss reached a new minimum, the weights and biases from the lowest validation loss were used for inference. The hyperparameters were identical in the DAE and N2N cases. The DAE and N2N models were trained on an NVIDIA Tesla V100 GPU, taking a total of 1–2 h for training. N2V was used in a slice-by-slice mode and the inputs were of size 192 × 239. However, it was trained separately for each velocity component. In the N2V approach, the whole geometry was used not just the aneurysm region as it led to better performance. The number of epochs was set to 200 with an initial learning rate of 10^{-3}, patch size of 32 × 32, number of masked pixels set to 32, and the U-net depth was 3. The N2V model was trained on an NVIDIA RTX A4500 GPU, taking less than an hour. In all deep learning algorithms, an l_1 loss was used, as it produced empirically better results than an l_2 loss. Interestingly, it was argued by Shit et al.[58] that l_1 loss is more suitable for 4D flow MRI type data.

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2.3.3 | Noise estimation

To properly use the 4D flow MRI data with a DAE, it is desirable for the AE to be trained on the same noise level as the 4D flow MRI data. To achieve this, a fully connected NN is trained for noise estimation, similar to Kahana et al.[87] The setup is shown in Figure 3 along with the subsequent denoising procedure. It takes a voxelized flow snapshot as input and outputs one scalar value, which is the noise level. The NN is trained in a supervised manner with artificially corrupted and voxelized CFD data, where the noise level σ_{train} is known. An MSE loss is used between the known σ_{train} and the network output σ_{net}. After training with noisy CFD data, this network can be then employed in inference mode to estimate the corruption levels for the 4D flow MRI data, noted with σ_{4DFlow} in the figure. In inference mode, the network takes as input a 4D flow MRI snapshot and outputs a predicted noise level. This is performed for all available 4D flow MRI snapshots, and the resulting noise level is averaged. The noise level fluctuates between different 4D flow MRI snapshots and with different random initialization of the network. However, this does not affect the denoising results significantly. The noise estimator network is a fully connected NN with six hidden layers with 256 neurons each. The input layer is the same size as the number of voxels, while the output dimension is 1. A ReLU activation function is applied after each layer. The ADAM optimizer is used for 300 epochs with an initial learning rate of 10^{-4}, decreasing by a factor of 10 after every 100 epochs. Three CFD simulations are used for training with $\text{Re} \in \{300,450,525\}$, while $\text{Re} = 375$ is used for validation. All simulations contain 100 snapshots, leading to 300 snapshots for training and 100 for validation. The training noise level is set to be between $0.01 < \sigma_{\text{train}} < 0.50$, while 99% of the data is assumed to be noisy. To avoid overfitting, the model is saved each time the validation loss decreases and the model corresponding to the lowest validation loss is used with the 4D flow MRI data in inference. The network was trained on an NVIDIA RTX A4500 GPU, taking less than 10 min.

The mean noise level estimated was $\sigma_{\text{4DFlow}} = 0.19 \pm 0.03$ among the 4D flow MRI snapshots and was utilized to generate training data for the supervised DAE. This is shown in Figure 3C and Figure 3D for training and inference

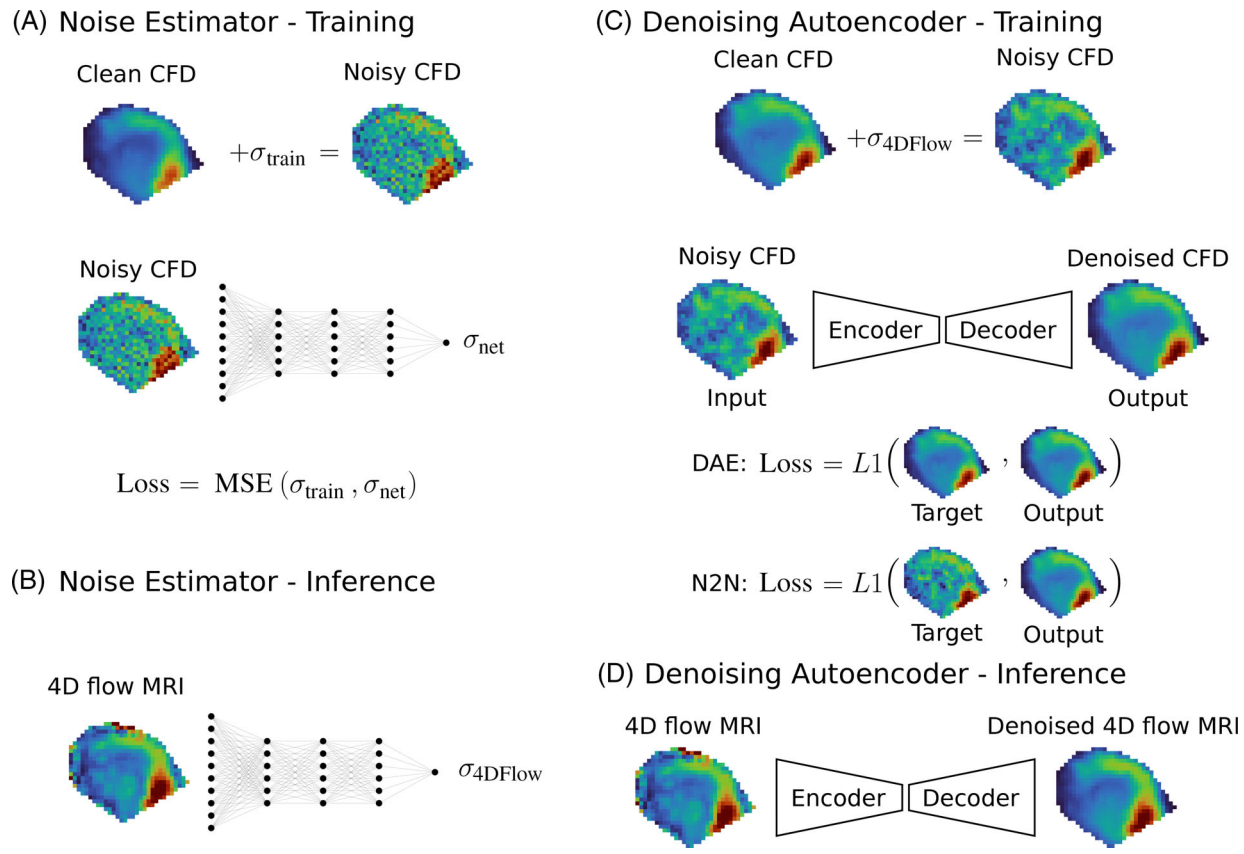

FIGURE 3 4D flow MRI data denoising framework. Noise estimator network is shown in (A) for training and in (B) for inference. The denoising autoencoder setup is shown in (C) for training and in (D) for inference. For N2N the twin-loss was used, and only one part is shown here.

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

modes, respectively. For training, CFD data is used with σ_{4DFlow} noise added, while in inference mode the 4D flow MRI snapshots are denoised. l_1 loss function is used between the network output and the target. The target can either be clean CFD data (DAE), or a second noisy sample (N2N). These approaches are also illustrated in Figure 3C. It should be noted that real 4D flow MRI noise for the velocity vector field is not Gaussian and the noise is Gaussian in the frequency domain (k-space).[56,88] Nonetheless, by learning an approximate Gaussian noise level by the noise estimator network, it is possible to generate flow fields that are realistic enough for the purpose of training the DAE models. The main goal of the noise estimator network is to show that the noise level added to the synthetic data for training can be pre-assumed in a reasonable fashion.

3 | RESULTS

The presented imputation and denoising techniques were applied to the synthetic and in vitro 4D flow MRI data sets. For filling in missing values, only the synthetic data was used with three different missing data patterns being investigated. These three patterns cover a wide range of scenarios from theoretic to more realistic cases. Using synthetic data also provided ground-truth values, which were used for calculating error metrics. Noise removal was done on both synthetic and in vitro 4D flow MRI data. For the latter, in lack of real ground-truth values, a high-fidelity CFD simulation was taken as a pseudo ground-truth. Various conventional and deep learning techniques were evaluated with some demonstrating excellent performance, while others showed less satisfactory results.

For validation of the CFD methodology, steady PIV results from a 2D plane through the aneurysm were compared with a separate steady CFD simulation. The PIV plane is shown in red in Figure 2C, and the main geometry is shown in opaque gray. The PIV and corresponding CFD results are shown in Figure 2D for the u and v velocity components inside the aneurysm region. Both results are shown on the color scale with the minimum and maximum values taken from the CFD simulation. The velocities from CFD results are projected to the plane shown in the figure. All settings for the CFD simulation were the same as presented above, except it was steady. The Reynolds number was set to 440. The PIV measurements were done in an in vitro model as described before. The u velocity results agree both qualitatively and quantitatively. The v velocities show some differences in the smaller details, but the main qualitative behavior is the same. Overall, the steady CFD and PIV velocity fields match qualitatively for both velocity components, the u velocities are quantitatively consistent, while the v velocities have slight discrepancies.

3.1 | Synthetic data imputation

The data imputation results for the first test case are shown in Figure 4 for time $t = 0.095$ s for a slice in the midplane of the aneurysm. Figure 4A shows the case where data is missing in a uniformly random manner in both space and time. The slices are colored by velocity magnitude and white voxels in the corrupt panel represent missing data. It can be seen that all methods achieve a reasonably good reconstruction even for a high missing percentage of $\alpha = 0.7$. These results are somewhat expected since the low-rank matrix completion methods are known to work well for low-rank structured data that are observed uniformly at random.[10] Figure 4B shows the second case, where data is missing at random locations in space but fixed in time. This is equivalent to complete rows missing in a data matrix. A corresponding real-life scenario could be faulty measurement sensors at certain locations that cannot collect any data or corrupted pixels in image acquisition. It is obvious that all three SVD-based methods completely fail in this case. itSVD and softImpute just assign zeros to all missing data, while PPCA assigns the mean. The AE approach can achieve a good reconstruction since it has been trained separately for each case with the corresponding missing data patterns. This is reasonable since it is possible to observe the patterns of missing data and create a training data set with similar missing data characteristics. A more realistic scenario is shown in Figure 4C where data is considered missing (unreliable) if the normalized velocity magnitude is $\| u^* \| < 0.15$. The SVD-based methods struggle again and there are areas in the flow where they incorrectly assign zero velocities. These methods have erroneous reconstructions near the walls where data is frequently missing. In Figure 4D, the velocity magnitude imputation relative reconstruction errors are shown as a function of the percentage of data missing α. The vertical axis is logarithmic to better represent the scales. The relative reconstruction error is defined as:

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

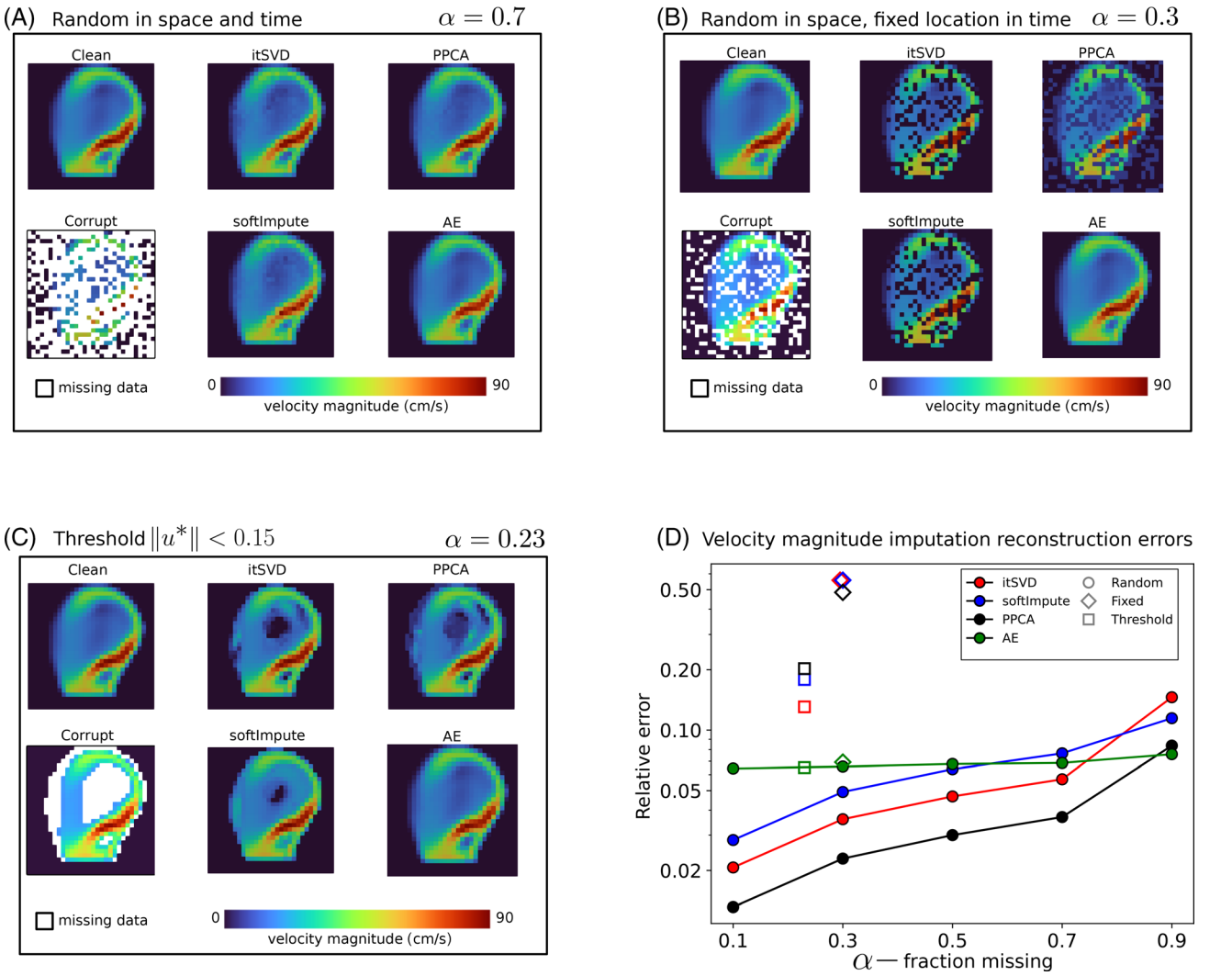

FIGURE 4 Imputation results for data (A) randomly missing in space and time, (B) randomly missing in space with fixed location in time, (C) missing below a normalized velocity magnitude threshold of $\| u^* \| < 0.15$ are shown. All panels are colored by velocity magnitude, and white pixels represent voxels with missing data points. All results are shown for time $t = 0.095$ s. The α values denote the fraction of data missing. (D) Relative reconstruction error is shown for different percentages of missing data. Circles represent cases where data was missing randomly in space and time, diamonds represent cases where the missing location was fixed in time, and squares represent cases where data was missing below a threshold. Colors correspond to different imputation techniques. The vertical axis has a logarithmic scale.

$$\varepsilon_{\text{rec}} = \frac{\| X_{\text{r}} - X_{\text{GT}} \|}{\| X_{\text{GT}} \|}, \tag{13}$$

where the subscript GT refers to the clean CFD taken as ground truth, X_{R} is the reconstructed data, and $\| \cdot \|$ here is the Frobenius norm. For the case where data is missing uniformly at random in space and time, 5 different fractions of missing data were investigated, $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, as shown with circles in the figure. Colors correspond to different imputation algorithms. Diamonds correspond to the case of random missing locations in space but fixed in time with a fraction of $\alpha = 0.3$ missing. Squares represent the threshold case with a total fraction of $\alpha = 0.23$ missing. Note that in this case, no data is missing outside the flow region, so the actual percentage missing inside the aneurysm is greater. For the uniformly random case, PPCA consistently has the lowest error out of the SVD-based methods, and the AE is slightly better only when 90% of the data is missing. The AE shows a low dependency on the fraction of the missing data where its results are only slightly worse when $\alpha = 0.9$ is missing compared with $\alpha = 0.1$. The AE method shows similar performance for the two nonuniformly random cases, while the other methods give severely worse results. This is because AE has been retrained separately for each noise fraction and missing data pattern and also SVD based methods have limitations when data are not randomly missing.

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

3.2 | Synthetic data denoising

Denoising results for the same test case (voxelized ICA aneurysm) are shown in Figure 5. The results are visualized in Figure 5A for a slice in the midplane of the aneurysm for time $t = 0.095$ s. The case shown corresponds to $\sigma = 0.1$ noise level and $\alpha = 0.3$ fraction of data has noise. Overall all five methods manage to denoise the flow field, however, there are minor differences in the reconstructions. A high-velocity jet-like structure can be observed that creates a counter-clockwise rotating vortex inside the aneurysm. Filtering clearly reduces the magnitude of this jet and smooths it out too much. N2V captures this structure well, but it shows slight non-smooth behavior in the lower velocity regions. N2N and DAE achieve similar reconstructions and both have minor issues with assigning small but nonzero velocities to voxels outside the flow region. Note that none of the methods use the explicit information that the velocities should be exactly zero outside the flow region. In Figure 5B, the relative errors are shown as a function of α, the percentage of data where noise was added. The noise fraction α varied between 0.1 and 0.7 keeping the noise level constant at $\sigma = 0.1$. The relative reconstruction errors were calculated on the scaled velocities, where all three velocity components were scaled together to be between 0 and 1. The velocity components were stacked column-wise, then the relative reconstruction error was calculated based on Equation 13. Overall RPCA has the lowest errors for most cases. For the most corrupted case, the two supervised NN models (DAE and N2N) have the best reconstruction. Traditional filtering has the highest error for all noise fractions except 0.7. This is mainly due to the extra dampening that median and Gaussian filters cause. Interestingly, there is not much difference between DAE and N2N results. This suggests that having a set of clean data for training is not much more beneficial than having a second set of noisy samples. This is known in the computer vision community and has been shown by multiple studies.[36,73,89]

3.3 | 4D flow MRI denoising

Results for denoising the in vitro 4D flow MRI data are shown in Figure 6. The first panel shows the denoised flow fields in the midplane of the aneurysm for $t = 0.2$ s. Figure 6B shows streamlines from the original and reconstructed flow fields colored by velocity magnitude for the same snapshot. It is obvious that RPCA fails to remove the noise in this case and it also dampens the velocity magnitude excessively. In the streamlines plot, the raw 4D flow MRI data shows many streamlines abruptly stopping or going in seemingly random locations and similar nonphysical phenomena can be observed in the RPCA streamlines. The other four methods achieve similar reconstructions visually and produce reasonably-looking streamlines as compared with the ground-truth CFD reference.

To better assess the temporal variation of the reconstructions, a video is included in the Supporting Information (Video S1) that shows the same comparison as Figure 6A but in time. From the video, it is seen that N2V and filtering do not produce smooth reconstructions in time. In both of these methods, there is a jittering effect that cannot be observed from individual snapshots but it is clearly seen in the video. This effect is most prominent in the center of the

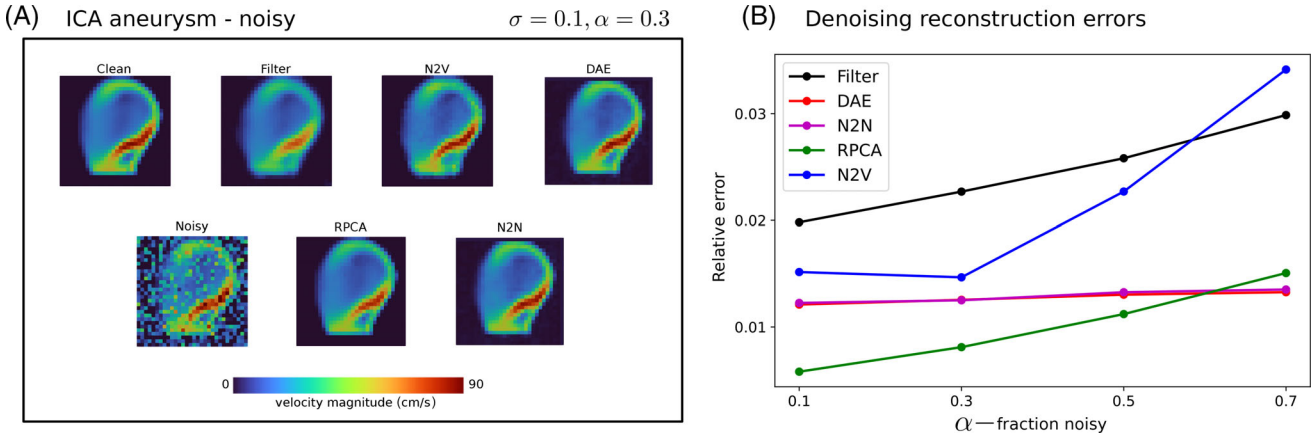

FIGURE 5 Denoising results for the synthetically corrupted ICA aneurysm case. Results obtained with different denoising methods are shown in (A) for a slice in the midplane of the aneurysm, colored by velocity magnitude for time $t = 0.095$ s. The relative reconstruction errors are shown in (B) as a function of the fraction of data noisy α. Colors correspond to different denoising techniques.

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

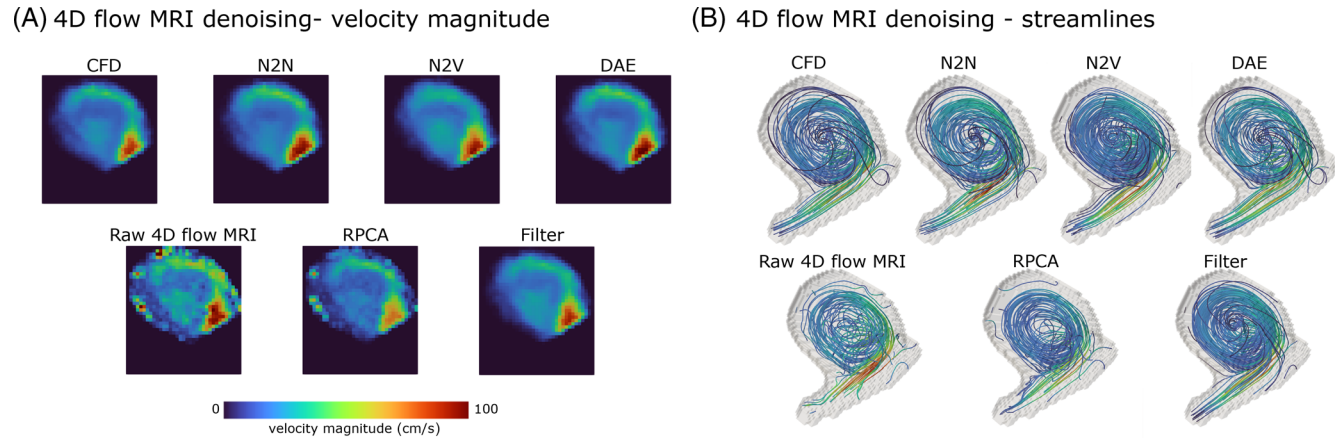

FIGURE 6 Denoising results for the in vitro 4D flow MRI case. Results obtained with different denoising methods are shown in (A) for a slice in the midplane of the aneurysm, colored by velocity magnitude for the 3rd timestep. (B) Streamlines are shown for the same timestep, colored by velocity magnitude. The aneurysm geometry is shown with opaque gray outlines.

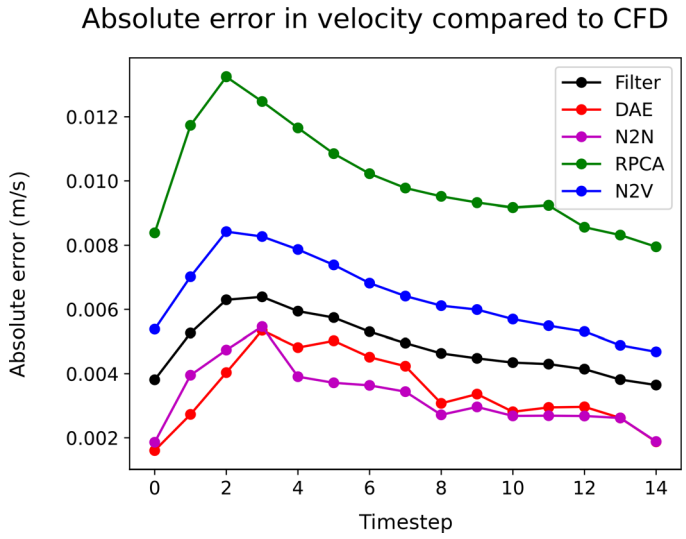

FIGURE 7 The absolute error of the reconstructed in vitro 4D flow MRI velocity field compared with the CFD simulation is shown as a function of time. Colors correspond to different denoising techniques.

aneurysm and in the near-wall regions. It stems from different reconstructions for consecutive time instances and is not desirable. The N2N and DAE approaches do not have this effect and produce much smoother reconstructions in time, similar to the reference CFD data.

In Figure 7, the absolute error in the velocity field compared to the reference CFD simulation is shown as a function of time. The reconstructed velocity components were stacked column-wise into a matrix $X_r \in \mathbb{R}^{(3n\times m)}$, then the absolute error for the j-th snapshot was calculated as:

$$e_j = \sum_{i=0}^{3n} \frac{\left|X_r^{ij} - X_{\mathrm{GT}}^{ij}\right|}{3n}. \tag{14}$$

The error curves mostly follow the shape of the inlet flow rate waveforms with the highest errors in systole. As was seen from the visual comparisons, RPCA has the highest error. The DAE and N2N methods have smaller errors compared with traditional filtering. Similar to the artificially noisy ICA aneurysm case, the difference between DAE and

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 In vitro 4D flow MRI denoising results for the three velocity components by the different denoising methods.

Relative errors: $\|X_r - X_{GT}\| / \|X_{GT}\|$						
Velocity	**Raw 4D flow MRI**	**Filter**	**DAE**	**N2N**	**RPCA**	**N2V**
u	1.420	0.244	**0.158**	0.161	0.562	0.307
v	1.122	0.222	0.207	**0.181**	0.485	0.389
w	1.387	0.279	0.213	**0.189**	0.574	0.378

Note: Bold values highlight the best results for each velocity component.

N2N is insignificant. In the early snapshots, DAE has the lowest error and later in the cardiac cycle N2N produces lower error than DAE. The relative reconstruction errors for each velocity component are provided in Table 2 for the different denoising techniques. The lowest errors are highlighted in bold for each row. DAE has the lowest error for the u component, while N2N has the lowest error for v and w.

4 | DISCUSSION

Postprocessing of experimental data like 4D flow MRI or PIV is standard practice, but emphasis is mostly placed on the rapidity of the techniques rather than their accuracy. Herein, different SVD and deep learning-based methods were compared for handling artificially corrupted CFD data and in vitro 4D flow MRI data. It was shown that in artificial cases, the SVD-based methods (especially PPCA and RPCA) provided robust reconstructions with respect to missing and noisy data. However, for the in vitro data, deep learning approaches outperformed the SVD-based ones. Traditional filtering certainly has its advantages by being fully unsupervised and computationally cheap. Nevertheless, as seen for the in vitro 4D flow MRI case, it produces reasonably clean velocity fields, but it cannot get rid of the temporal jittering effect shown in the Video S1.

SVD-based matrix completion worked great for data missing completely at random in space and time as expected but failed for the two other missing data patterns. This limitation is well known and has been discussed in more theoretical settings.[90] The case with data missing randomly in space but fixed in time and the case of data missing below a certain threshold could appear in real-life scenarios such as faulty measurement sensors or low-fidelity data in low-flow regions, respectively. It was demonstrated that AEs are capable of imputing blood flow data with high accuracy under all missing data patterns, since they provide the flexibility to be trained with arbitrary missing data arrangements.

It was shown that for the synthetically noisy ICA aneurysm case, all methods provided better reconstructions than the traditional filtering. Specifically, RPCA produces excellent noise-free velocity fields. This performance did not translate to the in vitro 4D flow MRI data, where RPCA produced the worst reconstruction. A conceptual advantage of RPCA over the other methods is that it leverages the temporal dimension of the data as well. RPCA uses all of the data at the same time, while all other methods do denoising in a snapshot by snapshot manner. Hence, these methods do not use any temporal information and cannot leverage the fact that snapshots close in time should probably be similar. This behavior has both advantages and disadvantages. If the flow data that is to be denoised have a high temporal resolution, then probably denoising algorithms should take advantage of temporal behavior. On the other side, if the temporal resolution is low (e.g., in the case of 4D flow MRI), then methods that rely highly on temporal information will perform poorly, and snapshot-to-snapshot denoising will be superior. Therefore, one reason behind the poor performance of RPCA for the 4D flow MRI case is that the temporal resolution of the 4D flow MRI data is low compared with the artificially noisy CFD data. The size of the data matrix is $X \in \mathbb{R}^{(3n \times m)}$, where for the 4D flow MRI data the number of voxels is $n = 50,505$ and the number snapshots is $m = 15$. Therefore, X is extremely tall-and-skinny and its rank is at most 15, which is already small compared with the number of rows n. Since the RPCA algorithm tries to drive down the rank even further, there were only 3 singular values that were larger than 10^{-12}, achieved by setting a limit of 300 on the number of iterations in the principal component pursuit algorithm. Doing further iterations would result in an even lower rank. This leads to an oversimplification of the problem and loss of relevant information. Based on previous works,[18,27] such a low number of modes cannot accurately describe flow phenomena in intracranial aneurysms. In other experimental cases (e.g., PIV), where RPCA has shown promising results,[33,44] the number of snapshots was at least one order of magnitude higher than 4D flow MRI. It should be noted that all the other methods investigated here

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

work in a snapshot-by-snapshot manner, therefore the number of total snapshots does not influence the results, assuming there was enough data for training.

A key challenge of the deep learning-based methods compared with the SVD-based approaches is that they need training data. In this study, we treated the deep learning methods as an algorithm rather than a pre-trained method that can provide real-time solutions. That is, the geometry was fixed and the deep learning methods were trained in a patient-specific manner. Therefore, 300 snapshots based on a few simulations at different Reynolds numbers were sufficient for the training, which is a small training dataset in machine learning standards. In other words, our deep learning strategy was aimed toward denoising flow in a given patient-specific geometry rather than developing a general denoising method that could be applied to any geometry. Another consequence of this strategy was that the NN were small and relatively simple, which not only reduces the chance of overfitting but also simplifies hyperparameter tunning and the learning process. However, this approach has its drawbacks too, as additional hyperparameter tuning may be necessary for new geometries, in addition to creating CFD simulations for the new training dataset.

In real-world scenarios, data often necessitates not just denoising and imputation, but also super-resolution and artifact handling. These processes should be integrated to effectively address the complex nature of real-world data. In this study, we focused solely on denoising and imputation, and we addressed them separately. To enhance the practical applicability of our findings, future research should explore combined approaches that simultaneously tackle all these aspects. Additionally, efficient hyperparameter tuning strategies for new geometries should be explored. By addressing these limitations, the methods can become more robust and broadly applicable in practical scenarios.

In this work, we used high-resolution CFD data as ground truth for evaluating the errors, which is a common practice in similar studies.[26,59] In some studies,[51] the authors use other experimental techniques (e.g., PIV) to cross-validate the results. However, this poses additional challenges when quantitative errors are being calculated since PIV also comes with its own limitations, particularly in complex cardiovascular geometries.[62,91] Alternative approaches include checking the divergence-free condition[58] or developing unsupervised metrics for evaluating denoising methods without ground truth.[92]

The success of the N2N method in denoising 4D flow MRI was an interesting observation as no clean data was used in its training. This might be surprising at first, but well known in other fields, and the mathematical details are explained by Lehtinen et al.[36] Loosely speaking, the network has the task of learning a noisy target from a noisy input. This task is impossible unless the network is extremely large and complex, which is not the case here. Therefore, the network is naturally underfitting and learns to output an "average" snapshot. This idea is behind another popular denoising algorithm, deep image prior (DIP),[93] as well. This phenomenon is highly undesirable in super-resolution tasks as the output will not be sharp but it is beneficial in denoising. In many denoising scenarios, the clean sample is obtained as a statistical measure at the pixel level (e.g., mean or median) across the stochastic corruption process. The simplest example would be zero-mean noise. If we have an image with multiple corruptions from the same zero-mean noise distribution, we can recover the clean image by simply taking the average of the noisy samples (assuming we have a large number of samples). Leveraging these properties with a NN framework requires the use of an appropriate loss function, and different loss functions might work well for different cases.[36] This was seen here as well, where for the synthetic ICA aneurysm case the l_2 loss (mean) worked best, while for the 4D flow MRI data the l_1 loss (median) produced significantly better results. This was somewhat expected, as l_1 is known to better handle outliers.[9]

The success of N2N suggests that there is a great opportunity to leverage larger amounts of noisy fluid flow data in the future to train deep learning models. For example, if multiple 4D flow MRI acquisitions of the same geometry are available, they could be used in a N2N scenario. Similarly, time-resolved PIV measurements for multiple cardiac cycles could also be thought of as different samples from the same underlying noise distribution. To the best of our knowledge, our study is one of the first uses of N2N and N2V methods in fluid dynamics. Even though some of the underlying ideas are quite similar, it was observed that N2V underperformed N2N for both cases investigated. N2V's performance is likely more influenced by the noise distribution. In contrast, N2N, which uses pairs of noisy data, can more flexibly adapt to the actual noise characteristics present in the data. To overcome some of these limitations, more advanced self-supervised denoising methods[39,41] could also be sought in the future. For, example, several improvements have been made to the original N2V formulation to further enhance the results in certain scenarios, for example, Structured N2V (StructN2V)[71] uses a specific mask structure to remove structured noise, Probabilistic N2V (PN2V)[73] and Parametrized PN2V (PPN2V)[38] estimate the noise distribution before denoising, achieving better results.

In addition to investigating improved versions of N2N and N2V or customizing these approaches to physics-based data such as fluid flow, our study could also be extended in other directions. Inspired by the recent paradigm of physics-informed machine learning,[94] enforcing physical laws could be done in a similar finite-difference manner to,[52]

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

while keeping the main framework presented here. However, since the high-resolution CFD data is resampled to a voxelized grid, even the voxelized CFD data might not satisfy the conservation equations, due to interpolation errors. Therefore, enforcing conservation laws will pose a challenge, particularly if the voxel size is large compared with the computational mesh size. In our methods, the treatment of missing and noisy data was separated. Designing algorithms that can simultaneously tackle missing and noisy data is another research direction. For example, there have been theoretical results and algorithms proposed for noisy low-rank data missing at random.[95–97] Leveraging concepts from compressed sensing in data reconstruction is another interesting research opportunity.[18,98,99] Other filtering techniques suitable for denoising 4D flow MRI data such as anisotropic diffusion filtering[100] or wavelet transforms[101] can also be compared with machine learning in future studies. Finally, deformable walls, in vivo data, and application to other cardiovascular flow problems remain to be investigated.

5 | CONCLUSIONS

In conclusion, supervised and unsupervised machine learning methods were applied for denoising and imputing cardiovascular flow data. It was shown that more conventional SVD-based methods can work well for artificial test cases. However, they failed in filling in missing entries when data were missing not completely at random as well as in denoising in vitro 4D flow MRI data. The DAE and N2N algorithms achieved better reconstruction errors than traditional filtering and managed to remove the temporal jittering effect that was not possible with N2V and filtering methods. The close performance of N2N and DAE suggests that there is a large potential in using only noisy data for training and still getting noise-free outputs from a NN. This study sheds light on the advantages and disadvantages of many different data enhancement methods. Our results encourage further studies to analyze the practical utility of these machine-learning techniques in denoising or imputing various cardiovascular flow measurement modalities.

ACKNOWLEDGMENTS

This study was supported by a collaborative National Science Foundation (NSF) grant under Grant No. 2103560 and 2246916.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

The Python source codes used to generate the results in the manuscript are available on GitHub: https://github.com/amir-cardiolab/ML_4Dflow.

ORCID

Hunor Csala https://orcid.org/0000-0003-2993-8759
Amirhossein Arzani https://orcid.org/0000-0002-3706-7909

REFERENCES

1. Bissell MM, Raimondi F, Ait Ali L, et al. 4D flow cardiovascular magnetic resonance consensus statement: 2023 update. *J Cardiovasc Magn Reson*. 2023;25(1):40.
2. El Sayed R, Sharifi A, Park CC, Haussen DC, Allen JW, Oshinski JN. Optimization of 4D flow MRI spatial and temporal resolution for examining complex hemodynamics in the carotid artery bifurcation. *Cardiovascular. Eng Technol*. 2023;14:1-13.
3. Zhou A, Jagadeesan BD, Moen SL, Grande AW, Moortele VPF. Three-dimensional vortex characterization in small intracranial aneurysms based on four dimensional flow magnetic resonance imaging at 7 tesla. *AIP Adv*. 2021;11(9).
4. Xue Z, Charonko JJ, Vlachos PP. Particle image velocimetry correlation signal-to-noise ratio metrics and measurement uncertainty quantification. *Meas Sci Technol*. 2014;25(11):115301.
5. Raffel M, Willert CE, Scarano F, Kähler CJ, Wereley ST, Kompenhans J. *Particle Image Velocimetry: a Practical Guide*. 3rd ed., Springer; 2018.
6. Sun Y, Li J, Xu Y, Zhang T, Weang X. Deep learning versus conventional methods for missing data imputation: A review and comparative study. *Expert Syst Appl*. 2023;227:120201.
7. Fan L, Zhang F, Fan H, Zhang C. Brief review of image denoising techniques. *Vis Comput Ind Biomed Art*. 2019;2:1-12.
8. Qin Z, Zeng Q, Zong Y, Xu F. Image inpainting based on deep learning: a review. *Displays*. 2021;69:102028.

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

9. Brunton SL, Kutz JN. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press; 2019.
10. Candès EJ, Tao T. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans Information Theory*. 2010;56(5):2053-2080.
11. Willcox K. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Comput Fluid*. 2006;35(2):208-226.
12. Nekkanti A, Schmidt OT. Gappy spectral proper orthogonal decomposition. *J Comput Phys*. 2023;478:111950.
13. Lee K, Mavris DN. Unifying perspective for gappy proper orthogonal decomposition and probabilistic principal component analysis. *AIAA J*. 2010;48(6):1117-1129.
14. Shu D, Li Z, Farimani AB. A physics-informed diffusion model for high-fidelity flow field reconstruction. *J Comput Phys*. 2023;478:111972.
15. Werhahn M, Xie Y, Chu M, Thuerey N. A multi-pass GAN for fluid flow super-resolution. *Proc ACM Comput Graph Interact Tech*. 2019;2(2):1-21.
16. Yu L, Yousif MZ, Zhang M, Hoyas S, Vinuesa R, Lim HC. Three-dimensional ESRGAN for super-resolution reconstruction of turbulent flows with tricubic interpolation-based transfer learning. *Phys Fluids*. 2022;34(12).
17. Bertalmio M, Bertozzi AL, Sapiro G. Navier-stokes, fluid dynamics, and image and video inpainting. In: 1. IEEE:I–I. 2001.
18. Arzani A, Dawson STM. Data-driven cardiovascular flow modelling: examples and opportunities. *J Royal Soc Interface*. 2021;18(175):20200802.
19. Gunn CA, Hu X, Vandenberghe L. Artifact rejection and missing data imputation in cerebral blood flow velocity signals via trace norm minimization. *Physiol Meas*. 2020;41(11):114003.
20. Valvano G, Martini N, Huber A, et al. Accelerating 4D flow MRI by exploiting low-rank matrix structure and hadamard sparsity. *Magn Reson Med*. 2017;78(4):1330-1341.
21. Zhang J, Brindise MC, Rothenberger SM, Markl M, Rayz VL, Vlachos PP. A multi-modality approach for enhancing 4D flow magnetic resonance imaging via sparse representation. *J Royal Soc Interface*. 2022;19(186):20210751.
22. Chatterjee P, Milanfar P. Is denoising dead? *IEEE Trans Image Process*. 2009;19(4):895-911.
23. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT press; 2016.
24. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA, Bottou L. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res*. 2010;11(12):3371-3408.
25. Ahmed SE, Pawar S, San O, Rasheed A, Iliescu T, Noack BR. On closures for reduced order models—a spectrum of first-principle to machine-learned avenues. *Phys Fluids*. 2021;33(9).
26. Luo Z, Wang L, Xu J, Chen M, Yuan J, Tan ACC. Flow reconstruction from sparse sensors based on reduced-order autoencoder state estimation. *Phys Fluids*. 2023;35(7).
27. Csala H, Dawson S, Arzani A. Comparing different nonlinear dimensionality reduction techniques for data-driven unsteady fluid flow modeling. *Phys Fluids*. 2022;34(11).
28. Fukami K, Nakamura T, Fukagata K. Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data. *Phys Fluids*. 2020;32(9).
29. Tauro F, Grimaldi S, Porfiri M. Unraveling flow patterns through nonlinear manifold learning. *PLoS One*. 2014;9(3):e91131.
30. Wu J, Wang J, Xiao H, Ling J. Visualization of high dimensional turbulence simulation data using t-SNE. 2017.
31. Vidal R, Ma Y, Sastry SS. *Generalized Principal Components Analysis*. Springer; 2016.
32. Brindise MC, Vlachos PP. Proper orthogonal decomposition truncation method for data denoising and order reduction. *Exp Fluids*. 2017;58:1-18.
33. Scherl I, Strom B, Shang JK, Williams O, Polagye BL, Brunton SL. Robust principal component analysis for modal decomposition of corrupt fluid flows. *Phys Rev Fluids*. 2020;5:054401. doi:10.1103/PhysRevFluids.5.054401
34. Gondara L. Medical image denoising using convolutional denoising autoencoders. *IEEE*. 2016;241-246.
35. Iqbal N. DeepSeg: deep segmental denoising neural network for seismic data. *IEEE Trans Neural Netw Learn Syst*. 2022;34(7):3397-3404.
36. Lehtinen J, Munkberg J, Hasselgren J, et al. Noise2Noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*. 2018.
37. Krull A, Buchholz T, Jug F. Noise2void-learning denoising from single noisy images. *IEEE CVPR*. 2019;2129-2137.
38. Prakash M, Lalit M, Tomancak P, Krull A, Jug F. Fully unsupervised probabilistic noise2void. *IEEE ISBI*. 2020;154-158.
39. Prakash M, Delbracio M, Milanfar P, Jug F. Interpretable unsupervised diversity denoising and artefact removal. *arXiv preprint arXiv:2104.01374*. 2021.
40. Huang T, Li S, Jia X, Lu H, Liu J. Neighbor2neighbor: self-supervised denoising from single noisy images. *IEEE CVPR*. 2021;14781-14790.
41. Mansour Y, Heckel R. Zero-shot Noise2Noise: efficient image denoising without any data. *IEEE CVPR*. 2023;14018-14027.
42. Quan Y, Chen M, Pang T, Ji H. Self2self with dropout: learning self-supervised denoising from single image. *IEEE CVPR*. 2020;1890-1898.
43. Candès EJ, Li X, Ma Y, Wright J. Robust principal component analysis? *Journal of the ACM (JACM)*. 2011;58(3):1-37.
44. Chatpattanasiri C, Franzetti G, Bonfanti M, Diaz-Zuccarini V, Balabani S. Towards reduced order models via robust proper orthogonal decomposition to capture personalised aortic Haemodynamics. *bioRxiv*. 2023.

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

45. Zhou C, Paffenroth RC. Anomaly detection with robust deep autoencoders. 2017.
46. Vétel J, Garon A, Pelletier D. Denoising methods for time-resolved PIV measurements. *Exp Fluids*. 2011;51:893-916.
47. Vlasenko A, Schnorr C. Physically consistent and efficient variational denoising of image fluid flow estimates. *IEEE Trans Image Process*. 2009;19(3):586-595.
48. Nekkanti A, Schmidt OT. Frequency–time analysis, low-rank reconstruction and denoising of turbulent flows using SPOD. *J Fluid Mech*. 2021;926:A26.
49. Gu P, Han J, Chen DZ, Wang C. Reconstructing unsteady flow data from representative streamlines via diffusion and deep-learning-based denoising. *IEEE Comput Graph Appl*. 2021;41(6):111-121.
50. Yousif MZ, Zhang M, Yu L, Yang Y, Zhou H, Lim H. Physics-constrained deep reinforcement learning for flow field denoising. *J Fluid Mech*. 2023;973:A12.
51. Fathi MF, Perez-Raya I, Baghaie A, et al. Super-resolution and denoising of 4D-flow MRI using physics-informed deep neural nets. *Comput Methods Programs Biomed*. 2020;197:105729.
52. Gao H, Sun L, Wang JX. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Phys Fluids*. 2021;33(7):073603.
53. Shone F, Ravikumar N, Lassila T, et al. Deep physics-informed super-resolution of cardiac 4D-flow MRI. *Information Processing in Medical Imaging*. Springer; 2023:511-522.
54. Bakhshinejad A, Baghaie A, Vali A, Saloner D, Rayz VL, D'Souza RM. Merging computational fluid dynamics and 4D flow MRI using proper orthogonal decomposition and ridge regression. *J Biomech*. 2017;58:162-173.
55. Hoon dNHLC, Jalba AC, Farag ES, et al. Data assimilation for full 4D PC-MRI measurements: physics-based denoising and interpolation. *Comput Graph Forum*. 2020;39:496-512.
56. Ferdian E, Suinesiaputra A, Dubowitz DJ, et al. 4DFlowNet: super-resolution 4D flow MRI using deep learning and computational fluid dynamics. *Front Phys*. 2020;8:138.
57. Saitta S, Carioni M, Mukherjee S, Schönlieb CB, Redaelli A. Implicit neural representations for unsupervised super-resolution and denoising of 4D flow MRI. *arXiv preprint arXiv:2302.12835*. 2023.
58. Shit S, Zimmermann J, Ezhov I, et al. SRflow: deep learning based super-resolution of 4D-flow MRI data. *Front Artif Intell*. 2022;5:928181.
59. Rutkowski DR, Roldán-Alzate A, Johnson KM. Enhancement of cerebrovascular 4D flow MRI velocity fields using machine learning and computational fluid dynamics simulation data. *Sci Rep*. 2021;11(1):10240.
60. Arzani A, Wang JX, Sacks MS, Shadden SC. Machine learning for cardiovascular biomechanics modeling: challenges and beyond. *Ann Biomed Eng*. 2022;50(6):615-627.
61. Westerweel J, Scarano F. Universal outlier detection for PIV data. *Exp Fluid*. 2005;39:1096-1100.
62. Brindise MC, Rothenberger S, Dickerhoff B, et al. Multi-modality cerebral aneurysm haemodynamic analysis: in vivo 4D flow MRI, in vitro volumetric particle velocimetry and in silico computational fluid dynamics. *J Royal Soc Interface*. 2019;16(158):20190465. doi: 10.1098/rsif.2019.0465
63. Gao Q, Liu X, Wang H, et al. Optimization of 4D flow MRI velocity field in the aorta with divergence-free smoothing. *Med Biol Eng Comput*. 2021;59:2237-2252.
64. Koltukluoğlu TS, Blanco PJ. Boundary control in computational haemodynamics. *J Fluid Mech*. 2018;847:329-364.
65. Koltukluoglu TS, Hirsch S, Binter C, Kozerke S, Székely G, Laadhari A. A robust comparison approach of velocity data between MRI and CFD based on divergence-free space projection. *IEEE ISBI*. 2015;1393-1397.
66. Sache A, Reymond P, Brina O, Jung B, Farhat M, Vargas MI. Near-wall hemodynamic parameters quantification in in vitro intracranial aneurysms with 7 T PC-MRI. *Magn Reson Mater Phys Biol Med*. 2023;36(2):295-308.
67. Zhang J, Rothenberger SM, Brindise MC, et al. Divergence-free constrained phase unwrapping and denoising for 4D flow MRI using weighted least-squares. *IEEE Trans Med Imaging*. 2021;40(12):3389-3399.
68. Van Ooij P, Guédon A, Poelma C, et al. Complex flow patterns in a real-size intracranial aneurysm phantom: phase contrast MRI compared with particle image velocimetry and computational fluid dynamics. *NMR Biomed*. 2012;25(1):14-26.
69. Amili O, Schiavazzi D, Moen S, Jagadeesan B, Moortele VPF, Coletti F. Hemodynamics in a giant intracranial aneurysm characterized by in vitro 4D flow MRI. *PLoS One*. 2018;13(1):1-25. doi:10.1371/journal.pone.0188323
70. Sobral A, Bouwmans T, Zahzah E. LRSLibrary: low-rank and sparse tools for background modeling and subtraction in videos CRC press, Taylor and Francis Group. 2015.
71. Broaddus C, Krull A, Weigert M, Schmidt U, Myers G. Removing structured noise with self-supervised blind-spot networks. *IEEE ISBI*. 2020;159-163.
72. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*. 2015;234-241.
73. Krull A, Vičar T, Prakash M, Lalit M, Jug F. Probabilistic noise2void: unsupervised content-aware denoising. *Front Comput Sci*. 2020;2:5.
74. Troyanskaya O, Cantor M, Sherlock G, et al. Missing value estimation methods for DNA microarrays. *Bioinformatics*. 2001;17(6):520-525.
75. Mazumder R, Hastie T, Tibshirani R. Spectral regularization algorithms for learning large incomplete matrices. *J Mach Learn Res*. 2010;11:2287-2322.

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

76. Rubinsteyn A, Feldman S. fancyimpute: An Imputation Library for Python. 2016.
77. Tipping ME, Bishop CM. Probabilistic principal component analysis. *J R Stat Soc Series B Stat Methodol*. 1999;61(3):611-622.
78. Roweis S. EM algorithms for PCA and SPCA. *Adv Neural Information Process Syst*. 1997;10:626-632.
79. Bishop CM, Nasrabadi NM. *Pattern Recognition and Machine Learning. 4*. Springer; 2006.
80. Green S. pyppca. https://github.com/shergreen/pyppca 2019.
81. Porta JM, Verbeek JJ, Kröse BJA. Active appearance-based robot localization using stereo vision. *Auton Robots*. 2005;18:59-80.
82. Li Y, Amili O, Moen S, et al. Flow residence time in intracranial aneurysms evaluated by in vitro 4D flow MRI. *J Biomech*. 2022;141:111211.
83. Updegrove A, Wilson NM, Merkow J, Lan H, Marsden AL, Shadden SC. SimVascular: an open source pipeline for cardiovascular simulation. *Ann Biomed Eng*. 2017;45:525-541.
84. Sangalli LM, Secchi P, Vantini S. AneuRisk65: a dataset of three-dimensional cerebral vascular geometries. *Electron J Statist*. 2014;8(2):1879-1890.
85. Hoi Y, Wasserman BA, Xie YJ, et al. Characterization of volumetric flow rate waveforms at the carotid bifurcations of older adults. *Physiol Meas*. 2010;31(3):291-302.
86. Valen-Sendstad K, Piccinelli M, KrishnankuttyRema R, Steinman DA. Estimation of inlet flow rates for image-based aneurysm CFD models: where and how to begin? *Ann Biomed Eng*. 2015;43:1422-1431.
87. Kahana A, Turkel E, Dekel S, Givoli D. A physically-informed deep-learning model using time-reversal for locating a source from sparse and highly noisy sensors data. *J Comput Phys*. 2022;470:111592.
88. Gudbjartsson H, Patz S. The Rician distribution of noisy MRI data. *Magn Reson Med*. 1995;34(6):910-914.
89. Kim K, Ye JC. Noise2score: tweedie's approach to self-supervised image denoising without clean images. *Adv Neural Information Process Syst*. 2021;34:864-874.
90. Naik R, Trivedi N, Tarzanagh DA, Balzano L. Truncated matrix completion-an empirical study. *IEEE EUSIPCO*. 2022;847-851.
91. Rayz VL, Cohen-Gadol AA. Hemodynamics of cerebral aneurysms: connecting medical imaging and biomechanical analysis. *Annu Rev Biomed Eng*. 2020;22:231-256.
92. Marcos-Morales A, Leibovich M, Mohan S, et al. Evaluating Unsupervised Denoising Requires Unsupervised Metrics. *arXiv preprint arXiv:2210.05553*. 2022.
93. Ulyanov D, Vedaldi A, Lempitsky V. Deep Image Prior. *IEEE CVPR*. 2018;9446-9454.
94. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. *Nat Rev Phys*. 2021;3(6):422-440.
95. Candes EJ, Plan Y. Matrix completion with noise. *Proc IEEE*. 2010;98(6):925-936.
96. Chen Y, Chi Y, Fan J, Ma C, Yan Y. Noisy matrix completion: understanding statistical guarantees for convex relaxation via nonconvex optimization. *SIAM J Optim*. 2020;30(4):3098-3121.
97. Chen Y, Fan J, Ma C, Yan Y. Bridging convex and nonconvex optimization in robust PCA: noise, outliers, and missing data. *Ann Stat*. 2021;49(5):2948.
98. Callaham JL, Maeda K, Brunton SL. Robust flow reconstruction from limited measurements via sparse representation. *Phys Rev Fluids*. 2019;4(10):103907.
99. Zhang X, Ji T, Xie F, Zheng H, Zheng Y. Unsteady flow prediction from sparse measurements by compressed sensing reduced order modeling. *Comput Methods Appl Mech Eng*. 2022;393:114800.
100. Krissian K, Aja-Fernández S. Noise-driven anisotropic diffusion filtering of MRI. *IEEE Trans Image Process*. 2009;18(10):2265-2274.
101. Ong F, Uecker M, Tariq U, et al. Robust 4D flow denoising using divergence-free wavelet transform. *Magn Reson Med*. 2015;73(2):828-842.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Csala H, Amili O, D'Souza RM, Arzani A. A comparison of machine learning methods for recovering noisy and missing 4D flow MRI data. *Int J Numer Meth Biomed Engng*. 2024;40(11):e3858. doi:10.1002/cnm.3858

20407947, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cnm.3858 by University Of Utah, Wiley Online Library on [28/01/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License