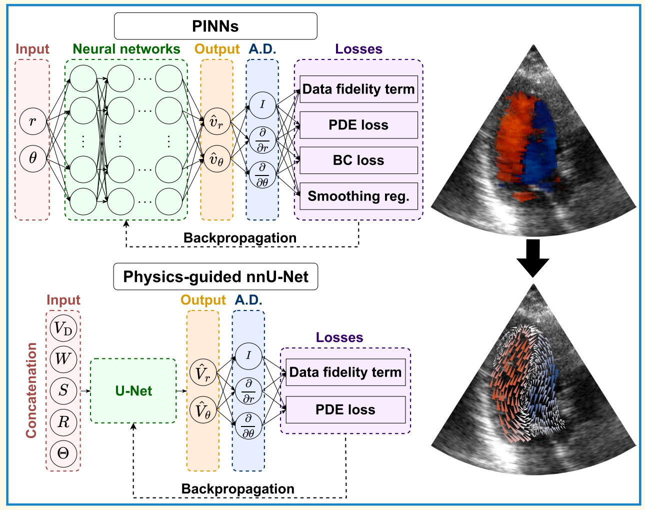
Physics-Guided Neural Networks for Intraventricular Vector Flow Mapping

Hang Jung Ling, Salomé Bru, Julia Puig, Florian Vixège, Simon Mendez, Franck Nicoud, Pierre-Yves Courand, Olivier Bernard, and Damien Garcia

***Abstract*—Intraventricular vector flow mapping (*i*VFM) seeks to enhance and quantify color Doppler in cardiac imaging. In this study, we propose novel alternatives to the traditional *i*VFM optimization scheme using physics-informed neural networks (PINNs) and a physics-guided nnU-Net-based supervised approach. When evaluated on simulated color Doppler images derived from a patient-specific computational fluid dynamics (CFD) model and in vivo Doppler acquisitions, both the approaches demonstrate comparable reconstruction performance to the original *i*VFM algorithm. The efficiency of PINNs is boosted through dual-stage optimization and pre-optimized weights. On the other hand, the nnU-Net method excels in generalizability and real-time capabilities. Notably, nnU-Net shows superior robustness on sparse and truncated Doppler data while maintaining independence from explicit boundary conditions. Overall, our results highlight the effectiveness of these methods in reconstructing intraventricular vector blood flow. The study also suggests potential applications of PINNs in ultrafast color Doppler imaging and the incorporation of fluid dynamics equations to derive biomarkers for cardiovascular diseases based on blood flow.**



***Index Terms*—Cardiac flow, color Doppler, deep learning (DL), echocardiography, physics-guided neural networks (PGNNs), physics-informed neural networks (PINNs), ultrasound, vector flow imaging.**

Manuscript received 19 March 2024; accepted 5 June 2024. Date of publication 10 June 2024; date of current version 27 November 2024. This work was supported by: MEGA Doctoral School (ED 162); French National Research Agency (ANR) through the "4D-iVFM" [ANR-21-CE19-0034-01] and "ORCHID" [ANR-22-CE45-0029-01] projects; LABEX PRIMES [ANR-11-LABX-0063] and LABEX CELYA [ANR-10-LABX-0060] of Université de Lyon within the program "Investissements d'Avenir" [ANR-11-IDEX-0007]; GENCI-IDRIS (HPC resources) [2023-AD010313603R1]. *(Corresponding author: Hang Jung Ling.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Human Ethical Review Committee of the University of Montreal Hospital Research Center (CRCHUM).

Hang Jung Ling, Julia Puig, Olivier Bernard, and Damien Garcia are with INSA-Lyon, Universite Claude Bernard Lyon 1, CNRS, Inserm, CREATIS UMR 5220, U1294, F-69621 Lyon, France (e-mail: hangjung.ling@gmail.com; hang-jung.ling@insa-lyon.fr; damien.garcia@creatis.insa-lyon.fr).

Salomé Bru and Simon Mendez are with IMAG UMR 5149, University of Montpellier, Montpellier, France.

Florian Vixège is with Université Grenoble Alpes, CNRS, Grenoble INP, LEGI, Grenoble, France.

Franck Nicoud is with IMAG UMR 5149, University of Montpellier, Montpellier, France, and also with the Institut Universitaire de France (IUF), Paris, France.

Pierre-Yves Courand is with INSA-Lyon, Universite Claude Bernard Lyon 1, CNRS, Inserm, CREATIS UMR 5220, U1294, F-69621 Lyon, France, and also with the Cardiology Department, Hôpital Lyon Sud, Hospices Civils de Lyon, Lyon, France.

This article has supplementary downloadable material available at https://doi.org/10.1109/TUFFC.2024.3411718, provided by the authors.

Digital Object Identifier 10.1109/TUFFC.2024.3411718

I. Introduction

AMONG the methods aiming to perform intracardiac flow imaging by color Doppler, intraventricular vector flow mapping (*i*VFM) [1], [2], [3], [4], [5] stands out as a postprocessing approach applicable to clinical color Doppler acquisitions. The *i*VFM algorithm relies on a constrained least-squares optimization scheme (see Section II-A for more details).

Recently, physics-informed neural networks (PINNs) [6] have emerged as a novel approach for data-driven optimization by integrating neural networks (NNs) and the laws of physics during the optimization process. The physical laws, often described by partial differential equations (PDEs), are incorporated into the loss function to enforce the correctness of the solutions. Automatic differentiation [7] has proven to be efficient in computing partial derivatives in PINNs. In cases involving strong nonlinear PDEs in the spatiotemporal domain, extensions to PINNs, such as conservative PINNs (cPINNs) and extended PINNs (XPINNs), have been proposed [8].

PINNs have found applications predominantly in fluid mechanics [8]. In the medical field, Arzani et al. [9] used PINNs to recover blood flow from sparse data in 2-D stenosis and aneurysm models. Kissas et al. [10] applied PINNs to predict arterial blood pressure from 4-D flow MRI data. In the

Highlights

- **Our PINNs, with dual-stage optimization and pre-optimized weights, demonstrated flexibility and performance comparable to the original *i*VFM approach for vector flow mapping using color Doppler.**
- **Our physics-guided nnU-Net-based supervised approach achieved robust intraventricular blood flow reconstruction with quasi-real-time inference, even on sparse and truncated Doppler data.**
- **Our study introduced innovative AI-driven and physics-guided approaches for clinical vector flow mapping, paving the way for enhanced diagnostic accuracy of cardiovascular diseases.**

ultrasound domain, PINNs have been primarily used for modeling wave propagation [11], shear wave elastography [12], and regularizing velocity field given by ultrafast vector flow imaging [13].

While PINNs have demonstrated effectiveness on sparse and incomplete data, their application remains unexplored in scenarios where one or more velocity components are missing, as is the case in *i*VFM. Color Doppler imaging provides only scalar information—the Doppler velocity, representing the noisy radial velocity—from which we aim to derive both the radial and angular velocity components of intraventricular blood flow.

PINNs often require reoptimization for new cases with different initial or boundary conditions, which can be time-consuming. A potential solution is physics-guided supervised learning [14], which produces output that adheres to the laws of physics using a physics-constrained training dataset, with optional physical regularization terms. Once trained, inference can be performed seamlessly on unseen data, provided their distribution closely resembles that of the training dataset.

In this article, we investigated the feasibility of using physics-based NNs for vector flow mapping, exploring both a physics-guided supervised approach implemented through the nnU-Net framework [15] and two variants of PINNs. Our contributions included the following:

1) training a physics-guided supervised approach based on nnU-Net, which showed high robustness on sparse and truncated data with nearly real-time inference speed;
2) implementing two PINN variants based on the penalty method to perform vector flow mapping, achieving performance comparable to the original *i*VFM algorithm;
3) using dual-stage optimization and pre-optimized weights from a selected Doppler frame, which enhanced PINNs' performance and reduced the optimization time of PINNs by up to 3.5 times.

II. Related Work

A. Intraventricular Vector Flow Mapping

As introduced in [1], a vector blood flow map within the left ventricle can be obtained from clinical color Doppler echocardiography by solving a minimization problem. This optimization task is governed by two equality constraints: C_1, representing the mass conservation equation, and C_2, representing the free-slip boundary conditions. The first constraint ensures the 2-D free divergence of the optimized velocity field, while the second constraint enforces that the normal component of the blood velocity is zero relative to the endocardial surface. In addition, a smoothing regularization, further detailed in Section III-A, is incorporated to impose spatial smoothness of the velocity field. Equation (1) expresses the mathematical formulation of this problem.

In (1), $(\hat{v}_r, \hat{v}_\theta)$ denote the estimated radial and angular blood velocity components. Here, Ω stands for the domain of interest, i.e., the left ventricle cavity, with its endocardial boundary denoted by $\partial\Omega$. The term ω indicates the weights of the data fidelity term. Weights equal to normalized Doppler power values in the range of $[0, 1]$ were used with in vivo Doppler data, as they reflect the reliability of the Doppler velocity. For simulated data, we used ω equal to 1. v_D refers to the sign-inverted Doppler velocity (positive velocities for movement away from the probe) to ensure the sign compatibility between v_D and v_r. The vectors $\boldsymbol{n}_\mathrm{W} = (n_{\mathrm{W}_r}, n_{\mathrm{W}_\theta})$ and $\boldsymbol{v}_\mathrm{W} = (v_{\mathrm{W}_r}, v_{\mathrm{W}_\theta})$ represent a unit vector perpendicular to the endocardial wall and a velocity vector of the endocardial wall, respectively.

The *i*VFM method [1] linearizes the constrained problem (1) and solves it using Lagrange multipliers and a least-squares optimization scheme, which reduces the number of supervisedly determined parameters to just one, namely, the smoothing regularization weight

$$\hat{\boldsymbol{v}} = (\hat{v}_r, \hat{v}_\theta) = \arg\min_{(v_r, v_\theta)} \underbrace{\int_\Omega \omega \| v_r - v_\mathrm{D} \| \, d\Omega}_{\text{closely match the Doppler data}}$$

$$\text{s.t.:} \quad \begin{cases} C_1 = r\,\mathrm{div}(\hat{\boldsymbol{v}}) = r\dfrac{\partial \hat{v}_r}{\partial r} + \hat{v}_r + \dfrac{\partial \hat{v}_\theta}{\partial \theta} = 0, & \text{on } \Omega \\ C_2 = (\hat{\boldsymbol{v}} - \boldsymbol{v}_\mathrm{W}) \cdot \boldsymbol{n}_\mathrm{W} = (\hat{v}_r - v_{\mathrm{W}_r}) n_{\mathrm{W}_r} \\ \quad + (\hat{v}_\theta - v_{\mathrm{W}_\theta}) n_{\mathrm{W}_\theta} = 0, & \text{on } \partial\Omega. \end{cases} \tag{1}$$

B. Physics-Informed Neural Networks

Unlike the conventional least-squares Lagrangian optimization, such as in *i*VFM, PINNs use an iterative scheme with multilayer perceptrons (MLPs) to iteratively refine and reach the optimal solution. PINNs offer the advantage of being flexible, especially in the optimization scheme, regardless of the linearity of the problem [6]. When additional complex physical constraints need to be incorporated, PINNs require minimal architectural modifications, typically requiring only the adaptation of the loss function. However, this process can

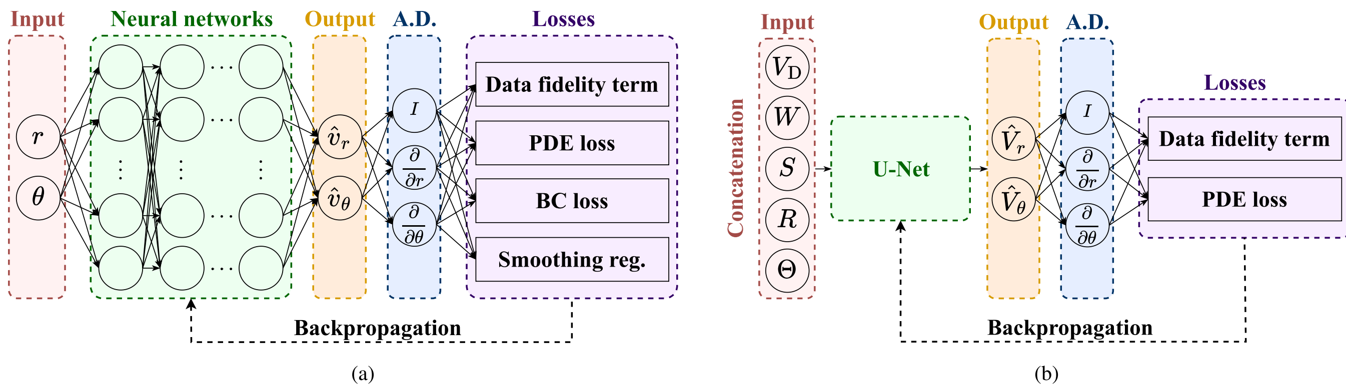


Fig. 1. Architectures of (a) PINNs and (b) physics-guided nnU-Net. A.D. refers to automatic differentiation. In (a), the 2-D input of PINNs has a shape of ($B \times 2$), where B is the batch size. r and θ denote radial and angular coordinates, respectively. In (b), nnU-Net takes a 4-D input of shape ($B \times 5 \times 192 \times 40$), which is the concatenation of sign-inverted dealiased Doppler velocity V_D, weight matrix W, left ventricular segmentation S, radial coordinate array R, and angular coordinate array Θ.

be challenging when applied to standard approaches involving nonlinear constraints.

When addressing a constrained optimization problem using PINNs, the problem is reformulated into a series of loss functions, which often involve conflicting objectives. To manage multiobjective optimization in PINNs, a linear scalarization of the losses is commonly used

$$\mathcal{L}_{\mu}(\theta_{\text{NN}}) = \sum_{j=1}^{N} \mu_j \mathcal{L}_j(\theta_{\text{NN}}), \quad \mu_j \in \mathbb{R}_{>0} \tag{2}$$

with μ_j representing the penalty coefficients or the loss weights, $\mathcal{L}_{1,\ldots,N}$ being the multiple losses derived from the original constrained optimization problem, and θ_{NN} representing the network parameters. All the losses involved in PINNs' optimization are functions of θ_{NN}. However, for better readability of the equations, θ_{NN} is omitted in the subsequent loss expressions.

Among the methods with linear scalarization, two notable approaches are the soft constraints and penalty methods. The soft constraints approach uses fixed penalty coefficients throughout the optimization. However, determining optimal coefficients can be challenging, especially as the number of objectives (N) increases, making this approach generally less favored in PINNs' optimization.

In contrast to the soft constraints approach, the penalty method involves varying coefficients. Various techniques have been proposed to adapt these coefficients during optimization. Examples include GradNorm [16], SoftAdapt [17], or ReLoBRaLo [18]. GradNorm and SoftAdapt dynamically adjust the loss weights based on the relative training rates of different losses. ReLoBRaLo can be seen as a combination of the former two techniques, which incorporates a moving average for loss weights and a random look-back mechanism. The random look-back mechanism is controlled by a variable that determines whether the loss statistics of the previous steps or those of the first step are used to compute the coefficients.

Recently, an alternative method called augmented Lagrangian (AL) has been proposed for solving constrained optimization problems using PINNs [19], [20]. Similar to the penalty method, the AL approach involves penalty terms, but it also introduces a term designed to mimic a Lagrange multiplier.

In situations where optimizing initial or boundary conditions is difficult, some studies have proposed imposing those conditions as hard constraints using a distance function and an analytical approximation of the conditions [21], [22]. Although it is feasible to impose hard constraints on the output of NNs, it is often challenging and more appropriate for problems with several initial or boundary conditions.

III. Methods

In this study, we addressed the constrained optimization problem of *i*VFM (1) through NNs aided by physics: a physics-guided supervised approach based on nnU-Net and PINNs using the penalty method. Specifically, we studied two variants of PINNs: 1) PINNs with the ReLoBRaLo weight-adapting strategy (RB-PINNs) and 2) AL PINNs (AL-PINNs). Schematic representations of the general architectures of PINNs and nnU-Net for intraventricular vector flow reconstruction are shown in Fig. 1(a) and (b), respectively.

The following sections introduce the loss functions to be optimized in PINNs (see Section III-A), provide implementation details for PINNs (see Sections III-B–III-E), discuss the physics-guided nnU-Net approach (see Section III-F), and present the evaluation metrics (see Section III-G).

A. PINNs' Loss Functions

In line with the previous *i*VFM method, we decomposed the mathematical formulation in (1) into several objectives to be optimized [see (3)–(6)]: 1) $\mathcal{L}_1$: data fidelity term; 2) $\mathcal{L}_2$: mass conservation residual loss (PDE loss); 3) $\mathcal{L}_3$: boundary condition residual loss (BC loss); and 4) $\mathcal{L}_4$: smoothing regularization.

For the PDE loss, namely, $\mathcal{L}_2$, the partial derivatives were computed using automatic differentiation. On the contrary, for $\mathcal{L}_4$, the partial derivatives were obtained using finite difference methods with 2-D convolution kernels, as spatial smoothness could not be computed with automatic differentiation. This was done by setting the weights of the 3×3 convolution kernels to the central finite difference coefficients with second-order accuracy.

The norm $\|\cdot\|$ used for computing $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ was Smooth L1 loss or Huber loss with $\beta = 1.0$ and *sum* reduction over all the samples unless otherwise stated. The Smooth L1 loss uses a squared term if the absolute error falls below β and an absolute term otherwise, making it less sensitive to outliers than the mean squared error

$$\begin{cases} \mathcal{L}_1 = \omega \left\| \hat{v}_r - v_{\mathrm{D}} \right\|, & \text{on } \Omega \quad (3) \\ \mathcal{L}_2 = \left\| r \dfrac{\partial \hat{v}_r}{\partial r} + \hat{v}_r + \dfrac{\partial \hat{v}_\theta}{\partial \theta} \right\|, & \text{on } \Omega \quad (4) \\ \mathcal{L}_3 = \left\| (\hat{v}_r - v_{\mathrm{W}_r}) n_{\mathrm{W}_r} + (\hat{v}_\theta - v_{\mathrm{W}_\theta}) n_{\mathrm{W}_\theta} \right\|, & \text{on } \partial\Omega \quad (5) \\ \mathcal{L}_4 = \displaystyle\sum_{k \in \{r,\theta\}} \left\{ \left(r^2 \frac{\partial^2 v_k}{\partial r^2} \right)^2 + 2 \left(r \frac{\partial^2 v_k}{\partial r \partial \theta} \right)^2 + \left(\frac{\partial^2 v_k}{\partial \theta^2} \right)^2 \right\}, & \text{on } \Omega. \quad (6) \end{cases}$$

B. RB-PINNs

1) Global Loss: The global loss function to be optimized in RB-PINNs was defined as

$$\mathcal{L}_{\mu,\theta_{\mathrm{NN}}} = \underbrace{\mu_1 \mathcal{L}_1}_{\substack{\text{data fidelity} \\ \text{term}}} + \underbrace{\mu_2 \mathcal{L}_2 + \mu_3 \mathcal{L}_3}_{\text{PDE \& BC loss}} + \underbrace{\mu_4 \mathcal{L}_4}_{\substack{\text{smoothing} \\ \text{reg.}}}, \quad (7)$$

where $\mu_1, \mu_2, \mu_3 \in \mathbb{R}_{>0}$ are adaptive penalty coefficients, and μ_4 is the smoothing regularization weight. We heuristically set μ_4 to $10^{-7.5}$.

Algorithm 1 ReLoBRaLo Update Strategy

Initialize $\mu_j(1) = 1$, $j \in \{1, 2, 3\}$.
for $i = 1, \ldots, I$ **do**
  Forward pass and compute losses
  $\mathcal{L}_j(i) \leftarrow \mathcal{L}_j$
  if $i >= 2$ **then**
    $\hat{\mu}_j^{(i,i-1)} \leftarrow n_{\mathrm{loss}} \times \mathrm{Softmax}\left(\frac{\mathcal{L}_j(i)}{\mathcal{T}\mathcal{L}_j(i-1)+\epsilon}\right)$
    $\hat{\mu}_j^{(i,1)} \leftarrow n_{\mathrm{loss}} \times \mathrm{Softmax}\left(\frac{\mathcal{L}_j(i)}{\mathcal{T}\mathcal{L}_j(1)+\epsilon}\right)$
    $\mu_j(i) \leftarrow \alpha\left(\rho \mu_j(i-1) + (1-\rho)\hat{\mu}_j^{(i,1)}\right) + (1-\alpha)\hat{\mu}_j^{(i,i-1)}$
  end
  Compute final loss using (7) and do backpropagation to update network parameters:
  $\boldsymbol{\theta}_{\mathrm{NN}} \leftarrow \boldsymbol{\theta}_{\mathrm{NN}} - \eta_{\theta_{\mathrm{NN}}} \nabla_{\theta_{\mathrm{NN}}} \mathcal{L}_{\mu,\theta_{\mathrm{NN}}}(i)$
end

---

2) Update Strategy for Loss Weights: Algorithm 1 details the ReLoBRaLo update strategy introduced in [18] for determining the loss weights, i.e., μ_1, μ_2, and μ_3. In this algorithm, n_{loss} represents the total number of losses for which the loss weights are updated; in our case, $n_{\mathrm{loss}} = 3$. $\hat{\mu}_j^{(i,i')}$ computes the scaling based on the relative improvement of $\mathcal{L}_j$ between the iterations i' and i. $\mu_j(i)$ is defined as the weight for $\mathcal{L}_j$ at the ith iteration, obtained through an exponential decay. The algorithm's hyperparameters consist of α for the exponential decay rate, ρ for a Bernoulli random variable with an expected value close to 1, $\mathcal{T}$ for temperature, and I for the total number of iterations. We heuristically set $\alpha = 0.999$, $\mathbb{E}(\rho) = 0.999$, and $\mathcal{T} = 1.0$, as this combination yielded the best results for our problem. θ_{NN} denotes the learnable network parameters, $\eta_{\theta_{\mathrm{NN}}}$ is the learning rate used for updating network parameters, and $\nabla_{\theta_{\mathrm{NN}}}$ is the gradient of the final loss with respect to θ_{NN}.

C. AL-PINNs

1) Global Loss: For AL-PINNs, we defined its global loss as

$$\mathcal{L}_{\lambda,\mu,\theta_{\mathrm{NN}}} = \underbrace{\mathcal{L}_1}_{\substack{\text{data fidelity} \\ \text{term}}} + \underbrace{\langle \boldsymbol{\lambda_1}, \boldsymbol{C_1} \rangle + \langle \boldsymbol{\lambda_2}, \boldsymbol{C_2} \rangle}_{\substack{\text{PDE \& BC loss} \\ \text{(Lagrange multipliers)}}} + \underbrace{0.5 \times \mu \times (\mathcal{L}_2 + \mathcal{L}_3)}_{\text{PDE \& BC loss (penalty)}} + \underbrace{\mu_4 \mathcal{L}_4.}_{\substack{\text{smoothing} \\ \text{reg.}}} \quad (8)$$

In this equation, λ_1 and λ_2 are learnable real Lagrange multipliers related to the two constraints: the mass conservation, C_1, and the free-slip boundary condition, C_2. The notation $\langle \cdot, \cdot \rangle$ refers to the inner product of two vectors. The learnable penalty coefficient for the two physical constraints is denoted by $\mu \in \mathbb{R}_{>0}$. Similar to RB-PINNs, μ_4 was set heuristically to $10^{-7.5}$.

Algorithm 2 AL Update Strategy

Initialize $\boldsymbol{\lambda_j} = \vec{0}$, $\mu = 2$, $j \in \{1, 2\}$.
for $i = 1, \ldots, I$ **do**
  Forward pass and compute losses
  $\mathcal{L}_{\lambda,\mu,\theta_{\mathrm{NN}}}(i) \leftarrow \mathcal{L}_{\lambda,\mu,\theta_{\mathrm{NN}}}$
  Compute final loss using (8) and do backpropagation to simultaneously update network parameters, learnable $\boldsymbol{\lambda_j}$ and μ:
  $$\begin{cases} \boldsymbol{\theta}_{\mathrm{NN}} \leftarrow \boldsymbol{\theta}_{\mathrm{NN}} - \eta_{\theta_{\mathrm{NN}}} \nabla_{\theta_{\mathrm{NN}}} \mathcal{L}_{\lambda,\mu,\theta_{\mathrm{NN}}}(i) \\ \boldsymbol{\lambda_j} \leftarrow \boldsymbol{\lambda_j} + \eta_\lambda \nabla_{\lambda_j} \mathcal{L}_{\lambda,\mu,\theta_{\mathrm{NN}}}(i) \\ \mu \leftarrow \mu + \eta_\mu \nabla_\mu \mathcal{L}_{\lambda,\mu,\theta_{\mathrm{NN}}}(i) \end{cases}$$
end

---

2) Update Strategy for Loss Weights: We applied the gradient ascent method [20] to update λ_1, λ_2, and μ, as decribed in Algorithm 2. In the original approach [20], μ remains constant throughout the optimization process, but we proposed to update this penalty coefficient to better adhere to the original AL method [23]. In our experiments, the gradient ascent method showed higher optimization stability and was less prone to gradient explosion compared with the original AL update rule proposed in [23]. In Algorithm 2, θ_{NN}, $\eta_{\theta_{\mathrm{NN}}}$, and $\nabla_{\theta_{\mathrm{NN}}}$ represent the learnable network parameters, the learning rate for updating network parameters, and the gradient of the final loss with respect to θ_{NN}, respectively. $\nabla_{\lambda_j}\cdot$ and $\nabla_\mu\cdot$ denote the gradient of the final loss with respect to λ_j and μ, respectively; I indicates the total number of iterations; and η_λ and η_μ are the learning rates for the learnable Lagrange multipliers λ_j and μ, respectively. The selection of appropriate learning rates is critical in preventing gradient overflow when dealing with physical losses that involve unbounded Lagrange multipliers. Their values are discussed in Section IV-B1.

D. Dual-Stage Optimization

To improve the convergence of our PINNs, we introduced a dual-stage optimization strategy: 1) an optimization stage using the AdamW [24] optimizer for the first 90% of the iterations to converge to a rough solution and 2) a fine-tuning stage using the L-BFGS [25] optimizer, which is not sensitive to learning rates, for the remaining iterations to obtain an optimal final solution. This approach aimed to reduce the optimization time of PINNs. An ablation study was performed using RB-PINNs to assess the potential improvement of this strategy.

E. PINNs' Architecture, Weight Initialization, and Sampling Strategy

1) Network Architecture: For both the PINNs implemented in this article, we used an MLP with six hidden layers, each containing 60 neurons, with the *tanh* activation function. This architecture resulted in approximately 18.6k trainable parameters.

2) Weight Initialization: We first applied the dual-stage optimization to a Doppler frame selected at the end of the early filling phase using RB-PINNs. The resulting weights were then saved as pre-optimized weights and used as initialization for all subsequent PINN models before optimization on new Doppler data. This initialization technique aimed to accelerate the optimization process of our PINNs and enhance their performance. A second ablation study was carried out to justify this choice.

3) Sampling Strategy: Leveraging the regularly spaced polar grid of color Doppler imaging and its relatively small size, we used all the sample points within the left ventricle on the grid for both data and collocation points. These data points were used to compute the data fidelity term, while the PDE loss and smoothness term were evaluated from the collocation points. For the BC loss, all the extracted points on the boundary were considered.

F. Physics-Guided nnU-Net

We trained a physics-guided nnU-Net (refer to Fig. 1(b) for its architecture) with configurations similar to those described in [26, Table I] on both simulations and in vivo data. We adapted the loss function to L1 loss for supervised regression. In addition, to enforce mass conservation in the predicted velocity field, we incorporated $\mathcal{L}_2$ in the loss function as a physical regularization term with a weight of $\gamma = 10^{-3}$. Both supervised and regularization terms were masked with the left ventricular binary segmentation, restricting loss computation to the region of interest. The final loss was expressed as

$$\mathcal{L}_{\gamma,\theta_{\text{NN}}} = \underbrace{\left\| \hat{V}_r - V_{r_{\text{ref}}} \right\|_1 + \left\| \hat{V}_\theta - V_{\theta_{\text{ref}}} \right\|_1}_{\text{data fidelity term}} + \gamma \underbrace{\left\| r\frac{\partial \hat{V}_r}{\partial r} + \hat{V}_r + \frac{\partial \hat{V}_\theta}{\partial \theta} \right\|_1}_{\text{PDE loss}} \tag{9}$$

where $(V_{r_{\text{ref}}}, V_{\theta_{\text{ref}}})$ represent the reference velocity field given by simulations or predicted velocity field by *i*VFM [1], considered as the gold standard for in vivo data.

Unlike PINNs that directly take coordinates (r, θ) as input, nnU-Net requires image data. Our nnU-Net's input was a concatenation of: 1) dealiased color Doppler image before scan conversion for in vivo data or alias-free image for simulated data; 2) a weight matrix with normalized Doppler powers in the range of $[0, 1]$ for in vivo data or containing ones for simulated data; 3) binary segmentation of the left ventricle cavity; 4) radial coordinate array; and 5) angular coordinate array. More details about the training dataset are given in Section IV-A2.

During training, we applied data augmentations, including random rotation ($[-15°, 15°]$), random zoom ($[0.7, 1.4]$), and random scanline masking. With the latter, a block of n consecutive scanlines was randomly masked out with a step size of m. In our experiments, $m = 10$ and n was a random integer between 0 and 9. This strategy simulated sparse Doppler data, enhancing the model's robustness and generalizability.

G. Evaluation Metrics

We assessed the performance of RB-PINNs, AL-PINNs, and nnU-Net using simulated Doppler images from a patient-specific computational fluid dynamics (CFD) model (see Section IV-A1). Evaluation metrics, including squared correlation (r^2) and normalized root-mean-square error (nRMSE), were computed by comparing predicted and ground-truth velocity fields within the left ventricle.

1) Squared Correlation: We defined the squared correlation as follows:

$$r^2_{v_k} = \text{Corr}\left(\hat{v}_k, v_{k_{\text{CFD}}}\right)^2, \quad k \in \{r, \theta\}, \tag{10}$$

where Corr is the Pearson correlation coefficient.

2) nRMSE: For both the radial and angular components, we computed the root-mean-square errors normalized by the maximum velocity defined by

$$\text{nRMSE} = \frac{1}{\max\|v_{\text{CFD}}\|_2} \sqrt{\frac{1}{n}\sum_{k=1}^{n} \left\| \hat{v}_k - v_{\text{CFD}_k} \right\|_2^2}, \tag{11}$$

where n stands for the number of velocity samples in the left ventricular cavity. For the nRMSE metrics shown in Tables I, II, and IV, we considered both the velocity components and reported them as $(\tilde{x} \pm \sigma_{\text{rob}})$. Here, $\tilde{x}$ signifies the median, while $\sigma_{\text{rob}} = 1.4826 \times$ MAD represents the robust standard deviation (std.), with MAD denoting the mean absolute deviation.

IV. Experimental Setup and Results

A. Dataset

1) Patient-Specific Computational Fluid Dynamics Heart Model: To validate our approaches, we used a new patient-specific physiological CFD model of cardiac flow developed by the IMAG laboratory. This model features a more realistic mitral valve compared with the previous version [27], [28]. We followed the same method described in [1, Sec. 2.3] to generate 100 simulated Doppler images evenly

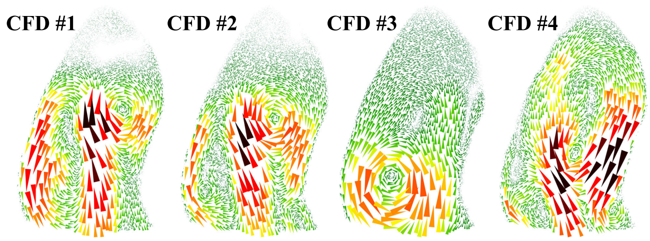


Fig. 2. Simulated color Doppler image during early filling derived from patient-specific CFD heart models with four variants of mitral valves. CFD #1–3 represent cases following mitral valve replacement with a bioprosthetic valve, while CFD #4 is a normal case.

TABLE I
ABLATION STUDY ON 100 SIMULATED DOPPLER IMAGES USING RB-PINNS. OPTIMIZATION TIME IS CONSISTENT (100 s/frame) ACROSS ALL CONFIGURATION COMBINATIONS. DEFAULT SETTINGS ARE HIGHLIGHTED IN PURPLE

Pre-optimized weights	Dual-stage optimization	$r^2(\uparrow)$		nRMSE [%]($\downarrow$)
		v_r	v_θ	$(\tilde{x} \pm \sigma_{\text{rob}})$
✘	✘	0.88	0.23	4.3 ± 2.2
✘	✔	0.97	0.57	2.8 ± 1.2
✔	✘	0.96	0.58	2.4 ± 1.0
✔	✔	**0.99**	**0.66**	$\mathbf{2.2} \pm 1.0$

distributed over a cardiac cycle, with a signal-to-noise ratio (SNR) equal to 50 dB. Each image comprised 80 scanlines with 200 samples per scanline. As Doppler power information was not available, we set the weight for the data fidelity term in both PINNs and *i*VFM to one, i.e., $\omega = W = 1$.

2) CFD and In Vivo Apical Three-Chamber Training Dataset for nnU-Net : Based on the CFD model described in Section IV-A1, we introduced variations in the mitral valve geometry, by modifying opening angle and orientation, to simulate three distinct cases following mitral valve replacement with a bioprosthetic valve (CFD #1–3) [29]. CFD #1 and #2 corresponded to two different inflow jet orientations, with a slight increase in mitral valve cross-sectional area in CFD #2. CFD #3 mimicked blood flow with a wide-opened mitral valve, resulting in a weak jet with limited penetration. CFD #1–3 were included as training/validation data for the nnU-Net, while the unaltered model representing a normal case (CFD #4) was used for testing. Fig. 2 provides an example of a simulated color Doppler image during early filling for each CFD model. Although these four models shared the underlying cardiac geometry, modifications to the mitral valve resulted in sufficiently diverse intraventricular flows to reduce the training bias.

Due to the limited availability of simulated Doppler data for training a supervised model, we chose to include in vivo apical three-chamber (A3C) duplex (B-mode + color Doppler) data in our training dataset. This decision was further elaborated in Section V-A. These data aligned with the dataset of prior studies [26], [30], acquired using a Vivid 7 ultrasound system (GE Healthcare, USA) with a GE 5S cardiac sector probe (bandwidth = 2–5 MHz). Further details about this dataset can be found in [26, Sec. III-A.1]. . We processed the in vivo A3C data to ensure high-quality training data. We initially filtered out low-quality data, resulting in a compilation of 92 Doppler echocardiographic cineloops from 37 patients, totaling 2668 frames. Subsequently, we performed preprocessing using ASCENT [26], [31]. This process involved segmenting the left ventricle cavity on B-mode images to define the region of interest and boundary conditions, which varied across frames, and correcting aliased pixels on the corresponding color Doppler images. Finally, we applied the *i*VFM method to reconstruct the 2-D vector field in the left ventricle, serving as the gold standard for training the nnU-Net. For training purposes, the physics-constrained dataset was split subjectwise into 74/10/8 clinical cineloops plus 2/1/1 CFD simulations, resulting in 2037/434/197 in vivo and 200/100/100 simulated images for training/validation/testing.

B. Training Strategies

All the methods were implemented in the same PyTorch-based framework to ensure consistent training and optimization. The training configurations for each approach were as follows.

1) RB-PINNs and AL-PINNs : For both RB-PINNs and AL-PINNs, we used a dual-stage optimization strategy, involving two stages with a total of $I = 2500$ iterations. In the first stage, we applied AdamW optimization for $0.9 \times I$ iterations, updating all the learnable parameters with a learning rate of $\eta_{\theta_{\text{NN}}} = \eta_\lambda = \eta_\mu = 10^{-5}$. Then, in the fine-tuning stage, which comprised the remaining 10% of the iterations, we used L-BFGS optimization. In this stage, only the network parameters were updated, while the learnable loss weights from the first stage were retained. The L-BFGS optimizer was configured with a maximum of ten iterations per optimization step and the strong Wolfe line search conditions.

This strategy ensured a balance between accuracy and optimization duration, enhancing the stability and efficiency of PINNs' optimization process. The advantage of this strategy was further demonstrated in Section IV-C1.

2) Physics-Guided nnU-Net : Our physics-guided nnU-Net underwent 1000 epochs of training with the following configurations: a patch size of (192 × 40) pixels, a batch size of 4, and SGD optimizer with an initial learning rate of 0.01, paired with a linear decay scheduler.

C. Experimental Results

1) Pre-Optimized Weights and Dual-Stage Optimization Enhanced PINNs' Performance: The ablation study presented in Table I highlights that the combination of pre-optimized weights and dual-stage optimization in RB-PINNs yielded the best performance within a fixed optimization time. The dual-stage optimization strategy significantly reduced the optimization time of PINN methods. Fig. 3 provides a visual representation of a case where optimization was conducted using pre-optimized weights with and without dual-stage optimization. In this example, single-stage optimization with AdamW required 3.5× more optimization time to achieve a visually similar solution compared with dual-stage optimization. Subsequent experiments with PINNs followed

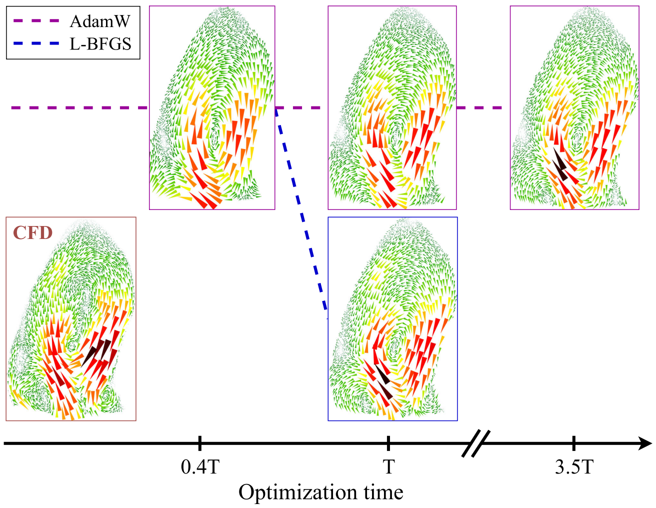


Fig. 3. Dual-stage (AdamW + L-BFGS) versus single-stage (AdamW only) optimization using RB-PINNs initialized with pre-optimized weights. *T* refers to the total amount of time required for dual-stage optimization. In this example, 3.5× more time is needed for single-stage optimization (top right) to converge to a similar solution given by dual-stage optimization (bottom right).

this optimization strategy—dual-stage optimization with pre-optimized weight initialization.

2) NN-Based Approaches Aligned With the Original iVFM : All the methods achieved high correlation in the radial velocity estimation on the 100 simulated color Doppler images derived from CFD #4, with $r^2_{v_r} > 0.98$ (see Fig. 4). For angular velocity correlation, both PINNs, RB-PINNs and AL-PINNs performed similar to *i*VFM, $r^2_{v_\theta} = 0.659$ and 0.669 versus 0.694, while nnU-Net surpassed *i*VFM ($r^2_{v_\theta} = 0.744$). This suggests the effectiveness of a supervised approach in learning intraventricular blood flow patterns. However, nnU-Net tended to be less precise when estimating highly negative radial velocities on simulated Doppler data, potentially due to the limited CFD training samples. Interestingly, all NN methods exhibited more errors for the radial component than *i*VFM, highlighting the high robustness and precision of the physics-constrained *i*VFM approach (see Fig. 5). The nRMSE of *i*VFM ranged between 0.2% and 1.6% and 1.4% and 21.3% for the radial and angular velocities, respectively. Among all the methods, AL-PINNs had the highest nRMSE for angular velocities (2.2%–23.3%). Despite having the highest nRMSE for radial velocities (3.8%–6.7%), nnU-Net produced the least errors in angular velocity estimation (2.3%–13.3%). A cineloop showing the reconstructed field by PINNs and nnU-Net versus CFD can be found in the Supplementary Material.

The final optimized values of the penalty coefficients were (median ± robust std. [min, max]): $\mu_1 = 0.90 \pm 0.11$ $[0.32, 1.00]$, $\mu_2 = 0.90 \pm 0.12$ $[0.32, 1.02]$, $\mu_3 = 1.19 \pm 0.22$ $[0.99, 2.36]$ for RB-PINNs, and $\mu = 2.02 \pm 0.01$ $[2.01, 2.07]$ for AL-PINNs.

3) nnU-Net Demonstrated Better Generalizability and Robustness on Sparse Doppler Data: Table II presents metrics for each method on both full and sparse simulated Doppler images. In the evaluation on full data, *i*VFM achieved the highest correlation for radial velocities and the lowest nRMSE, while nnU-Net, trained with the physical regularization term (PDE loss), excelled in the correlation of angular velocities. This implies that incorporating physical regularization helps constrain the nnU-Net's output to better adhere to the laws of physics.

TABLE II
METRICS COMPUTED ON 100 FULL AND SPARSE SIMULATED DOPPLER IMAGES

Methods	Full data			Sparse data*		
	$r^2(\uparrow)$		nRMSE [%]($\downarrow$)	$r^2(\uparrow)$		nRMSE [%]($\downarrow$)
	v_r	v_θ	($\tilde{x} \pm \sigma_{\text{rob}}$)	v_r	v_θ	($\tilde{x} \pm \sigma_{\text{rob}}$)
*i*VFM	**1.00**	0.69	**1.7** ± 1.0	0.85	0.58	3.2 ± 2.3
RB-PINNs	0.99	0.66	2.2 ± 1.0	0.86	0.56	3.4 ± 1.6
AL-PINNs	0.99	0.67	2.5 ± 1.0	0.80	0.58	3.5 ± 1.4
nnU-Net‡	0.99	0.60	2.3 ± 0.9	0.67	0.12	6.0 ± 3.0
nnU-Net†	0.99	**0.74**	2.1 ± 0.9	0.64	0.49	6.0 ± 3.3
nnU-Net	0.98	**0.74**	2.1 ± 0.9	**0.88**	**0.71**	**2.4** ± 1.0

* indicates data masked every 9 out of 10 scanlines from the center to the borders.
‡ means training without both physical regularization term (PDE loss) and random scanline masking augmentation.
† signifies training with physical regularization term (PDE loss), but without random scanline masking augmentation.

TABLE III
COMPARISON OF TRAINING, OPTIMIZATION, AND INFERENCE TIMES FOR NN-BASED METHODS AND *i*VFM

Methods	Device	No. trainable parameters	Training time	Optimization/ inference time per frame
*i*VFM	CPU	-	-	0.2 s
RB-PINNs	GPU	18.6 k	-	100 s
AL-PINNs	GPU	18.6 k	-	100 s
nnU-Net	GPU	7 M	12 h	**0.05 s**

As expected, the performance of all the methods was significantly impacted when evaluated on sparse data, where nine out of ten scanlines were masked. Remarkably, nnU-Net, which was trained with both the physical regularization term and the random scanline masking augmentation, demonstrated the least decline in performance. It maintained a high correlation for both radial and angular velocities while achieving the least nRMSE. This finding underscores the benefit of this augmentation in supervised learning for enhanced generalization. Despite having lower correlations than nnU-Net, *i*VFM remained robust, producing a lower nRMSE than RB-PINNs and AL-PINNs.

4) nnU-Net Exhibited Superior Reconstruction Speed: Table III provides a comparison of the training, optimization, and inference times for NN-based approaches against *i*VFM. These metrics were computed using a 16-GB V100 GPU for NN methods and an Intel i5-11500H CPU for *i*VFM. As the only supervised approach in the comparison, nnU-Net required 12 h of training but achieved the fastest per-frame inference time, taking only 0.05 s. *i*VFM ranked second, with a reconstruction time of 0.2 s/frame. Notably, both the PINNs necessitated longer optimization times, around 100 s, which is a recognized drawback of this approach.

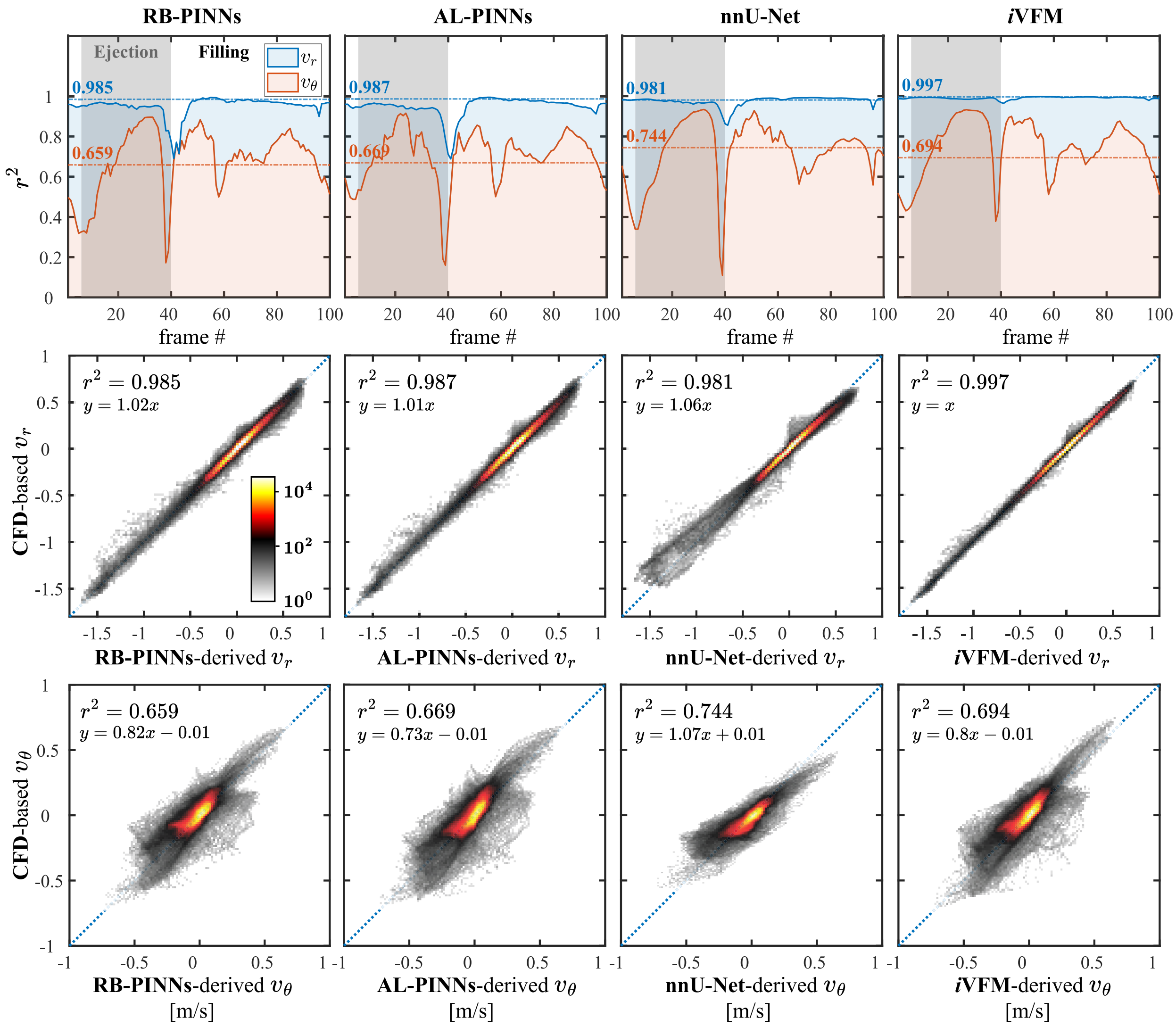


Fig. 4. Top row: Time-varying squared correlation between CFD-based velocities and reconstructed velocities by each method. Mid and bottom rows: CFD-based velocities versus estimated velocities derived from various methods. For mid and bottom rows, velocity data from 100 simulated color Doppler images were pooled. The binned scatter plots show the number of velocity occurrences.

5) Clinical Application of Vector Blood Flow Mapping: Fig. 6 showcases the intraventricular vector blood flow mapping by various methods on an in vivo case at different cardiac phases, including ejection, early filling, diastasis, and late filling. The reconstructed flow patterns by all the methods appear relatively similar, with *i*VFM generating the smoothest flow patterns. Although the prominent vortex was less visible during early filling, it became more pronounced at the center of the left ventricle cavity during diastasis. Another example of vector blood flow reconstruction by all four methods on in vivo data is given in the Supplementary Material.

6) Robustness of nnU-Net on Truncated Clinical Doppler Data: Unlike other methods that required explicit boundary conditions, nnU-Net learned these conditions implicitly during training. This offered an advantage as it reduced the need for specific knowledge about the flow at the endocardium. Table IV illustrates nnU-Net's behavior under Doppler scanline truncation, achieved by progressively cutting scanlines from both sides toward the center. The metrics were computed within the common region remaining after truncation. The results show stable performance up to a 50% reduction. However, beyond this threshold, a more pronounced decrease in performance was observed. Fig. 7 visually demonstrates nnU-Net's ability to consistently produce accurate intraventricular vector blood flow reconstructions, even with a significant 70% truncation.

V. Discussion

Our study introduces alternative approaches to the physics-constrained *i*VFM algorithm [1], leveraging the power of NNs: PINNs (RB-PINNs and AL-PINNs) and a physics-guided supervised technique (nnU-Net). These methods offer distinct strategies for the inherent constrained optimization problem in *i*VFM.

In the PINNs' framework, we addressed the same optimization problem as in *i*VFM, but solved it differently using gradient descent and NNs. By incorporating governing equations, such as mass conservation and boundary conditions,

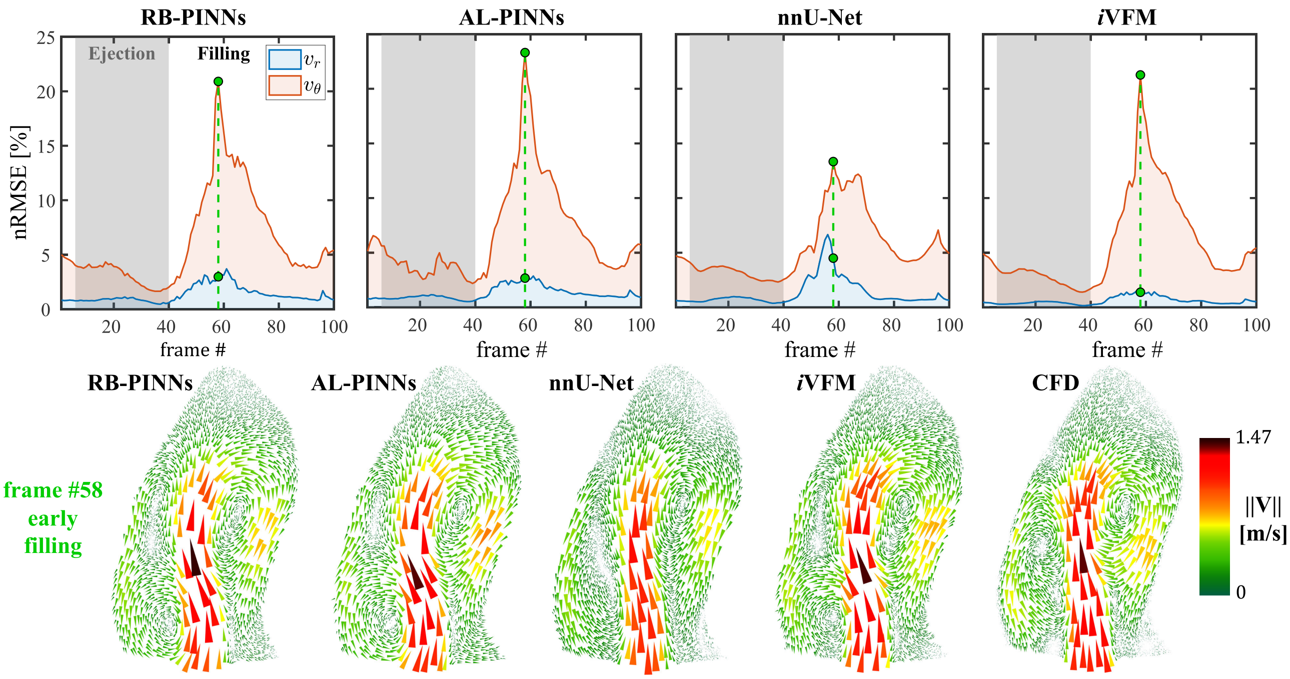


Fig. 5. nRMSEs between CFD-based and estimated velocity vectors by different techniques.

TABLE IV
NNU-NET'S METRICS COMPUTED ON EIGHT IN VIVO TEST CINELOOPS OF 197 FRAMES WITH DIFFERENT PERCENTAGES OF SCANLINE TRUNCATION

Percentage of truncation [%]	$r^2(\uparrow)$		nRMSE [%]($\downarrow$)
	v_r	v_θ	($\tilde{x} \pm \sigma_{\text{rob}}$)
20	0.97	0.96	2.4 ± 1.2
40	0.99	0.92	3.7 ± 1.5
50	0.99	0.87	4.3 ± 1.6
60	0.97	0.75	6.5 ± 2.7
70	0.94	0.59	8.4 ± 2.9

Note: Comparison made with nnU-Net's estimated velocity fields on full scanline data.

PINNs inherently enforce physical laws during optimization, potentially leading to optimized intraventricular vector velocity fields.

On the other hand, the supervised approach (nnU-Net) was trained on patient-specific CFD-derived simulations and *i*VFM-estimated velocity field on in vivo Doppler data. The network learned the underlying flow patterns while adhering to physical principles through the use of physics-constrained labels and the physical regularization term in the loss function. This approach demonstrated robustness to data limitations, such as missing scanlines.

A. PINNs Versus Physics-Guided nnU-Net

The application of PINNs deviates from the conventional optimization methods by leveraging NNs to find the optimal solution, which can be advantageous in complex physical problems. Although PINNs may not necessarily outperform analytical or numerical methods in computational efficiency, approximation accuracy, or convergence guarantees [32], they offer a unique advantage in terms of flexibility. This flexibility allows their architecture to remain relatively consistent across various physical optimization problems by adapting the loss functions to be optimized.

In our case of intraventricular blood flow reconstruction, we successfully improved the computational efficiency of PINNs while maintaining accuracy comparable to *i*VFM. This was achieved by implementing a dual-stage optimization with the use of pre-optimized weights. For future exploration, both the PINNs' architectures could benefit from imposing hard boundary conditions rather than optimizing them in the form of soft constraints. In addition, better strategies for automatically determining or learning the optimal smoothing regularization weight, rather than relying on heuristic search methods, will be investigated to further improve the robustness of PINNs.

Unlike PINNs, nnU-Net operates within a supervised learning framework, heavily relying on labeled training data. In this study, CFD-derived simulations played a crucial role in providing ground-truth velocity fields for training. This explains the high squared correlation achieved by nnU-Net, as the simulated training samples shared the exact heart geometry despite variations in flow patterns due to different mitral valve conditions.

However, when trained exclusively on simulated data, nnU-Net struggled to correctly estimate vector blood flow in in vivo color Doppler data due to a distribution shift between the simulated and real data. This arose from limitations in the current physiological spectrum of the simulations. To bridge this gap, we included *i*VFM-estimated velocities in our training data, allowing nnU-Net to learn from solutions representative of clinical Doppler data generated by the established *i*VFM method. As shown in Fig. 6, nnU-Net effectively learned the underlying flow pattern and physical properties,

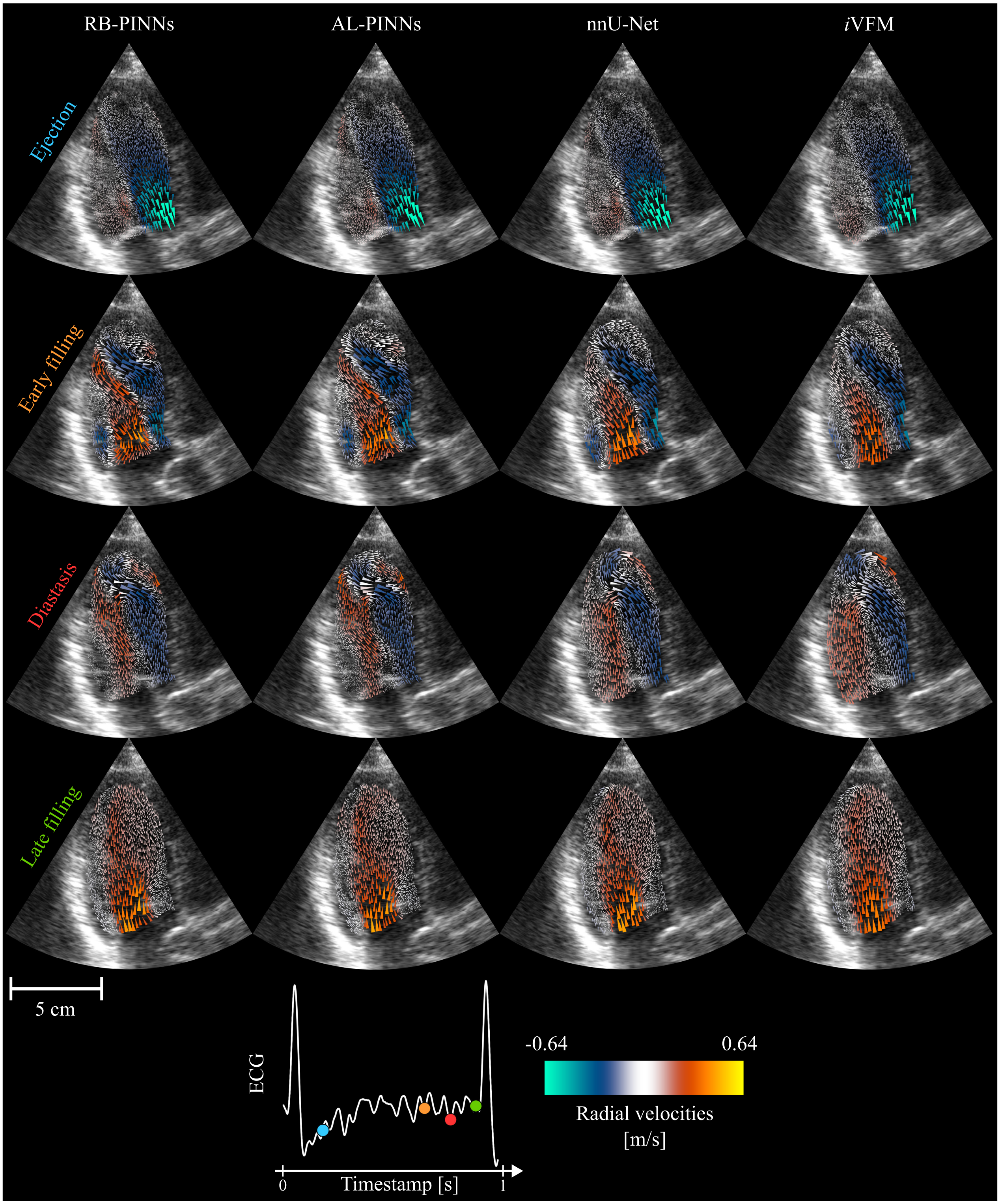


Fig. 6. Reconstruction of intraventricular vector blood flow in a patient using NN-based approaches and *i*VFM. The color of the arrows represents the estimated radial velocity fields.

i.e., the free-divergence and boundary conditions, from the training samples generated by *i*VFM.

Moreover, as illustrated in Fig. 7, nnU-Net can precisely reconstruct intraventricular flow on truncated Doppler data, where PINNs and *i*VFM cannot be directly applied as such due to incomplete and unknown boundary conditions. This potentially makes nnU-Net the preferred candidate for clinical applications, especially considering that most clinical Doppler acquisitions do not capture the entire left ventricular cavity due to limitations in probe placement or patient anatomy. With the added advantage of the shortest inference time, nnU-Net has real-time capabilities suitable for

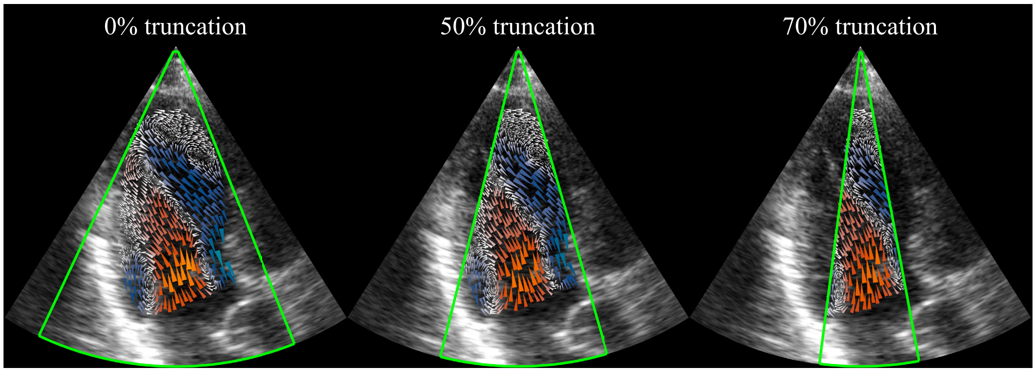


Fig. 7. Intraventricular vector blood flow reconstruction from Doppler data with varying percentages of scanline truncation using physics-guided nnU-Net. The color of the arrows represents the estimated radial velocity fields.

clinical settings. Future work will focus on generating more patient-specific CFD models and creating more realistic simulated Doppler data [33] to avoid the bias associated with using *i*VFM estimates as a reference to train our model.

B. Limitations of Color Doppler and Vector Flow Mapping

Conventional color Doppler echocardiography is subject to various limitations, posing challenges for accurate vector flow mapping. These limitations include clutter signals arising from myocardial tissue and valve leaflets, aliasing artifacts caused by Doppler velocity overshooting beyond Nyquist velocity, and low spatial and temporal resolutions. While the impact of clutter signal filtering on flow reconstruction is acknowledged, its specific effects were not investigated in this study. Our input data were already clutter-filtered (in vivo non scan-converted data from a GE scanner) or clutter-free (simulated data). Some researchers address clutter filtering in color flow imaging using deep learning (DL) techniques [34].

The clinical color Doppler data used for training and testing in this study contained only single aliasing, corrected by a DL-based unwrapping algorithm from our previous work [26]. It is important to note that this algorithm may face limitations in scenarios involving multiple aliasing, particularly in valvular disease. A potential solution could involve training a supervised DL model with multialiased data and their alias-free labels, such as in interferometric imaging [35] or color Doppler imaging of the femoral bifurcation [36].

The temporal resolution of clinical color Doppler, typically ranging from 10 to 15 frames/s, significantly restricts the use of temporal information. Consequently, the application of physical constraints is limited to those that are not time-dependent, such as mass conservation for an incompressible fluid. Around 25 harmonics are necessary to accurately record pressure time derivatives within the left ventricle with a 5% margin of error [37]. Hence, a color Doppler frame rate of 25 should be ideally sought to characterize intracardiac blood flow. Although we did not address such a strategy in this study, one approach would be to use Doppler information from two or three successive cardiac cycles. To overcome these limitations, advances in increasing the frame rate of color Doppler imaging, such as diverging scan sequences [38], [39], have been explored. Another promising avenue involves using multiline transmission [40], especially given the robust performance of our models on sparse Doppler data, notably nnU-Net. Higher frame rates offer the possibility of incorporating more complex physical constraints, including vorticity, Euler, or Navier–Stokes equations, potentially enhancing flow reconstruction accuracy. While this might pose challenges for the original *i*VFM due to the nonlinear terms in the equations, it aligns seamlessly with PINNs, requiring minimal changes to the loss function.

C. Future Directions

The successful application of PINNs in mapping intraventricular vector flow from color Doppler paves the way for future investigations. Upcoming research will prioritize the integration of high-frame-rate color Doppler with PINNs while incorporating the governing Navier–Stokes equations. This combination aims to leverage the temporal information to obtain a more accurate velocity field and the pressure gradient within the left ventricle.

Furthermore, with the development of our fully automated and robust tools, including left ventricular segmentation, dealiasing, and velocity field reconstruction using NNs, we anticipate extracting potential biomarkers from intracardiac vector blood flow for enhanced clinical insights and diagnostic capabilities.

VI. Conclusion

Our study presents novel approaches based on NNs for *i*VFM, using gradient-based optimization through PINNs (RB-PINNs and AL-PINNs) or physics-guided supervised learning (nnU-Net). These methods offer contrasting strategies to tackle the ill-posed inverse problem for vector flow reconstruction. Our results demonstrate the remarkable capabilities of both PINNs and nnU-Net in reconstructing intraventricular vector blood flow fields. Particularly noteworthy is nnU-Net's performance, demonstrating quasi-real-time capability, robustness on sparse Doppler data, and independence from explicit boundary conditions. These characteristics position nnU-Net as a promising solution for real-time clinical applications.

Acknowledgment

Olivier Bernard and Damien Garcia share the last authorship for this work. The authors thank François Tournoux (Montreal University—CHUM—CRCHUM) for providing access to the in vivo data used in this study.

REFERENCES

[1] F. Vixège et al., "Physics-constrained intraventricular vector flow mapping by color Doppler," *Phys. Med. Biol.*, vol. 66, no. 24, Dec. 2021, Art. no. 245019.

[2] K. C. Assi et al., "Intraventricular vector flow mapping—A Doppler-based regularized problem with automatic model selection," *Phys. Med. Biol.*, vol. 62, no. 17, pp. 7131–7147, Aug. 2017.

[3] R. Asami, T. Tanaka, K.-I. Kawabata, K. Hashiba, T. Okada, and T. Nishiyama, "Accuracy and limitations of vector flow mapping: Left ventricular phantom validation using stereo particle image velocimetory," *J. Echocardiography*, vol. 15, no. 2, pp. 57–66, Jun. 2017.

[4] B. A. Meyers, C. J. Goergen, P. Segers, and P. P. Vlachos, "Colour-Doppler echocardiography flow field velocity reconstruction using a streamfunction–vorticity formulation," *J. Roy. Soc. Interface*, vol. 17, no. 173, Dec. 2020, Art. no. 20200741.

[5] A. S. Daae, M. S. Wigen, S. Fadnes, L. Løvstakken, and A. Støylen, "Intraventricular vector flow imaging with blood speckle tracking in adults: Feasibility, normal physiology and mechanisms in healthy volunteers," *Ultrasound Med. Biol.*, vol. 47, no. 12, pp. 3501–3513, Dec. 2021.

[6] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.

[7] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, pp. 1–43, Apr. 2018.

[8] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for fluid mechanics: A review," *Acta Mechanica Sinica*, vol. 37, no. 12, pp. 1727–1738, Dec. 2021.

[9] A. Arzani, J.-X. Wang, and R. M. D'Souza, "Uncovering near-wall blood flow from sparse data with physics-informed neural networks," *Phys. Fluids*, vol. 33, no. 7, Jul. 2021, Art. no. 071905.

[10] G. Kissas, Y. Yang, E. Hwuang, W. R. Witschey, J. A. Detre, and P. Perdikaris, "Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks," *Comput. Methods Appl. Mech. Eng.*, vol. 358, Jan. 2020, Art. no. 112623.

[11] L. Wang, H. Wang, L. Liang, J. Li, Z. Zeng, and Y. Liu, "Physics-informed neural networks for transcranial ultrasound wave propagation," *Ultrasonics*, vol. 132, Jul. 2023, Art. no. 107026.

[12] Z. Yin, G.-Y. Li, Z. Zhang, Y. Zheng, and Y. Cao, "SWENet: A physics-informed deep neural network (PINN) for shear wave elastography," *IEEE Trans. Med. Imag.*, vol. 43, no. 4, pp. 1434–1448, Apr. 2024.

[13] H. Guan, J. Dong, and W.-N. Lee, "Towards real-time training of physics-informed neural networks: Applications in ultrafast ultrasound blood flow imaging," 2023, *arXiv:2309.04755*.

[14] S. A. Faroughi et al., "Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: Fluid and solid mechanics," *J. Comput. Inf. Sci. Eng.*, vol. 24, no. 4, Jan. 2024, Art. no. 040802.

[15] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, Feb. 2021.

[16] Z. Chen, V. Badrinarayanan, C. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *Proc. ICML*, 2018, pp. 794–803.

[17] A. Ali Heydari, C. A. Thompson, and A. Mehmood, "SoftAdapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions," 2019, *arXiv:1912.12355*.

[18] R. Bischof and M. Kraus, "Multi-objective loss balancing for physics-informed deep learning," 2021, *arXiv:2110.09813*.

[19] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, "Physics-informed neural networks with hard constraints for inverse design," *SIAM J. Sci. Comput.*, vol. 43, no. 6, pp. B1105–B1132, Jan. 2021.

[20] H. Son, S. W. Cho, and H. J. Hwang, "Enhanced physics-informed neural networks with augmented Lagrangian relaxation method (AL-PINNs)," *Neurocomputing*, vol. 548, Sep. 2023.

[21] J. Berg and K. Nyström, "A unified deep artificial neural network approach to partial differential equations in complex geometries," *Neurocomputing*, vol. 317, pp. 28–41, Nov. 2018.

[22] N. Sukumar and A. Srivastava, "Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks," *Comput. Methods Appl. Mech. Eng.*, vol. 389, Feb. 2022, Art. no. 114333.

[23] M. R. Hestenes, "Multiplier and gradient methods," *J. Optim. Theory Appl.*, vol. 4, no. 5, pp. 303–320, Nov. 1969.

[24] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.

[25] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, nos. 1–3, pp. 503–528, Aug. 1989.

[26] H. J. Ling, O. Bernard, N. Ducros, and D. Garcia, "Phase unwrapping of color Doppler echocardiography using deep learning," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 70, no. 8, pp. 810–820, Aug. 2023.

[27] C. Chnafa, S. Mendez, and F. Nicoud, "Image-based large-eddy simulation in a realistic left heart," *Comput. Fluids*, vol. 94, pp. 173–187, May 2014.

[28] C. Chnafa, S. Mendez, and F. Nicoud, "Image-based simulations show important flow fluctuations in a normal left ventricle: What could be the implications?" *Ann. Biomed. Eng.*, vol. 44, no. 11, pp. 3346–3358, Nov. 2016.

[29] R. Faludi et al., "Left ventricular flow patterns in healthy subjects and patients with prosthetic mitral valves: An in vivo study using echocardiographic particle image velocimetry," *J. Thoracic Cardiovascular Surg.*, vol. 139, no. 6, pp. 1501–1510, Jun. 2010.

[30] F. Mehregan et al., "Doppler vortography: A color Doppler approach to quantification of intraventricular blood flow vortices," *Ultrasound Med. Biol.*, vol. 40, no. 1, pp. 210–221, Jan. 2014.

[31] H. J. Ling, N. Painchaud, P.-Y. Courand, P.-M. Jodoin, D. Garcia, and O. Bernard, "Extraction of volumetric indices from echocardiography: Which deep learning solution for clinical use?" in *Proc. FIMH*, 2023, pp. 245–254.

[32] T. G. Grossmann, U. Julia Komorowska, J. Latz, and C.-B. Schönlieb, "Can physics-informed neural networks beat the finite element method?" 2023, *arXiv:2302.04107*.

[33] Y. Sun et al., "A pipeline for the generation of synthetic cardiac color Doppler," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 69, no. 3, pp. 932–941, Mar. 2022.

[34] H. Wang, S. Gao, M. Mozumi, M. Omura, R. Nagaoka, and H. Hasegawa, "Preliminary investigation on clutter filtering based on deep learning," *Jpn. J. Appl. Phys.*, vol. 60, May 2021.

[35] G. E. Spoorthi, R. K. S. S. Gorthi, and S. Gorthi, "PhaseNet 2.0: Phase unwrapping of noisy data based on deep learning approach," *IEEE Trans. Image Process.*, vol. 29, pp. 4862–4872, 2020.

[36] H. Nahas, J. S. Au, T. Ishii, B. Y. S. Yiu, A. J. Y. Chee, and A. C. H. Yu, "A deep learning approach to resolve aliasing artifacts in ultrasound color flow imaging," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 67, no. 12, pp. 2615–2628, Dec. 2020.

[37] L. J. Krovetz and S. D. Goldbloom, "Frequency content of intravascular and intracardiac pressures and their time derivatives," *IEEE Trans. Biomed. Eng.*, vol. BME-21, no. 6, pp. 498–501, Nov. 1974.

[38] D. Posada et al., "Staggered multiple-PRF ultrafast color Doppler," *IEEE Trans. Med. Imag.*, vol. 35, no. 6, pp. 1510–1521, Jun. 2016.

[39] A. Ramalli, A. Rodriguez-Molares, J. Avdal, J. D'hooge, and L. Løvstakken, "High-frame-rate color Doppler echocardiography: A quantitative comparison of different approaches," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 67, no. 5, pp. 923–933, May 2020.

[40] A. Ramalli et al., "Real-time high-frame-rate cardiac B-mode and tissue Doppler imaging based on multiline transmission and multiline acquisition," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 65, no. 11, pp. 2030–2041, Nov. 2018.