

Physics-Informed Neural Networks

Abstract

In this report, we provide insight into the application of Physics-Informed Neural Networks (PINNs) in biomedical contexts. Specifically, we present some motivations behind the implementation of PINNs, some detailed PINN frameworks in various healthcare applications, and their existing limitations, which need improvements in future work.

1 Motivations

Nowadays, with the rapid development of AI, AI-driven approaches have created immense opportunities to harness the proliferation of biomedical data to advance biomedical research and support clinical decision-making. In particular, Physics-Informed Neural Networks (PINNs) have emerged as a promising direction due to several advantages in biomedical settings compared to traditional computational models and even many other modern Machine Learning (ML) and Deep Learning (DL) counterparts. To name a few,

- Training ML and DL models to achieve reasonably good performance in some cases requires significant amounts of ground truth training data. However, in many biomedical settings, collecting such a large amount of subject-level information from a vast number of patients is very difficult, burdensome, or even infeasible (e.g., measuring blood pressure using a conventional cuff-based sphygmomanometer is very time-consuming on a large scale). Since PINNs can augment the standard supervised loss with an unsupervised loss constructed based on the underlying physical laws of biological systems, they thus reduce their reliance on labeled data. As a result, they have proven to be a potential candidate in biomedical applications with limited labeled data.
- PINNs have been demonstrated to provide a non-invasive, superior diagnostic tool for many diseases compared to conventional computational methods. For example,
 - One of the most widely used and effective methods for the early detection of breast cancer is mammography. However, this method requires exposing patients to ionizing radiation, which can be harmful with frequent exposure. O. Mukhmetov et al. [3] proposed implementing a PINN that leverages thermograms collected from non-invasive IR cameras—showing the temperature distribution on the surface of the breast—as well as 3D breast models obtained using a non-invasive 3D scanner to predict temperature distributions in breast tissues and thereby identify potential abnormal regions indicating the presence of tumors. This approach has been shown to be 12 times faster than the traditional Finite Element Analysis (FEA) in simulating thermal distributions.
 - Computed tomography angiography (CTA) is commonly used to provide geometrical features and branching patterns of the entire cerebrovascular architecture, playing a crucial role in the diagnosis and treatment of various cerebrovascular diseases. However, the high radiation dose and administration of exogenous contrast agents in CTA raise health

concerns. Meanwhile, Transcranial Doppler (TCD) ultrasound is a non-invasive tool for evaluating blood flow velocity within the cerebral arteries, but it is limited by its spatial coverage, as it can assess only a handful of locations across the cerebrovasculature. M. Sarabian et al. [4] proposed a PINN that utilizes data from TCD and MRI angiography to provide high-accuracy maps of velocity, area, and pressure in the entire brain vasculature, even in the absence of boundary conditions—a significant limitation of conventional purely physics-based computational models.

- PINNs outperform traditional mesh-based methods in solving inverse problems, thanks to automatic differentiation, especially in complex systems. By embedding physical laws into neural network training and treating unknown parameters as trainable, PINNs allow the discovery of their most optimal values that both fit the data and satisfy the underlying physical constraints.
 - Estimating Glioblastoma (GBM) infiltration from MRI is crucial for personalized radiotherapy. However, predicting tumor cell distribution requires patient-specific parameters, making it a challenging inverse problem. R. Zhang et al. [7] used PINNs to estimate these parameters from a single 3D MRI snapshot, achieving promising results.
 - Blood flow modeling, governed by the Navier–Stokes and Windkessel models, presents challenges for traditional mathematical methods, which struggle to solve these equations due to incomplete boundary conditions. To overcome this limitation, J. Garay et al. [8] leveraged PINNs to accurately estimate velocity, pressure fields, and Windkessel parameters from medical imaging data, despite the presence of missing information.

2 Physics-Informed Neural Networks (PINNs)

As stated in the previous section, typical machine learning and deep learning approaches rely solely on data and lack built-in physical constraints, often leading to inconsistent predictions, especially with scarce or noisy data. In contrast, mathematical models like PDEs and ODEs provide structured representations but may contain unknown parameters, be analytically intractable, or require costly numerical solutions.

Physics-Informed Neural Networks (PINNs) address these limitations by integrating physical laws directly into the learning process. The key idea is to embed the governing equations of a system—such as PDEs, ODEs, or other mathematical models—into the neural network’s architecture or loss function. This ensures that the model optimizes both data fidelity and physical consistency during training.

By enforcing the model constraints, PINNs effectively combine the strengths of physics-based modeling with data-driven learning, resulting in models that are not only more accurate and generalizable but also inherently interpretable, while reducing reliance on labeled data.

3 Methods

This section presents the implementation of PINNs in various biomedical applications.

3.1 Cuffless blood pressure estimation

3.1.1 Context

Tracking blood pressure frequently is crucial in some cases (e.g., for individuals with high blood pressure, to prevent heart attacks or strokes in a timely manner). However, measuring blood pressure using a conventional cuff-based sphygmomanometer on a frequent basis can be challenging and inconvenient. Meanwhile, wearable devices equipped with bioimpedance sensors are now widely available, providing a vast amount of bioimpedance data. Thus, a PINN is employed to extract useful cardiovascular information from these data.

3.1.2 Method

Bioimpedance signals are segmented based on the start and end of a cardiac cycle. Let $\vec{\mathbf{x}}_i = (\mathbf{x}_i^1, \dots, \mathbf{x}_i^N)$ be the N -dimensional time series bioimpedance data of the i -th cardiac cycle (N points are sampled for each segment). From these bioimpedance signals $\vec{\mathbf{x}}_i$, M physiological features $\vec{\mathbf{u}}_i = (\mathbf{u}_i^1, \dots, \mathbf{u}_i^M)$ are extracted. Specifically, in this application, three features are extracted:

- The level of the first amplitude drop in the bioimpedance waveform, which provides a proxy to an increase in blood volume.
- The inverse of the time difference between the first and second drop in bioimpedance amplitude, representing the blood pulse velocity.
- The time difference between the beginning and the end of the waveform, indicating the heart rate.

These $\vec{\mathbf{x}}_i$ and $\vec{\mathbf{u}}_i$ serve as inputs of a neural network f_{NN} to estimate the blood pressure $y_{\text{NN}}(\vec{\mathbf{x}}_i, \vec{\mathbf{u}}_i; \Theta) = f_{\text{NN}}(\vec{\mathbf{x}}_i, \vec{\mathbf{u}}_i; \Theta)$, where Θ denotes the trainable parameters of the neural network. For the construction of the unsupervised loss function, a first-order Taylor approximation of the function $f_{\text{NN}}(\cdot, \cdot; \Theta)$ around $(\vec{\mathbf{x}}_i, \vec{\mathbf{u}}_i)$ is then introduced:

$$P_i(\vec{\mathbf{x}}, \vec{\mathbf{u}}; \Theta) = f_{\text{NN}}(\vec{\mathbf{x}}_i, \vec{\mathbf{u}}_i; \Theta) + \sum_{k=1}^M \frac{\partial f_{\text{NN}}}{\partial \mathbf{u}^k}(\vec{\mathbf{x}}_i, \mathbf{u}^k - \mathbf{u}_i^k; \Theta)$$

where $\frac{\partial f_{\text{NN}}}{\partial \mathbf{u}^k}(\vec{\mathbf{x}}_i, \mathbf{u}^k - \mathbf{u}_i^k; \Theta)$ can be calculated using auto-differentiation.

Since the i -th and $(i + 1)$ -th segments represent consecutive beats, this sequential nature could be leveraged to compute the residual resulting from the difference between the neural network prediction and the Taylor polynomial P_i evaluated at the $(i + 1)$ -th segment:

$$h_i(\vec{\mathbf{x}}_{i+1}, \vec{\mathbf{u}}_{i+1}; \Theta) = f_{\text{NN}}(\vec{\mathbf{x}}_{i+1}, \vec{\mathbf{u}}_{i+1}; \Theta) - P_i(\vec{\mathbf{x}}_{i+1}, \vec{\mathbf{u}}_{i+1}; \Theta)$$

Hence, the loss function used to train the neural network is a combination of the conventional supervised loss and an unsupervised loss based on these residuals:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{conventional}} + \beta \mathcal{L}_{\text{physics}}$$

where

$$\mathcal{L}_{\text{conventional}} = \frac{1}{S} \sum_{i=1}^S (\mathbf{y}_{\text{NN}}(\vec{\mathbf{x}}_i, \vec{\mathbf{u}}_i; \Theta) - \mathbf{y}_{\text{true}}(\vec{\mathbf{x}}_i, \vec{\mathbf{u}}_i))^2$$

and

$$\mathcal{L}_{\text{physics}} = \frac{1}{(R-1)} \sum_{i=1}^{R-1} (h_i(\vec{\mathbf{x}}_{i+1}, \vec{\mathbf{u}}_{i+1}; \Theta))^2$$

In the above equations, S and R denote the number of labeled training samples and the total number of training samples, respectively. The weights assigned to each type of loss, α and β , are hyperparameters that must be tuned. A comprehensive visualization of the framework can be found in Figure 1.

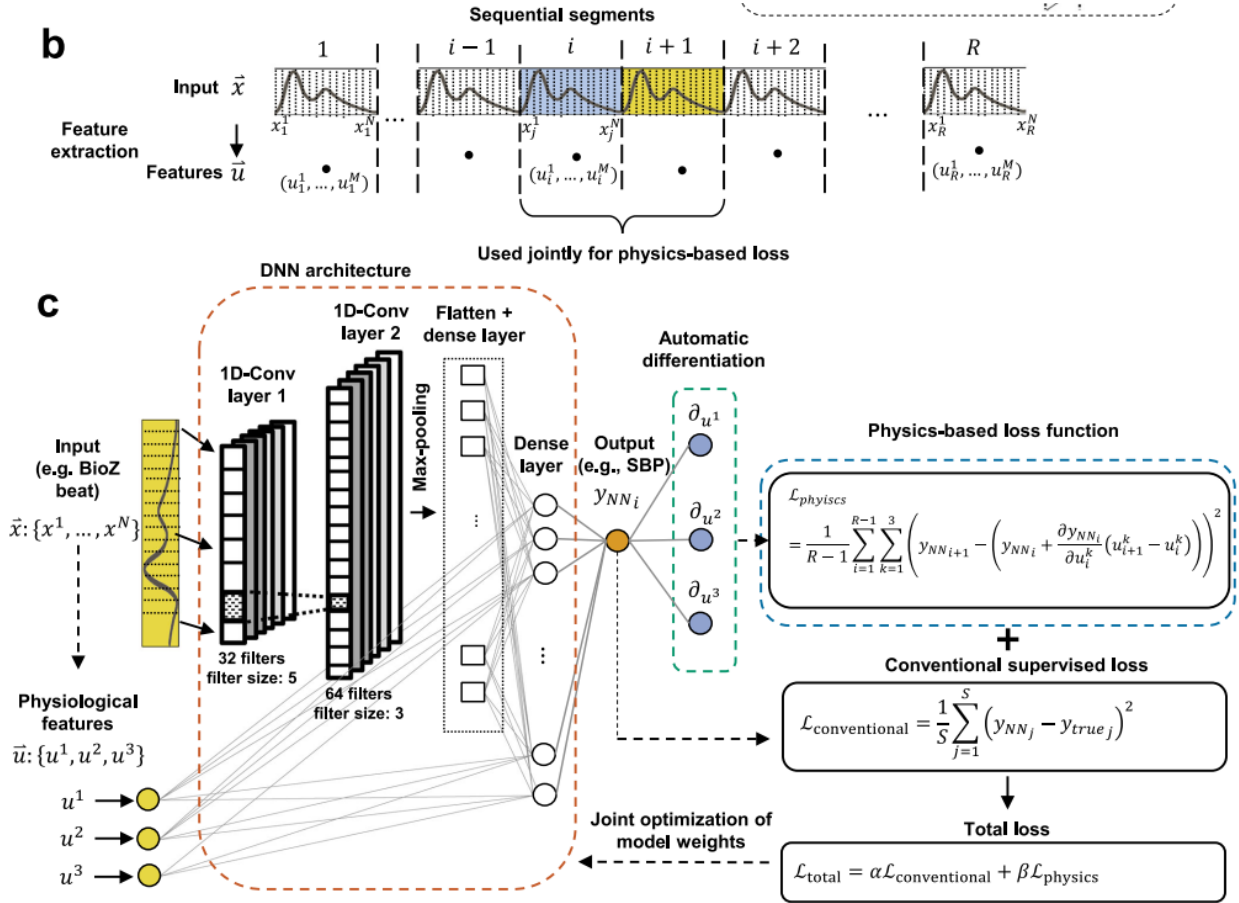


Figure 1: PINN framework in Section 3.1. Image take from the original paper [1]

3.2 Myocardial perfusion MRI quantification

3.2.1 Context

Dynamic contrast-enhanced magnetic resonance imaging (DCE-MRI) is becoming an increasingly popular technique for the non-invasive assessment of patients with suspected coronary artery disease (CAD). However, in some cases, images are difficult to interpret, and imaging might not fully capture the severity of CAD in patients with multi-vessel disease. Therefore, a PINN is implemented to quantify the myocardial blood flow using the data extracted from DCE-MRI and thus serves as a quantitative complement to the visual assessment.

3.2.2 Method

The basis for myocardial perfusion quantification is the use of a tracer-kinetic model to describe the physics of the transport process of a contrast agent across a myocardial tissue. In this application, a two-compartment exchange model (2CXM) has been proposed, in which the perfusion unit (represented by a single pixel in DCE-MRI) is modeled as a system of two interacting compartments, the plasma and the interstitium. The evolution of the contrast agent concentration over time in both compartments is then described by a pair of coupled ordinary differential equations (ODEs):

$$\nu_p \frac{dC_p(t)}{dt} = F_p(C_{\text{AIF}}(t) - C_p(t)) + PS(C_e(t) - C_p(t)) \quad (1)$$

$$\nu_e \frac{dC_e(t)}{dt} = PS(C_p(t) - C_e(t)) \quad (2)$$

Here, $C_p(t)$ and $C_e(t)$ are the concentrations of the agent in the plasma and interstitial space at time t , respectively. $C_{\text{AIF}}(t)$ represents the arterial input function, which is the concentration of the agent in the blood entering the tissue from the left ventricle. F_p is the plasma flow, while ν_p and ν_e are the fractional plasma and interstitial volumes, respectively. PS is the permeability-surface area product, which indicates how easily the agent moves between compartments. The concentration of the agent in the tissue, $C_{\text{myo}}(t)$, is a weighted sum of the concentrations within these separate compartments:

$$C_{\text{myo}}(t) = \nu_p C_p(t) + \nu_e C_e(t) \quad (3)$$

A neural network, f_θ , is then trained using the observed imaging data $C_p(t)$, $C_e(t)$, and $C_{\text{AIF}}(t)$ to infer the key myocardial parameters ν_p , ν_e , PS , and F_p . Specifically, the network takes time t as the input and produces the estimated arterial input function as well as the estimated concentrations of the agent in the plasma and interstitium for all K perfusion units at time t :

$$t \xrightarrow{f_\theta} \begin{bmatrix} C_p^1(t) \\ C_p^2(t) \\ \vdots \\ C_p^K(t) \end{bmatrix}, \begin{bmatrix} C_e^1(t) \\ C_e^2(t) \\ \vdots \\ C_e^K(t) \end{bmatrix}, C_{\text{AIF}}(t)$$

Here, the myocardial parameters $\eta^j = (\nu_p^j, \nu_e^j, PS^j, F_p^j)$ ($1 \leq j \leq K$) are set as trainable so that, after training, their optimal values, which best fit the observed data while satisfying the physical laws governing the system (as described in Eqs. 1, 2, and 3), can be obtained.

The loss function for training the neural network is defined as follows:

$$\mathcal{L} = \frac{1}{K} \sum_{j=1}^K \left(w_r \mathcal{L}_r^j + w_C \mathcal{L}_C^j + w_b \mathcal{L}_b^j + w_{\text{reg}} \mathcal{L}_{\text{reg}}^j \right)$$

where the hyperparameters w_C , w_r , w_b , and w_{reg} are the weightings for each term in the combined loss, K is the total number of pixels, and the exact form of each term is detailed as follows:

$$\mathcal{L}_C^j = \frac{1}{N_C} \sum_{i=1}^{N_C} \left(C_{\text{myo}}^j(t_i) - \hat{C}_{\text{myo}}^j(t_i; v_p^j, v_e^j, \theta) \right)^2 + \frac{1}{N_C} \sum_{i=1}^{N_C} \left(C_{\text{AIF}}^j(t_i) - \hat{C}_{\text{AIF}}^j(t_i; \theta) \right)^2$$

is the supervised loss, capturing the differences between the predicted and the observed concentrations.

$$\mathcal{L}_r^j = \frac{1}{N_r} \sum_{i=1}^{N_r} \left(r_p(t_i; \eta^j, \theta) \right)^2 + \frac{1}{N_r} \sum_{i=1}^{N_r} \left(r_e(t_i; \eta^j, \theta) \right)^2$$

is the unsupervised loss, constructed from the residuals corresponding to the 2CXM described in Eqs. 1 and 2. The differentials in these equations can be computed via auto-differentiation.

$$\mathcal{L}_b^j = \hat{C}_p^j(0; \theta)^2 + \hat{C}_e^j(0; \theta)^2 + \hat{C}_{\text{myo}}^j(0; v_p^j, v_e^j, \theta)^2 + \hat{C}_{\text{AIF}}^j(0; \theta)^2$$

is the initial conditions which enforces that there is no contrast in the system at time $t = 0$.

$$\mathcal{L}_{\text{reg}}^j = \frac{1}{N_r} \sum_{i=1}^{N_r} \left(\min \left(\hat{C}_p^j(t_i; \theta), 0 \right) \right)^2 + \frac{1}{N_r} \sum_{i=1}^{N_r} \left(\min \left(\hat{C}_e^j(t_i; \theta), 0 \right) \right)^2$$

is a regularization term to enforce non-negative concentration values.

In the above equations, N_C denotes the total number of contrast agent concentration measurements derived from the DCE-MRI. N_r is the number of randomly chosen time-domain points for which the residuals of the ODEs are calculated. $\hat{C}_p^j(t_i; \theta)$, $\hat{C}_e^j(t_i; \theta)$, $\hat{C}_{\text{myo}}^j(t_i; v_p^j, v_e^j, \theta)$, and $\hat{C}_{\text{AIF}}^j(t_i; \theta)$ are the PINN-predicted plasma, interstitial, myocardial concentrations, and AIF at time t_i . A comprehensive visualization of the framework can be found in Figure 2.

3.3 COVID-19 transmission dynamics analysis

3.3.1 Context

Modeling and predicting the behavior of infectious diseases is crucial for early warning systems and effective public health responses. Compartmental models, which formulate transmission dynamics as a system of ordinary differential equations (ODEs), are widely used for this purpose. Solving these models using traditional methods presents significant challenges due to the complexity of viral spread, the presence noisy and limited data, and the fact that many parameters in compartmental models are often unknown and can vary over time in real-world scenarios, making them difficult to estimate. To address this, X. Ning et al. [5] proposed treating these parameters as time-dependent and made use of PINNs to estimate their temporal variations from observed data.

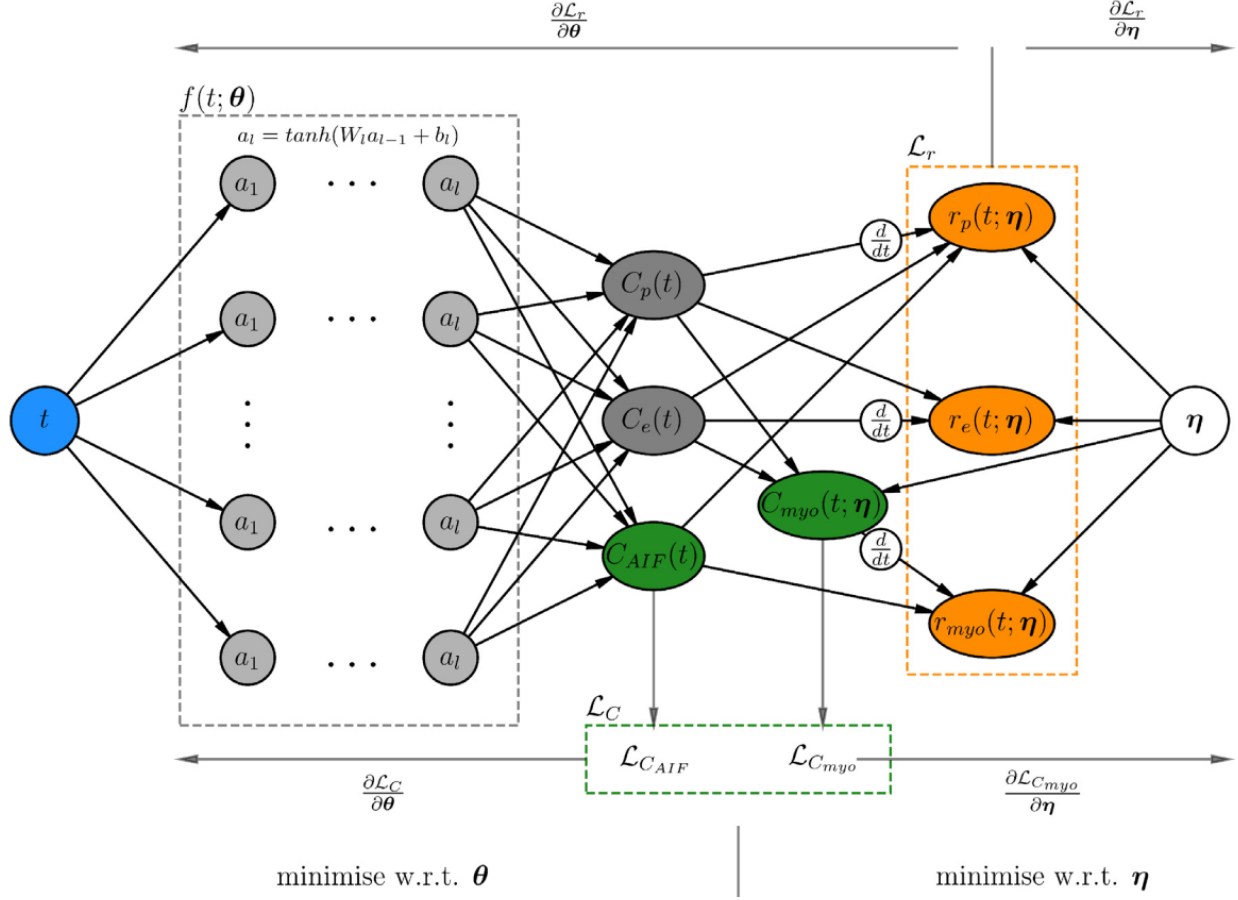


Figure 2: PINN framework in Section 3.2. Image take from the original paper [2]

3.3.2 Method

The authors in [5] proposed using the SEIRD compartment model to describe the transmission dynamics of COVID-19. This model divides the total population into five compartments: Susceptible (people vulnerable to infection), Exposed (latent or asymptomatic infectives), Infectious, Recovered, and Death. The evolution of these compartments is governed by the following system of ordinary differential equations (ODEs):

$$\begin{cases}
 \frac{dS(t)}{dt} = -\frac{\beta(t)S(t)(\epsilon E(t)+I(t))}{N} \\
 \frac{dE(t)}{dt} = \frac{\beta(t)S(t)(\epsilon E(t)+I(t))}{N} - \frac{E(t)}{\alpha} \\
 \frac{dI(t)}{dt} = \frac{E(t)}{\alpha} - \gamma(t)I(t) - \mu(t)I(t) \\
 \frac{dR(t)}{dt} = \gamma(t)I(t) \\
 \frac{dD(t)}{dt} = \mu(t)I(t) \\
 N = S(t) + E(t) + I(t) + R(t) + D(t) \\
 S(0) = S_0, E(0) = E_0, I(0) = I_0, R(0) = R_0, D(0) = D_0
 \end{cases} \quad (4)$$

where N is the total population, and $S(t)$, $E(t)$, $I(t)$, $R(t)$, and $D(t)$ represent the number of individuals in each compartment at time t . As stated, the transmission rate $\beta(t)$, recovery rate $\gamma(t)$, and death rate $\mu(t)$ are treated as time-dependent functions, reflecting the effects of interventions by authorities and/or virus mutations. The rate of progression from exposed to infectious, α , and the parameter ϵ (introduced to account for differences in transmission capacity between exposed and onset populations) are constants, with α chosen as $21 \cdot \tanh(x)$ and ϵ chosen as $\tanh(x)$, where x is a random sample drawn from the uniform distribution $\mathcal{U}_{[0,3]}$.

Given data on susceptible, exposed, infectious, recovered, and deceased cases, the aim is to train a PINN to predict the future trends of the disease. In addition to forecasting, the model must also estimate time-dependent parameters to better understand disease transmission dynamics and assess the effectiveness of interventions. Specifically, the PINN architecture consists of two separate deep neural networks (Figure 3):

1. The first network learns the state $U_{NN}(t) = (S_{NN}(t), E_{NN}(t), I_{NN}(t), R_{NN}(t), D_{NN}(t))$ of the five SEIRD compartments. Using automatic differentiation to compute the derivatives of the network's output with respect to its input gives rise to the residuals, which are used in the unsupervised loss function defined later.
2. The second network learns the time-variant parameters $\beta(t)$, $\gamma(t)$, and $\mu(t)$.

The loss function used to train the model is defined as follows:

$$\mathcal{L} = \omega_u MSE_u + \omega_r MSE_r$$

where ω_u and ω_r are hyperparameters representing the weights assigned to each loss term and

$$MSE_u = \frac{1}{M} \sum_{j=1}^M \|U_{NN}(t_j) - U(t_j)\|^2$$

$$MSE_r = \frac{1}{M} \sum_{j=1}^M \|R_{NN}(t_j)\|^2$$

correspond to the conventional supervised loss and the unsupervised loss, respectively, with the latter arising from the governing laws of the system described in 4. In the above equation, M denotes the total number of training samples, $U(t_j)$ represents the observed data, and $R_{NN}(t_j)$ is the residuals obtained from the five ODEs in 4.

3.4 Tumor growth forecasting

3.4.1 Context

Accurate tumor progression prediction is crucial for prognosis and treatment decisions. Nowadays, advances in bioimaging have improved the clinical outcomes of patients with cancer, but diagnosis and prognosis still rely heavily on physician expertise. Therefore, there is a pressing need to explore computational methods to reduce that reliance. While traditional mathematical methods only require limited imaging data to estimate tumor growth, they struggle with noisy and incomplete data—a common issue in medical imaging. On the other hand, AI and computer-aided diagnosis

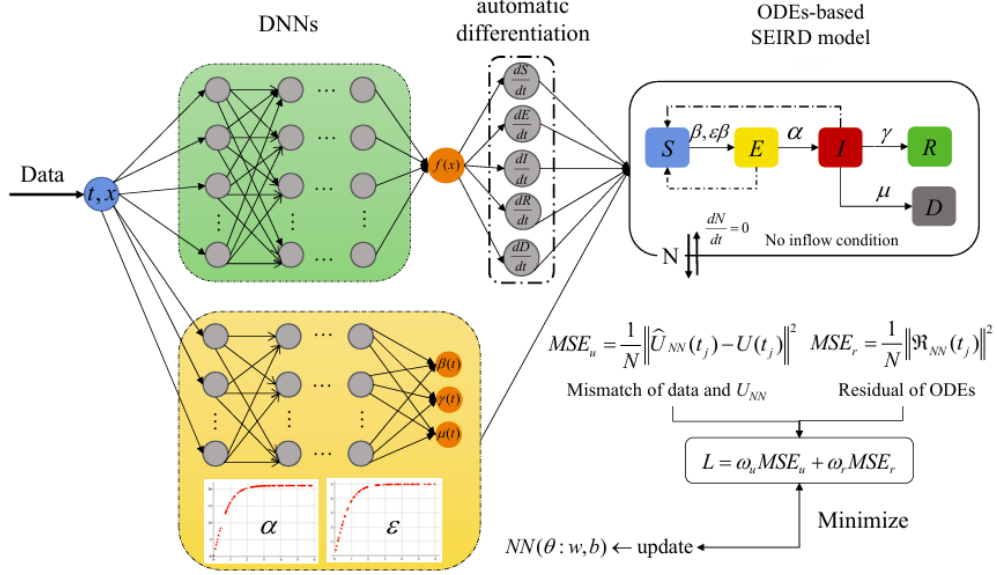


Figure 3: PINN framework in Section 3.3. Image take from the original paper [5]

systems can extract critical features from biomedical images that are invisible to the naked eye while reducing observer bias but require large datasets from longitudinal studies, which are often unavailable. To address this, TGM-Nets, a deep learning model inspired by the PINN framework, was proposed to integrate mathematical tumor growth models into training, enhancing prediction accuracy with limited bioimaging data.

3.4.2 Method

The mathematical model for Tumor Growth Models (TGMs) consists of two coupled PDE equations: a non-conservative Allen–Cahn equation, which uses a parameter ϕ to characterize cellular microstructure, and a diffusion-reaction equation that governs nutrient concentration σ . The tumor growth dynamics are described by the following phase-field equation, with details available in [9], **Section 2.2.1** and **Section 2.2.3**:

$$\begin{aligned}
 \frac{\partial \phi}{\partial t} &= \lambda \nabla^2 \phi - 2\phi(1 - \phi)f(\phi, \sigma) \\
 \frac{\partial \sigma}{\partial t} &= \eta \nabla^2 \sigma + S_h(1 - \phi) + S_c\phi - (\gamma_h(1 - \phi) + \gamma_c\phi)\sigma \\
 f(\phi, \sigma) &= M(1 - 2\phi - 3m(\sigma)) \\
 m(\sigma) &= m_{ref} \left(\frac{\rho + A}{2} + \frac{\rho - A}{\pi} \arctan \left(\frac{\sigma - \sigma_l}{\sigma_r} \right) \right)
 \end{aligned} \tag{5}$$

where

- $\phi(x, t)$ represents the density of tumor growth and $\sigma(x, t)$ represents the concentration of nutrient fields.

- λ is the diffusivity of the tumor phase-field and η is the diffusion constant of the nutrient.
- S_h and S_c are the nutrient supply in healthy tissue and tumor cells, respectively.
- γ_h and γ_c are the nutrient uptake by healthy tissue and tumor cells, respectively.
- $2\phi(1 - \phi)f(\phi, \sigma)$ is the derivative of the double-well potential $\Psi(\phi) = M\phi^2(1 - \phi)^2 - Mm(\sigma)\phi^2(3 - 2\phi)$ and $m(\sigma)$ is a tilting function in the phase-field literature, which can be interpreted as a function describing the tumor net proliferation rate.

Given temporal (t) and spatial coordinates (x, y) of training points as inputs, the aim to train a model to predict ϕ and σ , and accurately estimate some parameters of the PDEs. Specifically, the model needs to estimate two important parameters of the TGM, ρ and A , which represent the tumor cell division (proliferation) rate and the programmed cell death (apoptosis) rate, respectively. These two parameters will thus be treated as learnable parameters, while all other parameters will be preselected constants. The TGM-Nets approach employs three key techniques: Fourier transform for feature extraction, sequential learning to capture phase-field morphological changes over time, and a two-stage training process with physics-based fine-tuning for improved extrapolation and prediction accuracy.

Random Fourier feature mappings were employed for both spatial and temporal inputs. Let k_s and k_t denote the hyperparameters controlling the variance of these mappings for spatial and temporal features, respectively. Weight matrices B^s and B^t were constructed, with elements drawn from normal distributions $\mathcal{N}(0, k_s^2)$ and $\mathcal{N}(0, k_t^2)$, respectively. These matrices are fixed throughout the training process. The spatial and temporal Fourier features are then computed as:

$$\gamma^s(x) = \begin{pmatrix} \cos(2\pi B^s x) \\ \sin(2\pi B^s x) \end{pmatrix}, \quad \gamma^t(t) = \begin{pmatrix} \cos(2\pi B^t t) \\ \sin(2\pi B^t t) \end{pmatrix}$$

$\gamma^s(x)$ and $\gamma^t(t)$ are then fed into the forward pass of the network to output $\hat{\phi}$ and $\hat{\sigma}$.

Similar to the usual PINN approach, the overall loss is obtained by combining the PDE loss and the data loss:

$$Loss = \omega_{PDE} Loss_{PDE} + \omega_{data} Loss_{data},$$

ω_{PDE} and ω_{data} are hyperparameters representing the weight coefficients for loss term.

The PDE loss is defined as:

$$Loss_{PDE} = \frac{1}{N_{PDE}} \sum_{i=1}^{N_{PDE}} (e_{1i}^2 + e_{2i}^2)$$

where N_{PDE} denotes the number of collocation points, and e_{1i} and e_{2i} are the residuals resulting from the PDEs in 5. The differentials in these PDEs can be calculated using auto-differentiation.

The data loss is defined as:

$$Loss_{data} = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} [(\hat{\phi}_i - \phi_i)^2 + (\hat{\sigma}_i - \sigma_i)^2]$$

where N_{data} denotes the number of labeled data points, and $\hat{\phi}_i$, ϕ_i , $\hat{\sigma}_i$, and σ_i are the predicted and actual values of ϕ and σ , respectively.

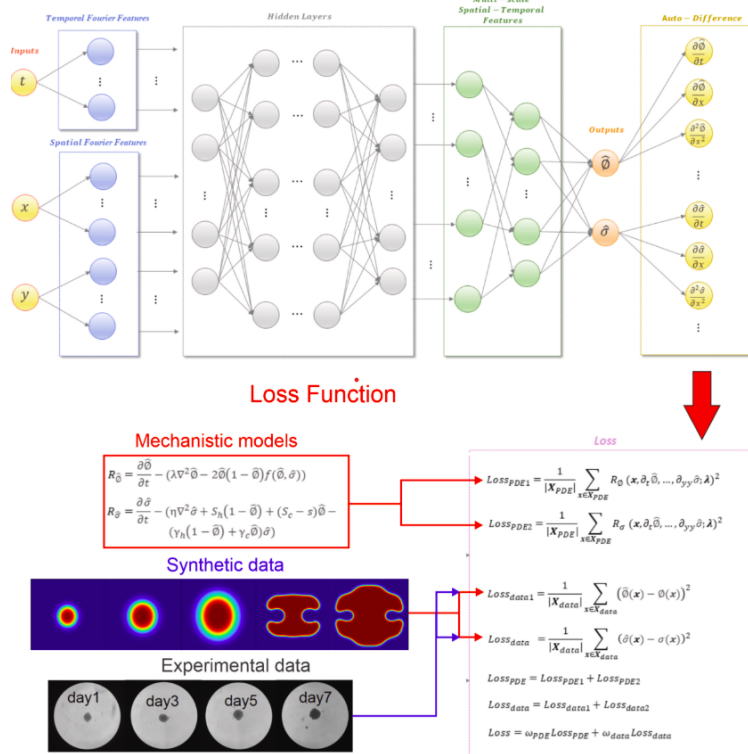


Figure 4: PINN framework in Section 3.4. Image take from the original paper [6]

4 Limitations

Besides their advantages, PINNs also have some limitations that require improvements in future work.

- In some cases, the optimized parameters of the trained PINNs are subject-specific due to personalized biological characteristics. For example, in 3.1, the model still requires calibration points (the start and end blood pressure in a cardiac cycle, as well as the minimum and maximum blood pressure) to be provided by end-users in order to perform well on their bioimpedance data. Hence, this poses a challenge for the practical deployment of the method, as identifying the minimum and maximum points is not feasible in a real-world scenario.
- In some applications, training PINNs requires dedicated hyperparameter tuning, as in the quantification of myocardial perfusion MRI 3.2, which can lead to a computational burden.
- PINNs still require certain assumption on the model, meaning that if underlying laws are not well understood or if the available data is not consistent with the assumption, the model may not perform well. We can take for example the assumption of Gaussian distribution in 3.4.

References

- [1] K. Sel, A. Mohammadi, R.I. Pettigrew, R. Jafari, *Physics-informed neural networks for modeling physiological time series for cuffless blood pressure estimation*, **npj Digit. Med.** **6**, 2023. <https://doi.org/10.1038/s41746-023-00853-4>
- [2] R.L.M. van Herten, A. Chiribiri, M. Breeuwer, M. Veta, C.M. Scannell, *Physics-informed neural networks for myocardial perfusion MRI quantification*, **Medical Image Analysis** **8**, 2022. <https://doi.org/10.1016/j.media.2022.102399>
- [3] O. Mukhmetov, Y. Zhao, A. Mashekova, V. Zarikas, E.Y.K. Ng, N. Aidossov, *Physics-informed neural network for fast prediction of temperature distributions in cancerous breasts as a potential efficient portable AI-based diagnostic tool*, **Computer Methods and Programs in Biomedicine** **242**, 2023. <https://doi.org/10.1016/j.cmpb.2023.107834>
- [4] M. Sarabian, H. Babaei, K. Laksari, *Physics-informed neural networks for brain hemodynamic predictions using medical imaging*, **IEEE Transactions on Medical Imaging** **41**, 2022. <https://doi.org/10.1109/TMI.2022.3161653>
- [5] X. Ning, J. Guan, X.A. Li, Y. Wei, F. Chen, *Physics-Informed Neural Networks Integrating Compartmental Model for Analyzing COVID-19 Transmission Dynamics*, **Viruses** **15**, 2023. <https://doi.org/10.3390/v15081749>
- [6] Q. Chen, Q. Ye, W. Zhang, H. Lie, X. Zheng, *TGM-Nets: A deep learning framework for enhanced forecasting of tumor growth by integrating imaging and modeling*, **Engineering Applications of Artificial Intelligence** **126**, 2023. <https://doi.org/10.1016/j.engappai.2023.106867>
- [7] R.Z. Zhang, I. Ezhov, M. Balcerak, A. Zhu, B. Wiestler, B. Menze, J. S. Lowengrub, *Personalized predictions of Glioblastoma infiltration: Mathematical models, Physics-Informed Neural Networks and multimodal scans*, **Medical Image Analysis** **101**, 2025. <https://doi.org/10.1016/j.media.2024.103423>
- [8] J. Garay, J. Dunstand, S. Uribe, F.S. Costabal, *Physics-informed neural networks for parameter estimation in bloodflow models*, **Computers in Biology and Medicine** **178**, 2024. <https://doi.org/10.1016/j.combiomed.2024.108706>
- [9] P. Colli, H. Gomez, G. Lorenzo, G. Marinoschi, A. Reali, and E. Rocca, *Mathematical analysis and simulation study of a phase-field model of prostate cancer growth with chemotherapy and antiangiogenic therapy effects*, **Mathematical Models and Methods in Applied Sciences** **30**, 2020. <https://doi.org/10.1142/S0218202520500220>