2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 979-8-3503-8622-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/BIBM62325.2024.10822632

Inverse Problem Antidote (IPA): Modeling of Systems Biology Model with Invertible Neural Networks

Linlin Li*
Weldon School of Biomedical Engineering
Purdue University
West Lafayette, IN, USA
lilinlin@purdue.edu

Shenyu Lu*
Elmore Family School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
lu876@purdue.edu

David M. Umulis**
Weldon School of Biomedical Engineering
Purdue University
West Lafayette, IN, USA
dumulis@purdue.edu

Xiaoqian Wang**
Elmore Family School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
joywang@purdue.edu

***Abstract*—In computational biology, accurately modeling biological systems is essential for understanding the underlying mechanisms of biological processes. One of the primary challenges in modeling lies in the inability to measure certain biological parameters directly. The task of identifying the parameters is known as the inverse problem and it entails estimating these unobservable parameters from available data. Existing modeling methods primarily focus on single-direction prediction. These methods include traditional Partial Differential Equation (PDE) modeling and machine learning approaches where given parameters are provided to predict the output and further compare the output with experimental evidence. However, these unidirectional methods struggle to effectively apply the experimental evidence directly to address the inverse problem. We contend that a single biological process should be modeled bidirectionally to simultaneously address the inverse problem. To this end, we propose leveraging the capabilities of the invertible neural network (INN) to establish a connection between the input parameter space and the model output space. We meticulously designed a bidirectional training technique that effectively applies real-world experimental data in guiding the INN in modeling the biological process. We tested our approach on a PDE-based Bone Morphogenic Protein (BMP) signaling network system in the zebrafish embryo and found the bidirectional modeling approach significantly enhances the alignment between simulation and experimental data. This method achieves a 94.65% reduction in Root Mean Square Error (RMSE) compared to the single-direction model when reconstructing experimental data using simulation with INN-identified parameters. Moreover, our method is rapid to implement, facilitating the precise identification of parameter ranges from experimental data. It makes parameter optimization feasible and provides a guideline for determining simulation parameter ranges, bypassing the traditional and laborious trial-and-error method.**

***Index Terms*—Invertible modeling, PDE, Systems Biology Model**

This work is based upon efforts supported by the EMBRIO Institute, contract #2120200, a National Science Foundation (NSF) Biology Integration Institute.

*Author Linlin Li and Shenyu Lu are equal contributors to this work and designated as co-first authors.

**Author David M. Umulis and Xiaoqian Wang are equal contributors to this work and designated as co-corresponding authors.

I. Introduction

Computational modeling has been widely used in studying biological systems through mathematics, physics, and computer science to improve the understanding of complex systems. Computational models have been applied at various levels of biological organization, from gene regulation networks and molecular interactions to tissue and organ-level changes. Depending on the specific system, computational modeling studies typically involve using mathematical equations to predict and complement experimental observations, aiding in the understanding of system complexity. Additionally, computational models can conduct existing experimental data to facilitate the development of new experiments or identify the most important variables to investigate [1].

When analyzing complex biological processes and constructing the mathematical model, a major challenge is the inability to directly measure the parameters of certain processes and accurately determine the model parameters [2]. Calibrating the parameters for these models to experimental evidence is crucial to ensure that they accurately represent biological phenomena and nonlinear relationships. The common process is to perform parameter optimization in biophysical parameter space to find the best-fit parameter for the experimental data. However, this process is usually computationally expensive due to the large parameter space or system complexity [3], [4]. For instance, a partial differential equation (PDE) system requires computation across both spatial and temporal domains.

Machine learning-based artificial intelligence (AI) methods have been increasingly valuable tools in parameter optimization for biological computational models in overcoming the challenge posed by the complexity and high dimensionality of the biological system [5]. Previously, various techniques have been applied to perform parameter optimization in biological computation models. Artificial Neural Networks (ANNs) models, particularly deep learning models advanced in managing high-dimension data are used for parameter estimation in cognitive models and systems biology models [6], [7]. Physics-Informed Neural Networks (PINNs) integrate the physical laws governing biological systems into the training process. By embedding differential equations directly into the neural network's loss function, PINNs can infer hidden dynamics and unknown parameters, offering robustness to noise and scattered data [5]. Instead of predicting or identifying the direct mapping between input parameter space and model output, Bayesian Methods integrate with a comprehensive framework for uncertainty quantification, using Markov Chain Monte Carlo (MCMC) sampling to fully characterize posterior parameter distributions [3]. This method is particularly useful when addressing challenges related to non-identifiable parameters.

Additionally, experimental data are crucial for calibrating and validating the model. However, most of the existing Machine Learning (ML) optimization methods struggle to effectively integrate both simulation results and experimental evidence through active search, often relying instead on bulk screening approaches. For instance, the ANNs model usually relies on training the model to achieve the one-directional mapping between parameter space and the simulation results without knowledge of the initial experimental input or output. Concentrating exclusively on a single direction can lead to overfitting in that specific direction, resulting in suboptimal performance for bidirectional prediction tasks. After obtaining the well-trained model, there is still a requirement for additional post-processing of parameter analysis and result evaluation against the experimental output, therefore, extensive parameter screening is still necessary. In this paper, our goal is to leverage advanced ML techniques to train a bi-directional model that can map both parameter space to simulation results and vice versa. By directly incorporating experimental data, we aim to enable an active search for identifying the most sensitive and best-fit parameters within the model.

Building upon the bidirectional mapping in the biological model, we propose utilizing an Invertible Neural Network (INN) to accurately represent this process. Specifically, the INN establishes a bidirectional mapping between the input and output spaces, as depicted in Figure 2. We designed a forward prediction model that significantly accelerates the PDE solving process, achieving the computation of approximately 6000 samples per second—this means a 6000-time faster compared to using traditional PDE solvers, which process only one sample per second. For backward predictions, we introduced a bidirectional training objective that enables the tracing of parameters used for the simulation based on the experiment curves, a functionality not available in traditional PDE models. The bidirectional approach utilizes a single architecture that faithfully models the biological process.

To test our approach, we adapted a systems biology model that studies the Bone Morphogenetic Protein (BMP) patterning in early zebrafish embryos. In zebrafish, Bone Morphogenetic Proteins (BMPs), part of the TGF-β super-family, regulate gene expression along the dorsal-ventral (DV) axis. During early embryonic development, BMP signaling shapes the DV axis in both invertebrates and vertebrates. BMP inhibitors like Chordin, Noggin, and others bind BMP ligands to prevent receptor interaction. Chordin can be cleaved by metalloproteases like Tolloid, releasing BMP ligands to activate receptors. BMP signaling involves forming receptor complexes, leading to the phosphorylation of Smad proteins, which then regulate gene expression in the nucleus. Our model uses multiple PDEs describing the process of protein molecular level of diffusion and reactions of BMP signaling regulation crossing the margin region of the zebrafish embryo [7], [8]. The governing equation of the BMP regulation network has been listed in Figure 1D.

Our results indicate that our proposed bidirectional method is significantly more robust than existing single-directional methods. Additionally, our method demonstrates superior performance in aligning simulation outcomes with experimental data. By harnessing experimental data to accurately trace simulation parameters, our approach secures a 94.65% reduction in RMSE when reconstructing and comparing simulation outputs to experimental observations, outperforming conventional Multi-Layer Perceptron (MLP) with single-directional capabilities.

The remainder of the paper is organized as follows. In Section II, we introduce the technical details of our approach in terms of problem setting and objective function, along with adapting to a systems biology model that studies the Bone Morphogenetic Protein (BMP) patterning in early zebrafish embryos. In Section III, we provide extensive results on both simulation and experiment data to validate our approach. We further illustrate the explanation results from our approach. Then in Section IV, we conclude our work.

II. Materials and Methods

A. Training Data

As described in the previous section, our PDE system simulates the BMP signaling network in the marginal region of the zebrafish embryo. The simulation domain is assumed to be half of the circumference of the marginal region (700 μm), as we consider the BMP pattern to be symmetric along the lateral direction. The BMP expression region spans 350 μm ventrally, while the Chordin(Chd) expression region covers 150 μm dorsally, based on previous studies. The governing equations involve 12 parameters that require calibration: D_B (BMP diffusion rate), D_C (Chd diffusion rate), D_{BC} (BMP-Chd complex diffusion rate), k_1 (forward reaction rate for BMP and Chd), k_{-1} (backward reaction rate for BMP and Chd), K_B (BMP decay rate), K_C (Chd decay rate), K_{BC} (BMP-Chd complex decay rate), λ_C (Tld processing rate of

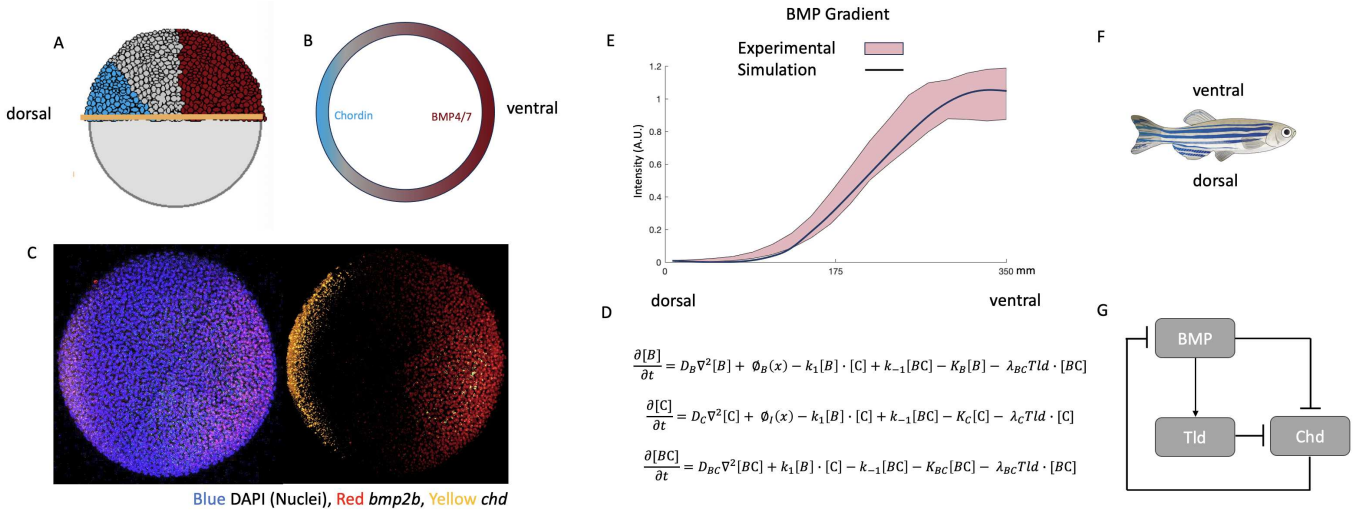


Fig. 1: Computational model of BMP patterning formation in zebrafish embryo. A, Zebrafish embryo illustration with BMP(Red) and Chordin(Blue) expression region. Figure adapted with permission from [9]. B, Cross section of marginal region. C. mRNA whole mount expression imaged by confocal microscope D. Governing equation of BMP regulatory network. [B]: BMP, [C]:Chordin, [BC]:BMP-Chordin complex, Tld: Tolliod. E. The shaded region indicates the downstream pSmad5 relative intensity profile of BMP signaling on the marginal region. The solid line is an example of a good fit simulation results. F. Figure of adult Zebrafish. G. Model illustration of BMP signaling regulatory network.

Chd), λ_{BC} (Tld processing rate of BMP-Chd complex), j_1 (BMP production rate), j_2 (Chd production rate). Assuming that the forward and backward reaction rates for B and Chd are identical, we have 11 unknown parameters to screen. The output domain is defined by 36 nodes along the 700 µm region, making the input size for our neural network system 11 and the output size 36.

B. Problem formulation

Notations. Denote $\mathbf{y} \in \mathbb{R}^d$ as the 11 input unknown parameters. The order in which the parameters follow is k_1, K_B, K_C, K_{BC}, D_B, D_C, D_{BC}, λ_C, λ_{BC}, j_1, j_2, which serve as inputs for the simulation (input $d = 11$ in our NN experiments). $\mathbf{x} \in \mathbb{R}^m$ represents the BMP concentration profile in 36 nodes over the embryo margin, which is the result obtained from the PDE solver (output $m = 36$ in our INN experiments). We define an convertible function $\mathbf{f}_\theta$ parameterized by θ such that $\mathbf{f}_\theta : \mathbf{y} \leftrightarrow \mathbf{x}$. For clarity, we denote the forward process as $\mathbf{f}_{fw} : \mathbf{y} \to \mathbf{x}$ and the backward process as $\mathbf{f}_{bw} : \mathbf{x} \to \mathbf{y}$. It also satisfies the relationship $\mathbf{f}_{fw} = \mathbf{f}_{bw}^{-1}$.

Our learning object is to train a function $\mathbf{f}$ using paired training data. Given the bijective nature of the mapping between inputs and outputs, it necessitates that $d = m$. However, this condition conflicts with our data structure where $d < m$. To address this, we augment the dimensionality of the input space by introducing a noise vector $noise \in \mathbb{R}^{m-d}$. The complete input is then the concatenation of the original input and the noise vector, expressed as $[\mathbf{y}, \text{noise}]$. Consequently, the output $\mathbf{x}$ is defined by $\mathbf{x} = \mathbf{f}_{\text{fw}}([\mathbf{y}, \text{noise}])$.

C. Invertible neural network architecture

In this study, we employ an invertible neural network (INN) to model the invertible function $\mathbf{f}$. Specifically, we utilize RealNVP [10] for implementing the INN.

Forward computation. For an arbitrary m dimensional input $\mathbf{z}$. We split $\mathbf{x}$ into two parts: $\mathbf{z}_{1:n}$ and $\mathbf{z}_{n+1:m}$, where $\mathbf{z}_{1:n}$ is the first n dimensions and $\mathbf{z}_{n+1:m}$ is the remaining $m - n$ dimensions. The INN consists of a series of coupling layers. A coupling layer can be computed as:

$$\begin{aligned} \mathbf{r}_{1:n} &= \mathbf{z}_{1:n} \\ \mathbf{r}_{n+1:m} &= \mathbf{z}_{n+1:m} \odot \exp(s(\mathbf{z}_{1:n})) + t(\mathbf{z}_{1:n}) \end{aligned} \tag{1}$$

where $s(\cdot)$ and $t(\cdot)$ are nonlinear neural networks that output scale and translation parameters.

A single coupling layer is not sufficient to transform an input to a desired latent representation, stacking them can be expressive. However, just stacking would always leave $\mathbf{z}_{1:d}$ unchanged. To resolve this, INN permutes the dimensions after each coupling layer, ensuring all dimensions get transformed as the data passes through multiple coupling layers.

Inverse computation. For a coupling layer of an INN, the inverse computation is

$$\begin{aligned} \mathbf{z}_{1:n} &= \mathbf{r}_{1:n} \\ \mathbf{z}_{n+1:m} &= (\mathbf{r}_{n+1:m} - t(\mathbf{r}_{1:n})) \odot \exp(-s(\mathbf{r}_{1:n})) \end{aligned} \tag{2}$$

D. Bi-directional training

We now introduce our approach for learning the function $\mathbf{f}$, which is designed to model the biological process. To guide the learning procedure, we employ four specific loss functions.

Expressive loss. We consider that the function $\mathbf{f}$ can predict the output with high accuracy. Given the input $[\mathbf{y}, \text{noise}]$, the output is expected to be highly expressive. The expressive loss is defined as:

$$\mathcal{L}_{\text{exp}} = \mathbb{E}[||\mathbf{f}_{\text{fw}}([\mathbf{y}, \text{noise}]) - \mathbf{x}||^2] \tag{3}$$

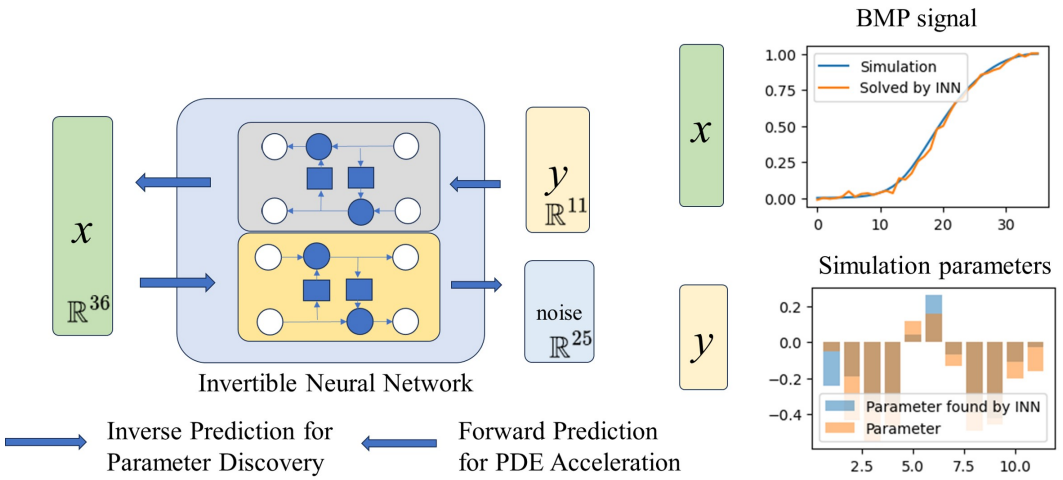


Fig. 2: Our proposed learning framework capitalizes on the invertibility of the Invertible Neural Network (INN), enhancing its versatility and enabling deployment in two distinct directions. For forward prediction, the INN serves as an accelerator for PDE solvers, efficiently handling large datasets without significant time costs. In the inverse prediction scenario, the INN facilitates parameter discovery, a crucial process for simulations that addresses the challenges of finding optimal input parameters via traditional PDE methods. This dual capability markedly improves both computational efficiency and analytical precision in complex predictive environments.

Reconstruction loss. The second term is the reconstruction loss, which enables the prediction of parameters in the input space based on a given output. This loss functions as the inverse of the expressive loss, it is defined as:

$$\mathcal{L}_{\text{back}} = \mathbb{E}[||\mathbf{f}_{\text{bw}}(\mathbf{x}) - [\mathbf{y}, \mathbf{0}]||^2], \tag{4}$$

The reconstruction loss aims to encode the useful information within the first d dimensions. Here, the tensor $\mathbf{0} \in \mathbb{R}^{m-d}$ is used to align the dimensions of the input and output.

Noise calibration loss. The third term is the noise calibration loss, which aims to minimize the magnitude of noise, serving as a robustness component within the learning objective. Given the high sensitivity of the INN to input, setting noise to $\mathbf{0}$ would establish an overly strict relationship between $\mathbf{y}$ and $\mathbf{x}$. This is undesirable, as slight fluctuations in either $\mathbf{y}$ or $\mathbf{x}$ could lead to significant prediction deviations. Therefore, we maintain that the noise should be small but not eliminated. The noise calibration loss is defined as follows:

$$\mathcal{L}_{\text{noi}} = \mathbb{E}[||\mathbf{x}_{\text{noise}}|| - \epsilon_n]_+, \tag{5}$$

where ϵ_n is a hyperparameter that controls the magnitude of the noise tensor. We detail the practical determination of this parameter in Section III-C2.

Noise prior loss. The fourth term aims to control the distribution of the noise term, ensuring it conforms to a predefined prior distribution. We denote the distribution of the noise term as noise $\sim \mathbb{P}$ and a target noise distribution as $\mathbf{z}_{\text{noise}} \sim \mathbb{N}$. Consequently, the noise prior loss is defined as follows:

$$\mathcal{L}_{\text{nop}} = \text{MMD}(\mathbb{P}, \mathbb{N}), \tag{6}$$

where $\mathbb{N}$ represents a Gaussian prior with zero mean and unit covariance matrix. Maximum Mean Discrepancy (MMD) is a statistical method for comparing two distributions that are accessible only through samples. A small MMD value indicates that the distribution $\mathbb{P}$ is converging towards $\mathbb{N}$. In other words, the noise vector is approaching the prior distribution.

The total objective function is the weighted sum of all the terms above:

$$\mathcal{L} = \lambda_{\text{exp}}\mathcal{L}_{\text{exp}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{noi}}\mathcal{L}_{\text{noi}} + \lambda_{\text{nop}}\mathcal{L}_{\text{nop}}. \tag{7}$$

In practice, we only tune λ_{nop} in the forward running. We determined λ_{nop} based on the lowest RMSE on the validation set. We set $\lambda_{\text{nop}} = 0.5$ for all tests. Other hyperparameters are set to 1.

III. Validation

In this section, we first assess our proposed method using a simulation test set generated by solving the PDE outlined in Section II-A. Subsequently, we evaluate the method using experiment data to determine its effectiveness in reversely identifying optimal parameters with our trained model. We also apply our approach to the experiment data to ascertain whether the parameter ranges are biologically relevant. Lastly, we employ explanatory methods to elucidate the best-fitted parameter predictions made by our model. Our code is publicly accessible on GitHub.

A. Setup

Data processing. The training dataset is generated from simulations and consists of 51491 instances, the validation and test sets contain 90000 and 26000 instances respectively. Given the significant scale discrepancies among different features, we apply the logarithmic transformation to the input features to ensure a consistent range across inputs. For the output, to align

the simulation results with the experimental data, we employ min-max normalization on each output. This ensures that the maximum value of each output is scaled to 1 and the minimum value to 0.

Network. We implement the invertible neural network using FrEIA library [11]. Our model comprises 15 fully connected coupling blocks, each with a three-layer MLP featuring ReLU activations. Two affine coupling functions are applied within each block. The network is optimized by AdamW optimizer, with a learning rate of 2^{-5}.

B. Results on Simulation Test

We evaluate the model performance bidirectionally: the forward pass maps parameters from an 11-dimensional space ($\mathbb{R}^{11}$) to predictions in a 36-dimensional space ($\mathbb{R}^{36}$), while the backward pass inversely maps predictions back to the parameter space. For our approach, we employ the same Invertible Neural Network (INN) for both directions. In contrast, due to the unidirectional capacity of Multilayer Perceptrons (MLPs), we train two separate MLPs: one for forward predictions (MLP Forward) and the other for backward predictions (MLP Backward), each specifically trained for direction-specific outcomes. Other baseline models utilize distinct network parameters, each uniquely trained for one direction.

For baseline comparisons, we employ commonly used regression methods, including Gradient Boosting Regressor (GBR), Decision Tree (DT), Support Vector Machine (SVM), Linear Regression (LR), Ridge Regression (RR), and MLPs. For GBR, DT, SVM, LR, and RR, we constructed models using the scikit-learn library [12]. For the MLP, we used a five-layer architecture using PyTorch, ensuring that the total number of trainable parameters is comparable to those of the INN model. The results can be found in Table I.

Method	Forward RMSE ↓	Backward RMSE ↓
GBR	0.4537	0.3919
DT	0.4949	0.5315
SVM	0.4273	0.4224
LR	0.5629	0.4409
RR	0.5629	0.4436
MLP (Forward)	0.0854	NA
MLP (Backward)	NA	0.3801
Our INN	0.2228	0.4088

TABLE I: Performance on single-direction prediction: Our proposed method closely approaches the performance of the MLP. Both models are executed bidirectionally using the same network architecture, which better simulates real biological processes. Approaching the performance of the MLP is considered satisfactory, given our stringent requirements for network construction and bidirectional functionality.

From Table I, we observe that our method achieves comparable performance with MLP in one-way prediction tasks. This result is noteworthy considering our model's complexity, as it is designed to handle both forward and backward predictions within a single framework. The dual functionality inherently introduces more constraints compared to the baseline single-direction networks, yet our approach remains highly competitive. This demonstrates the effectiveness and potential of our method in complex predictive scenarios.

We evaluate the robustness of each method through a reconstruction test. Specifically, we input parameters from an 11-dimensional input parameter space ($\mathbb{R}^{11}$) to generate predictions in a 36-dimensional output simulation results space ($\mathbb{R}^{36}$). Subsequently, these predictions are used as inputs to retrieve the original parameters. The robustness is quantified by computing the RMSE between the reconstructed parameters and the original inputs, providing a measure of each method's fidelity in preserving information through the transformation cycle. From Table II, we observe that while the MLP performs well in single-direction prediction, it tends to over-fit in this direction, resulting in deteriorated performance in terms of robustness. Regardless, owing to its invertibility, our proposed method demonstrates superior robustness compared to all other methods evaluated.

Method	Reconstruction RMSE ↓
GBR	0.4915
Decision tree	0.4327
SVM	0.5719
Linear regression	0.4213
Ridge regression	0.4270
MLP (MLP forward & MLP backward)	0.4752
Our INN	0.0000

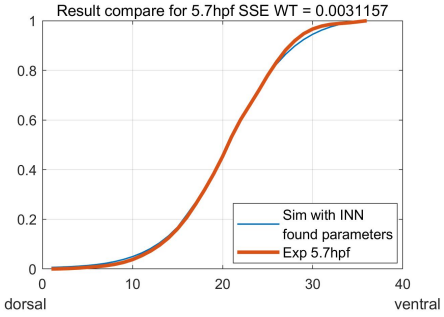
TABLE II: Performance on the robustness test. Summary of encoding-decoding test results, highlighting the robustness and fidelity of data reconstruction.

For the experimental data tests (presented in Sec III-C), we selected MLPs as our comparison method. This selection is based on the similarity of the number of parameters between MLPs and our model, ensuring a fair comparison. Furthermore, the strong unidirectional prediction capability of MLPs establishes them as a vital baseline for comparison.

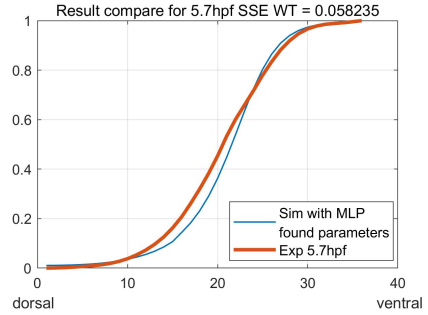
C. Results on Experiments Data Test

We further evaluate the model's performance using real-world experimental profiles of P-Smad (a downstream component of the BMP signaling network) obtained through fluorescence-labeled antibody immunostaining. The image data were collected using confocal microscope and processed with our previously developed image processing package, WaveletSEG [13]. The range of profile reflects the normalized intensity between multiple samples of P-Smad images across the marginal region shown in Fig. 1. We specifically focus on mean value lines that are interpolated into a 36-node profile to match the input size of our model. These values are input into the backward prediction modules of our proposed INN and a separately trained MLP (Backward) to estimate the simulation parameters for comparison. After obtaining the predicted best-fitted parameter set from both INN and MLP, we then apply the original PDE solver to simulate outcomes for evaluation. The simulation results are compared with the experimental mean

values again to evaluate the accuracy of each model, shown in Fig. 3. The RMSE between the simulation outcomes and experimental data is computed to quantify the precision of the INN and MLP models.



(a) Comparison between simulation results and experiment data, with simulation parameters determined by the INN.



(b) Comparison between simulation results and experiment data, with simulation parameters determined by the MLP (Backward).

Fig. 3: **Comparison of Experimental Data**: BMP signaling data is input into the network to identify the optimal parameters for simulation. Subsequently, simulations are conducted using parameters determined by both the INN and MLP, which address the specified equations of the BMP regulatory network. Comparisons are drawn between the simulated network outputs and the mean values of actual experimental data. A smaller discrepancy between these datasets indicates more precise parameter estimation in the model.

Results demonstrate that our method benefits from its bidirectional capabilities, establishing a more accurate relationship between input features and outputs. This leads to a substantially improved fit for real-world experimental data, achieving a 94.65% reduction in RMSE compared to the MLP. This significant improvement highlights the enhanced predictive accuracy and utility of our approach in practical applications.

1) Parameter range space discovery: In practice, experimental results do not yield a specific value but rather vary within a range, multiple parameter sets can possibly fit the experimental results within a minimal RMSE. Owing to the robustness of our method, we can leverage the experimental results to deduce optimized input ranges. Building on this, we can leverage our methods to perform and accelerate parameter optimization in the partial differential equation models. This range can also be a guide to design experiments for validation of the model at the laboratory. This will significantly reduce the laborious trial-and-error process traditionally required to determine the appropriate input range for simulations.

To achieve this, we extract the experimental profile falling within one standard deviation above and below the mean. Subsequently, we normalize these curves to match the region between [0,1] (since we are focusing on the signal profile shape instead of the amplitude of the signal level) and delineate the envelope encompassing the parameter region defined by these curves. We utilize the invertibility of the network to trace the input of each curve in the parameter space, as illustrated in Figure 4. This approach demonstrates our method's capability to identify the appropriate input ranges for simulations effectively. Then, we apply a grid search and linear sample 30 curves within that parameter region found by our INN model, fellows the forward prediction to obtain the INN predicted BMP profile based on refined parameter region. To further validate the identified input range, we input the traced parameter values into our PDE solver for simulation outputs. These outputs were then compared with actual experimental data, as well as data adjusted to include plus and minus one standard deviation. As shown in Figure 4, the simulations using our identified inputs consistently fall within the region bounded by one standard deviation above and below the experimental data, validating the accuracy of the input best-fitted parameter region identified by our method. These results indicate that our invertible network can utilize a limited amount of simulation data to archive a bi-directional mapping between parameter space and simulation result space. By leveraging this INN network, we can perform an active search for the best-fitted parameter set based on the experimental evidence.

2) Ablation Study: In this section, to further evaluate the model, we systematically investigate the contribution of each loss function employed in the training of the INN. By sequentially removing individual loss components while maintaining the others, we examine their impact on the model's performance. The effectiveness of each component is quantified by the accuracy in both forward and backward predictions on the simulation test set. The results are shown in Table III.

Method	Forward RMSE	Backward RMSE
w/o Expressive loss	0.2386	0.4100
w/o Reconstruction loss	0.2230	0.4090
w/o Noise prior loss	0.2339	0.4092
Our loss	0.2228	0.4088

TABLE III: Ablation Study on different loss components.

Table III indicates that the Expressive Loss is essential, as its removal leads to a significant drop in overall performance. Additionally, the Reconstruction Loss and Noise Prior Loss contribute to enhanced performance, with the system achieving optimal results when all loss terms are incorporated.

D. Feature explanation

Sensitivity analysis is a crucial process in the parameter analysis of the biophysical model. It allows the researchers

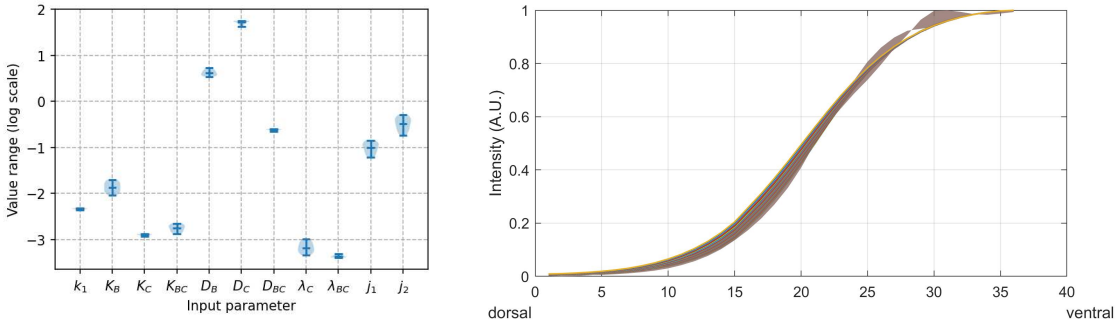


Fig. 4: **Parameter range identification and validation:** *Left:* The x-axis represents the indices (across 11 dimensions) of the simulation parameters, the y-axis indicates the permissible ranges for each specific parameter found by the reverse run of the INN. *Right:* We generated 30 input sequences based on the ranges depicted in the left figure. These sequences were used as inputs for simulations to generate BMP signals. The outputs of these simulations consistently fell within a narrow region (marked in brown), defined by one standard deviation above and below the mean.

to determine how the individual parameters influence the overall behavior of the model. Also, by identifying which parameters have the most significant influence on the system, sensitivity analysis helps prioritize efforts in model calibration and refinement, ensuring that the model accurately reflects the underlying biological processes. This process also contributes to designing specific experiments that guide and optimize the Model-Based Design of Experiments (MBDOE).

Compared to the traditional sensitivity analysis, our INN model inherently incorporates feature extraction during the training process. In this section, we aim to evaluate our trained INN model's ability to identify the significance of parameters in the original PDE model. To achieve this, we utilize explanatory tools [14] to interpret our model's predictive outcomes. Specifically, we implement a gradient-based method, Integrated Gradients [15], along with a perturbation-based method, Feature Ablation. These techniques quantitatively assess and attribute importance to the input features within our forward prediction model, enhancing our understanding of the model's decision-making process. Integrated Gradient calculates the gradient of the model's output with respect to each input feature across a series of steps from a baseline to the actual input. Feature ablation is a method used to assess the importance of individual features in a model's predictions by systematically removing these features and observing the impact on model performance. The feature explanation result can be found in Figure 5. The feature explanation results shown in Fig 5 demonstrated that the protein decay rates (K_B, K_C and K_{BC}), and the (Tld processing rate of Chd and BMP-Chd complex),(λ_C λ_{BC}) significantly contribute to our trained INN model.

Additionally, We conduct a sensitivity analysis of our method by selecting the test set sample with the lowest RMSE compared to the experimental data, ensuring a close match. Subsequently, we compute the gradient of the loss with respect to the input. The results are presented in Fig 6. the sensitivity analysis identifies a similar key parameter, with the inclusion of the production rate of BMP and Chd, (j_1 , j_2). These results are consistent with our previous conclusions in [8].

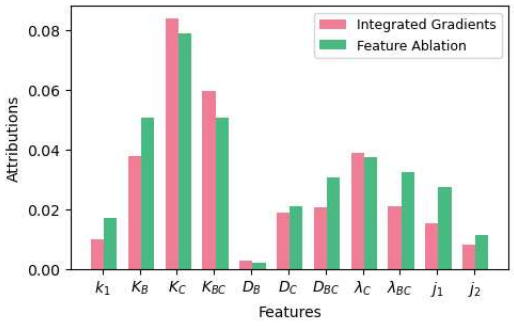


Fig. 5: Feature importance in INN predictions: Two explanation-based methods were employed to assess the importance of features in INN predictions. The consistent results across the K_B, K_C and K_{BC} features highlight their importance in the predictive process.

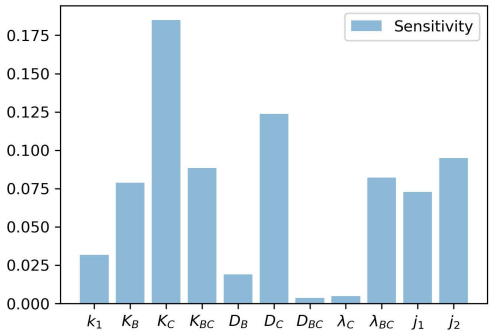


Fig. 6: **Sensitivity analysis of the input features**: The sensitivity of each input feature is quantified by calculating the gradient of the loss function with respect to the input.

In summary, we demonstrate the robustness and effectiveness of the Invertible Neural Network model to accelerate model calibration and parameter optimization for partial differ-

ential equation (PDE) models. By applying the experimental data within one standard deviation of the mean and utilizing the invertibility of our network, we can trace the corresponding parameter space and refine the model's input range. Our method leverages this variability to deduce optimized input ranges, reducing the traditional trial-and-error process. The results consistently show that simulations using our identified inputs fall within the expected experimental range, confirming the accuracy of our method. Additionally, we employ sensitivity analysis and explanatory tools to evaluate our model's performance, using techniques like Integrated Gradients and Feature Ablation to understand the model's decision-making process.

IV. Conclusion.

In this paper, we propose a method that utilizes an Invertible Neural Network (INN) to model biological processes. Our method uniquely employs a single architecture to model both the forward and backward processes, a capability not achieved by other methods. Compared to traditional machine learning algorithms like MLP, our approach demonstrates enhanced robustness and superior accuracy in parameter estimation from experimental data. Moreover, our method significantly outperforms PDE solvers in terms of efficiency. For example, during the forward process, our method can perform inference of 6000 samples in 1 second, whereas the PDE solver requires 1 second for a single sample. Beyond mere efficiency, our method also enables inverse inference, which was not feasible for a PDE solver. This comprehensive approach not only improves model calibration and refinement but also guides the design of experiments and enhances our understanding of the underlying biological processes. This approach can not only accelerate PDE solvers but also facilitate parameter discovery. It is a pioneering study in addressing inverse problems using INN.

In future research, we are planning to apply this method in further facilitating the modeling study such as Multi-objective optimization between organisms or high dimensional PDE models. In addition, including Evolutionary Algorithms, such as genetic algorithms and particle swarm optimization, mimic natural evolutionary processes to explore large parameter spaces efficiently, making them ideal for optimizing complex models like metabolic networks and gene regulatory systems.

Acknowledgement

This work is based upon efforts supported by the EMBRIO Institute, contract 2120200, a National Science Foundation (NSF) Biology Integration Institute. This research was supported in part by the NSF grant 2422229 awarded to L.L and D.M.U.

References

[1] S. Motta and F. Pappalardo, "Mathematical modeling of biological systems," *Briefings in Bioinformatics*, vol. 14, no. 4, pp. 411–422, 2013.

[2] G. Lillacci and M. Khammash, "Parameter estimation and model selection in computational biology," *PLoS computational biology*, vol. 6, no. 3, p. e1000696, 2010.

[3] N. J. Linden, B. Kramer, and P. Rangamani, "Bayesian parameter estimation for dynamical models in systems biology," *PLoS computational biology*, vol. 18, no. 10, p. e1010651, 2022.

[4] A. P. Browning and M. J. Simpson, "Geometric analysis enables biological insight from complex non-identifiable models using simple surrogates," *PLoS Computational Biology*, vol. 19, no. 1, p. e1010844, 2023.

[5] A. Yazdani, L. Lu, M. Raissi, and G. E. Karniadakis, "Systems biology informed deep learning for inferring parameters and hidden dynamics," *PLoS computational biology*, vol. 16, no. 11, p. e1007575, 2020.

[6] M. Rmus, T.-F. Pan, L. Xia, and A. G. Collins, "Artificial neural networks for model identification and parameter estimation in computational cognitive models," *PLOS Computational Biology*, vol. 20, no. 5, p. e1012119, 2024.

[7] L. Li, X. Wang, J. Chai, X. Wang, A. Buganza-Tepole, and D. M. Umulis, "Determining the role of advection in patterning by bone morphogenetic proteins through neural network model-based acceleration of a 3d finite element model of the zebrafish embryo," *Frontiers in Systems Biology*, vol. 2, p. 983372, 2022.

[8] L. Li, X. Wang, M. C. Mullins, and D. M. Umulis, "Evaluation of bmp-mediated patterning in a 3d mathematical model of the zebrafish blastula embryo," *Journal of mathematical biology*, vol. 80, pp. 505–520, 2020.

[9] A. Madamanchi, M. C. Mullins, and D. M. Umulis, "Diversity and robustness of bone morphogenetic protein pattern formation," *Development*, vol. 148, no. 7, p. dev192344, 04 2021. [Online]. Available: https://doi.org/10.1242/dev.192344

[10] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[11] L. Ardizzone, T. Bungert, F. Draxler, U. Köthe, J. Kruse, R. Schmier, and P. Sorrenson, "Framework for Easily Invertible Architectures (FrEIA)," 2018-2022. [Online]. Available: https://github.com/vislearn/FrEIA

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] T.-C. Wu, X. Wang, L. Li, Y. Bu, and D. M. Umulis, "Automatic wavelet-based 3d nuclei segmentation and analysis for multicellular embryo quantification," *Scientific reports*, vol. 11, no. 1, p. 9847, 2021.

[14] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson, "Captum: A unified and generic model interpretability library for pytorch," 2020.

[15] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.