

---

# DISCOVERING INTRINSIC MULTI-COMPARTMENT PHARMACOMETRIC MODELS USING PHYSICS INFORMED NEURAL NETWORKS

---

Imran Nasim<sup>\*1,2</sup> and Adam Nasim<sup>†2</sup>

<sup>1</sup>IBM, UK

<sup>2</sup>Department of Mathematics, University of Surrey, Guildford, GU2 7XH, Surrey, UK

## ABSTRACT

Pharmacometric models are pivotal across drug discovery and development, playing a decisive role in determining the progression of candidate molecules. However, the derivation of mathematical equations governing the system is a labor-intensive trial-and-error process, often constrained by tight timelines. In this study, we introduce PKINNs, a novel purely data-driven pharmacokinetic-informed neural network model. PKINNs efficiently discovers and models intrinsic multi-compartment-based pharmacometric structures, reliably forecasting their derivatives. The resulting models are both interpretable and explainable through Symbolic Regression methods. Our computational framework demonstrates the potential for closed-form model discovery in pharmacometric applications, addressing the labor-intensive nature of traditional model derivation. With the increasing availability of large datasets, this framework holds the potential to significantly enhance model-informed drug discovery.

*Keywords* physics informed neural networks · symbolic regression · model discovery

## 1 Introduction

Pharmacometrics, a vital tool in drug discovery, utilizes pharmacokinetic-pharmacodynamic (PKPD) models to employ ordinary differential equations (ODEs) for describing the relationship between dose, concentration, intensity, and response duration [8, 1, 20, 9]. These models are integral from early-phase target validation to optimizing lead compound development, scaling compounds for human dose predictions, and forecasting adverse events in toxicology. In clinical stage development, they aid in determining appropriate trial doses and predicting doses in untested populations, such as pediatric trials. Pharmacometric modeling is increasingly influential in drug submissions and clinical trial applications (CTAs)[21, 18, 16, 2, 11]. The landscape is evolving with the rise of large datasets and the growing use of artificial intelligence (AI), with a keen interest in purely data driven approaches to model discovery.

Inverse problems widely arise across many areas in computational science [12, 19], where the challenge frequently revolves around deducing a specific set of parameters based on certain observations or measurements. Traditional approaches based on optimization methods have shown some promise in specific areas but typically require specific domain knowledge and often have high computational costs [10, 5]. Physics informed neural networks (PINNs) have shown great promise in being applied to the inverse problem [14, 22, 13] by incorporating prior knowledge about the physics into the neural networks (NN). A limiting assumption used in the standard PINN studies is that the functional form of the differential equation is known, which is a particular issue for problems requiring the identification of relevant models. Very recent works have attempted to address this problem by combing the PINNs framework with symbolic regression (SR) methods [6, 23]. The objective of this study is to discover an underlying mathematical model based on pharmacometrics data, in a purely data driven way. To this end, we present a novel pharmacokinetic informed neural network (PKINNs hereafter) which combines PINNs and SR, enabling the network to possess important features of interpretability and generalizability, to discover intrinsic mechanistic models from noisy data in addition

---

\*imran.nasim@ibm.com, m20319@surrey.ac.uk

†m20331@surrey.ac.uk

to estimating several unknown parameters. We demonstrate that our framework accurately and robustly predicts the intrinsic derivatives of the underlying PK model and performs well in extrapolation prediction scenarios.

## 2 Methodology and Experiments

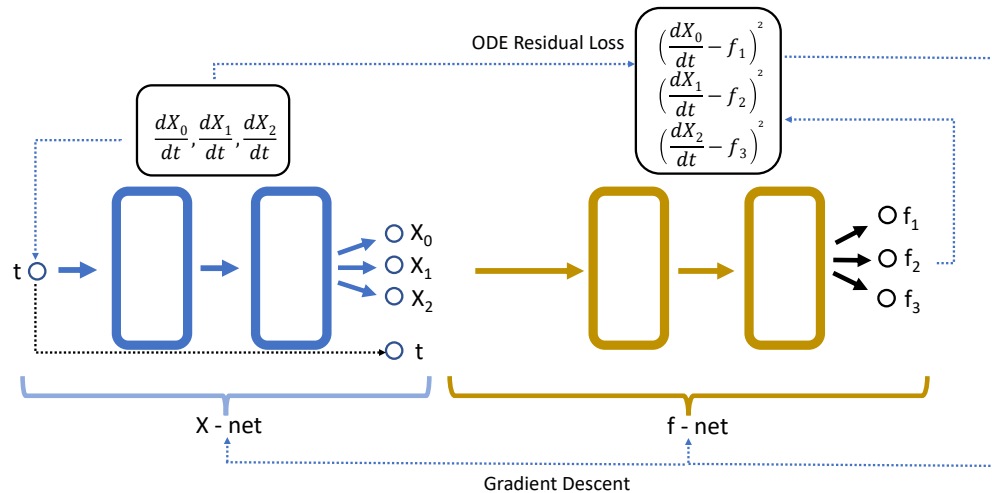


Figure 1: Schematic of the PKINNs architecture.

**Physics-informed Neural Network.** The vanilla PINN approach [17] tackles differential equation based problems by developing neural networks as substitute models for key quantities. It involves creating a loss function in relation to the differential equation that characterizes the physical phenomena adjusting the neural network parameters to minimize this loss function. The schematic of our PKINNs architecture is presented in Fig 1. The overall network comprises of two sub-networks  $X$ -net and  $f$ -net having parameters  $\theta$  and  $\phi$  respectively. The main PINN,  $X$ -net, predicts  $\mathbf{X} = \{X_0, X_1, X_2\}$  using a single input  $t$ . The auxiliary network,  $f$ -net, takes in the input from  $X$ -net and predicts  $\mathbf{f} = \{f_1, f_2, f_3\}$  which are unknown functions. The loss function to be minimized can be written as  $\mathcal{L}_{total} = \lambda_{data}\mathcal{L}_{data} + \lambda_{ODE}\mathcal{L}_{ODE} + \lambda_{IC}\mathcal{L}_{IC}$ , where  $\lambda_{data}$ ,  $\lambda_{ODE}$ ,  $\lambda_{IC}$  are weights for the data, ODE and IC components of the loss function. A point worth noting is that this architecture is flexible and allows for unknown parameters to be jointly optimized with the weights and biases of the network. We found that  $\lambda_{data} = 1$ ,  $\lambda_{ODE} = 2$ ,  $\lambda_{IC} = 1$  yielded the best results in our experiments. For further details on loss function, see Apx A.

**Symbolic Regression.** For symbolic regression we consider two different methods, PySR [7] and SINDy [4]. PySR uses evolutionary optimization algorithms whereas SINDy uncovers concise governing equations by selecting prominent candidate functions from a complex, high-dimensional nonlinear function space, using sparse regression techniques.

**Data generation.** We implemented the canonical pharmacokinetic two-compartment model with first-order absorption and first-order elimination [3]

$$\begin{aligned} \frac{dX_0}{dt} &= -k_a \cdot X_0 \\ \frac{dX_1}{dt} &= k_a \cdot X_0 - \left(\frac{CL + Q}{V_1}\right) \cdot X_1 + \frac{Q}{V_2} \cdot X_2 \\ \frac{dX_2}{dt} &= \frac{Q}{V_1} \cdot X_1 - \frac{Q}{V_2} \cdot X_2 \end{aligned} \quad (1)$$

where  $X_0$ ,  $X_1$ , and  $X_2$  denote the drug quantities in the depot compartment, the central compartment, and the peripheral compartment, respectively. Initially,  $X_0 = 1$  and both  $X_1$  and  $X_2$  are zero in all simulations. All simulations were run to  $t = 10$  to generate the PK datasets. We initialize the parameters as described in [15], for details about the parameters please see Apx. B.1. We synthetically generate data sets by numerically solving Eq (1). To simulate a real world like dataset we add Gaussian noise of different strengths to the simulated data. For this study we choose three levels of Gaussian noise i)  $\mathcal{N}(0, 0.005)$  ii)  $\mathcal{N}(0, 0.01)$  iii)  $\mathcal{N}(0, 0.02)$  which yields us three unique PK datasets which we refer to as low noise, medium noise and high noise respectively.

**Implementation and Training** For  $X$ -net and  $f$ -net we choose two and three hidden layers respectively with each layer having 100 neurons. The activation function used for all layers was tanh. We implemented the Adam optimizer

with a learning rate of  $10^{-2}$  for all models where each model was ran for 1000 epochs. We initialised the five free parameters in Eq (1) as learnable parameters within the network and gave an initial guess of unity for all parameters. We used all of the data upto  $t = 8$  for training the models and kept the data from  $t = 8 - 10$  for testing the quality of the model predictions. We train all of the PKINNs models using 80% of the total data as training data ( $t < 8$ ) and keep the remaining 20% of the data as test data ( $t \geq 8$ ) to evaluate the PKINNs in an extrapolation scenario. Further implementation details and hyperparameter choices can be found in Apx. A.2.

### 3 Results

The raw data and the model predictions for the drug concentrations are presented in Fig 2. We observe that PKINNs is able to accurately model the drug concentration data, with profiles often lying on the raw data points. Interestingly, we find that PKINNs is robust to the noise in the raw data irrespective of the noise strength which is demonstrated by the ability to extract the smooth drug conservation curves which are intrinsic to the multi-compartment PK model. To further probe the ability of PKINNs to capture the underlying PK model, we compare the calculated derivatives against the predicted derivatives for each component which is presented in Fig 3. We observe a clear linear relationship showing a very good agreement between the calculated and predicted derivatives for all drug concentration components for the low and medium noise datasets (upper and middle panels of Fig 3). This characteristic is also observed in the profiles obtained from the high noise dataset, though we note a slight deviation from a pure linear relationship in the third component  $dX_2/dt$  vs  $f_3$  (bottom right panel of Fig 3). We find that this effect is not significant and overall PKINNs accurately models and captures the intrinsic derivatives of the underlying PK model. We have demonstrated that PKINNs performs very well in predicting the drug concentration curves for the training data, but a natural question is how well the model performs in an extrapolation setting which is important for predictions tasks in Pharmacology. The extrapolation drug concentration curves for PKINNs is represented as dashed lines in the shaded regions of Fig 2. For the low and medium noise datasets we find that the extrapolations by PKINNs are faithful to the raw data with PKINNs yielding smooth curves that qualitatively match the raw data. We find that in the case of high noise dataset, the extrapolation prediction from PKINNs is not quite as good as with the lower noise models. To get a quantitative measure of the extrapolation agreement, we measure the mean squared error between the raw data and the PKINNs prediction which is given in Table 2. From the MSE values, we see that the best agreement between the raw data and PKINNs happens for the low noise dataset<sup>3</sup> achieving loss values of order magnitude  $10^{-5}$  for  $X_0$  and  $X_1$  whereas the medium and high noise datasets loss values are often of order magnitude  $10^{-4}$ .

So far we have demonstrated that PKINNs accurately predicts the drug concentration curves very well capturing the intrinsic derivatives of the datasets and is robust to the noise present in the data, however the internals of PKINNs model remain a black box. We proceed to uncover the black box to provide an explainable model yielded by PKINNs using techniques from Symbolic Regression, specifically PySR and SINDy. We apply both PySR and SINDy to the predicted drug concentration data generated by PKINNs, the results are presented in Tab 1. We observe that the functional forms for the drug concentration components predicted by PySR appear to be less sensitive to noise present in the data compared to the functional forms predicted by SINDy. However we find that the functional forms predicted by SINDy better match the functional form of the multi-compartment PK model used to generate the datasets which naturally leads to a more interpretable model. Additionally, as SINDy retains a more faithful functional form we can more easily interpret the values of the constants in context of the PK model compared to the functional forms yielded by PySR. We note that both models do not capture the real intrinsic ODE but we find this is not entirely surprising as noise was synthetically added to the datasets to give a more realistic data distribution, similar to what is collected from experiments.

Table 1: Recovered expressions for the approximate ODE components  $\tilde{f}_1$ ,  $\tilde{f}_2$  and  $\tilde{f}_3$  from PKINNs using both PySR and SINDy. Each row corresponds to the results for different levels of noise present in the raw data (low, medium and high).

Noise	PySR			SINDy		
	$\tilde{f}_1$	$\tilde{f}_2$	$\tilde{f}_3$	$\tilde{f}_1$	$\tilde{f}_2$	$\tilde{f}_3$
Low	$-1.1X_0$	$(-0.6 + X_0)X_0$	$-1.1X_0 + 0.2X_2$	$-X_0 - 0.8X_1$	$0.4X_0 - 3.4X_1 - 0.2X_2$	$2.3X_1 - 0.3X_2$
Medium	$-1.2X_0$	$(-0.6 + X_0)X_0$	$0.2X_0$	$-1.2X_0$	$0.4X_0 - 2.8X_1$	$2.2X_1 - 0.5X_2$
High	$-1.2X_0$	$(-0.6 + X_0)X_0 + X_1$	$0.2X_0$	$-1.1X_0 - 0.4X_1 - 0.3X_2$	$0.2X_0 - 1.3X_1 - 0.3X_2$	$0.2X_0 - 0.2X_2$

<sup>3</sup>While we have shown PKINNs performs well in a extrapolation prediction scenario, we stress that the model was not designed specifically for this use case

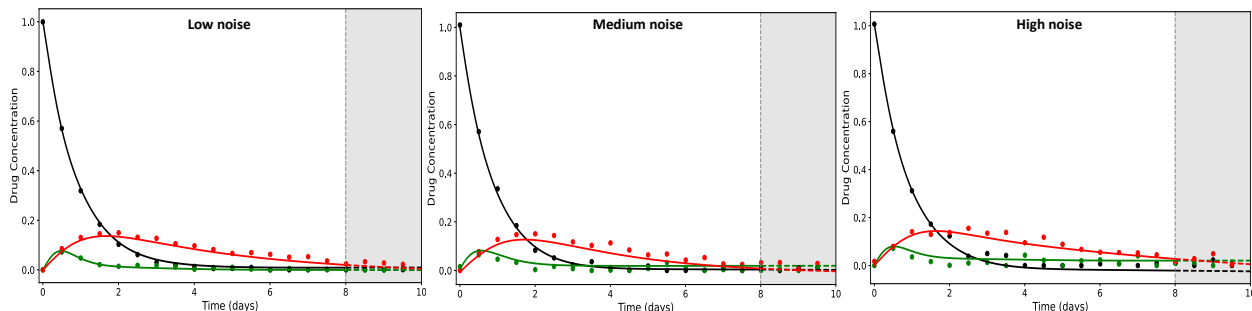


Figure 2: Drug concentration curves for the datasets. Raw data is represented as filled circles and the PKINNs model are solid lines, where the extrapolated region of the model is shown by dashed lines in the shaded region. Line colours:  $X_1$  (black),  $X_2$  (green),  $X_3$  (red).

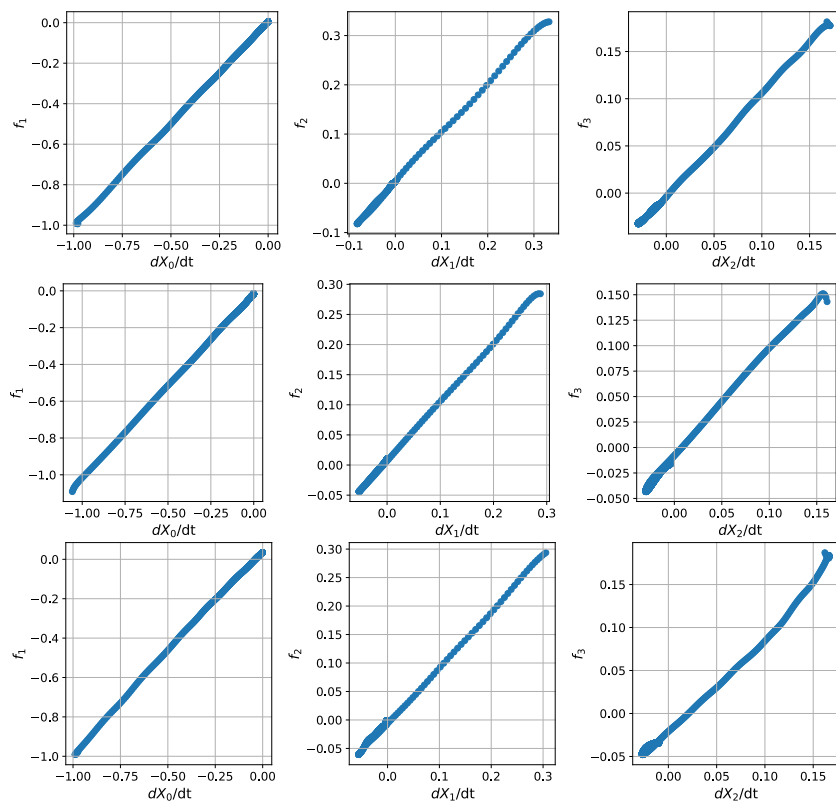


Figure 3: Comparison between the calculated derivatives and predicted derivatives of PKINNs. Low noise (upper panels), medium noise (middle panels) and high noise (lower panels).

## 4 Impact statement

In this work we have developed a novel pharmacokinetic informed neural network model called PKINNs, which is used to model drug concentration data yielded from pharmacokinetic experiments. We demonstrate that PKINNs accurately and robustly predicts the intrinsic derivatives of the underlying PK model and performs well in extrapolation predictions scenarios. We show that the dynamical models yielded by PKINNs are interpretable and explainable using techniques from SR, namely PySR and SINDy, and find the functional forms from SINDy to well predict the intrinsic PK model. Our goal is to enhance PKINNs to handle arbitrary compartment-based PKPD and complex quantitative systems pharmacology (QSP) models. This extension is particularly impactful as manually deriving such models is labor-intensive. The data-driven approach presented here explores function space, offering various parsimonious models for practitioners to choose from. Even if an expert disagrees with the suggested system, the framework serves as a valuable starting point for further derivation and selection in model-informed drug discovery.

## References

- [1] B. C. Bender, E. Schindler, and L. E. Friberg. Population pharmacokinetic-pharmacodynamic modelling in oncology: a tool for predicting clinical response. *Br J Clin Pharmacol*, 2015.
- [2] V. A. Bhattaram, C. Bonapace, D. M. Chilukuri, J. Z. Duan, C. Garnett, J. V. Gobburu, S. H. Jang, L. Kenna, L. J. Lesko, R. Madabushi, Y. Men, J. R. Powell, W. Qiu, R. P. Ramchandani, C. W. Tornoe, Y. Wang, and J. J. Zheng. Impact of pharmacometric reviews on new drug approval and labeling decisions—a survey of 31 new drug applications submitted between 2005 and 2006. *Clin Pharmacol Ther*, 2007.
- [3] David W A Bourne. *Mathematical modeling of pharmacokinetic data*. Routledge, 2018.
- [4] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.
- [5] Tan Bui-Thanh, Omar Ghattas, James Martin, and Georg Stadler. A computational framework for infinite-dimensional bayesian inverse problems part i: The linearized case, with application to global seismic inversion. *SIAM Journal on Scientific Computing*, 35(6):A2494–A2523, 2013.
- [6] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature communications*, 12(1):6136, 2021.
- [7] Miles Cranmer. Interpretable machine learning for science with pysr and symbolic regression. *arXiv preprint arXiv:2305.01582*, 2023.
- [8] H. Derendorf and B. Meibohm. Modeling of pharmacokinetic/pharmacodynamic (PK/PD) relationships: concepts and perspectives. *Pharm Res*, 1999.
- [9] L. Gibiansky and E. Gibiansky. Target-mediated drug disposition model: approximations, identifiability of model parameters and applications to the population pharmacokinetic-pharmacodynamic modeling of biologics. *Expert Opin Drug Metab Toxicol*, 2009.
- [10] Dan Givoli. A tutorial on the adjoint method for inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 380:113810, 2021.
- [11] J. V. Gobburu and P. J. Marroum. Utilisation of pharmacokinetic-pharmacodynamic modelling and simulation in regulatory decision-making. *Clin Pharmacokinet*, 2001.
- [12] Marco A Iglesias, Kody JH Law, and Andrew M Stuart. Ensemble kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, 2013.
- [13] Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations. *Journal of Computational Physics*, 426:109951, 2021.
- [14] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. hp-vpinns: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547, 2021.
- [15] Dan Lu, Sandhya Girish, Yuying Gao, Bei Wang, Joo-Hee Yi, Ellie Guardino, Meghna Samant, Melody Cobleigh, Mothaffar Rimawi, Pierfranco Conte, et al. Population pharmacokinetics of trastuzumab emtansine (t-dm1), a her2-targeted antibody–drug conjugate, in patients with her2-positive metastatic breast cancer: clinical implications of the effect of covariates. *Cancer chemotherapy and pharmacology*, 74:399–410, 2014.
- [16] J. R. Powell and J. V. Gobburu. Pharmacometrics at FDA: evolution and impact on decisions. *Clin Pharmacol Ther*, 2007.
- [17] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [18] M. Rowland, L. J. Lesko, and A. Rostami-Hodjegan. Physiologically Based Pharmacokinetics Is Impacting Drug Development and Regulatory Decision Making. *CPT Pharmacometrics Syst Pharmacol*, 2015.
- [19] Malcolm Sambridge and Klaus Mosegaard. Monte carlo methods in geophysical inverse problems. *Reviews of Geophysics*, 40(3):3–1, 2002.
- [20] T. Tuntland, B. Ethell, T. Kosaka, F. Blasco, R. X. Zang, M. Jain, T. Gould, and K. Hoffmaster. Implementation of pharmacokinetic and pharmacodynamic strategies in early research phases of drug discovery and development at Novartis Institute of Biomedical Research. *Front Pharmacol*, 2014.

- [21] M. Usman, S. Khadka, M. Saleem, H. Rasheed, B. Kunwar, and M. Ali. Pharmacometrics: A New Era of Pharmacotherapy and Drug Development in Low- and Middle-Income Countries. *Adv Pharmacol Pharm Sci*, 2023.
- [22] Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021.
- [23] Zhen Zhang, Zongren Zou, Ellen Kuhl, and George Em Karniadakis. Discovering a reaction–diffusion model for alzheimer’s disease by combining pinns with symbolic regression. *Computer Methods in Applied Mechanics and Engineering*, 419:116647, 2024.

## A PKINNs details

### A.1 Loss function

Given a model that predicts  $\mathbf{X}$  and  $\mathbf{f}$  for a temporal coordinate  $t$ , where  $\mathbf{X}$  is a vector function with three components  $X_0, X_1, X_2$ , the residuals for each of these components are calculated as follows:

$$\begin{aligned}\text{res}_1 &= \frac{dX_0}{dt} - f_1, \\ \text{res}_2 &= \frac{dX_1}{dt} - f_2, \\ \text{res}_3 &= \frac{dX_2}{dt} - f_3,\end{aligned}$$

where  $f_i$  are the predicted ODE components from PKINNs. These residuals represent the difference between the derivatives of the model predictions and the predicted values. The ODE loss function can be formulated as the mean square of these residuals:

$$\mathcal{L}_{ODE} = \frac{1}{N} \sum_{i=1}^{N_{ODE}} \left( \sum_{i=1}^{N_c} \left[ \frac{dX_{i-1}}{dt} - f_i \right] \right)^2 \quad (2)$$

where  $N_{ODE}$  is the number of collocation points enforcing the ODE residual loss and  $N_c$  are the number of compartments of the PK model.

The loss function of the initial condition is defined as the error between a constant initial condition vector and the predicted values, it can be represented as:

$$\mathcal{L}_{IC} = \sqrt{\sum_{i=0}^{N_c} (X_i - X'_i)^2}. \quad (3)$$

The final component of the loss functions is the data loss term, defined as the mean squared error between the true values  $\mathbf{X}$  and the predicted values  $\mathbf{X}'$  from the model:

$$\mathcal{L}_{data} = \frac{1}{N} \sum_{i=0}^{N_{data}} (X_i - X')^2. \quad (4)$$

The total loss function combines all the separate components with component specific weighting, which can be written as:

$$\mathcal{L}_{total} = \lambda_{data}\mathcal{L}_{data} + \lambda_{ODE}\mathcal{L}_{ODE} + \lambda_{IC}\mathcal{L}_{IC}. \quad (5)$$

Throughout our numerical experiments, we empirically found that  $\lambda_{data} = 1$ ,  $\lambda_{ODE} = 2$ ,  $\lambda_{IC} = 1$  yielded the best performance for PKINNs.

## A.2 Model training details

All of the deep learning models were implemented using KERAS in TENSORFLOW. The input dimension into  $X$ -net was one and the output dimension was three, to match the number of variables within the PK model studied. The input dimensions into  $f$ -net was set to four and the output dimension was set to three to match the number of ODEs of the PK model. We experimented with a number of learning rates using the Adam optimizer but we found that  $10^{-2}$  yielded us the best performance. We also found that 1000 epochs was sufficient to achieve convergence of the PKINNs model for the experiments we ran. Interestingly, we found a notable improvement in the model performance when we set  $\lambda_{ODE} = 2$  from  $\lambda_{ODE} = 1$ .

## A.3 Extrapolation Error

We measure the mean squared error of the raw data against the predictions made by PKINNs. The results are given in in Table 2.

Table 2: MSE of the PKINNs extrapolation prediction against the raw data for the different drug concentration components  $X_0$ ,  $X_1$  and  $X_2$ .

Noise Level	MSE $X_0$	MSE $X_1$	MSE $X_2$
Low	$6.2 \times 10^{-5}$	$1.9 \times 10^{-5}$	$1.6 \times 10^{-4}$
Medium	$3.1 \times 10^{-5}$	$2.6 \times 10^{-4}$	$5.7 \times 10^{-4}$
High	$7.1 \times 10^{-4}$	$1.7 \times 10^{-4}$	$5.3 \times 10^{-4}$

## B Pharmacology model details

### B.1 Parameters

The model parameters are as follows:  $k_a=1.14$  represents the absorption rate constant;  $CL=3.57$  is the elimination clearance rate;  $Q=1.14$  denotes the inter-compartment distribution; and  $V_1=0.454$ ,  $V_2=2.87$  are the central and peripheral volumes of distribution, respectively.