Physics-Informed Hybrid Quantum Neural Networks for Clinical Trial Optimization via Pharmacokinetic and Pharmacodynamic Modeling and Simulation

Achraf Boussahi[1, 2, *] and Abir Chekroun[2]

[1] *Intern, Constantine Quantum Technologies, University Constantine 1 Frères Mentouri, Constantine, Algeria*
[2] *École Supérieure en Informatique 08 Mai 1945, Sidi Bel Abbes, Algeria*

Pharmacokinetics and Pharmacodynamics (PK/PD) modelling is an essential tool in today's drug development pipeline, and it provides critical guidance for optimizing clinical trials and identifying efficient drug dosing regimens. The current pharmacometrics modeling approaches face major challenges of generalization and efficiency due to the limited data provided by the early trial phases. In this work, we present the current methodologies and basic notions of this field. We propose a physics-informed hybrid quantum neural network with a parallel architecture to model PK/PD time profiles at both subject and population levels. This architecture is fed with pharmacological prior knowledge using a chosen structural model expressed as ordinary differential equations. The hybrid nature of this approach has been shown in the literature to be efficient in avoiding information bottlenecks, highly effective for oscillating and harmonic observations such as those caused by multiple dosing, and powerful at generalizing to unseen data due to its quantum processing layer. The additional classical layer can overcome shockwaves and irregularities observed in the data where the QPC struggles. To the best of our knowledge, this approach has not yet been explored for this problem, and the proposed architecture, along with its possible variants and modifications, shows a promising area of research.

> *"All things are poison, and nothing is without poison; the dosage alone makes it so a thing is not a poison."*
> *—Paracelsus, 1538*

I. INTRODUCTION

For centuries, pharmacologists and medical researchers have been devoted to the discovery of healing remedies to treat diseases and preserve human lives [1]. But, as critical as this is, drug research and development (R&D) requires a long and costly process with strict testing and multiple phases of clinical trials to ensure the efficacy and safety of a candidate medicine [2], where, before guaranteeing any commercialization approval, a drug goes through years of *in vitro* laboratory research to identify its composition, followed by three phases of human trials (with possible adaptive phases) [3]. These complex procedures become cost- and time-inefficient [4], as 67% of candidates do not make it through Phase II [5].

To overcome these limitations, significant efforts were conducted to enhance R&D productivity such as predicting drug exposure prior to any *in vivo* experiments [6–9] and optimizing clinical trial designs relying on the limited data available from early phases [10, 11]. Accordingly, the pharmacometrics discipline appeared [12, 13] as a framework to quantify any related information about drugs, diseases, and trials through biologically-inspired mathematical, statistical and computational models that aid the efficient drug development and any regulatory decision-making related to it [14].

Within pharmacometrics, it is important to define two central pharmacological concepts: Pharmacokinetics (PK) and Pharmacodynamics (PD). PK models what the body does to the drug during its Absorption, Distribution, Metabolism, and Elimination phases (known as the ADME cycle). PD does the inverse and describes how the drug affects the body by following the changes of the targeted biomarker (e.g., blood pressure, enzyme activity, tumor size, or hormone levels) relative to the therapeutic concentration [15]. Together, these two concepts are combined into a PK/PD model, as shown in Figure. 1, which is widely used in clinical trial design and dose determination [15–17].

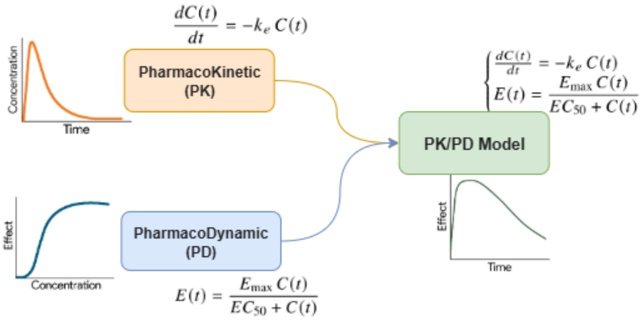

$\frac{dC(t)}{dt} = -k_e\, C(t)$
Concentration
Time
PharmacoKinetic (PK)
$\begin{cases} \frac{dC(t)}{dt} = -k_e\, C(t) \\ E(t) = \frac{E_{\max}\, C(t)}{EC_{50} + C(t)} \end{cases}$
PK/PD Model
Effect
Concentration
PharmacoDynamic (PD)
$E(t) = \frac{E_{\max}\, C(t)}{EC_{50} + C(t)}$
Effect
Time

FIG. 1. Schematic representation of Pharmacokinetics (PK), and Pharmacodynamics (PD), modeled using a simple ODEs. Once both are connected, they form a PK/PD model.

The drug behavior and its associated PK/PD time profiles can be quantified through several statistical, mechanistic, and data-driven approaches. Statistical methods, like the classical Non-Compartmental Analysis (NCA), require no biological knowledge and use structural moments (Area Under the Curve – AUC, Variance of Residence Time – VRT, . . .) to extract relationships between the time course and the PK profiles [18]. However, it is usually not suitable for sparse data [19] and cannot predict the course of PK data [17, 20].

Mechanistic models are widely used due to their high interpretability of pharmacological and physiological principles embedded within their structure. The most common are compartmental models, which divide the body into different interconnected areas called compartments, where within each

[*] Corresponding author: a.boussahi@esi-sba.dz

the concentration of a drug is assumed to be homogeneous. Usually, the bloodstream and the well-perfused organs (e.g., heart, brain, lungs, liver, and kidneys) are lumped together into a central compartment [21], which is of special interest [22], as all administered drugs must reach this compartment in the course of their distribution. There are also additional peripheral compartments, where the drug distributes more slowly, representing the lesser-perfused tissues, such as fat and muscle [23].

Pharmacokinetic models are usually described by one-, two-, or three-compartment structures at most [17], where the movement of the drug between the different compartments is explained by differential equations that incorporate the drug's PK parameters [23]. The most basic model is the one-compartment model that only includes the central compartment [22], as illustrated in Figure 2(a). In this case, the amount of drug in the central compartment, A_c, can be modeled using the following differential equation:

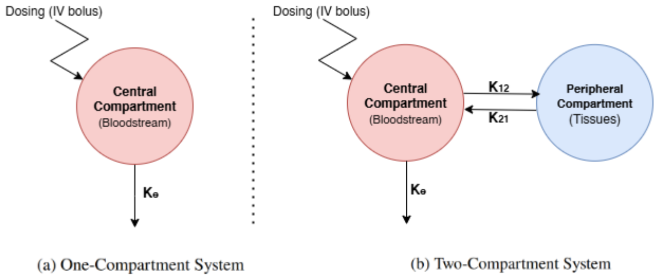
$$\frac{dA_c}{dt} = -k_e A_c \tag{1}$$

where k_e is the elimination rate constant, considering that the drug was given directly into the bloodstream as an intravenous bolus (IV bolus), in which case absorption is instantaneous.

In some cases, a multicompartmental model is required to describe the course of distribution and elimination of a drug, where the exchange between the central and peripheral compartments needs to be considered. Figure 2(b) shows a Two-Compartments model and it described through the following system of ordinary differential equations:

$$\begin{cases} \frac{dA_c}{dt} = -(k_e + k_{12})A_c + k_{21}A_p \\ \frac{dA_p}{dt} = k_{12}A_c - k_{21}A_p \end{cases} \tag{2}$$

where A_c and A_p are the amount of drug in the central and peripheral compartments respectively and k_{12}, k_{21} are the transfer rate constants between compartments, and k_e the elimination rate constant from the central compartment.


Dosing (IV bolus)
Central Compartment (Bloodstream)
Ke
Dosing (IV bolus)
Central Compartment (Bloodstream)
K12
K21
Peripheral Compartment (Tissues)
Ke

(a) One-Compartment System

(b) Two-Compartment System

FIG. 2. Illustration of one- and two-compartment models with zero-order absorption where the drug is directly injected into the bloodstream

Usually, the above-mentioned forms of compartmental models are applied when the drug is administered directly into the bloodstream as an intravenous (IV) bolus, and they can only capture the distribution and elimination phases in the multi-compartment models. To capture the absorption step of oral (or any extravascular) dosing, an additional absorption compartment is added, as shown in Figure 3. This allows the modeling of the drug's course before reaching the central compartment, generally using first-order kinetics with the following differential equation:

$$\frac{dA_g}{dt} = -k_a A_g, \tag{3}$$

where A_g is the amount of drug in the absorption (gut) compartment and k_a is the first-order absorption rate constant.

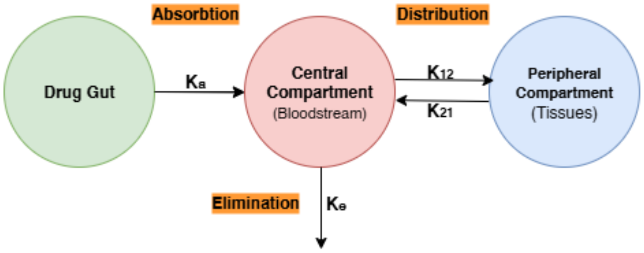

Absorbtion
Distribution
Drug Gut
Ka
Central Compartment (Bloodstream)
K12
K21
Peripheral Compartment (Tissues)
Elimination
Ke

FIG. 3. A two-compartment model with an additional first-order absorption compartment representing oral drug uptake.

We can directly add this absorption term in Eq. (28) to the system of one- or two-compartment equations, Eq. (1) and Eq. (2), respectively. In this case, the differential equation for the central compartment becomes (with F being the fraction of the drug that reaches the bloodstream, referred to as the bioavailability [22]):

$$\frac{dA_c}{dt} = \underbrace{F\,k_a A_g}_{\text{Absorption}} - \underbrace{k_e A_c}_{\text{Elimination}} + \underbrace{\big(-k_{12}A_c + k_{21}A_p\big)}_{\text{Distribution (2-Comp)}}. \tag{4}$$

Despite the useful insights that are obtained from compartmental models, they usually ignore the specific organ-level drug exposure [15, 24–26], especially in complex biological systems; hence, Physiology-Based Pharmacokinetic modeling (PBPK) was developed [24] for more realistic modeling through the integration of both drug-specific and drug-independent parameters [25–27]. Other advanced modeling frameworks, such as Quantitative Systems Pharmacology (QSP) [27] and Quantitative Structure–Activity Relationship (QSAR) [28] approaches, have also been proposed to intrinsically represent deeper biological knowledge within the models.

The most widely used approach for PK/PD modeling is the population-based method [27]. It accounts for determining the characteristics of a population of clinical trial subjects through considering their mean parameter values and their individual variability in exposure and response to a drug, as introduced by Sheiner et al. in the 1977 [29]. This approach is primarily used in direct patient care [30] to individualize drug choice [31] and design patient-specific dosing strategies [32], which are essential tools in drug development [33]. Population PK/PD is very useful for modeling sparse data [15], while still taking into consideration the subjects' differences in covariates (age, weight, sex, etc.) and their pharmacodynamic response to drug concentrations [32]. Several analytical and numerical methods have been introduced to help estimate the population characteristics, with the Non-Linear Mixed Effects (NLME) [34] approach being the most flexible to capture the Typical Population effects (population means and common values), Random Effects (inter-individual variability between subjects and their deviation from the population means), and also the residual error effects that occur within the model itself. Hence, the term Mixed Effects originates from this combination. We will discuss this approach in detail in Section III due to its importance for the course of this work.

Since we are dealing with data, Artificial Intelligence (AI) and Machine Learning (ML) have found a large playground to enhance the PK/PD modeling pipeline [27], where several approaches have been introduced in different directions, including classical ML algorithms [6, 17, 19], classical neural networks together with neural-ODE models [35], reinforcement learning [36], and more [27, 37].

As the setups we aim to model and simulate become more complex, the computations grow exponentially harder and more resource-demanding. At this stage, the most promising approach is to use quantum computers—an idea first proposed by Richard Feynman in 1981, when he argued that the non-classical nature of the world requires non-classical simulations based on the principles of quantum mechanics [38].

Since then, the evolution of quantum algorithms for simulation and modeling has often outperformed their classical counterparts, thanks to the intrinsic quantum properties of superposition and entanglement that arise naturally within and between the basic units of quantum computers, known as qubits. This has enabled quantum simulations to achieve significant breakthroughs [39] in quantum chemistry [40, 41], cosmology [42], nuclear physics [43], and healthcare [44].

This work is part of the Quantum Innovation Challenge 2025, where we aim to leverage the advantages of quantum algorithms—particularly their generalization and trainability—to enhance PK/PD modeling and support the design of optimal clinical trials. To achieve this, we repurpose the concept first introduced by Kordzanganeh et al. (2023) [45] and later extended with physics-informed modeling by Leong et al. (2025) [46], in order to introduce a Physics-Informed Hybrid Neural Network with a parallel architecture for PK/PD modeling and simulation.

This preliminary version of the work serves as an introductory documentation of the state of the art of the field for non-practitioners (as the authors discovered the field only a few weeks before the submission). The main goal is to prove the potential of feeding physical knowledge (in our case the governing ODE of the PK/PD models) into a hybrid quantum neural network as an area worth exploring. The knowledge may be fed through different forms (continuously, like neural ODEs, or discretely, like the physics-informed architecture used here) to later, in future phases, with the appropriate resources and accompaniment, explore the whole field.

In this paper, we first provide an overview of Pharmacokinetics and Pharmacodynamics modeling in the introduction. In Section II, we introduce the classical framework of population modeling, which will be useful for generalizing our method to population-level simulations. Section III defines both classical and quantum neural networks as the main components of our approach, which is also described in that section. Preliminary results obtained on synthetic data are presented in Section IV. In Section V, we discuss the computational resource requirements of our approach, particularly in the context of potential access to the Gefion supercomputer. Finally, we conclude with discussion and closing remarks.

II. POUPLATION PK/PD MODLING WITH MIXED-EFFECTS

Population Pharmacokinetics Pharmacodynamics (PopPK/PD) modeling aims to study the process of drug absorption, metabolism, elimination, and how it affects the body across a population of patients rather than just focusing on one individual, in order to characterize and model variability sources, and ensure patient differences. It is essential in clinical trial drug development and an optimal approach because in clinical practice, patients rarely get intensive blood sampling unlike in traditional PK modeling that uses a lot of samples, which also does not scale well, especially in large populations of patients. In addition, PopPK/PD extracts sparse and unbalanced data and still gets accurate and meaningful results (PK parameters).

PopPK/PD uses different classical approaches for statistical modeling such as naive pooling and fitting the average profile, which basically treats the population of patients as one entity. The first model concatenates all the given data in one graph and the other takes the average of the data across individuals at each time point. Although these methods seem quite simple and easy to calculate, they miss variability and result in biased estimates, plus nonlinear models do not behave well under averaging. Along with this, there is also the standard two-stage model that completes PK profiles from a certain number of subjects. It estimates PK parameters separately, then the population mean and the variance are computed (standard two-stage), where they will be used as empirical Bayes priors in individual estimation (iterative two-stage).

The most widely used statistical approach in PopPK/PD is nonlinear mixed-effects modeling (NLME). It is a much stronger quantitative framework that supports data variability, residual variability, and uncertainty. It is implemented by many software platforms such as NONMEM, Monolix, Phoenix NLME, and Stan. The model structure for object i observation at time j goes as follows:

$$y_{ij} = f(\theta, \eta_i, t_{ij}) + \varepsilon_{ij} \tag{5}$$

where y_{ij} is the observed measurement for individual i at time j, $f(\cdot)$ is a structural model (e.g., a PK model such as one-compartment with first-order absorption), θ are the fixed effects parameters (population means, like average clearance or volume of distribution), η_i are the random effects for individual i (captures between-subject variability),

$$\eta_i \sim N(0, \Omega), \tag{6}$$

and ε_{ij} is the residual error (captures within-subject variability and measurement noise),

$$\varepsilon_{ij} \sim N(0, \sigma^2). \tag{7}$$

In essence, the three pillars of population PK/PD modeling are: the fixed effects that represent the features common to the entire population, the covariates which refer to the individual-specific information that differs from each patient such as age, weight, organ function, or genetics influencing drug response, and also the random effects that extend the unexplained variability between individuals. It decomposes into three main types of stochastic models that define a statistical-mathematical expression relating subject-specific PK or PD model parameter values to random variables describing: between-subject variability, between-occasion variability, or residual unknown variability.

$$CL_i = CL_{\text{pop}} + \eta_i \tag{8}$$

where CL_i is the individualized value of drug clearance in the PK model, CL_{pop} is the population clearance, η_i is the random effect, usually drawn from a normal (=Gaussian) distribution with a mean of zero and a variance of ω^2:

$$\eta \sim \mathcal{N}(0, \omega^2) \tag{9}$$

$$CL_{ik} = \left[CL_{\text{pop}} + \kappa_k\right] + \eta_i \tag{10}$$

where CL_{ik} is the value of drug clearance in the PK model for the i^{th} individual on the k^{th} occasion, CL_{pop} is the population value of drug clearance in the PK model, η_i is the BSV (between-subject variability) random effect, usually

$$\eta \in \mathcal{N}(0, \omega^2)$$

(11)

κ_k is the BOV (between-occasion variability) random effect, usually

$$\kappa \in \mathcal{N}(0, \pi^2) \tag{12}$$

A simplified form for pharmacokinetic parameters with random effects is:

$$\text{CL}_i = \text{CL}_{pop} \cdot e^{\eta_{CL,i}} \tag{13}$$

$$\text{Vd}_i = \text{Vd}_{pop} \cdot e^{\eta_{Vd,i}} \tag{14}$$

where: CL_i, Vd_i = clearance and volume of distribution for individual i, CL_{pop}, Vd_{pop} = population average values (fixed effects), $\eta_{CL,i}$, $\eta_{Vd,i}$ = individual deviations from the mean (random effects).

$$Y(\text{time}) = F(\text{time}) + \varepsilon(\text{time}) \tag{15}$$

In all generality:

- Y: value of the observation (PK, PD, disease) at a certain time
- F: corresponding value of the model prediction
- ε: additive random variable describing the observation randomness not explained by the model

Usually,

$$\varepsilon \in \mathcal{N}(0, \sigma^2), \tag{16}$$

where σ^2 may or may not be a function of time.

In summary, population PK/PD modeling with mixed effects helps optimize drug dosing and improve clinical trial design while taking variability into consideration.

III. PHYISICS INFORMED HYBRID QNN FOR PK/PD MODELING AND SIMULATION

A. Classical & Quantum Neural Networks

Artificial Neural Networks (ANNs) are a powerful and fundamental algorithm in Deep Learning and Artificial Intelligence that is biologically inspired by the nervous system [27]. Their origins can be related to the work of the psychologist Frank Rosenblatt in 1958 [47], where he introduced the idea of a simple perceptron as a building block to understand and reproduce the human brain's capability of perceptual recognition, generalization, recall, and thinking [47]. The concept of the single perceptron was later extended and stacked into multilayer perceptrons (MLPs), also known as deep feedforward neural networks [48] as shown in Figure. 4.

Basically, a neural network is constructed of an input layer and an output layer, where the size of each is determined by the dimensionality of the input data and the targeted information, respectively, and hidden n-layers of m-neurons to map the input data vector into higher dimensions to capture the desired knowledge, and then reduce it back to the size of the targeted output vector. Each layer is fully connected to the next one through trainable weights, where the value of each node in the subsequent layer is determined by the weighted sum of all the neurons of the previous layer, as expressed in the following equation:

$$a_j^{(l)} = f\left(\sum_{i=1}^{m} w_{j,i}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right) \tag{17}$$

where $a_j^{(l)}$ is the final activated value of a neuron j in layer l, $a_i^{(l-1)}$ are the activated values from the previous layer, $w_{j,i}^{(l)}$ are the trainable weights, $b_j^{(l)}$ is an added bias, and $f(\cdot)$ is the activation predefined function that is used to introduce the non-linearity to the model, and so is determined the output nodes ym, this step is reffered to as the Feedforward step [48].

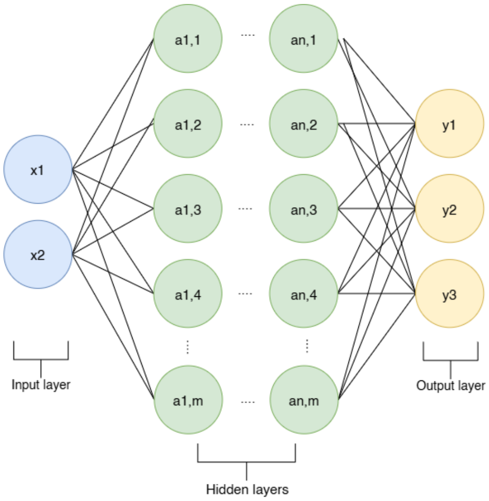


FIG. 4. A diagram of a Multilayer Perceptron (MLP) with 2 input neurons, 3 output neurons, and n hidden layers of m neurons each.

The weights and biases of such architectures are iteratively determined by going through the data and calculating the error between the predicted output in the feedforward step and the true target values using a loss function $L(y, \hat{y})$. Many different forms of loss functions exist to quantify the error by measuring the difference between the true output y and the predicted output $\hat{y}$ like the Mean Squared Error (MSE), defined as follows:

$$L(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{18}$$

This quantified error is then used through what is called the backpropagation step, which is the backbone of neural networks and gave them their popularity when utilized with MLPs [49]. It goes through the network backward and computes the gradients of the loss function with respect to each weight and bias. These parameters are then updated via an optimization algorithm such as gradient descent or one of its variants (such as the well-known Adam optimizer) [50], according to the following rule:

$$w_{new} \leftarrow w_{previous} - \eta \frac{\partial L}{\partial w} \tag{19}$$

where w is a trainable weight, η is the learning rate constant that is set to control the speed of learning, L is the loss function, and $\frac{\partial L}{\partial w}$ is the gradient of it with respect to the weight w.

Different architectures of neural networks were extended from this basic one for different problems and revolutionized Artificial Intelligence [51], such as Recurrent Neural Networks (RNNs) and Long Short-Term Memories (LSTMs) for natural language processing [52] and Convolutional Neural Networks (CNNs) for image classification [53]. Neural networks are also utilized in different tasks of Reinforcement Learning [54].

One of these variants that is of particular interest for this work is the Physics-Informed Neural Network, or PI-NN for short [55]. These neural networks are trained to solve supervised learning tasks while forcing them to respect any given law of physics described by differential equations. This will restrain the huge domain of search over which the neural network is trying to approximate its function over the data, which will optimize the learning process and improve generalization. This architecture found applications in different domains and gave significant results due to its high interpretability [56].

Physics-Informed Neural Networks are fed with physical knowledge, generally through their loss function, which the network always tries to minimize. Further details on the math behind this will be explained later in this work in Section III C. There is another family of neural networks that also have built-in ODE knowledge, called Neural Ordinary Differential Equations [57]. They are different from PINNs in that the ODEs are embedded into the neural network's architecture instead of the loss function, so the model learns the appropriate function continuously, which is more efficient with time-series data than the discreteness of PI-NNs.

Neural Ordinary Differential Equations have been used for PK/PD modeling and showed excellent results [35]. Although not considered in this version of the work, "quantizing" such an approach through a quantum-based Neural Ordinary Differential Equation [58] is an essential aspect to be explored in the upcoming phases.

B. Quantum Neural Networks

Quantum Neural Networks (QNNs) are an important component of today's Quantum Machine Learning paradigm [59]. Although they may take different forms [60, 61], they are often referred to in the literature as part of the family of Variational Quantum Circuits (VQCs) or Parameterized Quantum Circuits (PQCs) [46, 62].

The architecture of QNNs is mathematically different from classical neural networks that contain linear basis functions with nonlinear activation functions. QNNs are composed of rotational gates and entanglers, but they adopt optimization and training techniques from the classical approach. Naturally, these circuits fit a truncated Fourier circuit [46] because of the trigonometric rotational functions.

QNNs are known for their high expressibility [63] and their ability of generalization [59], where increasing the number of qubits in a variational layer (also called ansatz) increases the number of trainable parameters, and the addition of entangling gates (e.g., CNOTs) expands the effective Hilbert space dimension explored by the circuit, which enhances its expressivity [64].

A quantum node that creates the QNN circuit may contain two types of blocks: a feature map to encode the data into the quantum space, and the ansatz, which is the variational layer that needs to be tuned. In this work, we will use a simple angle encoding for embedding the classical data into our PQC, where we rotate the qubits using the value of our data on the Y-axis with an RY gate, as shown in Figure 6. Other encoding techniques may also be explored, such as amplitude encoding [65].

As for the ansatz, we will be using the strongly entangled ansatz, as shown in Figure 6, with Euler decomposition [66] using a combination of RX, RY, and RZ consecutive gates, followed by a fully entangling layer of CNOT gates. Other architectures may also be considered [67] to enhance the expressivity and optimize the hardware efficiency of the ansatz, or to incorporate more physical or prior knowledge into its architecture. The chosen circuit contributes to the parameterized circuit described by $U(x, \theta)$. The overall model can then be written as:

$$f(x, \theta) = \langle 0|^{\otimes n} U^{\dagger}(x, \theta)\, O\, U(x, \theta)|0\rangle^{\otimes n},$$

where n is the number of qubits, x denotes the classical input data, θ are the trainable parameters of the ansatz, $U(x, \theta)$ is the quantum node (feature map + ansatz), and O is the observable measured, in our case as the expectation value over the Pauli-Z basis, which results in a measurement range of $[-1, 1]$.

A variant of particular interest within QNNs is the Physics-Informed Quantum Neural Network (PI-QNN) [46, 64, 68–70], which has shown great results in the literature, especially when integrated within a hybrid architecture alongside classical neural networks [46, 68, 70]. It follows the same logic of feeding physical knowledge, described by differential equations, into the total loss function as described in the previous subsection.

C. Our approach for PK/PD modeling

We propose a Physics-Informed Hybrid Quantum Neural Network (PI-HQNN) for PK/PD modeling. The architecture is mainly inspired by the work of Kordzanganeh et al. [45], where they proposed a hybrid parallel architecture that relies on two branches for processing: a classical multilayer perceptron and a variational quantum neural network. Their work was later extended by Leong et al. [46], who enhanced the architecture with additional physics-informed processing by modifying the loss function to efficiently learn the governing differential equations of high-speed flow problems.

This hybrid architecture has been shown in the literature to be highly efficient for harmonic problems due to the quantum branch, which naturally fits the oscillating behavior with its built-in Fourier structure. The classical branch will then helps to generalize in non-harmonic situations, especially those containing shocks and discontinuities, where the QNN struggles to perform well [45, 46]. These features can be used in PK / PD modeling, in particular to overcome the problems of sparse data and the difficulty of generalizing to other scenarios. PI-HQNN can also help achieve better results with fewer trainable parameters in multiple dosing scenarios with oscillating data, and may also offer a useful tool for modeling population random effects due to the intrinsic probabilistic and truly random properties of quantum circuits [71]. We illustrate the general pipeline of the proposed model in Figure 5. To model PK profiles, the PI-HQNN aims to capture the relationship between the concentration of the drug in the body over time; therefore, its input is the different time steps. We can also consider injecting dosing information and covariates into the neural network through several gates to enhance the biological realism of the model, similar to the architecture used by Lu et al. [35]. (This part will be developed further in the next phases.)

The input will then be routed simultaneously into two different processing branches: a classical neural network, as described in III A and shown in Figure 4, and a Quantum Processing Unit that runs a variational quantum circuit. This circuit is constructed from a repeated architecture of parametrized ansatz and feature maps to encode the classical data. The circuit returns measurement outcomes as expectation values over the Pauli-Z basis, as illustrated in Figure 6.

The outputs of both branches are assumed to have the same dimensionality and expressibility. Therefore each node from the classical layer is linearly combined with its respective counterpart from the quantum layer into the same output node Oi as follows:

$$O_i = s_q \cdot q_i + s_c \cdot c_i \tag{20}$$

where s_q and s_c determine the contribution of each layer to the output. In this version of the work we consider them equal with $s_q = s_c = 1$. Future work may adjust these proportions manually or set them as trainable parameters to be optimized.

In Figure 5, we show the PI-HQNN architecture for a two-compartment model with an additional first-order absorption model, as it is the most general structure from which simpler model terms can be extracted. We note that the output meaning of this approach should remain flexible to the nature of the data and the experimental setup under consideration. In that figure, we demonstrate the outputs A_c, A_g, and A_p, representing the amount of drug in the central compartment, the gut, and the peripheral compartment, respectively. These are the traceable terms extracted from Equation 4.

The joined output of the PI-HQNN will contribute in different ways to the total loss function. In our general example, A_c will be compared to the data observations in order to compute the data loss. All the amounts of drug in the different compartments will be auto-differentiated to compute the physics loss using the descriptive Ordinary Differential Equations (ODEs), which is Equation 4 in our case, and the dosage information along with other assumptions will contribute to the initial condition loss. Therefore, the total loss function $\mathcal{L}_{\text{total}}$ of the PI-HQNN is defined as follows:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{data}} \cdot \mathcal{L}_{\text{data}} + \lambda_{\text{physics}} \cdot \mathcal{L}_{\text{physics}} + \lambda_{\text{IC}} \cdot \mathcal{L}_{\text{IC}} \tag{21}$$

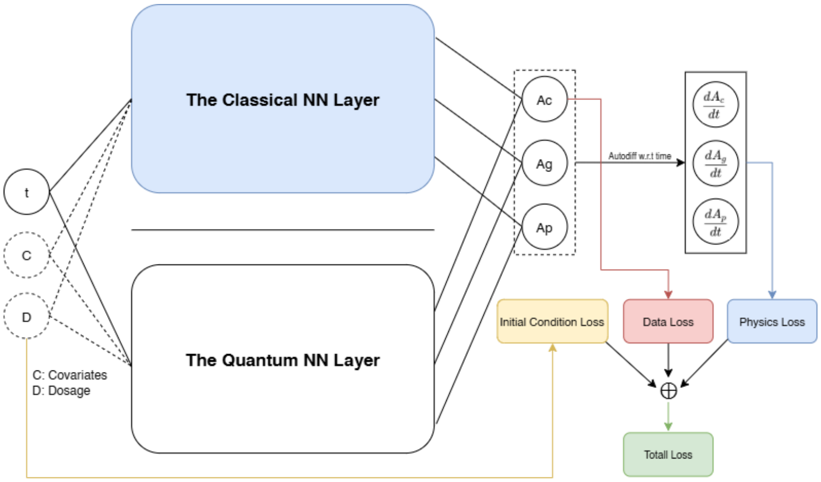


FIG. 5. The general architecture of the Physics-Informed Hybrid Quantum Neural Network (PI-HQNN) for a two-compartment model with first-order absorption

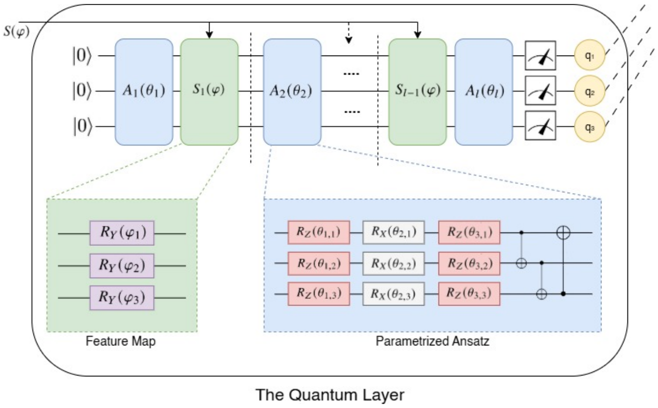


FIG. 6. Scheme for the quantum layer of the hybrid neural network with 3 qubits, showing chained feature maps and a parameterized ansatz

where $\mathcal{L}_{\text{data}}$, $\mathcal{L}_{\text{physics}}$, and $\mathcal{L}_{\text{IC}}$ are the data, physics, and initial condition losses, respectively. And λ_{data}, λ_{physics}, and λ_{IC} are coefficients that determine the contribution of each loss term to the total loss. For now, these coefficients are not trainable; they can either be set manually or assumed equal (e.g., $\lambda_{\text{data}} = \lambda_{\text{physics}} = \lambda_{\text{IC}} = 1$).

The Data Loss λ_{data} in this work is computed with a simple Mean Squared Error (MSE) function between the predicted amounts A_{pred} and the true observed ones A_{obs} at a given time step as follows:

$$\mathcal{L}_{\text{data}} = \frac{1}{N}\sum_{i=1}^{N}\left(A_{\text{pred}}(t_i) - A_{\text{obs}}(t_i)\right)^2 \tag{22}$$

where N is the total number of observed points at time steps t_i, and A is a vector of three observed amounts in this example:

$$A = \begin{pmatrix} A_c \\ A_g \\ A_p \end{pmatrix} \tag{23}$$

Note that this is a general theoretical form and can only be applied if we are able to monitor the real changes in all compartments, not only the central one. In practice, however, we usually only observe the amount of drug in the bloodstream, or more precisely, its concentration. The latter can be derived from the amount in the central compartment as follows:

$$C(t) = \frac{A_c(t)}{V} \tag{24}$$

where V is the volume of distribution of the drug. Thus, the data loss is only computed with the central compartment results.

The Physics Loss $\mathcal{L}_{\text{physics}}$ will enforce the ODE of the modeled situation, and it is computed as:

$$\mathcal{L}_{\text{physics}} = \frac{1}{M}\sum_{j=1}^{M}\left\|\dot{A}(t_j) - f\left(A(t_j)\right)\right\|_2^2 \tag{25}$$

where:

$$A(t) = \begin{pmatrix} A_c(t) \\ A_g(t) \\ A_p(t) \end{pmatrix} \tag{26}$$

and the vector field $f(A)$ for the two-compartment with first-order absorption model based on Equation 4 is

$$f(A(t)) = \begin{pmatrix} F\,k_a A_g(t) - (k_e + k_{12})A_c(t) + k_{21}A_p(t) \\ -\,k_a A_g(t) \\ k_{12}A_c(t) - k_{21}A_p(t) \end{pmatrix} \tag{27}$$

It is also possible to simplify: in the one-compartment model with first-order absorption, the network can output only the amount (or the concentration) of the central compartment and replace the traceable gut compartment with its analytical solution, where the closed form of Equation 28 can be written as:

$$A_g(t) \; = \; A_{g0}\, e^{-k_a t} \tag{28}$$

for a single bolus dose with $A_{g0} = \text{Dose}$. Thus the ODE of the system becomes

$$\frac{dA_c}{dt} \; = \; F\, k_a\, A_{g0} e^{-k_a t} - k_e A_c(t). \tag{29}$$

so the physics loss becomes

$$\mathcal{L}_{\text{physics}} \; = \; \frac{1}{M}\sum_{j=1}^{M}\left(\frac{dA_c(t_j)}{dt} - \left(F\, k_a\, A_{g0} e^{-k_a t_j} - k_e A_c(t_j)\right)\right)^2 \tag{30}$$

In the case of multiple dosing events (boluses at times t_n with amounts Dose_n) and the one-compartment model, the gut closed form becomes a sum of exponentials:

$$A_g(t) = \sum_{n=0}^{N-1} \text{Dose} \cdot e^{-k_a(t-n\tau)} \tag{31}$$

so the corresponding physics loss for multiple dosing is

$$\frac{1}{M}\sum_{j=1}^{M}\left(\frac{dA_c(t_j)}{dt} - \left(F k_a \sum_{n=0}^{N-1} \text{Dose}\, e^{-k_a(t_j-n\tau)} - k_e A_c(t_j)\right)\right)^2 \tag{32}$$

The Initial Condition Loss $\mathcal{L}_{\text{IC}}$ will be used to ensure that the model satisfies all the initial conditions of the described system at time 0 and other assumptions when the model will be extended to take the covariates into consideration (this will be given more attention in future works).

In the One-Compartment model with an IV bolus we know that at $t = 0$ the amount (or the concentration) in the central compartment is equal to the administrated drug's dose, i.e. $A_c(0) = \text{Dose}$. So the Initial Condition Loss will be:

$$\mathcal{L}_{\text{IC}} = \left(A_c(0) - \text{Dose}\right)^2 \tag{33}$$

If we add an absorption phase, then the amount of the gut compartment A_g will be introduced and the initial condition for it will be $A_g(0) = \text{Dose}$ while the central compartment is still empty $A_c(0) = 0$. In this case:

$$\mathcal{L}_{\text{IC}} = \left(A_g(0) - \text{Dose}\right)^2 + \left(A_c(0)\right)^2 \tag{34}$$

For multicompartmental models, the additional peripheral compartments are assumed empty at the beginning. e.g., a two-compartment model with absorption will have $A_g(0) = \text{Dose},\ A_c(0) = 0,\ A_p(0) = 0$, which leads to:

$$\mathcal{L}_{\text{IC}} = \left(A_g(0) - \text{Dose}\right)^2 + \left(A_c(0)\right)^2 + \left(A_p(0)\right)^2 \tag{35}$$

The total loss function $\mathcal{L}_{\text{total}}$ will be used in the training process, where the objective will be to minimize it by updating the parameters of both the classical and quantum neural networks. The loss function will be backpropagated through both branches of the model to calculate its gradients with respect to the trainable parameters using automatic differentiation. The calculated gradients will be used by a chosen optimizer (in this work, the Adam optimizer) to update the parameters in the appropriate direction and magnitude that guarantee the loss minimization, using a defined learning rate, as shown in Equation. 19.

Besides approximating the function that describes the changes of the amount of drug (or the concentration) over time, we are also interested in estimating the structural model parameters such as the elimination and absorption constants: k_e and k_a respectively, the intercompartmental rate constants $k_{12}, k_{21}, \ldots$, and even the volume of distribution V. In our approach, we consider them as independent trainable parameters to be optimized together with the neural networks' weights.

Each of these parameters is optimized first as an unconstrained trainable variable, e.g., θ_{ke}, and transformed using a softplus activation function to ensure positivity. For example, k_e is obtained as:

$$k_e = \text{softplus}(\theta_{ke}) = \ln(1 + e^{\theta_{ke}}) \tag{36}$$

We update the parameter using the same optimizer (Adam in our case), which basically relies on the gradient update rule stated in Equation 19, giving:

$$k_e^{(t+1)} = \text{softplus}(\theta_{ke}^{(t)} - \eta\, \frac{\partial \mathcal{L}_{\text{total}}}{\partial \theta_{ke}}) \tag{37}$$

The same procedure is applied to the other trainable parameters.

Usually, clinical trials are administered over a population of subjects. In the first phase, the sample size is around 40 to 60 persons [14]. However, as we have already stated in Section II, a clear inter-subject variability (random effects) is present within the population, which should be taken into consideration alongside the central tendency of the population (fixed effects).

In our approach, we propose that for population modeling, the random effect is introduced into the trainable parameters. In the mixed-effect solution, we model the subjects' variability with an additional term randomly sampled from a normal distribution with mean 0 (which gives meaning to the central tendency most of the time) and a standard deviation that we propose to be trainable over the population, as follows:

$$\theta_i = \theta + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \sigma^2) \tag{38}$$

where θ is the parameter in fixed effect (central tendency), η_i is the random effect for subject i, σ is the standard deviation which is our trainable parameter.

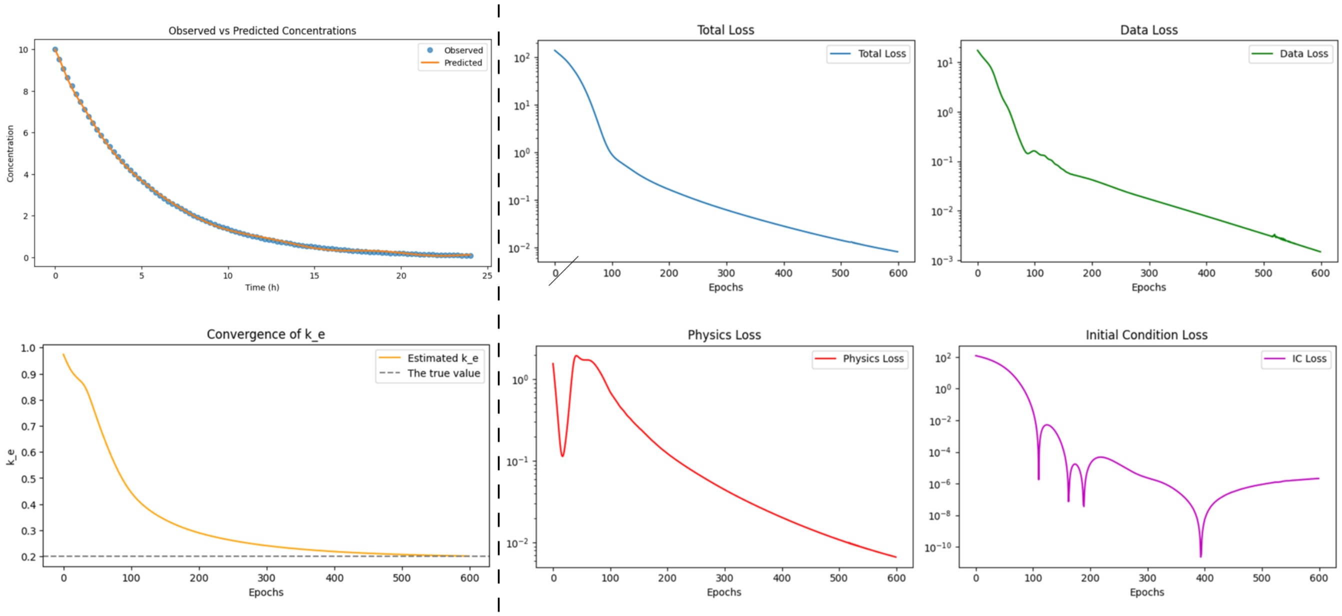


FIG. 7. Model fitting results for the one-compartment model with elimination rate constant $k_e = 0.2$. The Physics-Informed Hybrid Quantum Neural Network (PI-HQNN) achieved near-zero loss and accurately recovered the elimination constant.

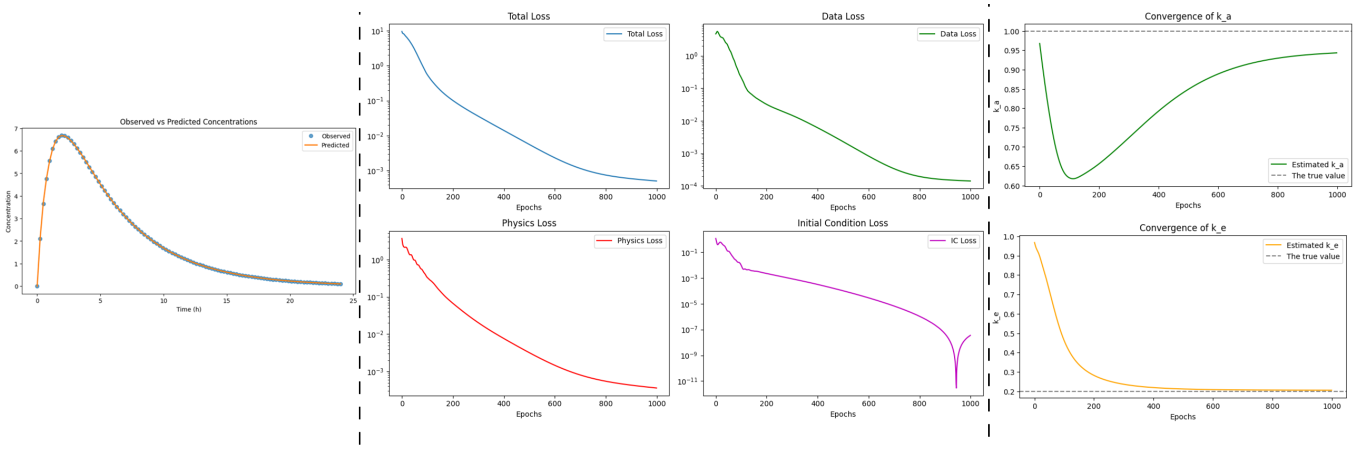


FIG. 8. Model fitting results for the one-compartment model with first-order absorption. The PI-HQNN achieved an excellent fit and accurate parameter estimation under numerical and hardware limitations.

subsectionSimulation and Dosing Design

After we train and fit our model, we can simulate other populations (we should consider the covariates as inputs when defining our architecture) or explore alternative dosing regimens (by inputting different doses) to predict their effects. For this purpose, we can employ Monte Carlo simulation.

The Monte Carlo method is a well-known simulation technique that models the behavior of systems through stochastic sampling and estimates likely outcomes using repeated random processes [72]. Due to the probabilistic nature of quantum systems, this approach can also be effectively extended to design quantum simulation algorithms, known as Quantum Monte Carlo (QMC) [73].

This subsection will be given more attention in the next phases.

IV. PRELIMINARY RESULTS

We tested our approach with the available resources on synthetic data of a single subject. First, we modeled a simple one-compartment model with a predefined $k_e = 0.2$. The results shown in Figure 7 demonstrate that the model perfectly fitted the data points, achieved a total loss close to zero, and almost perfectly predicted the elimination constant k_e.

We also tested the approach on a one-compartment model with first-order absorption. The results in Figure 8 show a near-perfect fit of the concentration–time curve and an almost exact estimation of the parameters, within numerical precision and hardware constraints.

Although we were not able to perform a full experiment due to time and hardware constraints, we demonstrated the contribution of the quantum branch in capturing harmonic patterns such as those observed in multiple dosing events. The quantum branch introduced oscillatory behavior to a certain extent, as illustrated in Figure 9.

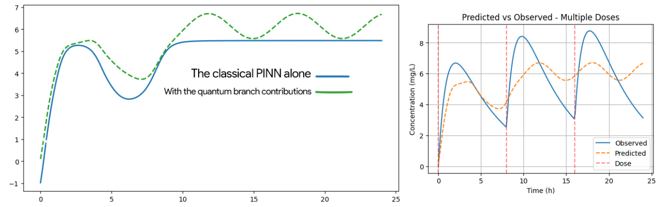


FIG. 9. Preliminary results on multiple dosing simulation. The quantum branch of the PI-HQNN contributed to capturing oscillatory behavior in the concentration–time profile.

The first version of this work is mainly theoretical. We plan to dedicate most of the effort in the next phases to complete the experimental validation and expand the scope of results.

V. COMPUTATIONAL ASSESSMENT OF THE APPROACH

We conducted a FLOPs analysis of the proposed PI-HQNN architecture. The corresponding code can be found in the supplementary notebook. Figure 10 illustrates the effect of increasing the number of qubits and the number of layers on the computational cost. Under the current setup, the architecture remains lightweight and moderately scalable. However, when moving towards real population-level simulations combined with deeper ansatz structures, access to high-performance computing resources such as supercomputers will be necessary.

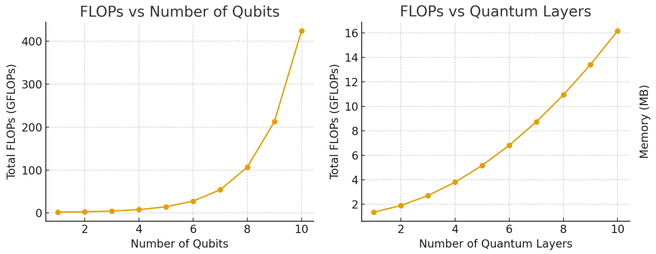


FIG. 10. FLOPs analysis of the PI-HQNN architecture showing the impact of qubit count and layer depth on computational cost.

VI. CONCLUSION

In this preliminary version of the work, we presented a theoretical framework of pharmacometrics and PK/PD modeling from different perspectives, highlighting how several efforts have been conducted to optimize this process due to its crucial role in drug development. We introduced population-based modeling to provide a basic understanding of how mixed effects can be implemented within the proposed approach. Furthermore, we defined both classical and quantum neural networks, and showed how they can be integrated with biological knowledge into our Physics-Informed Hybrid Quantum Neural Network (PI-HQNN). We proposed different variants of this architecture and discussed their potential for estimating mixed effects, as well as for simulation and dosing regimen design. Finally, we demonstrated preliminary results together with a computational assessment of the approach.

This work serves as a foundation and first step in the project, aiming to explore how physics-informed quantum and hybrid neural networks, in both continuous and discrete forms, can contribute to PK/PD modeling. Ultimately, this research supports our broader goal of developing a quantum-based software for clinical trial design: **Qlynix**.

[1] Medicine National Research Council (US) Committee to Update Science and Animals. *Science, Medicine, and Animals*. National Academies Press (US), Washington, DC, 2004. Safety Testing.

[2] National Academies of Sciences, Engineering, and Medicine. *The Future of Drug Safety: Promoting and Protecting the Health of the Public*. National Academies Press, Washington, DC, 2007.

[3] National Institutes of Health. Clinical trials: Phases of clinical trials. `https://www.nih.gov/health-information/nih-clinical-research-trials-you`, 2020. Accessed: 2025-09-17.

[4] Jack W. Scannell, Alex Blanckley, Helen Boldon, and Brian Warrington. Diagnosing the decline in pharmaceutical r&d efficiency. *Nature Reviews Drug Discovery*, 11(3):191–200, 2012.

[5] U.S. Food and Drug Administration. The drug development process: Step 3: Clinical research. `https://www.fda.gov/patients/drug-development-process/step-3-clinical-research`, 2022. Accessed: 2025-09-17.

[6] Xuelian Jia, Donato Teutonico, Saroj Dhakal, Yorgos M. Psarellis, Alexandra Abos, Hao Zhu, Panteleimon D. Mavroudis, and Nikhil Pillai. Application of machine learning and mechanistic modeling to predict intravenous pharmacokinetic profiles in humans. *Journal of Medicinal Chemistry*, 68(7):7737–7750, 2025. PMID: 40146185.

[7] M. Davies, R. D. O. Jones, K. Grime, R. Jansson-Löfmark, A. J. Fretland, S. Winiwarter, P. Morgan, and D. F. McGinnity. Improving the accuracy of predicted human pharmacokinetics: Lessons learned from the astrazeneca drug pipeline over two decades. *Trends in Pharmacological Sciences*, 41(6):390–408, 2020.

[8] Panteleimon D. Mavroudis, Donato Teutonico, Alexandra Abos, and Nikhil Pillai. Application of machine learning in combination with mechanistic modeling to predict plasma exposure of small molecules. *Frontiers in Systems Biology*, Volume 3 - 2023, 2023.

[9] Nikhil Pillai, Aparajita Dasgupta, Sirimas Sudsakorn, Jennifer

Fretland, and Panteleimon D. Mavroudis. Machine learning guided early drug discovery of small molecules. *Drug Discovery Today*, 27(8):2209–2215, 2022.

[10] Y. Zang, B. Guo, Y. Qiu, H. Liu, M. Opyrchal, and X. Lu. Adaptive phase i-ii clinical trial designs identifying optimal biological doses for targeted agents and immunotherapies. *Clinical Trials*, 21(3):298–307, June 2024.

[11] A. G. Chapple and P. F. Thall. A hybrid phase i-ii/iii clinical trial design allowing dose re-optimization in phase iii. *Biometrics*, 75(2):371–381, June 2019.

[12] Ene I. Ette and Paul J. Williams, editors. *Pharmacometrics: The Science of Quantitative Pharmacology*. John Wiley & Sons, Inc., 2007.

[13] Bhavatharini A., Deepalakshmi Mani, and Arun Parthasarathy. Pharmacometrics: The science applied from bench to bedside. *Journal of Applied Pharmaceutical Science*, 01 2022.

[14] Wang Y. United States Food and Drug Administration (FDA), Center for Drug Evaluation and Research. Division of pharmacometrics. `https://www.fda.gov/about-fda/cder-offices-and-divisions/division-pharmacometrics`, 2025. Accessed: 2025-09-18.

[15] Joseph F. Standing. Understanding and applying pharmacometric modelling and simulation in clinical practice and research. *British Journal of Clinical Pharmacology*, 83(2):247–254, 2017.

[16] Gerhard Levy, Milo Gibaldi, and William J. Jusko. Multicompartment pharmacokinetic models and pharmacologic effects. *Journal of Pharmaceutical Sciences*, 58(4):422–424, 1969.

[17] Albert Tang. Machine learning for pharmacokinetic/pharmacodynamic modeling. *Journal of Pharmaceutical Sciences*, 112(5):1460–1475, 2023.

[18] Kiyoshi Yamaoka, Terumichi Nakagawa, and Toyozo Uno. Statistical moments in pharmacokinetics. *Journal of Pharmacokinetics and Biopharmaceutics*, 6(6):547–558, 1978.

[19] Lina Keutzer, Huifang You, Ali Farnoud, Joakim Nyberg, Sebastian G. Wicha, Gareth Maher-Edwards, Georgios Vlasakakis, Gita Khalili Moghaddam, Elin M. Svensson, Michael P. Menden, Ulrika S. H. Simonsson, and on behalf of the UNITE4TB Consortium. Machine learning and pharmacometrics for prediction of pharmacokinetic data: Differences, similarities and challenges illustrated with rifampicin. *Pharmaceutics*, 14(8), 2022.

[20] Leslie Z. Benet and Renato L. Galeazzi. Noncompartmental determination of the steady-state volume of distribution. *Journal of Pharmaceutical Sciences*, 68(8):1071–1074, 1979.

[21] Tarek A. Ahmed. Pharmacokinetics of drugs following iv bolus, iv infusion, and oral administration. In Tarek A Ahmed, editor, *Basic Pharmacokinetic Concepts and Some Clinical Applications*, chapter 3. IntechOpen, London, 2015.

[22] {Stig Bousgaard} Mortensen, {Anna Helga} Jónsdóttir, Søren Klim, and Henrik Madsen. *Introduction to PK/PD modelling - with focus on PK and stochastic differential equations*. Number 2008-16 in D T U Compute. Technical Report. Technical University of Denmark, DTU Informatics, Building 321, 2008.

[23] P. Shenoy, M. Rao, S. Chokkadi, S. Bhatnagar, and N. Salins. Developing mathematical models to compare and analyse the pharmacokinetics of morphine and fentanyl. *Indian Journal of Anaesthesia*, 68(1):111–117, Jan 2024. Epub 2024 Jan 18.

[24] Kenneth B. Bischoff. Physiological pharmacokinetics. *Bulletin of Mathematical Biology*, 48(3):309–322, 1986.

[25] D. K. Shah and A. M. Betts. Towards a platform pbpk model to characterize the plasma and tissue disposition of monoclonal antibodies in preclinical species and human. *Journal of Pharmacokinetics and Pharmacodynamics*, 39(1):67–86, 2012.

[26] S. S. De Buck, V. K. Sinha, L. A. Fenu, M. J. Nijsen, C. E. Mackie, and R. A. Gilissen. Prediction of human pharmacokinetics using physiologically based modeling: a retrospective analysis of 26 clinically tested drugs. *Drug Metabolism and Disposition: The Biological Fate of Chemicals*, 35(10):1766–1780, 2007.

[27] M. McComb, R. Bies, and M. Ramanathan. Machine learning in pharmacometrics: Opportunities and challenges. *British Journal of Clinical Pharmacology*, 88(4):1482–1499, 2022.

[28] Alexander Golbraikh, Xiang Simon Wang, Hao Zhu, and Alexander Tropsha. *Predictive QSAR Modeling: Methods and Applications in Drug Discovery and Chemical Risk Assessment*, pages 2303–2340. Springer International Publishing, Cham, 2017.

[29] Lewis B. Sheiner, B. Rosenberg, and V. V. Marathe. Estimation of population characteristics of pharmacokinetic parameters from routine clinical data. *Journal of Pharmacokinetics and Biopharmaceutics*, 5(5):445–479, 1977.

[30] Ene I. Ette and Paul J. Williams. Population pharmacokinetics i: background, concepts, and models. *The Annals of Pharmacotherapy*, 38(10):1702–1706, 2004.

[31] Stephen B. Duffull, David F. Wright, and Holly R. Winter. Interpreting population pharmacokinetic-pharmacodynamic analyses–a clinical viewpoint. *British Journal of Clinical Pharmacology*, 71(6):807–814, 2011.

[32] Peter L. Bonate. Recommended reading in population pharmacokinetic pharmacodynamics. *The AAPS Journal*, 7(2):E363–E373, 2005.

[33] Diane R. Mould and Richard N. Upton. Basic concepts in population modeling, simulation, and model-based drug development. *CPT: Pharmacomet. Syst. Pharmacol.*, 1(9):e6, 2012.

[34] Reza Drikvandi. Nonlinear mixed-effects models for pharmacokinetic data analysis: assessment of the random-effects distribution. *Journal of Pharmacokinetics and Pharmacodynamics*, 44(3):223–232, 2017.

[35] James Lu, Brendan Bender, Jin Y. Jin, and Yuanfang Guan. Deep learning prediction of patient response time course from early data via neural-pharmacokinetic/pharmacodynamic modelling. *Nature Machine Intelligence*, 3(8):696–704, 2021.

[36] Benjamin Ribba, Sherri Dudal, Thierry Lavé, and Richard W. Peck. Model-informed artificial intelligence: Reinforcement learning for precision dosing. *Clinical Pharmacology & Therapeutics*, 107(4):853–857, 2020.

[37] J Cosgrove, J Butler, K Alden, M Read, V Kumar, L Cucurull-Sanchez, J Timmis, and M Coles. Agent-based modeling in systems pharmacology. *CPT: Pharmacometrics & Systems Pharmacology*, 4(11):615–629, 2015.

[38] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, June 1982.

[39] I. M. Georgescu, S. Ashhab, and Franco Nori. Quantum simulation. *Rev. Mod. Phys.*, 86:153–185, Mar 2014.

[40] Alan Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, September 2005.

[41] Honghui Shang, Yi Fan, Li Shen, Chu Guo, Jie Liu, Xiaohui Duan, Fang Li, and Zhenyu Li. Towards practical and massively parallel quantum computing emulation for quantum chemistry. *npj Quantum Information*, 2023.

[42] Marco D. Maceda and Carlos Sabín. Digital quantum simulation of cosmological particle creation with ibm quantum computers. *Scientific Reports*, 15(1):3476, 2025.

[43] José-Enrique García-Ramos, Álvaro Sáiz, J. Arias, Lucas

Lamata, and Pedro Perez-Fernandez. Nuclear physics in the era of quantum computing and quantum machine learning. *Advanced Quantum Technologies*, 05 2024.

[44] Biswaraj Baral, Reek Majumdar, Bhavika Bhalgamiya, and Taposh Dutta Roy. Evaluating quantum machine learning approaches for histopathological cancer detection: Classical, hybrid simulation, and ibm quantum computing. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 02, pages 238–239, 2023.

[45] Mo Kordzanganeh, Daria Kosichkina, and Alexey Melnikov. Parallel hybrid networks: An interplay between quantum and classical neural networks. *Intelligent Computing*, 2, January 2023.

[46] Fong Yew Leong, Wei-Bin Ewe, Tran Si Bui Quang, Zhongyuan Zhang, and Jun Yong Khoo. Hybrid quantum physics-informed neural network: Towards efficient learning of high-speed flows, 2025.

[47] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[48] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning Series. MIT Press, Cambridge, MA, 2016.

[49] Paul J. Werbos. Applications of advances in nonlinear sensitivity analysis. In R. F. Drenick and F. Kozin, editors, *System Modeling and Optimization*, pages 762–770, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.

[50] Mohamed Reyad, Amany M. Sarhan, and M. Arafa. A modified adam algorithm for deep neural network optimization. *Neural Computing and Applications*, 35(23):17095–17112, 2023.

[51] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021.

[52] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[53] Ir Purwono, Alfian Ma'arif, Wahyu Rahmaniar, Haris Imam, Haris Imam Karim Fathurrahman, Aufaclav Frisky, and Qazi Mazhar Ul Haq. Understanding of convolutional neural network (cnn): A review. *International Journal of Robotics and Control Systems*, 2:739–748, 01 2023.

[54] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018.

[55] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017.

[56] Amer Farea, Olli Yli-Harja, and Frank Emmert-Streib. Understanding physics-informed neural networks: Techniques, applications, trends, and challenges. *AI*, 5(3):1534–1557, 2024.

[57] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019.

[58] Yu Cao, Shi Jin, and Nana Liu. Quantum neural ordinary and partial differential equations, 2025.

[59] Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021.

[60] Kerstin Beer. Quantum neural networks. *leibniz informationszentrum technik und naturwissenschaften*, 2022.

[61] Min-Gang Zhou, Zhi-Ping Liu, Hua-Lei Yin, Chen-Long Li, Tong-Kai Xu, and Zeng-Bing Chen. Quantum neural network for quantum neural computing. *Research*, 6, January 2023.

[62] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors, 2018.

[63] Yadong Wu, Juan Yao, Pengfei Zhang, and Hui Zhai. Expressivity of quantum neural networks. *Phys. Rev. Res.*, 3:L032049, Aug 2021.

[64] Corey Trahan, Mark Loveland, and Samuel Dent. Quantum physics-informed neural networks. *Entropy*, 26(8), 2024.

[65] Naoki Mitsuda, Tatsuhiro Ichimura, Kouhei Nakaji, Yohichi Suzuki, Tomoki Tanaka, Rudy Raymond, Hiroyuki Tezuka, Tamiya Onodera, and Naoki Yamamoto. Approximate complex amplitude encoding algorithm and its application to data classification problems. *Phys. Rev. A*, 109:052423, May 2024.

[66] Onequbiteulerdecomposer — qiskit documentation. `https://quantum.cloud.ibm.com/docs/en/api/qiskit/qiskit.synthesis.OneQubitEulerDecomposer`, 2025. IBM Quantum, 2017–2025. Last visited: 2025-09-29.

[67] Xiaoyu Guo, Takahiro Muta, and Jianjun Zhao. Quantum circuit ansatz: Patterns of abstraction and reuse of quantum algorithm design, 2024.

[68] Alexandr Sedykh, Maninadh Podapaka, Asel Sagingalieva, Karan Pinto, Markus Pflitsch, and Alexey Melnikov. Hybrid quantum physics-informed neural networks for simulating computational fluid dynamics in complex shapes. *Machine Learning: Science and Technology*, 5(2):025045, May 2024.

[69] Abhishek Setty, Rasul Abdusalamov, and Felix Motzoi. Self-adaptive physics-informed quantum machine learning for solving differential equations. *Machine Learning: Science and Technology*, 6(1):015002, January 2025.

[70] Nahid Binandeh Dehaghani, A. Pedro Aguiar, and Rafal Wisniewski. A hybrid quantum-classical physics-informed neural network architecture for solving quantum optimal control problems, 2024.

[71] Mohamed Messaoud Louamri, Achraf Boussahi, Nacer Eddine Belaloui, Abdellah Tounsi, and Mohamed Taha Rouabah. A study of gate-based and boson sampling quantum random number generation on ibm and xanadu quantum devices, 2025.

[72] Peter Bonate. A brief introduction to monte carlo simulation. *Clinical pharmacokinetics*, 40:15–22, 02 2001.

[73] David Ceperley and Berni Alder. Quantum monte carlo. *Science*, 231(4738):555–560, 1986.