CPT: Pharmacometrics & Systems Pharmacology

ARTICLE OPEN ACCESS

Redefining Parameter Estimation and Covariate Selection via Variational Autoencoders: One Run Is All You Need

Jan Rohleff[1] | Freya Bachmann[1] | Uri Nahum[2,3] | Dominic Bräm[2] | Britta Steffens[2,4] | Marc Pfister[2] | Gilbert Koch[2] | Johannes Schropp[1]

[1]Department of Mathematics and Statistics, University of Konstanz, Konstanz, Germany | [2]Pediatric Pharmacology and Pharmacometrics, University Children's Hospital Basel (UKBB), University of Basel, Basel, Switzerland | [3]Institute of Biomedical Engineering and Medical Informatics, University of Applied Sciences and Arts Northwestern Switzerland (FHNW), Olten, Switzerland | [4]School of Life Sciences, University of Applied Sciences and Arts Northwestern Switzerland (FHNW), Olten, Switzerland

Correspondence: Jan Rohleff (jan.rohleff@uni-konstanz.de)

Received: 23 July 2025 | **Revised:** 25 September 2025 | **Accepted:** 29 September 2025

Funding: This study was supported by the Swiss National Science Foundation (SNSF) [grant number 10003647 and 229140] awarded to GK.

Keywords: Bayesian inference | covariate selection | data-driven modeling | generative artificial intelligence (AI) | machine learning | nonlinear mixed effects modeling | parameter estimation | variational autoencoder

ABSTRACT

Generative Artificial Intelligence (AI) frameworks, such as Variational Autoencoders (VAEs), have proven powerful in learning structured representations from complex, high-dimensional data. In pharmacometrics (PMX), nonlinear mixed effects (NLME) modeling is widely used to capture inter-individual variability and link covariates to characterize parameters with the goal of informing key decisions in drug research and development. This research combines the strengths of both approaches by introducing a VAE framework specifically designed for NLME modeling. The proposed method integrates the flexibility of generative AI with the interpretability and robustness of mechanism-based PMX modeling. To advance covariate selection in PMX, we replace the Evidence Lower Bound objective in VAEs with an objective function based on the corrected Bayesian information criterion. This enables the simultaneous evaluation of all potential covariate-parameter combinations, thereby allowing for automated and joint estimation of population parameters and covariate selection within a single run. Manual selection and repeated model fitting across covariate combinations are no longer required. We demonstrate the effectiveness of this combined AI-PMX approach with two representative cases. As the first generative AI-based optimization method for NLME modeling, the VAE achieves high-quality results in a single run, outperforming traditional stepwise procedures in terms of efficiency. As such, the presented approach facilitates automated model development, advancing PMX and its applications in model-informed drug development.

1 | Introduction

Model-informed drug development (MIDD) leverages pharmacometrics (PMX) modeling and simulation to support key decisions in drug research, development, and regulatory review [1]. Nonlinear mixed effects (NLME) modeling characterizes PMX mechanisms by accounting for fixed effects (population parameters) and random effects (inter-individual variability) through the maximization of the associated log-likelihood function [2], which cannot be computed in closed form. Classical solution

Gilbert Koch and Johannes Schropp Shared last authorship. Both authors contributed equally.

Previous Presentations: Parts of this work were presented in an oral presentation on June 5, 2025, at the PAGE (Population Approach Group Europe) 2025 conference in Thessaloniki, Greece.

Study Highlights

- What is the current knowledge on the topic?
  - Variational Autoencoders (VAEs) have been explored in pharmacometrics modeling. However, they have not yet been adapted to nonlinear mixed effects (NLME) frameworks or applied to covariate selection.
- What question did this study address?
  - Can we develop a robust, efficient, and flexible AI framework for NLME modeling using VAEs?
  - How competitive are VAEs compared to the established methods in estimating population parameters and selecting covariates?
  - Do they offer advantages in computational efficiency and automated covariate selection?
- What does this study add to our knowledge?
  - VAEs are a powerful AI framework for efficiently solving NLME modeling problems and opening new opportunities in pharmacometrics.
  - They estimate population parameters and select covariates simultaneously in one single run, making the modeling process less time-consuming while maintaining the quality of results.
- How might this change drug discovery, development, and/or therapeutics?
  - VAEs adapted to NLME frameworks represent a significant advancement in the field of PMX. Combining AI and PMX modeling can lead to more accurate and more efficient automated model assessment and building.

methods in NLME modeling include linearization-based approaches such as First-Order Conditional Estimation (FOCE) [3] and stochastic approximation maximization techniques such as the SAEM algorithm [4].

PMX plays a central role in MIDD, but model development remains a cumbersome and slow process due to complex model structures and often high computational demands. In particular, covariate selection is hard to automate and often results in suboptimal models.

In PMX, NLME modeling generally follows a sequential process [5]. First, the structural model is developed and the corresponding model parameters, that is, fixed and random effects, are estimated. Second, covariate selection is performed, which is often the most time-consuming part. Current automated covariate selection methods, such as SCM [6], SAMBA [7] and COSSAC [8], rely on iterative testing of potential covariate-parameter relationships. Each iteration requires estimating model parameters and potential covariate effects by maximizing the log-likelihood. Covariate selection is then based on information criteria such as the Akaike Information Criterion (AIC) [9], the Bayesian Information Criterion (BIC) [10], or the corrected BIC (BICc) [11].

NLME modeling can be viewed as a Bayesian inference problem [12, 13], since individual parameters are assumed to follow a population distribution (the prior), and the observed data update our knowledge about these parameters (the posterior). Variational autoencoders (VAEs) [14] are generative artificial intelligence (AI) methods that combine neural networks with Bayesian principles to estimate unobservable variables from the data. These properties make VAEs a powerful and flexible tool for solving complex Bayesian inference problems, such as those in NLME modeling. Initial applications of VAEs in PMX have been presented for PK modeling [15–17].

In this paper, we present an adapted VAE for NLME modeling. First, the VAE is configured to reflect the structure of NLME frameworks, incorporating a population prior and structuring the latent space around individual model parameters. Second, the VAE is augmented to perform automated covariate selection. This is achieved by estimating a full covariate effect matrix, enabling the evaluation of all possible covariate–parameter combinations in a single step. Delattre et al. [11] proposed using the BICc for covariate selection, as it properly accounts for the random effects structure. To evaluate all possible covariate–parameter combinations at once, our adapted VAE employs a BICc-based criterion as the loss function, enabling efficient covariate selection in one run.

In conclusion, our adapted VAE simultaneously estimates model parameters, that is, fixed and random effects, and performs automated covariate selection within the NLME framework. The proposed combined AI-PMX approach is demonstrated using two case studies, the well-known theophylline dataset [18], and a more complex neonatal weight progression dataset [19, 20]. The presented VAE, implemented in Python, represents the first generative AI-based optimization method for NLME modeling capable of completing both model parameter estimation and covariate selection in one single run.

2 | Methods

This Section first provides the theoretical basis of the NLME framework (Section 2.1), followed by a description of how VAEs can be integrated within NLME frameworks (Section 2.2). Finally, we present our augmented VAE approach designed for automated covariate selection (Section 2.3).

2.1 | Nonlinear Mixed Effects Model

Consider a dataset of N individuals, where the data for each individual i is given by

$$\mathcal{D}_i = \left(t_{ij}, x_{ij}, c_i; 1 \le j \le n_i\right)$$

for $1 \le i \le N$. Here, t_{ij} represents the time of the j-th observation $x_{ij} \in \mathbb{R}$, with $n_i \in \mathbb{N}$ denoting the total number of observations for individual i. The vector $c_i \in \mathbb{R}^{n_c}$ contains $n_c \in \mathbb{N}$ individual covariate (relations). Furthermore, $n_z \in \mathbb{N}$ denote the number of individual parameters applied in the model. For each individual i, the structural model reads

$$\frac{d}{dt} y_i(t) = f\left(t, y_i(t), \zeta_i\right) \quad \text{for } t \in (0, T], \quad y_i(0) = y_0\left(\zeta_i\right), \tag{1}$$

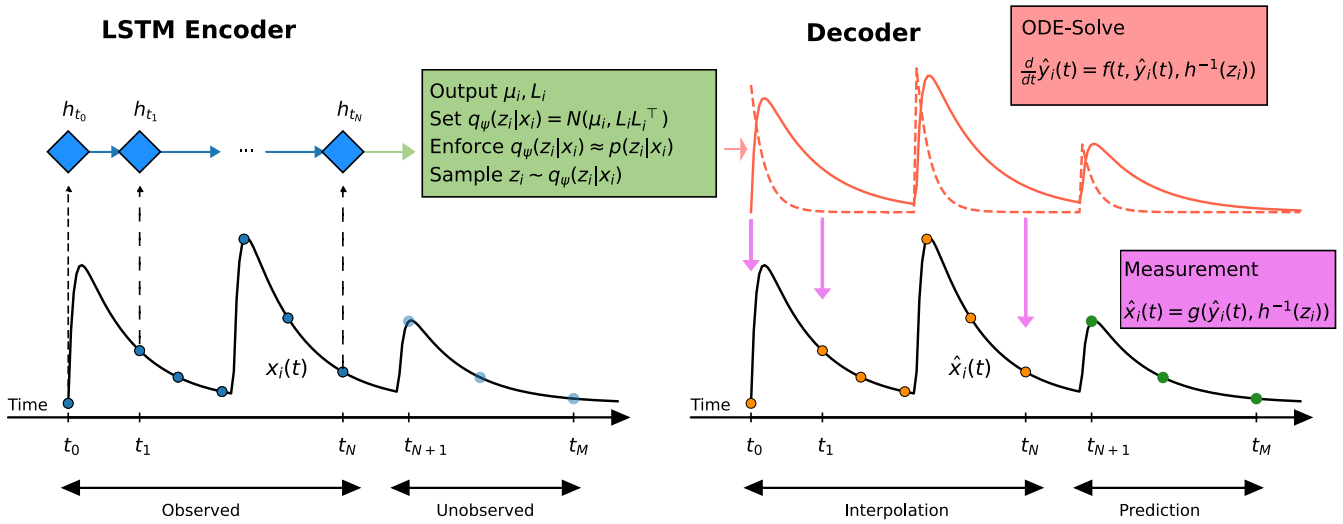

FIGURE 1 | The Variational Autoencoder (VAE) framework. In the left panel, the LSTM encoder infers the latent variable z_i from the measurements x_i. The encoder posterior distribution $q_\psi(z_i|x_i)$ is enforced to match the true posterior distribution $p(z_i|x_i)$. In the right panel, the decoder reconstructs the measurements $\hat{x}_i$ from the latent variable z_i.

where f represents the mechanism of the model. The error model satisfies

$$x_{ij} = g(t_{ij}, y_i(t_{ij}), \zeta_i) + \epsilon_{ij} \quad \text{for } j = 1, \ldots, n_i, \tag{2}$$

with an output function g. The residual errors are assumed to follow a normal distribution with variance a^2, that is, $\epsilon_{ij} \sim \mathcal{N}(0, a^2)$. For the sake of readability, we choose a constant error model, however, the framework is flexible and allows any error model specification. The variable ζ_i is given by the individual model (prior distribution)

$$\begin{aligned} h(\zeta_i) &= z_i \quad \text{for } i = 1, \ldots, N, \\ z_i &= z_{pop} + \beta c_i + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \Omega), \quad \Omega = \text{diag}\left(\omega_1^2, \ldots, \omega_{n_z}^2\right), \end{aligned} \tag{3}$$

where $z_{pop} \in \mathbb{R}^{n_z}$ denotes the fixed effects, $\beta = (\beta_1, \ldots, \beta_{n_c}) \in \mathbb{R}^{n_z \times n_c}$ is a matrix of possible covariate effect parameters, and h denotes a smooth invertible transformation to adjust the prior to more general non-Gaussian distributions. The population parameters of the model Equations (1–3) are given by $\theta = (z_{\text{pop}}, \beta, \Omega, a)$. The likelihood function of the observations $x = (x_{ij}; 1 \le i \le N, 1 \le j \le n_i)$ is expressed as

$$p(x) = p(x; \theta) = \int p(x, z) dz = \int p(x|z) p(z) dz.$$

We denote the corresponding log-likelihood by

$$\mathcal{LL}(\theta) = \mathcal{LL}(x; \theta) = \log p(x; \theta).$$

The objective is to maximize the log-likelihood function $\mathcal{LL}(\theta)$ with respect to the population parameters θ, that is,

$$\max_\theta \mathcal{LL}(\theta) \tag{4}$$

referred to as marginalized likelihood problem. A commonly used solution for this problem is the SAEM algorithm [4] available in Monolix 2024 [21]. In 2013, Kingma and Welling introduced in VAEs [14, 22] as a powerful AI-based alternative that offers an efficient way to approximate and optimize marginalized likelihoods. This motivates our application of VAEs in NLME modeling to solve Equation (4).

2.2 | VAEs for NLME Modeling

In this subsection, we design a VAE to solve marginalized likelihood problems Equation (4) specifically adapted to the structure of NLME frameworks. The VAE architecture consists of two main components, encoder and decoder. The encoder infers the latent variables z_i from the observed data $x_i = (x_{ij}; 1 \le j \le n_i)$, while the decoder reconstructs the data based on the latent representation. Training is performed by maximizing the Evidence Lower Bound (ELBO) of the marginal log-likelihood $\mathcal{LL}(\theta)$. An overview of the VAE architecture is illustrated in Figure 1.

2.2.1 | Encoder

The objective of the encoder is to propose a smooth, parametrized approximation $q_\psi(z_i|x_i)$ of the posterior distribution $p(z_i|x_i)$, which extracts the individual model parameters z_i from the observations x_i for every individual i. The distribution $q_\psi(z_i|x_i)$ is modeled using a Long Short-Term Memory (LSTM) neural network architecture (see [23], Ch. 10.10) with subject-independent network parameters ψ

$$\begin{aligned} (\mu_i, L_i) &= \text{EncoderLSTM}_\psi(\mathcal{D}_i) \quad \text{for } i = 1, \ldots, N, \\ q_\psi(z_i|x_i) &= \mathcal{N}(z_i; \mu_i, \Sigma_i) \end{aligned} \tag{5}$$

with $\Sigma_i = L_i L_i^\top$, where L_i a lower-triangular matrix with positive diagonal entries. The approximated posterior distribution $q_\psi(z_i|x_i) = \mathcal{N}(z_i; \mu_i, \Sigma_i)$ is utilized to sample the latent variable $z_i \in \mathbb{R}^{n_z}$, that is,

$$z_i = \mu_i + L_i \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, I), \tag{6}$$

or equivalently, $z_i \sim q_\psi(z_i|x_i)$.

2.2.2 | Decoder

The decoder utilizes the sampled latent variable z_i in the structural model Equation (1) to generate the model prediction $\hat{y}_i(t)$. From this, the predicted observation $\hat{x}_i$ is reconstructed according to the error model Equation (2).

2.2.3 | Loss Function and Training

An ELBO loss function, adapted to the NLME framework, is derived to enable VAE training. The error in the measurements is incorporated into the ELBO loss function using the log-likelihood, assuming $x_{ij} \mid z_i \sim \mathcal{N}\left(\hat{x}_{ij}, a^2\right)$. In case of a one-dimensional observation, this leads to

$$\log p(x|z) = \sum_{i=1}^{N} \log p\left(x_i|z_i\right) = -\frac{N_{\text{tot}}}{2}\left(\log(2\pi) + \log\left(a^2\right)\right) - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{n_i}\left(\frac{x_{ij} - \hat{x}_{ij}\left(z_i\right)}{a}\right)^2$$

with $N_{\text{tot}} = \sum_{i=1}^{N} n_i$. To ensure that the sampled latent variables z_i (see Equation (6)) fit into the population, that is,

$$z_i = z_{\text{pop}} + \beta c_i + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \Omega), \tag{7}$$

the ELBO loss function includes a Kullback–Leibler divergence term (see [23], Ch. 3.13)

$$D_{KL}\left(q_\psi\left(z_i|x_i\right) \parallel p\left(z_i\right)\right) = D_{KL}\left(\mathcal{N}\left(\mu_i, \Sigma_i\right) \parallel \mathcal{N}\left(z_{pop} + \beta c_i, \Omega\right)\right).$$

Hence, the ELBO loss function with the population parameter $\theta = \left(z_{\text{pop}}, \beta, \Omega, a\right)$ for the marginalized likelihood problem Equation (4) reads

$$\mathcal{L}_\psi(x;\theta) = \sum_{i=1}^{N} \mathbb{E}_{z_i \sim q_\psi(\cdot|x_i)}\left[\log p\left(x_i|z_i\right)\right] - \sum_{i=1}^{N} D_{KL}\left(q_\psi\left(z_i|x_i\right) \parallel p\left(z_i\right)\right). \tag{8}$$

The ELBO loss function is a lower bound of the log-likelihood function, that is, $\mathcal{L}_\psi(x;\theta) \leq \mathcal{LL}(x)$. More precisely, the relation

$$\mathcal{L}_\psi(x;\theta) = \mathcal{LL}(x;\theta) - D_{KL}\left(q_\psi(z|x) \parallel p(z|x)\right) \tag{9}$$

holds, where $p(z|x) = \prod_{i=1}^{N} p\left(z_i|x_i\right)$ and $q_\psi(z|x) = \prod_{i=1}^{N} q_\psi\left(z_i|x_i\right)$. Hence, the marginalized likelihood problem Equation (4) can be solved by maximizing the ELBO loss function in Equation (8) with respect to the population parameters $\theta = \left(z_{\text{pop}}, \beta, \Omega, a\right)$ and the network parameters ψ. Population parameters θ_k are updated done according to the global uniquely solvable maximization step

$$\theta^{(k+1)} = \arg\max_\theta \mathcal{L}_{\psi^{(k)}}(x;\theta). \tag{10}$$

2.2.4 | Implementation Details

VAEs are potentially sensitive to architectural hyperparameters (e.g., number of layers, hidden dimension) and parameter initialization, which can affect performance and stability. They can also get trapped in undesirable local minima of Equation (4) during training [14]. Thus, inspired by the SAEM algorithm, we incorporate a burn-in phase consisting of a priori NLME investigations and Kullback–Leibler annealing [24], effectively pretraining the LSTM encoder. This enhances stability and enables the VAE to escape undesired local minima.

2.2.5 | Accuracy and Prediction Quality of Solutions

The quality of the VAE solutions with respect to the likelihood function depends on the size of the gap between the ELBO loss function $\mathcal{L}_\psi(x;\theta)$ and the log-likelihood $\mathcal{LL}(\theta)$, which is called the tightness of the bound. According to Equation (9), this gap corresponds to the Kullback–Leibler divergence term between $q_\psi(z|x)$ and $p(z|x)$. This means, the better the encoder distribution $q_\psi(z|x)$ approximates the true posterior $p(z|x)$, the smaller the gap and the better the likelihood value of the VAE solutions.

Classical methods, such as FOCE, linearize the model by using a first-order Taylor expansion, which results in Gaussian posteriors. However, SAEM-based methods use Markov Chain Monte Carlo (MCMC) iterations to approximate the posterior. MCMC-based approximations are quite accurate, but computationally costly, especially for large populations. In contrast to classical methods, the VAE derives the encoder distribution $q_\psi(z|x)$ directly from the data, using a LSTM neural network smoothly parametrized by its network parameters ψ. Since the network parameters ψ are subject-independent, the VAE potentially improves data structure readout and prediction quality. Using a trained encoder $q_\psi(z|x)$, one can infer the full posterior distribution of the latent variables z, as well as perform maximum a posteriori (MAP) estimations, both by simple evaluations of the LSTM neural network without calculating any integrals or solving optimization problems.

2.3 | Augmented VAEs for Covariate Selection

Based on the VAE concept, we propose a novel approach for covariate selection. To assess the quality of a covariate model, it is not sufficient to maximize the log-likelihood $\mathcal{LL}(\theta)$ alone. Instead, a model selection criterion that balances goodness-of-fit and model complexity is required. For this purpose, we aim to minimize the BICc [11].

$$\text{BICc} = -2\mathcal{LL}(\theta) + \log(N) \cdot \dim\left(\theta_R\right) + \log\left(N_{\text{tot}}\right) \cdot \dim\left(\theta_F\right), \tag{11}$$

where $\dim\left(\theta_R\right)$ and $\dim\left(\theta_F\right)$ denote the number of the random and fixed effects, respectively.

Given covariate vectors c_i for individuals $i = 1, \ldots, N$, the augmented VAE estimates the full covariate effect matrix

$$\beta = \begin{pmatrix} \beta_1 & \cdots & \beta_{n_c} \end{pmatrix} \in \mathbb{R}^{n_z \times n_c},$$

where each column $\beta_j \in \mathbb{R}^{n_z}$, for $j \in \{1, \ldots, n_c\}$, describes how the j-th covariate affects the model parameters. If an entire column β_j is estimated to be zero, the corresponding j-th covariate is deemed irrelevant and excluded from the final model. Conversely,

if any entry in the column is non-zero, that is, $\beta_{jk} \neq 0$ for some $k \in \{1, \ldots, n_z\}$, the j-th covariate is expected to have an effect on the parameter z_k. Consequently, estimating the covariate effect matrix β corresponds to selecting the most appropriate combination of all possible covariate-parameter combinations.

2.3.1 | BICc-ELBO Loss Function

For the augmented VAE, we introduce the BICc-ELBO loss function that incorporates the appropriate penalty term for any possible covariate-parameter combination. Based on the structure of the BICc function (compare Equation (11)) the BICc-ELBO loss function reads

$$\mathcal{L}_\psi^{BICc}(x, \theta) = -2\mathcal{L}_\psi(x; \theta) + \log(N)\|\beta\|_0 + C \tag{12}$$

with C being a constant that does not affect the optimization. In Equation (12), the term $\|\beta\|_0$ denotes the L_0-(pseudo) norm of β, that is, the number of non-zero elements in the covariate effect matrix β, which coincides with the number of relevant covariate effects. Minimizing the BICc-ELBO loss function Equation (12) selects the best covariate combination with respect to BICc from all covariate-parameter combinations.

The BICc-ELBO is an upper bound for BICc (BICc $\leq \mathcal{L}_\psi^{BICc}(x, \theta)$), so we can optimize the BICc by minimizing the BICc-ELBO. Minimizing the BICc-ELBO loss function Equation (12) is similar to maximizing the ELBO loss function Equation (8), with a different update of the fixed effects (z_{pop}, β), while updates for Ω and a remain equivalent. For the ELBO loss function, the parameters $\theta = (z_{\text{pop}}, \beta, \Omega, a)$ are updated according to Equation (10). In contrast, the BICc-ELBO loss function requires solving the optimization problem

$$(z_{\text{pop}}, \beta)^{(k+1)} = \arg\min_{(z_{\text{pop}}, \beta)} \mathcal{L}_{\psi^{(k)}}^{BICc}(x; \theta) = \arg\min_{(z_{\text{pop}}, \beta)} -2\mathcal{L}_{\psi^{(k)}}(x; \theta) + \log(N)\|\beta\|_0, \tag{13}$$

which is NP-hard [25]. However, the quadratic structure of $\mathcal{L}_\psi$ in (z_{pop}, β) allows to efficiently compute global solutions of (13) using Mixed Integer Quadratic Programming (MIQP), even with hundreds of covariates [26]. Update Equation (13) allows to simultaneously estimate the relevant covariate effects (i.e., determine β) and the population parameters θ. Since the optimization problems defined in Equations (10) and (13) admit unique global solutions, the stability and convergence properties of the VAE are preserved when replacing the loss function $\mathcal{L}_\psi$ with $\mathcal{L}_\psi^{BICc}$.

3 | Results

Two real-world case studies are applied to assess the performance of the proposed VAE approach. The first case study focuses on the pharmacokinetics of theophylline [18]. The second investigates the weight progression of neonates during the first 7 days of life [19, 20].

For each of the two case studies, we describe the analysis dataset, the applied model, and the numerical results. For parameter estimation, we compare the VAE results with those obtained using the SAEM algorithm [4] for fixed covariates. Additionally, we benchmark our automated covariate selection against the SAEM-based methods SAMBA, COSSAC, and SCM using the BICc criterion computed via importance sampling.

Implementation is done in Python 3.13 using PyTorch [27], utilizing the package TorchODE [28] for parallelized ordinary differential equation (ODE) solving. The VAE employs a single LSTM layer, with the hidden dimensions depending on the number of individuals and model parameters. It is trained using the Adam optimizer [29], which automatically adapts the learning rate for each parameter. For reproducibility, the random seed was fixed. All learning parameters are provided in the Supporting Information S1. Computations were performed on a MacBook Pro equipped with an Apple M4 Pro chip (12-core CPU, 24 GB RAM). VAE examples are available on GitHub (https://github.com/janrohleff/vae_nlme).

3.1 | Case Study 1—Theophylline Pharmacokinetics

3.1.1 | Data

The dataset consists of theophylline concentration measurements in the blood plasma of $N = 12$ patients. All patients received a theophylline loading dose at time $t = 0$. Concentration measurements $C_i(t_{ij})$ were taken at various time points t_{ij} for each individual $i = 1, \ldots, N$ and measurement $j = 1, \ldots, n_i$. Additionally, each individual's baseline weight w_i and sex sex_i were recorded as covariates. The available data of individual i is represented by

$$\mathcal{D}_{ij} = \begin{pmatrix} t_{ij} \\ C_i(t_{ij}) \\ w_i \\ sex_i \end{pmatrix} \in \mathbb{R}^4 \quad \text{for } i = 1, \ldots, N \text{ and } j = 1, \ldots, n_i.$$

3.1.2 | Model Description

Theophylline concentrations are characterized by a one-compartment model with first-order absorption and elimination [18] given by

$$\begin{aligned} \frac{d}{dt}Abs(t) &= -k_a Abs(t), & Abs(0) &= D, \\ \frac{d}{dt}A(t) &= k_a Abs(t) - k_e A(t), & A(0) &= 0, \end{aligned} \tag{14}$$

where $Abs(t)$ and $A(t)$ denote the amount of theophylline in the absorption and central compartment. The parameters k_a and k_e are the absorption and elimination rate constants. D is the amount of drug given to the individual at time $t = 0$. The observed concentration in the central compartment, given by $g(A(t), V) = A(t)/V$, is

$$C(t) = \frac{A(t)}{V}.$$

We assume that the parameters $(k_{a,i}, k_{e,i}, V_i) \in \mathbb{R}^3$ follow a log-normal distribution, that is, $h(x) = \log(x)$ and

$$\log(k_{a,i}) = \log(k_{a,pop}) + \beta_{k_a} c_i + \eta_{k_a,i}, \quad \eta_{k_a,i} \sim \mathcal{N}\left(0, \omega^2_{k_a}\right),$$
$$\log(k_{e,i}) = \log(k_{e,pop}) + \beta_{k_e} c_i + \eta_{k_e,i}, \quad \eta_{k_e,i} \sim \mathcal{N}\left(0, \omega^2_{k_e}\right),$$
$$\log(V_i) = \log(V_{pop}) + \beta_V c_i + \eta_{V,i}, \quad \eta_{V,i} \sim \mathcal{N}\left(0, \omega^2_V\right).$$

Note that any invertible transformation h, as shown in Equation (3), can be chosen to model alternative parameter distributions. The covariate $c_i = \left(c_i^w, c_i^{sex}\right) \in \mathbb{R}^2$ is defined as

$$c_i^w = \log\left(\frac{w_i}{w_{pop}}\right) \quad \text{and} \quad c_i^{sex} = \begin{cases} 1 & \text{for individual } i \text{ is a male} \\ 0 & \text{for individual } i \text{ is a female} \end{cases},$$

and w_{pop} denotes the mean weight over the N individuals. The full covariate effect matrix reads

$$\beta = \begin{pmatrix} \beta_{k_a} \\ \beta_{k_e} \\ \beta_V \end{pmatrix} = \begin{pmatrix} \beta^w_{k_a} & \beta^{sex}_{k_a} \\ \beta^w_{k_e} & \beta^{sex}_{k_e} \\ \beta^w_V & \beta^{sex}_V \end{pmatrix} \in \mathbb{R}^{3\times 2}.$$

3.1.3 | Numerical Results

Running the VAE results in the final optimal covariate effect matrix β^* with non-zero covariate effects β^w_V, $\beta^w_{k_a}$, and all other covariate effects set to zero (see Figure 2). The VAE converges in up to 250 iterations. It is important to note that the VAE selection process follows the standard selection rules. In particular, the addition of a covariate effect at a given iteration is accompanied by a simultaneous reduction in the corresponding variance. At iteration 30, the uptake of β^w_V and $\beta^w_{k_a}$ coincides with a decrease in ω_V and ω_{k_a}, respectively (see Figure 2). Detailed results are summarized in Table 1. In this example, the estimated population parameters and log-likelihood values are similar across all methods, apart from some stochastic noise. The run is repeated in the Supporting Information S2 with different initial seeds and varying hidden dimensions in the LSTM layer. The results show that the VAE consistently computes the same solution up to negligible differences.

Looking at the covariate model, COSSAC and SCM select the same optimal covariate effect matrix β^* as the VAE, whereas SAMBA suggests only the covariate effect β^w_V. The BICc

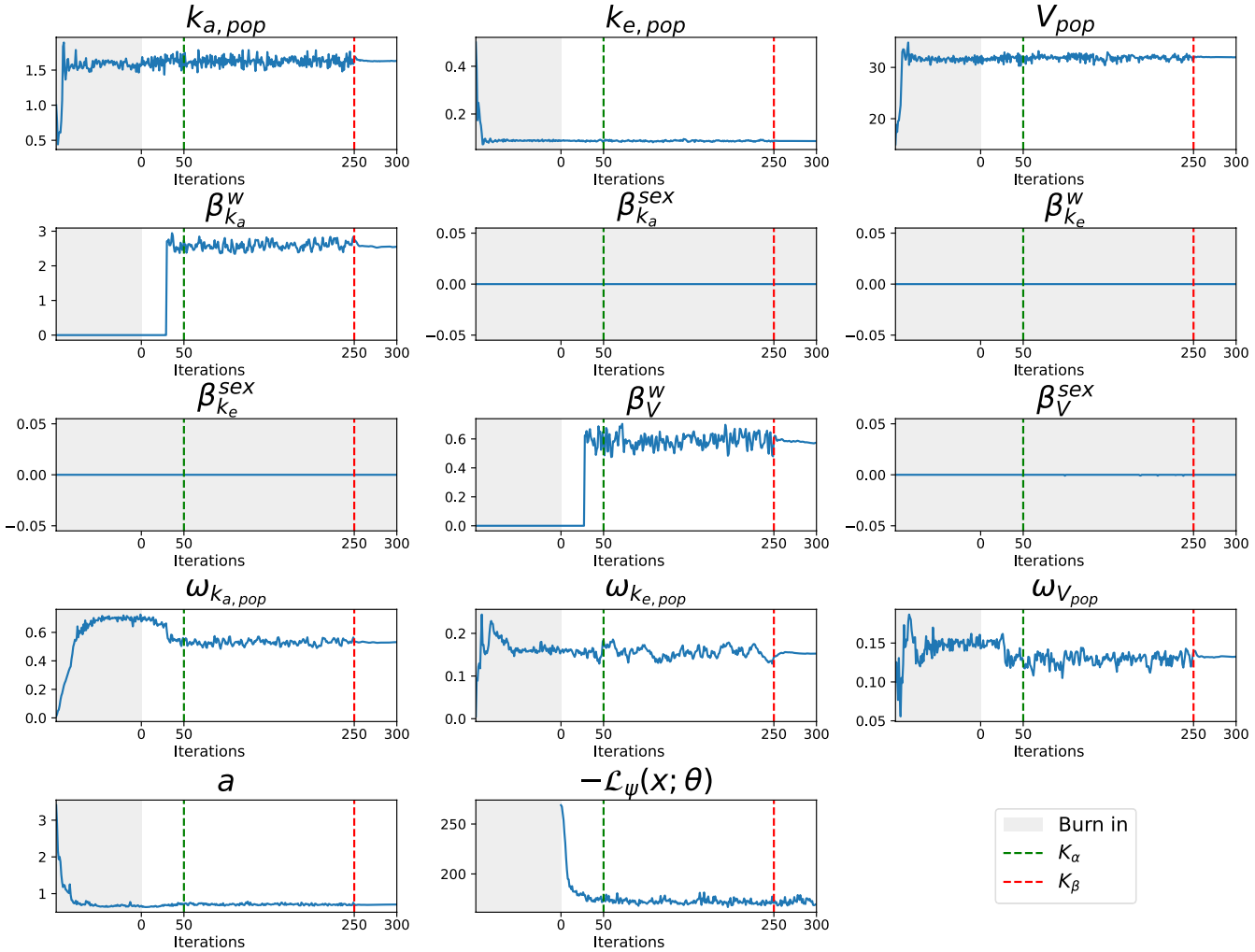

FIGURE 2 | Theophylline example. Convergence of the population parameters θ (including covariate effects β) for the theophylline data set. The VAE starts with a burn-in phase of 100-iteration (gray area), followed by the main training phase. The dashed green line marks the point where the Kullback–Leibler divergence term is fully annealed; the dashed red line marks the smoothing phase of the maximization step. Covariate effects excluded from the final model (i.e., converging to zero) are shown on a light gray background.

TABLE 1 | Theophylline example.

	VAE		COSSAC		SAMBA		SCM	
Fixed effects								
$k_{a,pop}$	1.63		1.60		1.64		1.60	
$k_{e,pop}$	0.086		0.087		0.084		0.087	
V_{pop}	31.97		31.82		32.21		31.82	
Covariates								
$\beta^{w}_{k_a}$	2.55		2.58		2.53		2.58	
$\beta^{sex}_{k_a}$	—		—		—		—	
$\beta^{w}_{k_e}$	—		—		—		—	
$\beta^{sex}_{k_e}$	—		—		—		—	
β^{w}_{V}	0.57		0.57		—		0.57	
β^{sex}_{V}	—		—		—		—	
Standard deviation								
ω_{k_a}	0.53		0.54		0.55		0.54	
ω_{k_e}	0.15		0.15		0.15		0.15	
ω_{V}	0.13		0.13		0.14		0.13	
Error model								
a	0.71		0.73		0.73		0.73	
Stat. criteria								
	Lin	**IS**	**Lin**	**IS**	**Lin**	**IS**	**Lin**	**IS**
$-2\mathcal{LL}$	331.1	332.2	330.4	331.9	333.4	335.1	330.4	331.9
BICc	362.7	363.8	362.0	363.5	362.5	364.2	362.0	363.5
Runs	1		3		2		16	

Note: Comparison of model parameters estimated by VAE, COSSAC, SAMBA and SCM. The likelihood function $-2\mathcal{LL}$ and the BICc criterion are computed by a linearization method (Lin) and by importance sampling (IS). The number of runs is given in the last row.

criterion indicates that the covariate model selected by the VAE, COSSAC and SCM is slightly better in terms of model quality.

The VAE stands out in terms of efficiency, requiring only one single run to simultaneously estimate population model parameters and select covariates. SAMBA and COSSAC also perform well but require multiple runs (two for SAMBA and three for COSSAC), each involving optimization and computing of the log-likelihood function for a fixed covariate model. SCM is the slowest, requiring 16 runs to reach similar results.

Some remarks: (i) In this example, a VAE population fit with fixed covariates, that is, estimating population parameters only, takes 5.8 s. If we include covariate selection, the total time is 6.2 s. This means, for this small example, covariate selection only adds about 6% CPU time. (ii) Multiple dosing can be handled by the VAE. We tested the same model under a multiple dosing regimen, administering 10 doses every 10 h. The results are consistent with those obtained from the single-dose setting. The corresponding experiment and implementation details are available on GitHub.

3.2 | Case Study 2—Neonatal Weight Progression

3.2.1 | Data

The aim was to develop a model to characterize weight progression over the first 7 days of life using data from $N = 2425$ neonates [20]. Each neonate's weight $W_i(t_{ij})$ was measured at various time points t_{ij}, where $i = 1, \ldots, N$ and $j = 1, \ldots, n_i$. The clinically relevant covariates include sex (*sex*), mode of delivery (*DelM*), gestational age (*GA*), maternal age (*Mage*), and parity $(Para_2)$. The available data is represented as

$$\mathcal{D}_{ij} = \begin{pmatrix} t_{ij} \\ W_i(t_{ij}) \\ sex_i \\ DelM_i \\ GA_i \\ Mage_i \\ Para_{2i} \end{pmatrix} \in \mathbb{R}^7 \quad \text{for } i = 1, \ldots, N \text{ and } j = 1, \ldots, n_i .$$

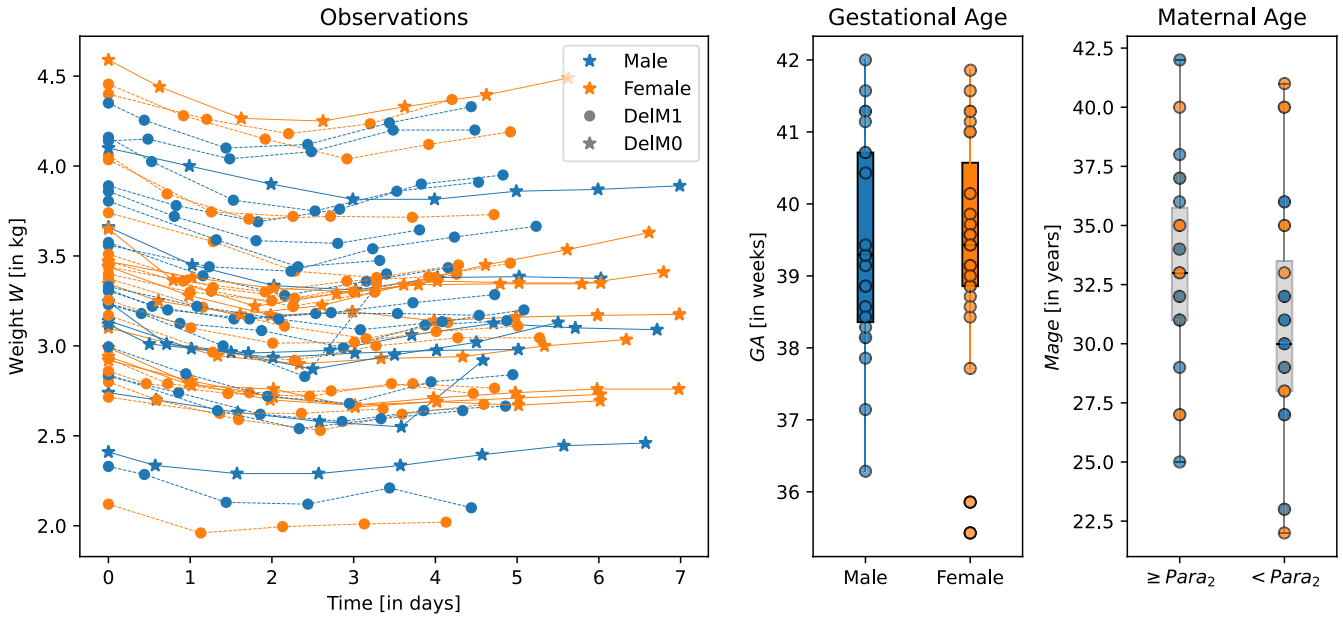

FIGURE 3 | Neonatal weight progression example. The neonate weight data set with clinical covariates sex (*sex*), mode of delivery (*DelM*), gestational age (*GA*), maternal age (*Mage*), and parity ($Para_2$).

The dataset for the first 50 neonates is visualized in Figure 3.

3.2.2 | Model Description

We consider a slightly updated version of the structural model from [20], in which the original discontinuity in the lag time has been smoothed out, which is more realistic from a physiological perspective and ensures differentiability. This modification enables gradient-based training of the VAE's encoder. Maturation effects are incorporated with postnatal age-dependent zero-order production and first-order elimination terms. The structural model reads

$$\frac{d}{dt}W(t) = k_{prod}(t) - k_{el}(t)W(t), \quad W(0) = W_0, \tag{15}$$

where

$$k_{prod}(t) = \frac{k_{in}}{1+\exp\big(-2\big(t-T_{lag}\big)\big)} \quad \text{and} \quad k_{el}(t) = k_{out}\left(1-\frac{t}{T_{50}+t}\right).$$

The production term $k_{prod}(t)$ is modeled by a modified sigmoid function and can be interpreted as a smooth delay of T_{lag} days. The structural model contains five parameters $\big(W_0, k_{in}, T_{lag}, k_{out}, T_{50}\big) \in \mathbb{R}^5$, each with inter-individual variability and assumed to follow a log-normal distribution. Covariates *sex*, *DelM*, $Para_2$ are categorical, and *GA* and *Mage* are continuous and centered around their mean value as in the previous example, that is, $c_i^{Mage} = log\big(Mage_i / Mage_{pop}\big)$ and $c_i^{GA} = log\big(GA_i / GA_{pop}\big)$. Five parameters with five covariates result in 2^{25} possible covariate-parameter combinations.

3.2.3 | Numerical Results

First, we perform a population fit of model Equation (15) without including any covariates to evaluate how well the Gaussian approximation captures the true posterior. Detailed results are provided in Supporting Information S3. The VAE achieves an objective function value of $-2\mathcal{LL} = 147682$, slightly higher compared to $-2\mathcal{LL} = 147468$ obtained by the SAEM, which corresponds to a small difference of less than 0.2%. The reason is that, in this example, the MCMC-based approximation used by SAEM matches the true posterior $p(z|x)$ slightly better than the Gaussian like approximation $q_\psi(z|x)$ of the VAE, see Equation (9). Despite this small difference in the approximation of the posterior distribution, both SAEM- and VAE-based methods are well suited to estimate the population parameters.

Second, we include covariate selection. The results are presented in Table 2, and the selected covariates are in Figure 4. The convergence of the covariate effects is shown in Figure 5.

After covariate selection, the small gap in the log-likelihood between the VAE and SAEM remains (see Table 2). Compared to its counterparts without covariates, the automatically selected covariates by the VAE in one run shows strong improvement like those of SAMBA, COSSAC or SCM. Combining MCMC posterior with fixed, VAE-selected covariates closes the gap. SAMBA, COSSAC, and SCM need 2, 33, and 244 runs, respectively, to select similar good covariates. The VAE reaches this result with just one run. The CPU time of this VAE run with covariate selection is 26% longer than without, which is still much faster than two runs (by SAMBA). In summary, the VAE matches the other methods in terms of covariate selection quality, while outperforming them in computational efficiency.

4 | Discussion

NLME modeling is an essential tool for key decisions in MIDD. However, traditional NLME modeling has its limitations

TABLE 2 | Neonatal weight progression example.

Log-likelihood and BICc values										
	VAE		**MCMC**		**COSSAC**		**SAMBA**		**SCM**	
	Lin	**IS**	**Lin**	**IS**	**Lin**	**IS**	**Lin**	**IS**	**Lin**	**IS**
$-2\mathcal{LL}$	146370	146406	146120	146154	146086	146132	146121	146177	146116	146123
BICc	146566	146602	146316	146351	146283	146329	146318	146374	**146296**	**146305**
Runs	**1**		**1**		33		2		244	

Note: Statistic criteria for the VAE, MCMC with VAE-fixed covariate model, COSSAC, SAMBA, and SCM. Left: Likelihood computed by linearization method (Lin). Right: Likelihood computed by importance sampling method (IS). Bold values indicate the lowest value.

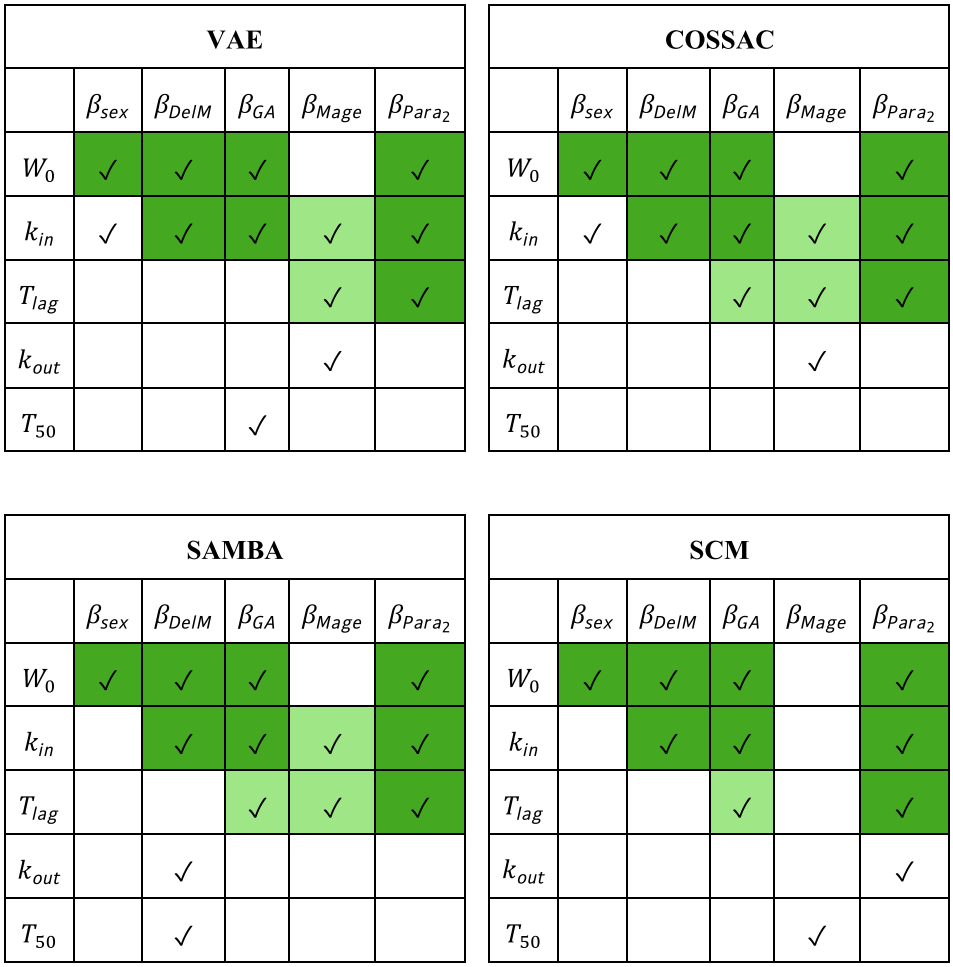

FIGURE 4 | Neonatal weight progression example. Selected covariates for the VAE, COSSAC, SAMBA and SCM models. *Dark green* cells indicate covariates selected by all four methods, while *light green* cells indicate covariates selected by three out of four methods.

regarding efficacy, especially when it comes to covariate selection, which is an iterative and time-consuming process.

To address this, we have established VAEs as a powerful AI-based framework to efficiently solve general NLME modeling problems in PMX. VAEs incorporate the flexibility of AI-based techniques into solving marginalized likelihood problems (see Equation (4)).

All NLME modeling approaches need approximations of the true posterior distribution $p(z|x)$. Unlike SAEM's MCMC-based inference, the VAE approximates the posterior distribution with a smooth Gaussian approximation $q_\psi(z|x)$. This approximation is obtained via a neural network encoder trained directly on the data.

We highlight two aspects of this approximation: (*i*) smoothness and (*ii*) Gaussianity.

i. The smooth posterior approximation assumes smooth model structures in the individual parameters. This assumption of smoothness aligns with biological intuition, supports gradient-based optimization, and enables efficient ELBO convergence. Moreover, future tasks in

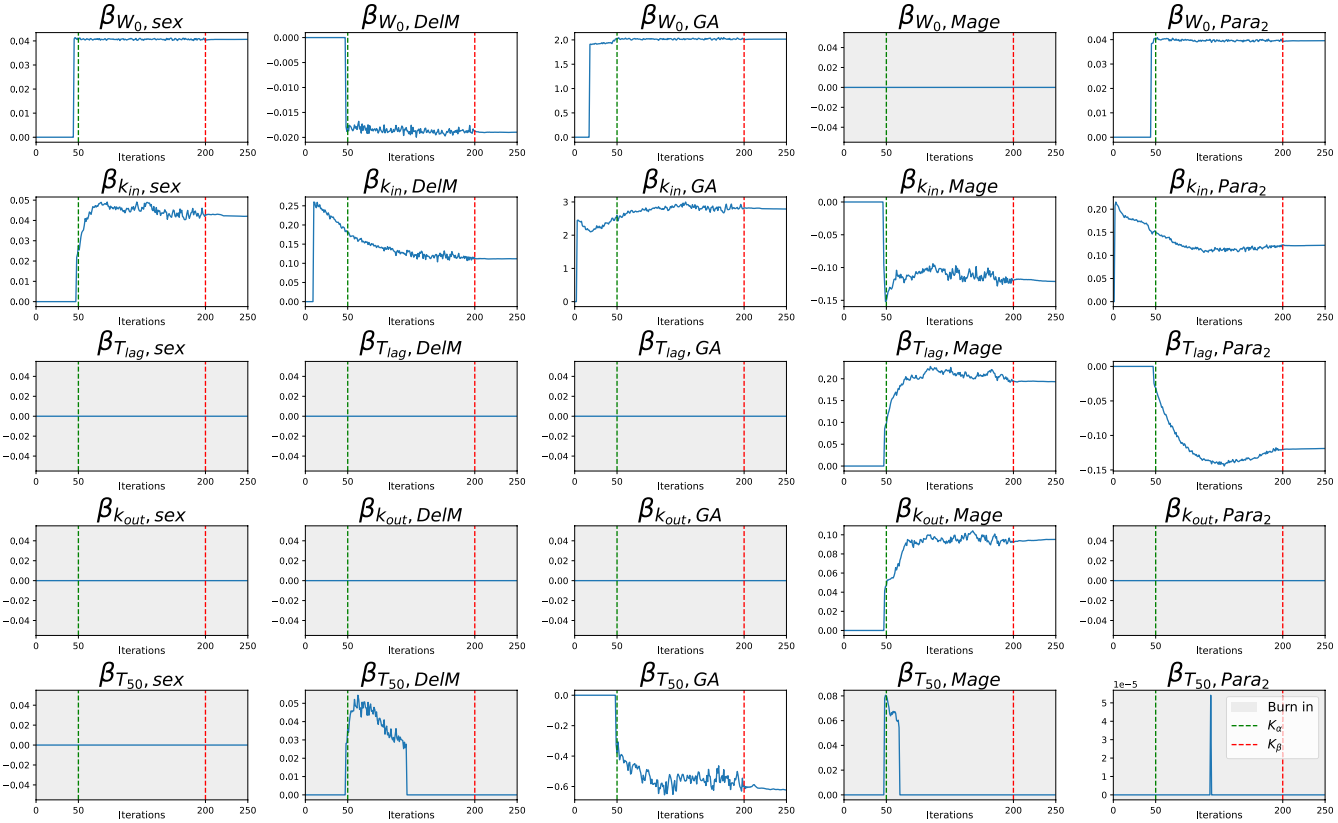

FIGURE 5 | Neonatal weight progression example. Convergence of the covariate effects β for all possible covariate-parameter combinations. Covariate effects that converge to zero are excluded from the final model are highlighted with a light gray background.

automated modeling (discussed later) benefit from this smooth structure or even rely on it by design.

ii. Gaussian approximations make inference faster and computationally cheaper. While SAEM requires repeated ODE solutions for each MCMC sample, the VAE only requires a single encoder pass per sample. Once trained, it can instantly predict posteriors for new individuals. In our experiments, we found that quite often Gaussian approximation closely matched the true posterior, consistent with observations in Janssen et al. [17]. Nevertheless, the VAE framework should ideally be extended in the future to capture non-Gaussian posterior distributions, potentially improving the accuracy of population parameter estimates in cases where the true posterior is not well approximated by a Gaussian. One possible direction of research are more flexible approximations, such as normalizing flows [30, 31] or mixtures of Gaussians.

The presented VAE-NLME framework offers a new automated approach to covariate selection by optimizing the adapted information criterion BICc-ELBO. This enables simultaneous estimation of population parameters and covariate selection in one single run, that is, one run is all you need.

SAMBA, COSSAC, as well as SCM require multiple runs, as they iteratively select a covariate model and compute the corresponding likelihood, testing potential covariate-parameter relationships. The neonatal weight progression example demonstrates that COSSAC and SCM require a large number of runs, resulting in high computation costs. This is not the case for the VAE. Due to its design and the efficiency of the MIQP solver, the VAE can handle NLME frameworks with hundreds of covariates and parameters. Consequently, the VAE stands out in terms of efficiency and its ability to handle a large number of covariates, while the quality of the selected covariates remains comparable across methods.

Thanks to the design of our VAE-NLME framework, the model is highly robust. It does not require extensive hyperparameter tuning, and the burn-in with Kullback–Leibler annealing enables the VAE to consistently find good solutions in our experiments.

A major computational bottleneck in all NLME frameworks is solving ODEs. While optimization is efficient, ODE evaluation dominates runtime. SAEM-based solutions need more ODE evaluations than other NLME frameworks due to their use of MCMC posterior approximations, compared to Janssen et al. [17]. Hence, faster ODE solvers would benefit all approaches.

The VAE is unlocking the potential of combining AI-based methods and PMX by offering great flexibility and enabling the integration of advanced AI techniques. Automated covariate selection is one example. Another example could be modification of the encoder to output continuous individual parameters, making it ideal for NLME modeling problems with time-dependent covariates. Furthermore, certain unknown components of the decoder can be replaced with neural networks, enabling the automated modeling of complex behaviors and offering a more

efficient approach to model development. Moreover, VAEs can be applied in PMX to integrate more general data types, such as medical images (e.g., tumor scans in oncology), as well as genomic and metabolomic data [32], as encoder inputs. The VAE paves the way to many other future applications.

In conclusion, the VAE-NLME framework represents a launch pad into a new era of PMX, where combining AI-based frameworks and PMX modeling can lead to more accurate and especially more efficient model building and assessment. The presented AI-PMX approach facilitates automated model development, advancing PMX and its applications in MIDD.

Author Contributions

J.R., F.B., G.K., and J.S. wrote the manuscript. J.R. and J.S. designed the research. J.R. and J.S. performed the research. J.R., D.B., B.S., U.N., M.P., and G.K. analyzed the data. D.B., B.S., U.N., M.P., and G.K. provided the data. J.R. developed and implemented the complete VAE codebase. All authors reviewed and approved the final manuscript.

Acknowledgments

The authors acknowledge the use of AI-based language models (OpenAI's ChatGPT) to improve readability and clarity of the text. The authors remain fully responsible for the content and interpretation of the work.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. R. Madabushi, P. Seo, L. Zhao, M. Tegenge, and H. Zhu, "Review: Role of Model-Informed Drug Development Approaches in the Lifecycle of Drug Development and Regulatory Decision-Making," *Pharmaceutical Research* 39, no. 8 (2022): 1669–1680, https://doi.org/10.1007/s11095-022-03288-w.

2. P. L. Bonate, *Pharmacokinetic-Pharmacodynamic Modeling and Simulation*, 2nd ed. (Springer, 2011).

3. M. L. Lindstrom and D. M. Bates, "Nonlinear Mixed Effects Models for Repeated Measures Data," *Biometrics* 46, no. 3 (1990): 673–687.

4. B. Delyon, M. Lavielle, and E. Moulines, "Convergence of a Stochastic Approximation Version of the EM Algorithm," *Annals of Statistics* 27, no. 1 (1999): 94–128.

5. D. Mould and R. Upton, "Basic Concepts in Population Modeling, Simulation, and Model-Based Drug Development," *CPT: Pharmacometrics & Systems Pharmacology* 1 (2012): 1–14, https://doi.org/10.1038/psp.2012.4.

6. E. N. Jonsson and M. O. Karlsson, "Automated Covariate Model Building Within NONMEM," *Pharmaceutical Research* 15 (1998): 1463–1468, https://doi.org/10.1023/A:1011970125687.

7. M. Prague and M. Lavielle, "SAMBA: A Novel Method for Fast Automatic Model Building in Nonlinear Mixed-Effects Models," *CPT: Pharmacometrics & Systems Pharmacology* 1 (2022): 161–172, https://doi.org/10.1002/psp4.12742.

8. G. Ayral, J. F. Si Abdallah, C. Magnard, and J. Chauvin, "A Novel Method Based on Unbiased Correlations Tests for Covariate Selection in Nonlinear Mixed Effects Models: The COSSAC Approach," *CPT: Pharmacometrics & Systems Pharmacology* 10 (2021): 318–329, https://doi.org/10.1002/psp4.12612.

9. H. Liang, H. Wu, and G. Zou, "A Note on Conditional AIC for Linear Mixed-Effects Models," *Biometrika* 95, no. 3 (2008): 773–778, https://doi.org/10.1093/biomet/asn023.

10. G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics* 6, no. 2 (1978): 461–464.

11. M. Delattre, M. Lavielle, and M. A. Poursat, "A Note on BIC in Mixed-Effects Models," *Electronic Journal of Statistics* 8, no. 1 (2014): 456–475.

12. M. Lavielle, *Mixed Effects Models for the Population Approach*, 1st ed. (Chapman and Hall/CRC, 2014).

13. J. Wakefield, L. Aarons, and A. Racine-Poon, *The Bayesian Approach to Population Pharmacokinetic/Pharmacodynamic Modeling* (Springer New York, 1999), 205–265.

14. D. P. Kingma and M. Welling, "An Introduction to Variational Inference," *Foundations and Trends in Machine Learning* 12, no. 4 (2019): 307–392.

15. J. Lu, K. Deng, X. Zhang, G. Liu, and Y. Guan, "Neural-ODE for Pharmacokinetics Modeling and Its Advantage to Alternative Machine Learning Models in Predicting New Dosing Regimens," *IScience* 24, no. 7 (2021): 102804.

16. A. Janssen, F. W. Leebeek, M. H. Cnossen, and R. A. Mathôt, "Deep Compartment Models: A Deep Learning Approach for the Reliable Prediction of Time-Series Data in Pharmacokinetic Modeling," *CPT: Pharmacometrics & Systems Pharmacology* 11 (2022): 934–945, https://doi.org/10.1002/psp4.12808.

17. A. Janssen, F. C. Bennis, M. H. Cnossen, et al., "Mixed Effect Estimation in Deep Compartment Models: Variational Methods Outperform First-Order Approximations," *Journal of Pharmacokinetics and Pharmacodynamics* 51, no. 6 (2024): 797–808.

18. J. C. Pinheiro and D. M. Bates, *Mixed-Effects Models in S and S-PLUS* (Springer New York, 2000).

19. M. Wilbaux, S. Kasser, S. Wellmann, et al., "Characterizing and Forecasting Individual Weight Changes in Term Neonates," *Journal of Pediatrics* 173 (2016): 101–107.

20. M. Wilbaux, S. Kasser, J. Gromann, et al., "Personalized Weight Change Prediction in the First Week of Life," *Clinical Nutrition* 38, no. 2 (2019): 689–696.

21. Monolix 2024R1, "Simulations Plus," https://doi.org/10.5281/zenodo.11401936.

22. D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv preprint*. 2013; arXiv: 1312.6114.

23. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016), https://www.deeplearningbook.org.

24. S. R. Bowman, L. Vilnis, O. Vinyals, et al., "Generating Sentences From a Continuous Space," *arXiv preprint*. 2016;arXiv: 1511.06349.

25. B. K. Natarajan, "Sparse Approximate Solutions to Linear Systems," *SIAM Journal on Computing* 24, no. 2 (1995): 227–234.

26. D. Bertsimas, A. King, and R. Mazumder, "Best Subset Selection via a Modern Optimization Lens," *Annals of Statistics* 44, no. 2 (2017): 813–852.

27. A. Paszke, S. Gross, F. Massa, et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *arXiv preprint*. 2019;arXiv:1912.01703.

28. M. Lienen and S. Günnemann, "torchode: A Parallel ODE Solver for PyTorch," *arXiv preprint*. 2023;arXiv:2210.12375.

29. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint*. 2017;arXiv: 1412.6980.

30. I. Kobyzev, S. Prince, and M. Brubaker, "Normalizing Flows: An Introduction and Review of Current Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, no. 11 (2021): 3964–3979.

31. D. P. Kingma, T. Salimans, R. Jozefowicz, et al., "Improving Variational Inference With Inverse Autoregressive Flow," *arXiv preprint*. 2017;arXiv:1606.04934.

32. C. Prakash, P. Moran, and R. Mahar, "Pharmacometabolomics: An Emerging Platform for Understanding the Pathophysiological Processes and Therapeutic Interventions," *International Journal of Pharmaceutics* 675 (2025): 125554, https://doi.org/10.1016/j.ijpharm.2025.125554.

Supporting Information

Additional supporting information can be found online in the Supporting Information section. **Data S1:** psp470129-sup-0001-Supinfo.zip.