Amortized In-Context Mixed Effect Transformer Models: A Zero-Shot Approach for Pharmacokinetics

César Ojeda University of Potsdam
Ramsés J. Sánchez University of Bonn
Wilhelm Huisinga University of Potsdam
Niklas Hartung Lamarr Institute

Abstract

Accurate dose–response forecasting under sparse sampling is central to precision pharmacotherapy. We present the Amortized In-Context Mixed-Effect Transformer (AICMET) model, a transformer-based, latent-variable framework that unifies mechanistic compartmental priors with amortized, in-context Bayesian inference. AICMET is *pre-trained* on hundreds of thousands of synthetic pharmacokinetic trajectories with Ornstein-Uhlenbeck priors over the parameters of compartment models, endowing the model with strong inductive biases and enabling *zero-shot adaptation* to new compounds. At inference time, AICMET is *conditioned on the collective context of previously profiled trial participants*, generating calibrated posterior predictions for newly enrolled patients after a few early drug concentration measurements. This capability collapses traditional model development cycles from weeks to seconds, while preserving some degree of expert modelling. Experiments across public datasets show that AICMET attains state-of-the-art predictive accuracy, and faithfully quantifies inter-patient variability — outperforming both nonlinear mixed-effects baselines and recent neural ODE variants.

Our code repository[1], pretrained model and tutorials[2] are available online.

[1] https://github.com/cesarali/aicmet
[2] https://fim4science.github.io/OpenFIM/intro.html

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

Recent advances in scalable, deep-learning techniques have brought the long-standing vision of truly personalized therapeutics — often described as the "holy grail" of precision medicine — within practical reach, positioning them as a cornerstone of modern drug development and therapeutic use. Yet a key challenge remains markedly underexplored in contemporary neural architectures: the principled modelling of an ensemble of longitudinal, patient-specific response trajectories, as a function of administered doses, and with few observations per individual.

Traditionally, pharmacokinetic (PK) modelling relies on systems of ordinary differential equations (ODEs) to describe the temporal evolution of drug concentrations within the body. In clinical trials, however, the available data are typically sparse — often consisting of only a handful of observations per individual. To compensate, classical approaches employ *Nonlinear Mixed-effects* (NLME) models, in which ODE parameters are decomposed into fixed effects (capturing population-level kinetics) and random effects (accounting for inter-individual variability). This hierarchical strategy (*i.e*, fixed vs random, population vs individual) enables the sharing of statistical strength across subjects, while preserving subject-specific inference, even under limited observation regimes (Lavielle, 2014).

Despite their successes, these traditional pipelines require manual specification of drug-specific kinetic systems — often crafted de novo for each compound — along with labor-intensive calibration and training workflows. Some deep learning alternatives have been proposed for individual-level inference in rich data settings, such as neural ODEs (Lu et al., 2021). However, the specific challenge of representing sparse hierarchical data, critical for robust population-level inference, has not been addressed so far.

A modern framework should retain the representational flexibility of neural-ODE-like models, while re-

maining data-efficient and population-aware. In particular, it must handle sparse and irregular sampling, and it should explicitly capture the latent structure of the study population — *i.e.*, shared dynamics across patients together with individual-specific variability. This enables generalization across compounds, subjects, and study designs, while adapting to patient-specific observations.

One promising family of methods that meets (some of) these requirements are *Amortized In-context Inference Models*: pretrained neural networks that adapt to new tasks directly from context data, enabling zero-shot generalization, and reducing the need for costly retraining (or finetuning) on every new dataset. Two recent representatives are Prior-fitted Networks (PFNs) and Foundation Inference Models (FIMs). PFNs are transformer-based models pretrained on large synthetic datasets to perform in-context estimation of *predictive posterior distributions* for tabular data, given only a few supervised examples (Müller et al., 2022; Hollmann et al., 2022; Müller et al., 2025). FIMs, by contrast, are trained on simulations of thousands of dynamical systems to map observed trajectories (*i.e.* the context) directly to their infinitesimal generators in a zero-shot fashion (Berghaus et al., 2024, 2026; Seifner et al., 2025b,a; Mauel et al., 2026; Hinz et al., 2025). Both PFNs and FIMs fall under the broader umbrella of *simulation-based inference*.

In this work, we translate these methodologies to mixed-effects pharmacokinetic models. Indeed, similar to FIMs, we place a broad *prior distribution* over compartmental PK models by assigning a stationary Ornstein–Uhlenbeck (OU) process to their parameters Overgaard et al. (2005). We use this prior to generate a large synthetic training corpus of PK trajectories, encoding strong pharmacological inductive biases. Unlike FIMs — but akin to PFNs — we focus on the posterior predictive distribution of *partially observed systems*. To this end, we introduce a hidden-variable formulation that operates in context, inspired by neural processes (Garnelo et al., 2018; Nguyen and Grover, 2022) but extended with a hierarchical structure: a global latent representation that captures population-level effects, while local latents account for patient-specific deviations. The model is trained to *match* the prior distribution over PK models.

This design allows the model to both generate synthetic patient trajectories, and deliver dose-conditioned predictions for newly enrolled patients, by exploiting the collective information from previously profiled trial participants *contained in its context*. In turn, it enables characterization of inter-patient variability, assessment of safety margins under covariate shifts, and quantification of predictive uncertainty — key requirements for adaptive trial design and regulatory submission.

Our main contributions are:

1. We introduce a simulator that places stochastic priors on compartmental PK models, and use it to generate a large corpus of synthetic PK trajectories. We empirically show that the resulting corpus encodes strong pharmacological inductive biases, enabling models trained on it to *generalize* to both clinical trials and pre-clinical studies.

2. We present the *Amortized In-Context Mixed-Effect Transformer* (AICMET) model, a transformer-based, hierarchical, latent-variable model for pharmacokinetics — to our knowledge, the first capable of amortized, in-context posterior prediction — pretrained to match our synthetic training distribution. We demonstrate that it achieves competitive *zero-shot* performance across compounds and subjects when compared to NLME and neural ODE baselines.

2 RELATED WORK

Machine learning solutions for NLME modelling span a wide range of methodologies, from traditional non-parametric Gaussian process regression methods (Shi et al., 2012; Leroy et al., 2022) to more scalable variants that condition the GP mean functions on neural networks (Chung et al., 2020). Neural ODE-based approaches have also been applied in this context (Nazarovs et al., 2022), although they typically impose a linear dependence on the fixed effects, and still rely on computationally expensive and often unstable adjoint methods for training. Specifically tailored solutions for pharmacokinetics (Lu et al., 2021) have demonstrated that neural ODEs can learn both complex latent dynamics and response times as a function of dosing schemes. However, these methods are generally trained on large datasets and ignore the hierarchical, population-level structure central to NLME modeling.

Closer to our setting, Arruda et al. (2024) follow the classical simulation-based inference pipeline, where the goal is to infer parameters of a fixed simulator tailored to NLME modeling. However, amortization occurs only at the simulator level. They still train a new population model for *each* dataset, and their method is restricted to population models with tractable densities. Our approach moves amortization one step further. Rather than learning a dataset-specific population model, we infer the *posterior predictive distribution* directly from the context data, in the spirit of

PFNs. As a result, the same pretrained model can be deployed *off-the-shelf* across downstream datasets.

Thus, our proposal provides an alternative to existing approaches by combining compartmental priors with amortized in-context inference, enabling zero-shot adaptation from sparse, irregularly sampled data while explicitly capturing both population- and individual-level variability.

3 PROBLEM DEFINITION

Let us assume we have access to a *pharmacokinetic study*, which consists of noisy concentration measurements collected from a cohort of I individuals. We denote this study by $\mathcal{S} = \{\mathcal{D}^1, \dots, \mathcal{D}^I\}$. Let $\mathcal{D}^i = (\mathbf{u}^i, \boldsymbol{\tau}^i, \mathbf{Y}^i)$ correspond to the data associated to the ith individual, with $\mathbf{u}^i$ the initial dosing information (a tuple containing its value (u_1^i) and dosing route (u_2^i)), $\boldsymbol{\tau}^i = (\tau_1^i, \dots, \tau_{T_i}^i)$ the observation times, $\mathbf{Y}^i = (y_1^i, \dots, y_{T_i}^i)$ the measured (plasma) concentrations, and T_i the number of available measurements. Note that observation times need not be shared across individuals, since measurements are typically sparse and irregularly sampled.

We focus on two complementary tasks:

1. **Population synthesis.** Without accessing any per-subject information beyond what is contained in $\mathcal{S}$, the goal is to generate new *virtual* individuals $\{\mathcal{D}^{I+1}, \dots, \mathcal{D}^{I+N}\}$ whose statistics are indistinguishable from those of $\mathcal{S}$. In other words, we seek a generative model capable of sampling from the implicit population distribution.

2. **Individual prediction.** Suppose that we are presented with a new subject $\mathcal{D}^n = (\mathbf{u}^n, \boldsymbol{\tau}^n, \mathbf{Y}^n)$. The goal is to forecast future concentrations y^* at any time $\tau^* > \tau_{T_n}^n$.

In the mixed-effects setting, the general strategy amounts to decomposing variability into *fixed effects*, which describe the population-level kinetics of the study, and *random effects*, which capture individual-specific deviations.

In what follows, we introduce the *Amortized In-Context Mixed-Effect Transforme* model (AICMET), a generative model that addresses both tasks *in-context*. That is, at deployment it receives either $\mathcal{S}$ (for Task 1) or $\mathcal{S} \cup \mathcal{D}^n$ (for Task 2) as prompt and returns either samples (Task 1) or forecasts (Task 2) *in a single forward pass* — without gradient–based optimization. This design contrasts with classical NLME workflows that perform explicit inference *for every new dataset*.

The AICMET framework consists of two components: a synthetic data generation model, and a hierarchical latent variable model. We describe these in the next sections.

4 AICMET: DATA GENERATION

In this section, we introduce a data generation model that combines compartmental PK systems with stochastic, time-varying parameters. We use this model to generate a large synthetic training dataset. We begin by presenting the compartmental PK systems (Section 4.1), followed by the prior over their parameterization (Section 4.2). We then describe the observation model (Section 4.3) and, finally, formally specify the full generative process as the probability of observing PK trajectories (Section 4.4).

4.1 Compartmental Pharmacokinetic Systems

Compartmental models are classically used in pharmacokinetics to describe the uptake, distribution and elimination of drug in the body, with kinetically similar organs represented as a *compartment*. The *central compartment* represents systemic blood circulation and organs in fast exchange with blood, *peripheral compartments* represent organs in slower exchange with blood, and the *gut compartment* represents the absorption of an orally administered drug.

Following this formulation, each simulated study $\mathcal{S}$ consists of I individuals, where each individual is characterized by a time-varying latent state $\mathbf{X}(t) \in \mathbb{R}^{2+P}$ that represents the drug amount in the gut, central (c), and P peripheral (per) compartments. That is

$$\mathbf{X}(t) = \Big(X_{\text{gut}},\, X_c,\, X_{\text{per},1}, \dots, X_{\text{per},P}\Big)^\top. \tag{1}$$

which solves the ODE system

$$\begin{aligned}
\dot{X}_{\text{gut}} &= -k_a\, X_{\text{gut}}, \\
\dot{X}_c &= k_a X_{\text{gut}} - \Big(k_e + \sum_{j=1}^{P} k_j^+\Big) X_c + \sum_{j=1}^{P} k_j^- X_{\text{per},j}, \\
\dot{X}_{\text{per},j} &= k_j^+ X_c - k_j^- X_{\text{per},j},
\end{aligned} \tag{2}$$

with j running from 1 to P. The initial condition is determined by the dose and the dosing route (i.e., $X_{u_2=c}(0) = u_1$ for intravenous dosing and $X_{u_2=\text{gut}}(0) = u_1$ for oral dosing), while all other states are initialized at zero. The number of individuals, the number of peripheral compartments, as well as the dose and dosing route, are randomly drawn for each simulated study. Next, we define the *kinetic parameters* $k_a, k_e, k_1^+, k_1^-, ..., k_P^+, k_P^-$ underlying these PK systems.

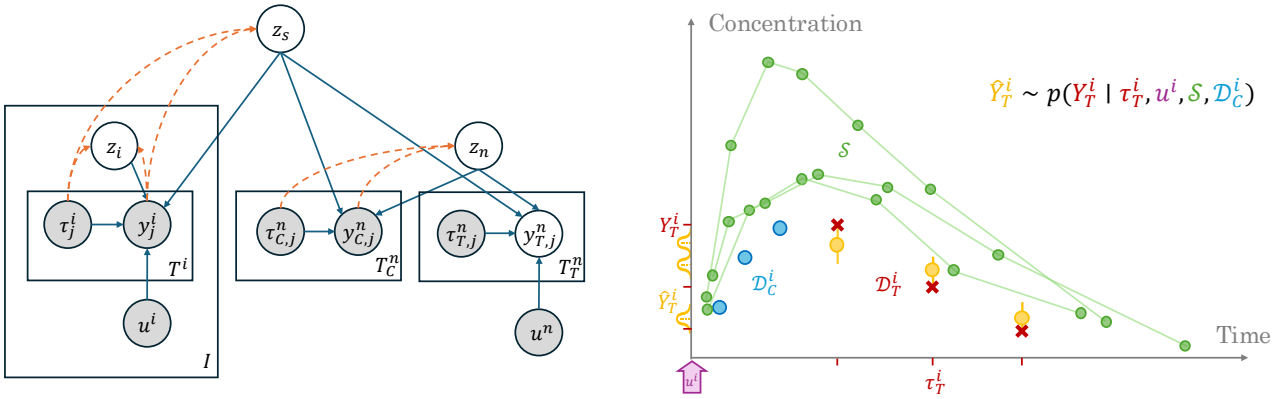


Figure 1: Left: hierarchical latent structure assumed in our AICMET model. Shaded nodes are observed. All latent representations (z_i, z_s, z_n) are continuous; solid blue arrows indicate conditional dependencies (decoder), orange dashed arrows indicate the recognition network (encoder). Right: different levels of context entering the predictive distribution (14).

4.2 Stochastic parameter model

Classically, PK models are built incrementally, starting with first-order kinetic rates and adding extra compartments or non-linear terms ad hoc, should they be indicated by the data. In our synthetic data generation approach, we extend the compartment model above by including *stochastically varying parameters*, thereby allowing for solutions that mimic realistic PK data.

Let $\theta_i(t) = \log(V^i, k_a^i, k_e^i, , k_1^{i+}, k_1^{i-}, ..., k_P^{i+}, k_P^{i-})$ denote the time-dependent vector of log-kinetic parameters (plus the volume of distribution V^i which appears in the observation model, Section 4.3, below) for the ith individual. Let each component $\theta_{i,k}$, with $k \in \{1, \ldots, 3+2P\} =: \mathcal{K}$, evolve as an independent OU process

$$d\theta_{i,k}(t) = -\lambda_k(\theta_{i,k}(t) - \mu_{i,k})\, dt + \sigma_k\, dW_{i,k}(t), \quad (3)$$

initialized at $\theta_{i,k}(0) \sim \mathcal{N}(\mu_{i,k}, \sigma_k^2/(2\lambda_k))$ to ensure stationarity.

The OU parameters $(\mu_{i,k}, \sigma_k, \lambda_k)$ are themselves random variables, sampled according to the following distributions

$$\mu_{i,k}|m_k, s_k \sim \mathcal{N}(m_k, s_k^2), \quad (4)$$

$$\lambda_k^2 \sim \mathcal{U}(a_\lambda, b_\lambda), \quad (5)$$

$$\sigma_k^2/2\lambda_k \sim \mathcal{U}(a_\sigma, b_\sigma), \quad (6)$$

where $m_k \sim \mathcal{U}(a_{m_k}, b_{m_k})$, $s_k \sim \mathcal{U}(a_{s_k}, b_{s_k})$, and the values of $a_{m_k}, b_{m_k}, a_{s_k}$ and b_{s_k} are chosen based on a meta-analysis of NLME model fits, and determine the shape of typical PK trajectories. Similarly, the parameters $a_\lambda, b_\lambda, a_\sigma$ and b_σ are set empirically to yield physiologically plausible PK profiles. We conclude this section by noting that if $\theta_k(t) \equiv \mu_k$, the data-generating process reduces to a standard compartmental PK model. Parameter stochasticity is induced through Eq. 3, precisely to enable generalization beyond this parametric family.

4.3 Observation Model

In this section, we first describe the random selection of observation times (τ) and, based on these, the construction of synthetic concentration samples (y). To ensure meaningful observation times for each study, we estimate two characteristic timescales: the time to reach peak plasma concentration T_{peak}, and the drug half-life T_{half}. These timescales are obtained using a one-compartment model with first-order absorption. To wit

$$T_{\text{peak}} = \frac{\log(\bar{k}_a) - \log(\bar{k}_e)}{\bar{k}_a - \bar{k}_e}, \qquad T_{\text{half}} = \frac{\log(2)}{\bar{k}_e},$$

for typical parameters at the study level (*i.e.*, $\bar{k}_a = e^{m_1}$ and $\bar{k}_e = e^{m_2}$).

Irregular plasma samples are then simulated at subject–specific times $0 \le \tau_1^i < \cdots < \tau_{T_i}^i \le \tau_{\max}$. These times are chosen to mimic the design of experimental studies. Specifically, we select four equally spaced observations before T_{peak}, and six observations after T_{peak}, placed at increasingly wider intervals determined by T_{half}. This design reflects typical sampling schedules. Finally, we randomly subsample the number of observations according to empirical sample-size distributions, and handle mismatched sizes through masking. The observed plasma concentrations (y) are obtained by evaluating simulated trajectories at the selected times, and applying a proportional measure-

$$
\begin{aligned}
&p(\boldsymbol{\eta}, T_{\text{peak}}, T_{\text{half}}, \boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\tau}, \mathbf{X}, \mathbf{Y}) \\
&= p(\boldsymbol{\eta})\; p(T_{\text{peak}}, T_{\text{half}} \mid \boldsymbol{\eta}) \prod_{i=1}^{I} p(\boldsymbol{\theta}_i(0), \boldsymbol{\mu}_i \mid \boldsymbol{\eta}) \\
&\quad \times p\left(\boldsymbol{X}^i(0)\right) p(\boldsymbol{\tau}^i \mid T_{\text{peak}}, T_{\text{half}}) \\
&\quad \times \prod_{t \in \mathcal{T}} p_{\text{OU}}(\boldsymbol{\theta}_i(t+\Delta t) \mid \boldsymbol{\theta}_i(t), \boldsymbol{\eta}, \boldsymbol{\mu}_i) \\
&\qquad \times p_{\text{ode}}\big(\mathbf{X}^i(t+\Delta t) \mid \mathbf{X}^i(t), \boldsymbol{\theta}_i(t)\big) \\
&\quad \times \prod_{j=1}^{T_i} p_{\text{obs}}\big(y_j^i \mid \mathbf{X}^i(\tau_j^i), \boldsymbol{\theta}_i(\tau_j^i)\big)\,. \qquad (8)
\end{aligned}
$$

Equation Block 1: AICMET Prior Distribution

ment error model. In equations, we write

$$
y_j^i \;=\; \frac{X_c^i(\tau_j^i)}{V^i(\tau_j^i)}(1 \,+\, \varepsilon_{i,j}), \quad \varepsilon_{i,j} \sim \mathcal{N}\big(0, \sigma_{\text{obs}}^2\big). \qquad (7)
$$

4.4 (Prior) Generative Data Model

Introducing $\eta_k = (m_k, s_k, \lambda_k, \sigma_k)$, and collecting these into $\boldsymbol{\eta} = (\eta_k)_{k \in \mathcal{K}}$, we can formally write the joint probability underlying the generation of our synthetic study $\mathcal{S} = \{\mathcal{D}^1, \ldots, \mathcal{D}^I\}$ as in Eq. 8. In this expression, $\mathcal{T}$ denotes a fine simulation grid, p_{OU} the OU transition density implied by Eq. 3, p_{ode} the flow defined by Eq. 2, and p_{obs} the observation model from Eq. 7. Equation (8) thus serves both as the blueprint for synthetic data generation, and as the conceptual backbone of our in-context learning objective. Appendix A contains all hyperparameters used to specify our synthetic prior distribution, along with a validation study against real-world data showing that the resulting synthetic prior generates physiologically plausible pharmacokinetic profiles. Next, we introduce a latent variable model that matches the *marginal* prior Eq. (8).

5 AICMET: HIERARCHICAL LATENT VARIABLE MODEL

In this section, we introduce our latent-variable model. We equip it with a hierarchical structure: a *global* study-level code, which plays the role of the fixed effects, and an *individual-specific* code, corresponding to the random effects for each subject. We view the model as a neural process, but without imposing permutation invariance. A new individual is represented by the dataset $\mathcal{D}^n$, and the full data are given by $\mathcal{S}^n = \mathcal{S} \cup \mathcal{D}^n$, which augments the study context with this newcomer. The model is trained by minimizing a loss that couples the global context with the new individual, followed by context–target splits at the individual level. As in neural processes and variational autoencoders, the **generative model** is composed of two components: a *prior over latent codes* and a *decoder (likelihood) function*. We define these below.

Prior over Latent Codes. We place a zero–mean isotropic Gaussian prior on the *global* study vector (encoding fixed effects) $\mathbf{z}_s \in \mathbb{R}^{d_s}$, as well as on each *individual* vector (encoding random effects) $\mathbf{z}_i \in \mathbb{R}^{d_i}$, including the unseen individual n:

$$
\mathbf{z}_s \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d_s}), \text{ and } \mathbf{z}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_{d_i}), \qquad (9)
$$

for $i \in \{1, \ldots, I\} \cup n$.

Decoder (Likelihood) Function. The joint density factorizes as

$$
\begin{aligned}
p(\mathbf{Y} \mid \mathbf{z}_s, \{\mathbf{z}_i, \mathbf{u}^i, \boldsymbol{\tau}^i\}_{i=1}^I, \mathbf{z}_n, \mathbf{u}^n, \boldsymbol{\tau}^n) = \\
\prod_{i=1}^{I} \prod_{j=1}^{T_i} \mathcal{N}\big(y_j^i \mid \mu_\theta(\tau_j^i, \mathbf{Z}_i),\, \sigma_\theta^2(\tau_j^i, \mathbf{Z}_i)\big) \\
\times \prod_{j=1}^{T_n} \mathcal{N}\big(y_j^n \mid \mu_\theta(\tau_j^n, \mathbf{Z}_n),\, \sigma_\theta^2(\tau_j^n, \mathbf{Z}_n)\big), \qquad (10)
\end{aligned}
$$

where we introduced the variable $\mathbf{Z}_i = (\mathbf{z}_s, \mathbf{z}_i, \mathbf{u}^i)$, with $i \in \{1, \ldots, I\} \cup n$, and where θ denotes trainable parameters. We require the decoder in Eq. 10 to handle the irregular sampling schedules that are typical in clinical datasets, while also leveraging the strong *context-learning* capabilities of modern transformers. To this end, we adopt a transformer decoder with *functional attention* (Seifner et al., 2025a). The requested prediction or sampling times (τ) are embedded as **queries**, whereas the dosing information $\mathbf{u}^n$, together with the population-level ($\mathbf{z}_s$) and individual-level ($\mathbf{z}_n$) latent variables are embedded as **keys** and **values**. Through self-attention, the model functions as a context learner: each query time selectively attends to the most informative dosing and latent-effect context, thereby defining a distribution over concentration–time functions conditioned on both $\mathbf{z}_s$ and $\mathbf{z}_n$. The left panel of Fig. 1 illustrates the graphical model underlying Eq. 10, while Fig. 2 depicts the neural network implementation of the decoder. The detailed equations specifying the neural network architecture are provided in the Supplementary Material.

Encoder Function as a Hierarchical Variational Posterior. Exact posterior inference under Eq. 10 is intractable because μ_θ and σ_θ are non-linear. We therefore introduce the following factorized, varia-

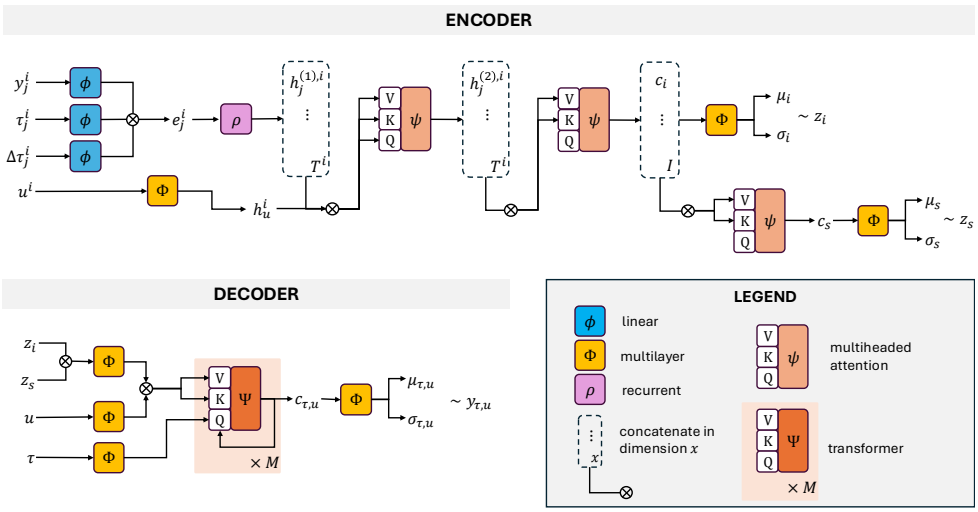


Figure 2: AICMET architecture. The encoder produces dynamic representations with a recurrent backbone, and attention mechanisms are applied to summarize these representations at both the individual and study levels. Our transformer-based decoder embeds the encoder representations alongside dose information. Finally, we define functional queries that allow us to evaluate the predictive distribution at any target time τ. By introducing $\mathbf{z}_i$ and $\mathbf{z}_s$, we can model fixed and random effects, enabling a population-aware, individualized characterization of dynamics.

tional approximation:

$$q_\phi\big(\mathbf{z}_s, \{\mathbf{z}_i\}_{i=1}^I, \mathbf{z}_n \,\big|\, \mathcal{S}^n\big) = q_\phi(\mathbf{z}_s \mid \mathcal{S}^n)\; q_\phi(\mathbf{z}_n \mid \mathcal{D}^n) \times \prod_{i=1}^{I} q_\phi(\mathbf{z}_i \mid \mathcal{D}^i). \quad (11)$$

Each factor is modeled as a Gaussian with mean and diagonal covariance parameterized by an encoder network with trainable parameters ϕ. The architecture of this encoder is illustrated in Fig. 2, and the detailed equations are provided in Appendix B. For clarity, we omit explicit reference to ϕ in what follows. Let us now introduce our *objective functions*.

New Individual Objective. Our first objective is to maximize the *predictive* likelihood of a new individual, marginalizing over all latent variables, $\log p(\mathbf{Y}^n | \boldsymbol{\tau}^n, \mathcal{S})$. Applying Jensen's inequality with the approximate posterior in Eq. 11 yields the evidence lower bound (ELBO)

$$\begin{aligned}\mathcal{L}_{\text{new}} = \mathbb{E}_q\left[\log p\big(\mathbf{Y}^n \mid \mathbf{z}_s, \mathbf{z}_n, \mathbf{u}^n, \boldsymbol{\tau}^n\big)\right] \\ - \mathrm{KL}\big[q(\mathbf{z}_s \mid \mathcal{S}) \,\big\|\, p(\mathbf{z}_s)\big] - \mathrm{KL}\big[q(\mathbf{z}_n \mid \mathcal{D}^n) \,\big\|\, p(\mathbf{z}_n)\big] \\ - \mathrm{KL}\big[q(\mathbf{z}_s \mid \mathcal{S}^n) \,\big\|\, q(\mathbf{z}_s \mid \mathcal{S})\big]. \quad (12)\end{aligned}$$

The first and second Kullback-Leibler (KL) terms in Eq. 12 regularize the global and new-individual encodings towards their respective priors, thereby enabling sampling. The final KL term enforces consistency between the representations with and without the new individual. During training, the new individual is obtained by randomly selecting one subject from the study, and excluding it from the study context.

Predictive Objective. For longitudinal prediction, each *observed* trajectory is split into *context* and *target* subsets, *i.e.* $\mathcal{D}^i = \mathcal{D}^i_{\mathrm{C}} \cup \mathcal{D}^i_{\mathrm{T}},$. The right panel of Figure 1 illustrates this splitting. Adapting Eq. 12 to this partially observed setting yields the following *predictive bound*

$$\begin{aligned}\mathcal{L}_{\mathrm{P}} = \mathbb{E}_{q(\mathbf{z}_s|\mathcal{S})}\Big[\sum_{i=1}^{I} \mathbb{E}_{q(\mathbf{z}_i|\mathcal{D}^i)}\big[\log p(\mathbf{Y}^i_{\mathrm{T}} | \,\mathbf{z}_s,\, \mathbf{z}_i, \mathbf{u}^i, \boldsymbol{\tau}^i_T)\big]\Big] \\ - \sum_{i=1}^{I} \mathrm{KL}\big[q(\mathbf{z}_i \mid \mathcal{D}^i) \,\big\|\, q(\mathbf{z}_i \mid \mathcal{D}^i_{\mathrm{C}})\big]. \quad (13)\end{aligned}$$

The KL term inside the sum encourages the individual encoder to refine its beliefs *after observing the targets*, thereby specializing the latent representation for extrapolative prediction. Jointly optimizing Eqs. 12 and 13 yields a study-level representation that is capable of (i) generating synthetic individuals whose statistics match those of $\mathcal{S}$; and (ii) forecasting future concentrations for any individual after only a handful of early observations. Details of the neural parameterizations used for q_ϕ, μ_θ, and σ_θ, as well as the training procedure, are provided in Appendices B and C.

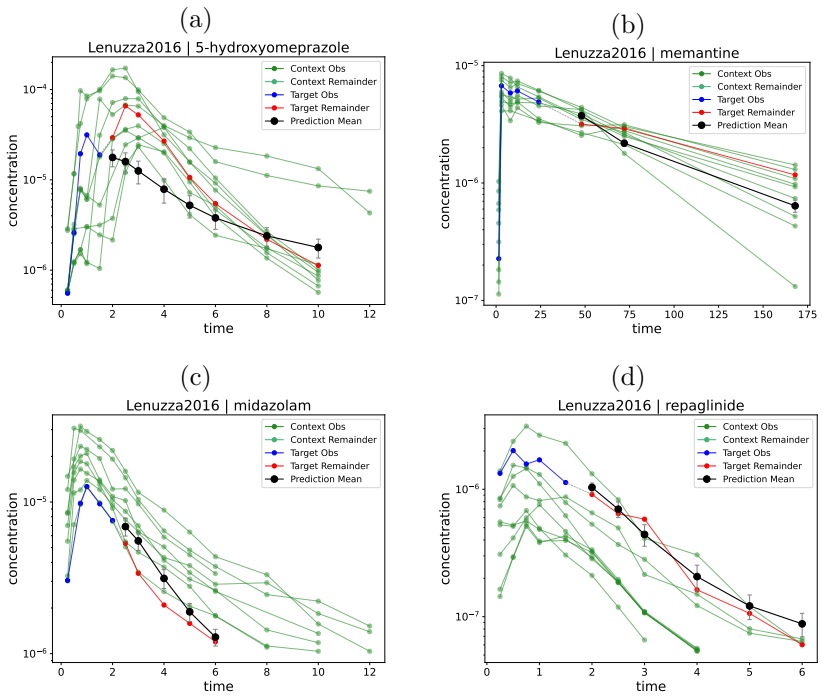


Figure 3: AICMET *zero-shot* prediction curves for different compounds. Each subplot shows the observed and target concentrations, and the context, together with simulation-derived prediction intervals. AICMET is indeed able to infer the main patterns characteristic of the study from its context.

Prediction. Finally, to estimate the *posterior predictive distribution* we calculate the integral

$$p(\mathbf{Y}_{\mathrm{T}}^i|\mathbf{u}^i, \boldsymbol{\tau}_{\mathrm{T}}^i, \mathcal{S}, \mathcal{D}_C^i) = \int p(\mathbf{Y}_{\mathrm{T}}^i|\, \mathbf{z}_s,\, \mathbf{z}_i, \mathbf{u}^i, \boldsymbol{\tau}_T^i) \times q(\mathbf{z}_s|\mathcal{S})\, q(\mathbf{z}_i|\mathcal{D}_C^i)\, d\mathbf{z}_i d\mathbf{z}_s, \quad (14)$$

via Monte Carlo by drawing samples from the encoders.

6 EXPERIMENTS

In this section, we begin by presenting the evaluation dataset and the baselines used for comparison, and then turn to a discussion of our main results.

6.1 Datasets

Evaluation data for assessing the performance of AICMET were obtained from the open-access database PK-DB, which contains data from both clinical and pre-clinical studies (Grzegorzewski et al., 2021). Specifically, we extracted plasma concentration measurements for 18 compounds (a total of 2019 valid concentration–time data points) from a Phase I clinical trial that investigated the safety and dosing of a combination of drugs and their metabolites in healthy volunteers (Lenuzza et al., 2016). The curated dataset includes 10 parent compounds (caffeine, dextromethorphan, digoxin, memantine, midazolam, omeprazole, paracetamol, repaglinide, rosuvastatin, and tolbutamide), together with their primary metabolites (paraxanthine, dextrorphan, 1-hydroxy-midazolam, 5-hydroxy-omeprazole, hydroxy-repaglinide, omeprazole sulfone, paracetamol glucuronide, and 4-hydroxy-tolbutamide). Importantly, both intra- and inter-substance sampling frequencies varied substantially, yielding irregularly sampled time series across subjects and compounds. Each compound was measured in 10 individuals, with up to 15 observations per individual. We also included two additional datasets on indometacin and theophylline pharmacokinetics (`Indometh` and `Theoph` from the R `datasets` package) (R Core Team, 2025), with 6 and 12 subjects, respectively.

6.2 Baselines and Ablations

We compare a *classical* pharmacometric baseline against one neural ODE model (NODE-PK), ablated variants of our approach, and finally against the full AICMET model. The classical baseline is nonlinear mixed-effects modelling (NLME) (Lavielle, 2014), which we fit *study by study*. That is, separately for each compound-specific dataset using conventional estimation procedures. In contrast, all neural mod-

Table 1: **Comparison of Log-RMSE Across Models.** For each compound, log-RMSE is reported for baseline models and the proposed AICMET model with selected ablations. The best-performing model for each compound is highlighted in **bold**.

Compound	NLME	NODE-PK	T-PK	SNODE-PK	ST-PK	AICME-NODE	AICMET
caffeine	**0.294 ± 0.294**	0.432 ± 0.209	0.575 ± 0.211	0.780 ± 0.286	0.451 ± 0.258	0.646 ± 0.243	0.399 ± 0.205
dextromethorphan	0.752 ± 0.752	0.605 ± 0.248	0.630 ± 0.227	1.702 ± 0.518	**0.395 ± 0.163**	0.640 ± 0.241	0.595 ± 0.364
digoxin	**0.300 ± 0.300**	0.661 ± 0.315	0.717 ± 0.264	0.501 ± 0.188	0.683 ± 0.242	0.569 ± 0.216	0.691 ± 0.464
indometacin	0.582 ± 0.582	**0.350 ± 0.291**	0.512 ± 0.247	0.441 ± 0.233	0.429 ± 0.328	0.487 ± 0.221	0.536 ± 0.203
memantine	**0.343 ± 0.343**	0.554 ± 0.386	0.799 ± 0.301	0.580 ± 0.221	0.513 ± 0.335	0.534 ± 0.196	0.475 ± 0.341
midazolam	0.596 ± 0.596	0.352 ± 0.224	0.735 ± 0.269	0.874 ± 0.314	0.288 ± 0.173	0.548 ± 0.209	**0.264 ± 0.151**
omeprazole	1.300 ± 1.300	1.003 ± 0.618	1.864 ± 0.623	1.267 ± 0.471	**0.842 ± 0.400**	1.395 ± 0.522	0.936 ± 0.942
paracetamol	**0.299 ± 0.299**	0.582 ± 0.302	0.825 ± 0.294	1.115 ± 0.406	0.447 ± 0.274	0.691 ± 0.248	0.352 ± 0.187
repaglinide	0.479 ± 0.479	0.693 ± 0.290	0.846 ± 0.309	1.514 ± 0.544	0.709 ± 0.268	0.562 ± 0.204	**0.339 ± 0.311**
rosuvastatin	**0.383 ± 0.383**	0.535 ± 0.303	0.748 ± 0.272	0.624 ± 0.231	0.502 ± 0.205	0.578 ± 0.217	0.558 ± 0.286
theophylline	0.732 ± 0.732	0.327 ± 0.144	0.401 ± 0.172	0.366 ± 0.155	0.356 ± 0.141	0.318 ± 0.126	**0.274 ± 0.109**
tolbutamide	0.723 ± 0.723	0.578 ± 0.263	0.816 ± 0.295	0.949 ± 0.346	0.678 ± 0.339	0.854 ± 0.316	**0.517 ± 0.254**
1-hydroxy-midazolam	–	0.737 ± 0.381	0.678 ± 0.241	1.395 ± 0.497	0.567 ± 0.327	0.935 ± 0.352	**0.491 ± 0.309**
4-hydroxy-tolbutamide	–	0.353 ± 0.182	0.898 ± 0.318	0.524 ± 0.191	0.354 ± 0.174	**0.274 ± 0.103**	0.335 ± 0.164
5-hydroxy-omeprazole	–	1.474 ± 0.745	1.683 ± 0.603	1.811 ± 0.654	1.286 ± 0.472	1.575 ± 0.581	**1.188 ± 0.451**
dextrorphan	–	0.834 ± 0.284	1.001 ± 0.362	0.904 ± 0.331	0.733 ± 0.445	0.614 ± 0.229	**0.563 ± 0.287**
hydroxy-repaglinide	–	**0.049 ± 0.156**	0.532 ± 0.194	0.059 ± 0.022	0.069 ± 0.217	0.095 ± 0.036	0.076 ± 0.240
omeprazole sulfone	–	1.390 ± 0.488	1.620 ± 0.577	1.529 ± 0.556	**1.054 ± 0.297**	1.438 ± 0.533	1.082 ± 0.509
paracetamol glucuronide	–	**0.297 ± 0.169**	0.423 ± 0.156	0.823 ± 0.301	0.357 ± 0.192	0.365 ± 0.137	0.307 ± 0.184
paraxanthine	–	**0.287 ± 0.135**	0.646 ± 0.233	0.653 ± 0.241	0.413 ± 0.181	0.409 ± 0.154	0.312 ± 0.138

els — including NODE-PK and our ablations (T-PK, SNODE-PK, ST-PK, AICME-NODE, AICMET) — are *pre-trained on our simulated PK datasets* and subsequently *evaluated in context (that is, zero-shot) on the empirical studies*, without any additional per-study parameter updates. Throughout, "NODE-PK" refers to our pre-trained implementation, ensuring that comparisons isolate architectural differences rather than data advantages. We now describe each model and ablation in detail.

Nonlinear Mixed-Effect (NLME) Modelling. For each of the 10 parent compounds in the evaluation dataset, one- and two-compartment NLME models were fitted (with first-order oral absorption in the case of orally dosed compounds). Inter-individual variability was included on all parameters, and a combined additive/proportional error model was used. The better-fitting model was selected by AIC and used as the baseline for that compound. Since metabolite production rates (i.e., dosing of metabolites) depend on the PK of the parent compound, these could not be directly modelled within this framework.

Neural ODEs (NODE-PK). The NODE-PK model follows Lu et al. (2021) and extends the latent neural ODE of Rubanova et al. (2019) with explicit handling of dosing. An RNN encoder produces a latent initial condition for each individual, which is then evolved through neural ODE dynamics between observation times. NODE-PK does not perform any study-level aggregation: individuals are treated independently, and predictions rely solely on their own past observations.

Ablations without KL terms. To isolate the contribution of architectural choices, we construct three ablations that all use only the prediction loss $\mathcal{L}_F$ (no KL terms or stochastic latents):

(i) **T-PK**: replaces the NODE-PK decoder with a functional attention (FA) transformer decoder, to be able to estimate concentrations at any observation time, yet still predicting individuals conditioned only on their own past observations.

(ii) **SNODE-PK**: adds a deterministic study-level representation to NODE-PK, obtained by aggregating individual encodings via attention, while keeping the NODE-PK decoder.

(iii) **ST-PK**: combines study-level aggregation with a FA-transformer decoder in place of the NODE-PK decoder.

Full AICMET Model. Finally, we evaluate our proposed amortized in-context mixed-effects model, which include stochastic study and individual latents, and is trained with all loss terms (Eqs. 12 and 13).

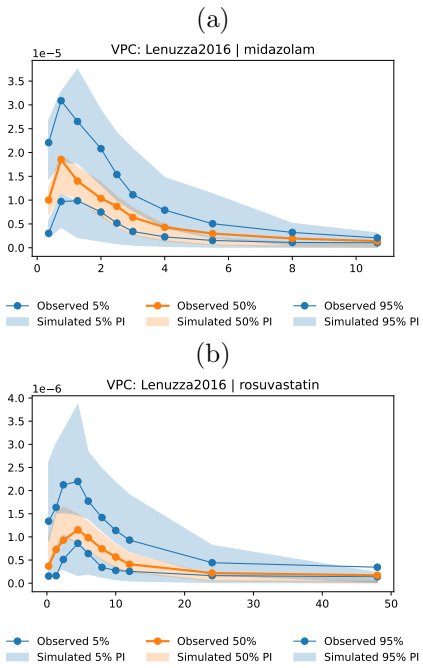


Figure 4: AICMET zero-shot visual predictive checks (VPCs) for two compounds. The panels compare observed summaries against simulation-based prediction intervals.

We consider two decoders: **AICME-NODE**, which uses the NODE-PK decoder, and **AICMET**, which replaces it with our functional attention transformer decoder. **AICMET** represents the complete methodology proposed in this work (Figure 2).

Thus, our ablations explicitly remove the decoder, the hierarchical latents, and the study-level aggregation, thereby testing whether the hierarchical structure captures meaningful variability.

6.3 Evaluation Metrics.

For each compound, predictive performance was assessed in a leave-one-individual-out setting: predictions for a given subject i were conditioned on the study context $\mathcal{S}$, the dosing information $\mathbf{u}^i$, and the first four PK samples of that subject $\mathcal{D}_{\mathrm{C}}^i$. Accuracy on the remaining samples $\mathcal{D}_{\mathrm{T}}^i$ was then measured by the root mean squared error on the log-concentration scale (log-RMSE).

To evaluate generative capability, we performed a simulation-based graphical diagnostic known as the *visual predictive check* (VPC), a standard tool in pharmacometrics (Holford, 2005). This diagnostic tool compares the variability of simulated plasma concentration profiles within a study via an overlay of simulation percentiles and observed data.

Implementation Details. Appendices B and C contain the full implementation details.

6.4 Discussion

As shown in Table 1, the pretrained zero-shot models (*i.e.*, from NODE-PK to AICMET) achieve state-of-the-art performance, improving upon the predictive accuracy (Task 2) of NLME models for 7 of the 12 parent compounds. This provides empirical evidence that our pretraining prior distribution (Eq. 8) encodes strong and useful inductive biases. Moreover, unlike NLME models, the pretrained models generalize to metabolites without requiring any additional model specification. Among them, AICMET is the clear best performer: it outperforms NODE-PK on 14 of the 20 target compounds, highlighting the benefits of its hierarchical, stochastic representation. Interestingly, the node-based models equipped with a study-level representation, whether deterministic or stochastic, are unable to effectively exploit this information. By contrast, transformer decoders appear easier to condition on study-level representations. Figure 3 further shows that AICMET yields accurate zero-shot predictions (Task 2) across concentration ranges spanning several orders of magnitude, a characteristic feature of PK data.

For population synthesis (Task 1), the VPCs in Figure 4 show that AICMET correctly captures the distribution of samples for new individuals from the context alone, in zero-shot mode. This is precisely what enables it to *generalize across compounds*. Thus, unlike NLME models, which require substantial compound-specific hyperparameter tuning, AICMET can infer predictive posterior distributions from PK studies in seconds, without tuning and with stronger performance. Beyond these main results, we also assessed the robustness of AICMET to study sizes larger than those seen during pretraining. The corresponding results are reported in Appendix D.

Limitations. Our current implementation is restricted to single-dose, covariate-free settings. Likewise, Phase-III-like sparse sampling regimes remain outside the present scope. Future work will explore whether joint training on both dense and sparse data, as is common in NLME when pooling data across trial phases, together with mixture-of-experts architectures, can help overcome these limitations. We also plan to extend the framework to account for multiple dosing and individual-specific covariates. We hypothesize that, in the latter case, zero-shot inference alone may no longer be sufficient and that finetuning may be required. Finally, more expressive generative models could be used as decoders, for example diffusion models, which may better capture complex distributions.

Acknowledgements

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG) — Project-ID 318763901 — SFB1294; and by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr Institute for Machine Learning and Artificial Intelligence.

References

Jonas Arruda, Yannik Schälte, Clemens Peiter, Olga Teplytska, Ulrich Jaehde, and Jan Hasenauer. An amortized approach to non-linear mixed-effects modeling based on neural posterior estimation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 1865–1901. PMLR, 21–27 Jul 2024.

David Berghaus, Kostadin Cvejoski, Patrick Seifner, César Ali Ojeda Marin, and Ramsés J Sánchez. Foundation inference models for markov jump processes. *Advances in Neural Information Processing Systems*, 37:129407–129442, 2024.

David Berghaus, Patrick Seifner, Kostadin Cvejoski, Cesar Ojeda, and Ramses J Sanchez. In-context learning of temporal point processes with foundation inference models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=h9HwUAODFP.

Ingyo Chung, Saehoon Kim, Juho Lee, Kwang Joon Kim, Sung Ju Hwang, and Eunho Yang. Deep mixed effect model using gaussian processes: a personalized and reliable prediction for healthcare. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3649–3657, 2020.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

J. Grzegorzewski, J. Brandhorst, K. Green, D. Eleftheriadou, Y. Duport, F. Bartsch, A. Köller, D. Y. J. Ke, S. De Angelis, and M. König. Pk-db: Pharmacokinetics database for individualized and stratified computational modeling. *Nucleic Acids Research*, 49(D1):D1358–D1364, 2021.

Manuel Hinz, Maximilian Mauel, Patrick Seifner, David Berghaus, Kostadin Cvejoski, and Ramses J Sanchez. Towards fast coarse-graining and equation discovery with foundation inference models. *arXiv preprint arXiv:2510.12618*, 2025.

Nick Holford. The visual predictive check—superiority to standard diagnostic (rorschach) plots. In *Population Approach Group in Europe (PAGE) Meeting*, volume 14, 2005. Abstract 738. https://www.page-meeting.org/?abstract=738.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

Marc Lavielle. *Mixed Effects Models for the Population Approach: Models, Tasks, Methods and Tools*. Chapman and Hall/CRC, 1st edition, 2014. doi: 10.1201/b17203. URL https://doi.org/10.1201/b17203.

N. Lenuzza, X. Duval, G. Nicolas, E. Thévenot, S. Job, O. Videau, C. Narjoz, M. A. Loriot, P. Beaune, L. Becquemont, F. Mentré, C. Funck-Brentano, L. Alavoine, P. Arnaud, M. Delaforge, and H. Bénech. Safety and pharmacokinetics of the cime combination of drugs and their metabolites after a single oral dosing in healthy volunteers. *European Journal of Drug Metabolism and Pharmacokinetics*, 41(2):125–138, 2016.

Arthur Leroy, Pierre Latouche, Benjamin Guedj, and Servane Gey. Magma: inference and prediction using multi-task gaussian processes with common mean. *Machine Learning*, 111(5):1821–1849, 2022.

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.

Maximilian Mauel, Johannes R Hübers, David Berghaus, Patrick Seifner, and Ramses J Sanchez. Foundation inference models for ordinary differential equations. *arXiv preprint arXiv:2602.08733*, 2026.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*,

2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Samuel Müller, Arik Reuter, Noah Hollmann, David Rügamer, and Frank Hutter. Position: The future of bayesian prediction is prior-fitted. *arXiv preprint arXiv:2505.23947*, 2025.

Jurijs Nazarovs, Rudrasis Chakraborty, Songwong Tasneeyapant, Sathya N Ravi, and Vikas Singh. Mixed effects neural ode: A variational approximation for analyzing the dynamics of panel data. *arXiv preprint arXiv:2202.09463*, 2022.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. *arXiv preprint arXiv:2207.04179*, 2022.

Rune V Overgaard, Niclas Jonsson, Christoffer W Tornøe, and Henrik Madsen. Non-linear mixed-effects models with stochastic differential equations: implementation of an estimation algorithm. *Journal of pharmacokinetics and pharmacodynamics*, 32(1): 85–107, 2005.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2025. URL https://www.R-project.org/.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

Patrick Seifner, Kostadin Cvejoski, David Berghaus, Cesar Ojeda, and Ramses J Sanchez. In-context learning of stochastic differential equations with foundation inference models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=ceCJPoZOKJ.

Patrick Seifner, Kostadin Cvejoski, Antonia Körner, and Ramses J Sanchez. Zero-shot imputation with foundation inference models for dynamical systems. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=NPSZ7V1CCY.

JQ Shi, B Wang, EJ Will, and RM West. Mixed-effects gaussian process functional regression models with application to dose–response curve prediction. *Statistics in medicine*, 31(26):3165–3177, 2012.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [No]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [No]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [No]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Amortized In-Context Mixed Effect Transformer Models: Supplementary Materials

A Synthetic PK Prior

A.1 Synthetic Prior Hyperparameters

Table 2 summarizes all hyperparameters needed for the simulation of the compartment models with OU priors (Eq. 8).

Table 2: Synthetic Prior (Meta-Study) Configuration Parameters (Condensed)

Parameter	Value / Range	Parameter	Value / Range
drug_id_options	Drug_A, B, C	solver_method	rk4
num_individuals_range	[5, 10]	time_start	0.0
num_peripherals_range	[1, 3]	time_stop	24.0
log_k_a_mean_range	[-1, 2]	time_num_steps	100
log_k_a_std_range	[0.2, 0.6]	log_k_e_mean_range	[-5, 0]
k_a_tmag_range	[0.01, 0.02]	log_k_e_std_range	[0.2, 0.6]
k_a_tscl_range	[1, 5]	k_e_tmag_range	[0.01, 0.02]
log_V_mean_range	[2, 8]	k_e_tscl_range	[1, 5]
log_V_std_range	[0.2, 0.6]	V_tmag_range	[0.001, 0.001]
V_tscl_range	[1, 5]	rel_ruv_range	[0.001, 0.01]
log_k_1p_mean_range	[-4, 0]	log_k_1p_std_range	[0.2, 0.6]
k_1p_tmag_range	[0.01, 0.02]	k_1p_tscl_range	[1, 5]
log_k_p1_mean_range	[-4, -1]	log_k_p1_std_range	[0.2, 0.6]
k_p1_tmag_range	[0.01, 0.02]	k_p1_tscl_range	[1, 5]

A.2 Synthetic Prior Validation

To validate the realism of our synthetic prior, we performed a systematic literature search on PubMed covering 114 bioequivalence studies reporting standardized pharmacokinetic summary metrics such as AUC (total drug exposure) and Cmax (peak plasma concentration). These studies provide a natural benchmark for our synthetic training distribution because (i) such measurements are routinely reported for most drugs on the market, and (ii) regulatory guidelines require applicants to provide these standardized quantities.

Table 3 compares the distribution of dose-normalized pharmacokinetic summaries observed in the literature with those induced by our simulations. Overall, the agreement indicates that our synthetic prior generates physiologically plausible pharmacokinetic profiles.

While the simulated values tend to be somewhat lower than those observed in the literature, they remain within a comparable range across percentiles for both Cmax and AUC. This suggests that the synthetic prior captures realistic orders of magnitude and variability in PK behavior, supporting its use as a training distribution for amortized inference.

B Architecture

In this section we provide all the details of the neural network architecture for the Amortized In Context Mixed Effect Transformer (AICMET) model, required to define both encoder and decoder. We can see that in order to create a faithful representation of the data, we first need to create a longitudinal representation per patient that

Table 3: Comparison between literature-derived and simulated dose-normalized pharmacokinetic summaries. Cmax is reported in 1/L and AUC in h/L.

Metric	Percentile	Literature	Simulation
Cmax	10th	6.7e-4	2.7e-4
Cmax	50th	9.5e-3	4.2e-3
Cmax	90th	6.8e-2	4.3e-2
AUC	10th	4.4e-3	1.3e-3
AUC	50th	5.5e-2	2.0e-2
AUC	90th	1.0e+0	0.2e-1

respects the time information per individual. Then, we aggregate the time dimension to obtain a representation per individual, and finally we aggregate all the individuals to obtain one representation per study. In the following, we use H as hidden dimension and Z_d as latent dimension for the unstructured latent variables. A superscript in parentheses, e.g. $\mathbf{h}^{(1)}$, denotes the *layer index*.

B.1 Neural-network primitives.

We denote by $\phi^\theta(\cdot)$ a linear projection $\mathbb{R}^m \to \mathbb{R}^H$, by $\Phi^\theta(\cdot)$ a feed-forward network $\mathbb{R}^H \to \mathbb{R}^H$, by $\rho^\theta(\cdot)$ a recurrent layer (GRU/LSTM), by $\psi^\theta(\mathbf{q}, \mathbf{K}, \mathbf{V})$ a multi-head attention module, and by $\Psi^\theta(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ a transformer encoder (linear attention). For both $\psi^\theta(\cdot,\cdot,\cdot)$ and $\Psi^\theta(\cdot,\cdot,\cdot)$ denote transformer encoders with linear attention (Katharopoulos et al., 2020), both of which take three arguments as inputs (i.e., queries, keys and values). We use the dot-product softmax kernel, i.e., $\psi^\theta(\mathbf{Q}, \mathbf{K}, \mathbf{V}) := \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$. Since PK data are often sparsely and irregularly sampled, we expand the features by defining the individual dataset as

$$\widetilde{\mathcal{D}}^{(i)} = \left\{ \left(y_j^{(i)},\, \tau_j^{(i)},\, \Delta\tau_j^{(i)}\right) \right\}_{j=1}^{T_i-1}, \qquad \Delta\tau_j^{(i)} := \tau_{j+1}^{(i)} - \tau_j^{(i)}.$$

This simplifies the task of the network of handling irregular time intervals, as this representation enforces this feature.

B.2 Encoder

We first introduce the encoder, which we construct by first introducing the individual representation and then aggregating for the studies representations.

Per-transition embeddings. We pass the basic features trough a linear layer and concatenate, thereby defining a basic data embedding:

$$\mathbf{e}_j^{(0),i} = \text{concat}\Big[\phi_y^\theta(y_j), \phi_\tau^\theta(\tau_j), \phi_{\Delta\tau}^\theta(\Delta\tau_j)\Big] \in \mathbb{R}^H.$$

Longitudinal Representation. To capture the dynamical information from the embeddings, we now create a representation per individual and per time step, using either a recurrent neural network (GRU/LSTM) or a transformer architecture:

$$\mathbf{h}_j^{(1),i} = \rho^\theta\big(\mathbf{e}_{1:j}^{(0),i}\big) \in \mathbb{R}^H.$$

Next, in order to judge the overall curve shape, a self-attention mechanism is applied over the longitudinal representations $\mathbf{h}_j^{(1),i}$. This allows the different native features of the drug profiles, such as the peak and slope of different sections around the peak, to be weighted and compared against each other. We first stack the representations $\mathbf{H}^{(1),i} = [\mathbf{h}_1^{(1),i}, \ldots, \mathbf{h}_{T_i-1}^{(1),i}] \in \mathbb{R}^{H\times(T_i-1)}$; then

$$\mathbf{H}^{(2),i} = \Psi^\theta\big(\mathbf{H}^{(1),i}, \mathbf{H}^{(1),i}, \mathbf{H}^{(1),i}\big).$$

Individual posterior. Now we aggregate with attention pooling so that we are able to obtain a representation per individual

$$\mathbf{c}_i = \psi^\theta\big(\mathbf{q}, \mathbf{H}^{(2),i}, \mathbf{H}^{(2),i}\big) \in \mathbb{R}^H.$$

We then differentiate the mean and variance of the encoder with a final MLP layer

$$q(\boldsymbol{z}_i \mid \mathcal{D}^i) = \mathcal{N}\Big(\boldsymbol{z}_i \in \mathbb{R}^{Z_d} \;\Big|\; \Phi^\theta_\mu(\mathbf{c}_i), \operatorname{diag}\big(e^{\,\Phi^\theta_\sigma(\mathbf{c}_i)}\big)\Big).$$

Study Posterior. To obtain a study-specific representation, we concatenate the individual summaries $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_I] \in \mathbb{R}^{H\times I}$, and then summarize with another attention pooling mechanism, $\mathbf{c}_s = \psi^\theta\big(\mathbf{q}_s, \mathbf{C}, \mathbf{C}\big)$. Finally the study encoder is obtained by

$$q(\boldsymbol{z}_s \mid \mathcal{S}) = \mathcal{N}\Big(\boldsymbol{z}_s \in \mathbb{R}^{Z_d} \;\Big|\; \Phi^\theta_\mu(\mathbf{c}_s), \operatorname{diag}\big(e^{\,\Phi^\theta_\sigma(\mathbf{c}_s)}\big)\Big).$$

B.3 Decoder

To design a decoder suited for PK data, we must accommodate the irregular sampling schedules typically observed in clinical datasets while simultaneously exploiting the strong *context-learning* capabilities of modern transformers. Conventional neural ODE approaches (e.g. Lu et al., 2021) rely on adjoint-sensitivity computations; this can hinder scalability for a data prior like ours that requires many samples to enforce expert based inductive bias. We therefore adopt a transformer decoder endowed with *functional attention* (Seifner et al., 2025a). The requested prediction/sample time points τ are embedded as **queries**, whereas the dosing information $\mathbf{u}^n$ together with the population-level ($\mathbf{z}_s$) and individual-level ($\mathbf{z}_n$) latent variables are embedded as **keys** and **values**. Through self-attention, the model acts as a context learner: each query time dynamically attends to the most informative dosing and latent-effect context, thereby defining a distribution over concentration-time functions conditioned on both, $\mathbf{z}_s$ and $\mathbf{z}_i$. For a new subject $\mathcal{D}^{(n)}$, we first sample $\mathbf{z}_n$ from its prior, then propagate the query-time set through the decoder to obtain the predictive concentration distribution at each requested time point. Specifically, for a new individual $\mathcal{D}^n$, we first sample $\boldsymbol{z}_n \sim q(\boldsymbol{z}_n|\mathcal{D}^n)$ and $\boldsymbol{z}_s \sim q(\boldsymbol{z}_s|\mathcal{S})$ for prediction, or $\boldsymbol{z}_n \sim \mathcal{N}(0, I)$ for generation and embed a query time τ:

$$\begin{aligned}
\mathbf{q}_\tau &= \phi^\theta_{\text{dec}}(\tau) \in \mathbb{R}^H,\\
\mathbf{K}_n &= \Phi^\theta_K\left([\boldsymbol{z}_n; \boldsymbol{z}_s; \mathbf{u}]\right) \in \mathbb{R}^{H\times 1},\\
\mathbf{V}_n &= \Phi^\theta_V\left([\boldsymbol{z}_n; \boldsymbol{z}_s; \mathbf{u}]\right) \in \mathbb{R}^{H\times 1},\\
\mathbf{c}_\tau &= \psi^\theta\left(\mathbf{q}_\tau, \mathbf{K}_n, \mathbf{V}_n\right) \in \mathbb{R}^H.
\end{aligned} \tag{15}$$

It only remains to define the final mean and variance heads, $(\mu_\tau, \log\sigma^2_\tau) = \Phi^\theta_{\text{dec}}(\mathbf{c}_\tau)$, and specify the distribution to a diagonal Gaussian $p\big(y_\tau \mid \tau, \boldsymbol{z}_n, \boldsymbol{z}_s\big) = \mathcal{N}\big(y_\tau \mid \mu_\tau, \sigma^2_\tau\big)$.

C Experimental Details

C.1 Neural Network Architecture

Figure 5 displays the implementation details of the AICMET Model.

C.2 Training Details

We used the ADAM optimizer with a learning rate of 0.0001 and a batch size of 128 across all datasets. Data were simulated on the fly and models were trained up until 5000 iterations. The history context length of the decoder $|\mathcal{D}^n_C|$ varies according to the distribution of the empirical data. All experiments are conducted on a single Nvidia V-100 GPU, and results are based on 5 runs per model. That is, all models (baselines, ablations, and AICMET variants) were trained five times with different random initializations. For each method, we selected the best-performing run, and report variability across seeds.

We rescaled the total loss with $\mathcal{L}_T = \sum_l e^{-U^\theta_l}\mathcal{L}^\theta_l(\mathcal{D}) - U^\theta_l$ as specified by Karras et al. (2024), where $\mathcal{L}^\theta_l$ specify all the different elements of the loss (KL terms treated separately), and U^θ_l are nuisance parameters whose purpose

```
AIMCETPK(
  encoder = RNNContextEncoder(
    input_encoder = TimeObsSeparateEncoder(
      time_encoder = MLP([1 -> 256 -> 256]),
      obs_encoder  = MLP([1 -> 256 -> 256]),
      layernorm = LayerNorm(512)
    ),
    rnn = GRU(512, 256, num_layers=4),
    self_attn = MultiheadAttention(256),
    summary_attn = MultiheadAttention(256),
    proj = Linear(256 -> 256),
    dose_proj = MLP([2 -> 256 -> 256])
  ),
  decoder = TransformerDecoder(
    time_proj = MLP([1 -> 512 x4]),
    z_proj    = MLP([256 -> 512 x4]),
    init_proj = MLP([4 -> 512 x4]),
    attn_blocks = 2 x CrossAttentionBlock(
      attn = MultiheadAttention(512),
      mlp  = MLP([512 -> 512])
    ),
    mean_head   = MLP([512 -> 512 -> 512 -> 1]),
    logvar_head = MLP([512 -> 512 -> 512 -> 1])
  ),
  aggregator = AttentionStudyAggregator(
    attn = MultiheadAttention(256)
  ),
  mu_s_layer        = Linear(256 -> 256),
  logvar_s_layer    = Linear(256 -> 256),
  mu_i_layer        = Linear(256 -> 256),
  logvar_i_layer    = Linear(256 -> 256),
  mu_init_layer     = Linear(256 -> 1),
  logvar_init_layer= Linear(256 -> 1),
  combine_mlp       = MLP([512 -> 256 -> 256]),
  loss_multihead    = MultiHeadLoss()
)
```

Figure 5: Architecture summary of the `AICMET` model. MLPs are shown with input/output dimensions; repeated blocks are compressed for clarity.

is to ensure equal loss contributions of each component during training. Further details about the architecture, training and simulations are in the Supplementary Material.

Finally, we sampled new systems for every batch so the model never sees one system twice. In total, the model sees 1280 systems per epoch, for 1000 epochs for a total of 1.280.000 different systems.

D Robustness with Respect to Study Sizes

To assess robustness with respect to cohort size, we sampled synthetic datasets with varying numbers of subjects and evaluated predictive performance using the root mean squared error (RMSE) on the log-concentration scale. The results are reported in Table 4.

Table 4: Predictive performance across different study sizes. Reported values correspond to the mean and standard deviation of the RMSE on the log-concentration scale.

Subjects in study	Mean RMSE	Std
10	0.532	0.375
20	0.410	0.234
30	0.344	0.093
40	0.318	0.154
50	0.407	0.185
60	0.356	0.126
70	0.252	0.112
80	0.304	0.118
90	0.321	0.154
100	0.408	0.160

As shown in Table 4, AICMET maintains strong predictive performance across a broad range of study sizes, indicating robustness to variations in cohort size. Although some fluctuation is observed, no systematic degradation appears as the number of subjects changes.

Regarding extremely sparse per-individual sampling, such as in late-phase trials, we note that additional methodological development will likely be required. In such regimes, a hybrid training strategy combining richly sampled early-phase synthetic data with sparsely sampled late-phase synthetic data may be necessary, potentially together with a mixture-of-experts approach.

E Extra Results

We include results for both the prediction and VPC plot for the rest of the compounds not seen in the main body of the text.

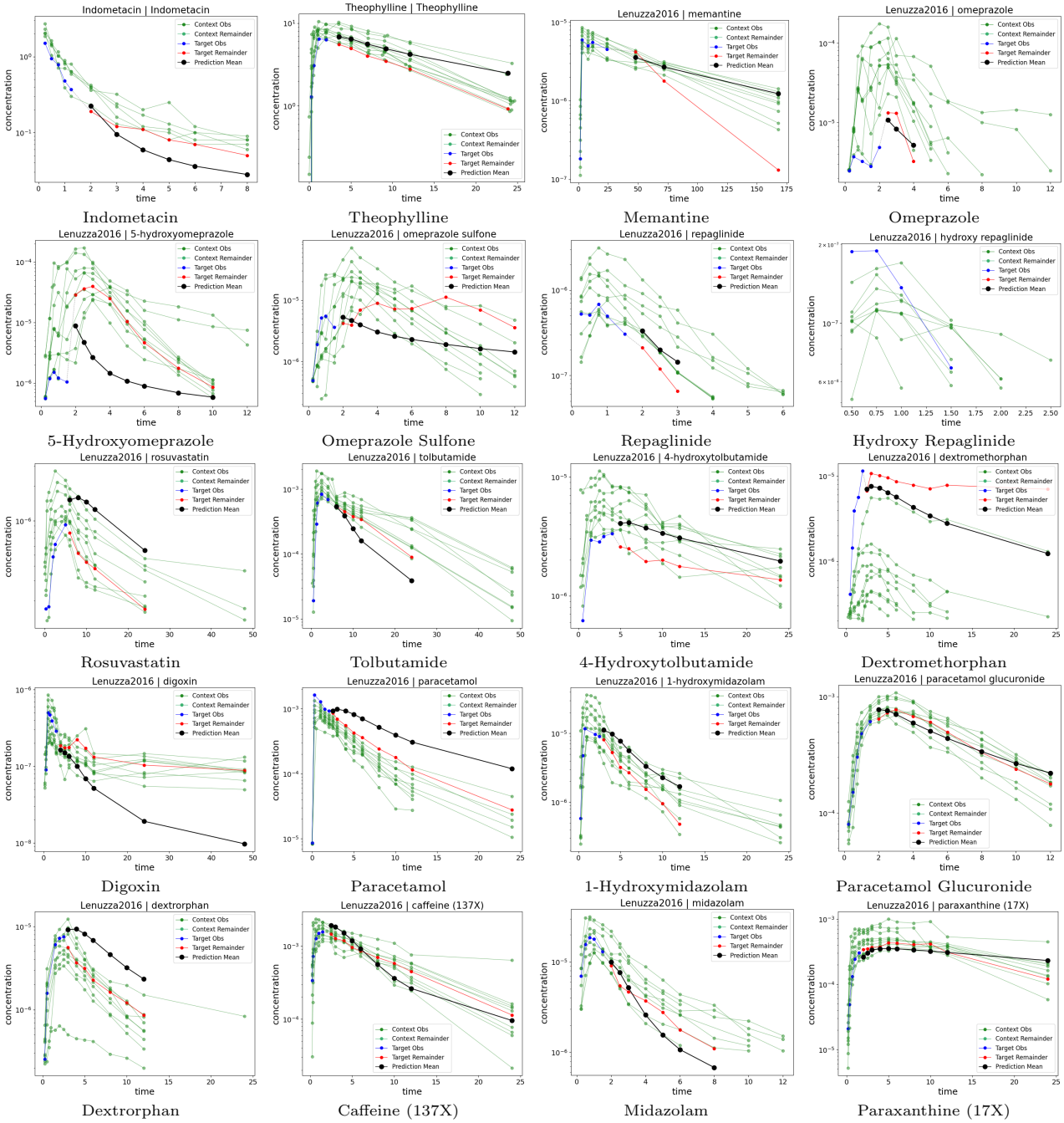
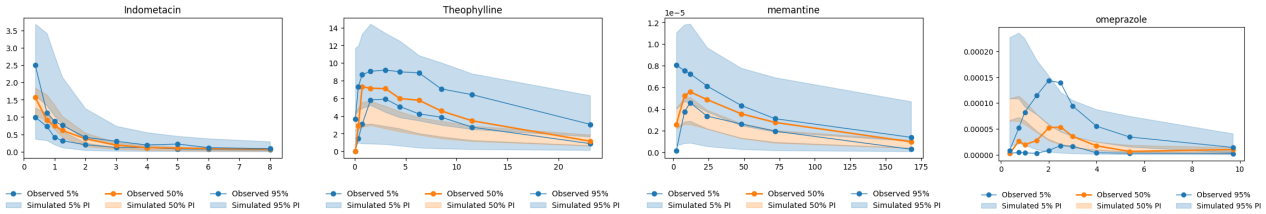


Figure 6: Predicted concentration-time profiles for all compounds in the study. Each panel shows a representative sample from the generative model.

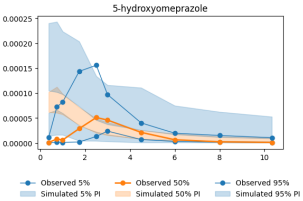


Indometacin

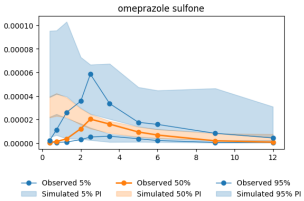
Theophylline

Lenuzza 2016 (000)

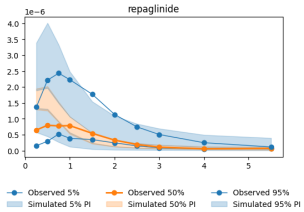
Lenuzza 2016 (001)



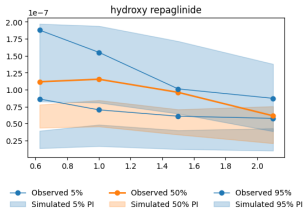
Lenuzza 2016 (002)



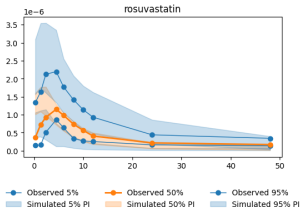
Lenuzza 2016 (003)



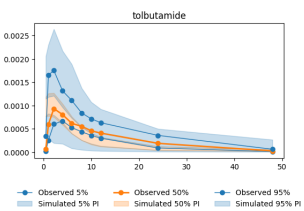
Lenuzza 2016 (004)



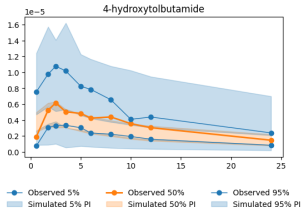
Lenuzza 2016 (005)



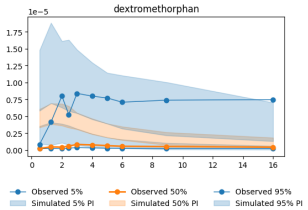
Lenuzza 2016 (006)



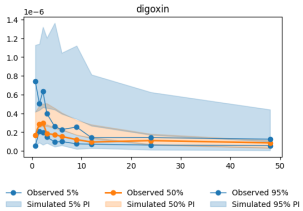
Lenuzza 2016 (007)



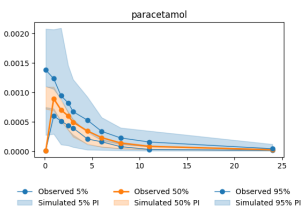
Lenuzza 2016 (008)



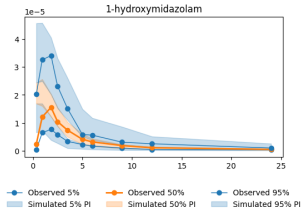
Lenuzza 2016 (009)



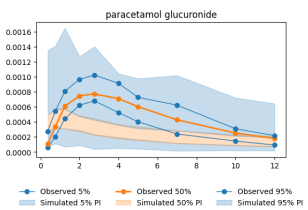
Lenuzza 2016 (010)



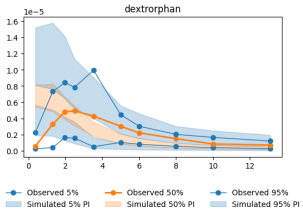
Lenuzza 2016 (011)



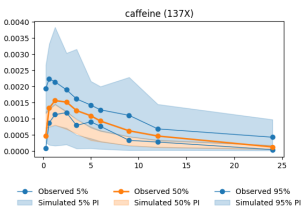
Lenuzza 2016 (012)



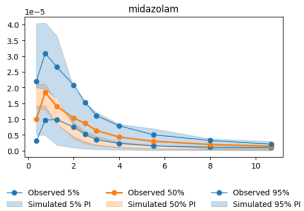
Lenuzza 2016 (013)



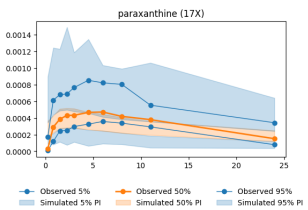
Lenuzza 2016 (014)



Lenuzza 2016 (015)



Lenuzza 2016 (016)



Lenuzza 2016 (017)

Figure 7: VPC visualizations for all compounds