Prior-Fitted Functional Flows: In-Context Generative Models for Pharmacokinetics

César Ojeda[1] **Niklas Hartung**[1] **Purity Kamene Kavwele**[1] **Tim Jahn**[2] **Piyush Kumar**[1] **Marian Klose**[3]
Wilhelm Huisinga[1] **Ramsés J. Sánchez**[4] **Darius A. Faroughy**[5]
[1]University of Potsdam [2]Technical University Berlin [3]Freie Universität Berlin
[4]Lamarr Institute & University of Bonn [5]NHETC, Rutgers University, USA

Abstract

We introduce Prior-Fitted Functional Flows, a generative foundation model for pharmacokinetics that enables zero-shot population synthesis and individual forecasting without manual parameter tuning. We learn functional vector fields, explicitly conditioned on the sparse, irregular data of an entire study population. This enables the generation of coherent virtual cohorts as well as forecasting of partially observed patient trajectories with calibrated uncertainty. We construct a new open-access literature corpus to inform our priors, and demonstrate state-of-the-art predictive accuracy on extensive real-world datasets. Our code repository [a], pretrained model, data and examples are available online.

[a] https://github.com/cesarali/pff

1 INTRODUCTION

Pharmacokinetics (PK) studies the dynamics of how the human body processes a drug. In a typical early-development setting, each subject receives a single dose, and the goal is to characterize how systemic drug concentration unfolds over time. This characterization requires not only representing temporal dynamics, but also capturing *heterogeneity* because, even under identical doses, different individuals can exhibit markedly different profiles, due to factors such as physiology, disease state, and co-medications.

The key difficulty is that these dynamics must be inferred from sparse and noisy measurements collected across a *study*: a cohort of individuals, each observed at an irregular, and often subject-specific set of time points. Operationally, one would like to (i) predict the concentration profile of a new individual, or (ii) forecast the unobserved future of a patient from only a few early measurements. PK formalizes this setting by treating each measured concentration as a noisy sample of an underlying, patient-specific, and continuous-time *concentration function* generated by a dynamical system [Gibaldi and Perrier, 1982]. In the population setting, this is most often expressed through nonlinear mixed-effects (NLME) models, which decompose variability into fixed effects shared across the cohort, and random effects capturing individual deviations [Lavielle, 2014]. From this viewpoint, the fundamental inferential object is a *distribution over plausible concentration functions*, conditioned on study information and (optionally) partial observations.

Recent work has begun to approximate such concentration functions with deep generative models, such as neural ODEs [Lu et al., 2021]. However, PK datasets are not only sparse and noisy, but also small (especially in Phase I and many Phase II settings). Training a high-capacity model from scratch for each new dataset is therefore both statistically fragile and operationally expensive. To bridge the gap between expressive modeling and data scarcity, we turn to *amortized inference*.

Amortization shifts the computational cost of inference from repeated, dataset-specific optimization to a single, up-front *pretraining* phase across large and diverse datasets — typically synthetically generated from *mechanistic priors*. The heterogeneous pretraining distributions encourage models to learn *reusable inference algorithms* that map a context (*e.g.*, the study and partial observations) directly to a target (posterior) distribution. This paradigm underlies prior-fitted networks (PFNs) for tabular data [Müller et al., 2022, Hollmann et al., 2023, 2025], foundation inference models (FIMs) for system identification [Berghaus et al., 2024, Seifner et al., 2025a, Berghaus et al., 2026, Hübers et al., 2026], and many foundation models for time-series imputation [Seifner et al., 2025b] and forecasting [Dooley et al., 2024, Bhethanabhotla et al., 2024, Hemmer and Durstewitz, 2025].

`AICMET` [Ojeda et al., 2026] represents a first step in this direction for PK. It was pretrained on synthetic studies gen-

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:5040–5059.

erated from stochastic priors on compartmental models, and performs *zero-shot posterior estimation* over concentration functions using a hierarchical mixed-effects representation. The latter is implemented through neural-process-style latent variables [Garnelo et al., 2018]. A limitation of this design is the reliance on a compressed representation of the study context, which can lead to underfitting when the context is complex [Kim et al., 2019], and to uncertainty that is not sharply calibrated [Wang et al., 2025].

In this work, we dispense from explicit mixed-effect parametrizations and instead learn the distribution over concentration functions directly in function space. We introduce *Prior-fitted Functional Flows* (PFFs), which leverages conditional optimal-transport (OT) flow matching [Kerrigan et al., 2024] to transport a simple reference measure over functions (a Gaussian process) into the conditional distribution over drug-concentration functions. The construction is conditional at two levels: first at the level of the study (*i.e.*, the cohort-level context), and second, at the level of a patient's early measurements for forecasting. We therefore make the following contributions:

1. We release an open-access reference dataset mined from clinical bioequivalence studies, containing inferred PK parameters and synthetic concentration–time samples used to calibrate physiologically plausible pretraining priors.
2. We introduce `PFF`, a prior-fitted functional flow model for zero-shot PK inference. PFF directly transports measures over concentration–time functions, combines study-level conditioning with GP posterior references for partially observed individuals, and uses continuum-attention neural operators [Calvello et al., 2024]. Experiments and ablations show improved forecasting and sample fidelity over NLME, `AICMET`, and PFF variants.

2 RELATED WORK

Classical PK and NLME Modelling. Population PK is typically formulated with nonlinear mixed-effects (NLME) models, where a compartmental ODE is shared across subjects and parameters decompose into fixed (population) and random (individual) effects [Lavielle, 2014]. Recent ML extensions include GP and neural mixed-effects formulations, including mixed-effects neural ODEs Kipnis et al. [2025], but these approaches are often developed for data-rich settings rather than the small, sparse studies common in early drug development. And different from us, focus on learning parameters of mechanistic models Häggström et al. [2025].

Amortized Inference for PK. Simulation-based amortization has recently been used to accelerate pharmacokinetic inference by pretraining neural estimators on simulated data [Arruda et al., 2024]. In Arruda et al., amortization is primarily performed in *parameter space*: the learned estimator approximates posterior inference over the scalar parameters of a specified PK simulator, which are then propagated through the mechanistic model to obtain concentration–time predictions. This is complementary to the notion of amortization pursued in `AICMET`, which maps study context directly to posterior-predictive concentration functions in a zero-shot manner [Ojeda et al., 2026], but retains a neural-process-style latent-variable representation of study and individual effects. Building on this line, `PFF` amortizes the posterior-predictive operator directly in function space. It learns a conditional flow over continuous-time concentration functions, conditioned on both study-level context and individual partial observations, and therefore avoids dataset-specific retraining while targeting the posterior-predictive distribution itself rather than first inferring simulator parameters.

Generative Modelling for Continuous Functions. Function-space generative models learn continuous-time transport maps between distributions over trajectories, including diffusion-based and flow-matching approaches [Biloš et al., 2023, Kerrigan et al., 2023, Kollovieh et al., 2024, Jahn et al., 2025]. Our method builds most directly on the dynamic conditional optimal-transport perspective for functional flow matching [Kerrigan et al., 2024].

3 PRELIMINARIES

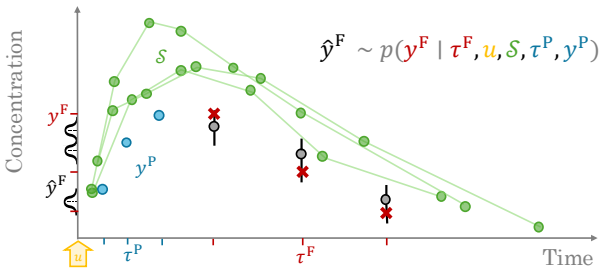


Figure 1: The model conditions on a population study (green) and dosing schedules (yellow). Based on partial observations (blue) of a new patient, it generates plausible continuations (black) to produce future forecasts (red).

In this section, we introduce the notation and formalize the amortized conditional generation problem we aim to solve. We then review the optimal-transport (OT) and flow-matching tools we build on, culminating in the conditional OT formulation that underlies our construction.

3.1 PROBLEM DEFINITION

We formalize pharmacokinetic (PK) data as sparse, noisy observations of latent, continuous-time concentration functions, and cast inference as learning a conditional probability measure over such functions.

Study data. A pharmacokinetic study consists of noisy concentration measurements collected from a cohort of I individuals $\mathcal{S} = \{\mathcal{D}^1, \dots, \mathcal{D}^I\}$. For individual i, we write $\mathcal{D}^i = (\mathbf{u}^i, \boldsymbol{\tau}^i, \mathbf{y}^i)$, where $\mathbf{u}^i = (a^i, r^i)$ is the dosing descriptor, with dose amount $a^i \in \mathbb{R}^+$ and administration route $r^i \in \{1, \dots, R\}$. The vector $\boldsymbol{\tau}^i = (\tau_1^i, \dots, \tau_{T_i}^i)$ contains the (irregular, subject-specific) observation times, and $\mathbf{y}^i = (y_1^i, \dots, y_{T_i}^i)$ contains the measured (*e.g.*, plasma) concentrations, with T_i the number of available measurements. Note that the observation times need not be shared across individuals, reflecting the sparse and irregular sampling typical in early-development studies.

Latent concentration functions. Let $\mathcal{T} = [0, T]$ denote the observation window of interest, with horizon time T. We model each individual's concentration trajectory as a *latent function* $f : \mathcal{T} \to \mathbb{R}$, and interpret the measurements $\mathbf{y}^i$ as noisy, irregular samples of f at times $\boldsymbol{\tau}^i$. To work in a well-defined function space, let $\mathcal{F}$ be a real, separable Hilbert space of functions equipped with its Borel σ-algebra $\mathcal{B}(\mathcal{F})$ and norm $\|\cdot\|_{\mathcal{F}}$. Let us also denote with $\mathscr{P}(\mathcal{F})$ the set of Borel probability measures on $\mathcal{F}$.

Conditional probability measures. PK inference is inherently conditional. Predictions for a new individual depend on the study $\mathcal{S}$, dosing information, and possibly a small number of early measurements for that individual. For now, we collect all such information into a single context variable $c \in \mathscr{C}$, where $\mathscr{C}$ is an abstract context space. The inferential object of interest is therefore a *conditional probability measure* over concentration functions $\nu(\cdot|c) \in \mathscr{P}(\mathcal{F})$. This measure should capture both the uncertainty due to sparse and noisy observations, and the heterogeneity across individuals implied by the context.

The amortized conditional generation problem. Let $\eta \in \mathscr{P}(\mathcal{F})$ be a tractable reference measure over functions (*e.g.*, a Gaussian process). Our goal is to learn a single conditional generator that, given η and a context c, produces samples distributed according to $\nu(\cdot|c)$. Concretely, we seek a measurable map $T^\star : \mathscr{C} \times \mathcal{F} \to \mathcal{F}$ such that, for (almost) every context $c \in \mathscr{C}$ we have $(T^\star(c, \cdot))_{\#}\eta = \nu(\cdot|c)$. In Section 3.3, we introduce conditional optimal transport on product spaces, and the class of triangular maps that formalize the conditional generation problem. Then, in Section 4, we instantiate c to reflect the PK settings of interest — that is, study-conditioned generation and, for forecasting, conditioning on partial observations of a new patient.

3.2 OT AND FLOW MATCHING

Our model builds on a standard link between optimal transport (OT) and flow-based generative modeling.

Unconditional OT. Let $\eta, \nu \in \mathscr{P}(\mathcal{F})$ be probability measures on the (separable Hilbert) function space $\mathcal{F}$. In the Monge formulation, OT seeks a measurable transport map $T : \mathcal{F} \to \mathcal{F}$ that pushes η to ν (*i.e*, $T_{\#}\eta = \nu$) while minimizing squared displacements. This yields the squared 2-Wasserstein distance

$$W_2^2(\eta, \nu) = \inf_{T:\mathcal{F}\to\mathcal{F}} \int_{\mathcal{F}} \|f - T(f)\|_{\mathcal{F}}^2 \, d\eta(f), \quad (1)$$

where the infimum is taken over all T which fulfill $T_{\#}\eta = \nu$.

The same quantity admits the dynamical Benamou-Brenier formulation [Benamou and Brenier, 2000]:

$$W_2^2(\eta, \nu) = \min_{(\gamma_t, v_t)} \int_0^1 \int_{\mathcal{F}} \|v_t(f)\|_{\mathcal{F}}^2 \, d\gamma_t(f) \, dt, \quad (2)$$

subject to the boundary conditions $\gamma_0 = \eta$, $\gamma_1 = \nu$, and the continuity equation

$$\partial_t \gamma_t + \mathrm{div}(v_t \gamma_t) = 0. \quad (3)$$

Here $(\gamma_t)_{t\in[0,1]} \subset \mathscr{P}(\mathcal{F})$ is a time-indexed path of measures, and $v_t : \mathcal{F} \to \mathcal{F}$ is a time-dependent *vector field*. Intuitively, v_t specifies how "particles" f move through $\mathcal{F}$, so that their resulting path measure $(\gamma_t)_{t\in[0,1]}$ evolves from η to ν. Thus the transport map T is parameterized through the vector field v_t.

The above two formulations are also known as static and dynamic optimal transport, respectively. In practice, directly solving the Benamou–Brenier minimization problem (Eq. 2) is intractable, as it requires optimizing over the space of all possible measure paths and vector fields. Flow Matching circumvents this by using the fact that any vector field satisfying the continuity equation (Eq. 3) generates a valid transport (though not ad hoc one of minimal costs). More specifically, Flow matching replaces direct OT optimization with a supervised regression problem, in which one (i) defines a family of simple conditional interpolation paths between paired samples $f_0 \sim \eta$ and $f_1 \sim \nu$, (ii) derives the corresponding conditional velocity field explicitly, and (iii) learns a parametric model $v_t^\theta(\cdot)$ to match this vector field. While in (i) any coupling ρ works, choosing an OT-aligned coupling (e.g., via mini-batch OT) is crucial; it ensures the interpolation paths are straight, which minimizes the kinetic energy and aligns the Flow Matching objective with the Benamou–Brenier objective [Pooladian et al., 2023, Tong et al., 2023]. We explain the functional version of Flow Matching in more detail in the following section.

Functional Flow Matching. [Kerrigan et al., 2023]. Let $f_0 \sim \eta$ and $f_1 \sim \nu$, and let $z = (f_0, f_1) \sim \rho$ for some coupling $\rho \in \Pi(\eta, \nu)$ (later to be chosen as minibatch-OT). To address step (i) from the previous section, Functional Flow Matching defines a Gaussian conditional path $\gamma_t(\cdot|z)$, which interpolates between source and target samples:

$$\gamma_t(\cdot \mid z) = \mathcal{N}\big((1-t)f_0 + t f_1, \ \Sigma\big), \quad (4)$$

where Σ is a trace-class covariance operator on $\mathcal{F}$ [Da Prato and Zabczyk, 2014]. This conditional path has the advantage

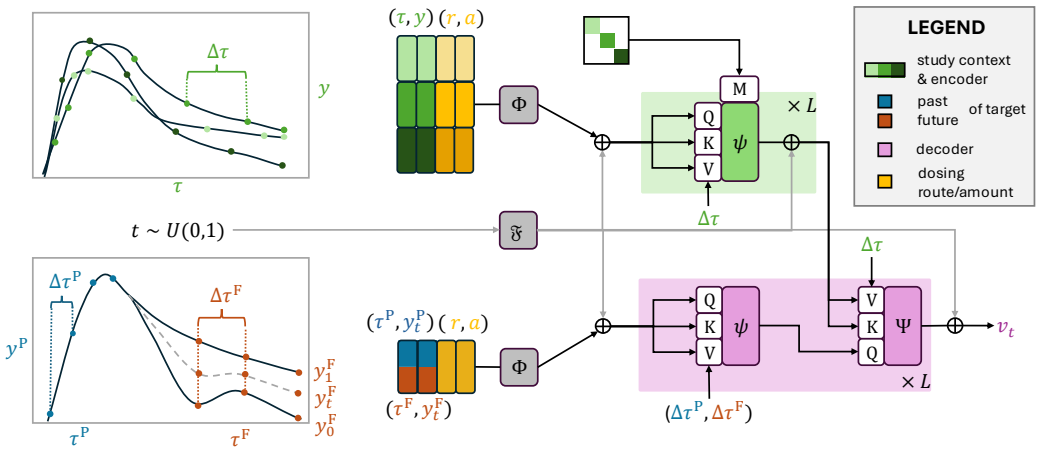


Figure 2: Top left: study context consists of time series of different individuals. Bottom left: data for a target individual are split into past observations $(\tau^{\mathrm{P}}, y^{\mathrm{P}})$ and future observations $(\tau^{\mathrm{F}}, y_1^{\mathrm{F}})$. With a uniformly distributed interpolation parameter t, y_t^{F} is obtained by interpolating between observed data y_1^{F} and a sample y_0^{F} from a Gaussian conditional path. Right: model architecture consisting of a study context encoder (top) and decoder (bottom). Abbreviations: τ: study time; y observed plasma concentration; r: route of administration; a: dosing amount; Φ, multilayer perceptron; $\mathfrak{F}$: Fourier features; ψ/Ψ: self-/cross-attention; Q/K/V/M/L: query/key/value/block diagonal mask/layers.

that it allows us to derive the associated conditional vector field $v_t(\cdot|z)$ corresponding to step (ii) explicitly:

$$v_t(\cdot \mid z) = f_1 - f_0. \tag{5}$$

While these conditional objects describe the trajectory of individual sample pairs, the marginal path γ_t describes the evolution of the full probability density from η to ν. It is obtained by mixing the conditional path over the coupling,

$$\gamma_t = \int \gamma_t(\cdot \mid z)\, d\rho(z), \tag{6}$$

and the corresponding marginal vector field can be written as

$$v_t(\cdot) = \int v_t(\cdot \mid z)\, \frac{d\gamma_t(\cdot \mid z)}{d\gamma_t}\, d\rho(z). \tag{7}$$

Crucially, the goal is to learn this marginal field v_t, which drives the global transport. To realize step (iii), one learns a parametric vector field $v_t^\theta(f)$ by solving the regression problem for the conditional vector field (Eq. 5)

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\, \mathbb{E}_{z\sim\rho}\, \mathbb{E}_{f\sim\gamma_t(\cdot|z)} \left[\|v_t^\theta(f) - (f_1 - f_0)\|_{\mathcal{F}}^2 \right].$$

Theoretical results guarantee that minimizing this objective has the same gradients as the (intractable) mean-squared error to the marginal field v_t (Eq. 7) [Tong et al., 2023].

3.3 CONDITIONAL OPTIMAL TRANSPORT IN FUNCTION SPACE

The previous section described Flow Matching in the *unconditional* setting. However, our problem is inherently conditional: we observe a context $c \in \mathscr{C}$ (e.g., patient covariates, dosing regimen) and seek to sample from a target conditional measure $\nu(\cdot \mid c) \in \mathscr{P}(\mathcal{F})$. Rather than solving a separate generative problem for every possible context, we formalize this as Conditional Optimal Transport. We seek a single *triangular* transport map on the product space $\mathscr{C} \times \mathcal{F}$ that generalizes across contexts, effectively amortizing the inference. As a warning, it is important to distinguish the notion of conditioning used in Flow Matching (conditioning on the path endpoints $z = (f_0, f_1)$ to define the vector field) from the conditioning on an external context $c \in \mathscr{C}$ (e.g., patient covariates).

Conditional OT. Let $\mu \in \mathscr{P}(\mathscr{C})$ denote a measure over contexts. We consider joint measures on $\mathscr{C} \times \mathcal{F}$ whose marginal on $\mathscr{C}$ equals μ,

$$\mathscr{P}_\mu(\mathscr{C} \times \mathcal{F}) = \left\{ \pi \in \mathscr{P}(\mathscr{C} \times \mathcal{F}) : \pi_{\mathscr{C}} = \mu \right\}, \tag{8}$$

where $\pi_{\mathscr{C}}$ denotes the $\mathscr{C}$-marginal of π. We take as reference joint measure the product $\eta = \mu \otimes \eta_{\mathcal{F}}$, where $\eta_{\mathcal{F}} \in \mathscr{P}(\mathcal{F})$ is an easy-to-sample base measure over functions. A target joint measure $\nu \in \mathscr{P}_\mu(\mathscr{C} \times \mathcal{F})$ is characterized by its conditional distributions $\nu(\cdot \mid c)$.

For conditional OT, we seek *triangular* transport maps $T : \mathscr{C} \times \mathcal{F} \to \mathscr{C} \times \mathcal{F}$. These maps leave the context component unchanged, formally defined as:

$$T(c, f) = \big(c,\, T_{\mathcal{F}}(c, f)\big), \tag{9}$$

where $T_{\mathcal{F}} : \mathscr{C} \times \mathcal{F} \to \mathcal{F}$. If T is triangular and satisfies the global pushforward constraint $T_{\#}\eta = \nu$, then the *fiberwise pushforward* identity holds for μ-almost every context $c \in \mathscr{C}$:

$$\big(T_{\mathcal{F}}(c, \cdot)\big)_{\#}\eta_{\mathcal{F}} = \nu(\cdot \mid c). \tag{10}$$

This formulation effectively amortizes the transport problem: learning a single global map T solves the conditional generation problem for all contexts simultaneously.

Triangular Functional Flows. In the dynamic Benamou–Brenier framework (Eq. 2), a triangular transport map corresponds to a flow generated by a *triangular vector field* $v_t : \mathscr{C} \times \mathcal{F} \to \mathscr{C} \times \mathcal{F}$. The context c is static , i.e. it represents fixed attributes that do not evolve during the generative process, which implies that its time derivative must be zero. This imposes a block structure (triangularity) on the vector field:

$$v_t(c, f) = \begin{pmatrix} 0 \\ v_t^{\mathcal{F}}(c, f) \end{pmatrix}, \tag{11}$$

where $v_t^{\mathcal{F}} : \mathscr{C} \times \mathcal{F} \to \mathcal{F}$ drives the evolution of the function f conditioned on c. Consequently, the learning problem reduces to estimating the functional component $v_t^{\mathcal{F}}$. We achieve this by parameterizing $v_t^{\mathcal{F}}$ with a neural network and minimizing the Flow Matching objective (Eq. 3.2), treating c as an additional conditioning input to the network.

4 PRIOR-FITTED FUNCTIONAL FLOWS

In this section, we introduce *Prior-Fitted Functional Flows* (`PFF`), a methodology for *zero-shot inference* of distributions (measures) over concentration functions from sparse PK studies. The core idea is to learn a context-conditioned transport map in function space that pushes a simple reference measure (a Gaussian process) to the *posterior predictive distribution* implied by a study context. This is achieved by training a neural operator to approximate the triangular vector field derived in Section 3.3. Rather than fitting scarce real-world datasets, we generate synthetic studies from a broad prior over physiologically plausible pharmacokinetic models. Crucially, while the data relies on a simulator, the training of the vector field is based on Flow Matching and therefore is *simulation-free*.

This yields an *amortized* model that can be applied off-the-shelf: given a new study (and, when available, a few early measurements from a new individual), `PFF` produces samples from a predictive distribution over full concentration trajectories in a single forward pass (plus ODE solve).

This approach is close in spirit to prior-fitted networks in that, rather than fitting a new model per dataset, we train once on a simulator-defined prior and learn an inference procedure that outputs the posterior predictive distribution over functions directly from context. We thus refer to our model as Prior-Fitted Functional Flows.

4.1 PRETRAINING PRIOR DISTRIBUTION

We construct a large-scale synthetic pretraining corpus using a hierarchical generative model P^{sim} that combines standard compartmental ODE systems with stochastic, time-varying parameters. This setup allows for the simulation of diverse study designs, including variable dosing routes, compartment structures, and irregular sampling schedules based on physiological timescales. A core contribution of this work is the rigorous validation of this generative prior against a large-scale dataset of pharmacokinetic parameters automatically mined from clinical literature, ensuring that our synthetic training data faithfully reflects real-world physiological variability. The full dataset and generation pipeline are made publicly available, with comprehensive details provided in Appendix A.2.

4.2 CONDITIONAL FUNCTIONAL FLOW MATCHING MODEL

We now instantiate the abstract conditional generation problem from Section 3 in the PK setting, and describe the conditional flow-matching construction used by Prior-Fitted Functional Flows (`PFF`). The key objects are: (i) a *simulator-induced* target measure over concentration trajectories, and (ii) a single learnable vector field in function space whose induced flow maps a tractable reference measure to the target in zero-shot mode. Crucially, `PFF` is trained once and used for both population synthesis and forecasting. The two modes differ only in the context provided at inference time, and in the corresponding choice of reference measure.

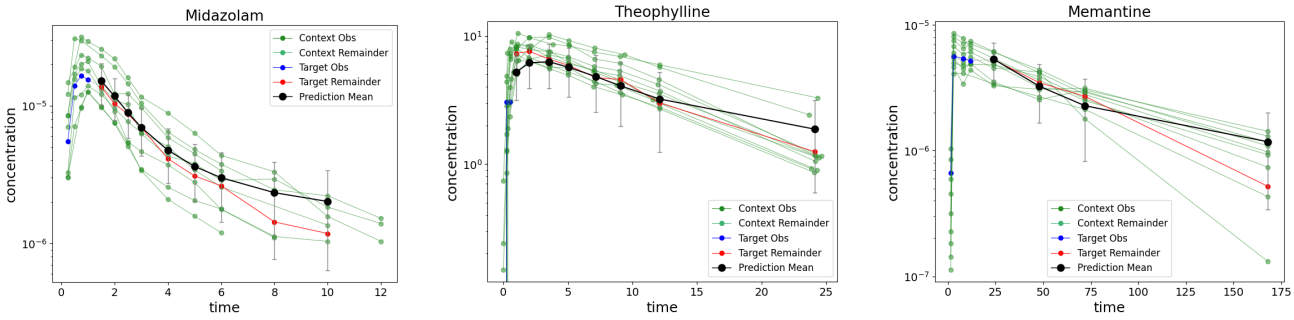
Thus, our methodology distinguishes two conditioning levels

1. *Study-level conditioning* on $\mathcal{S}$, which is always present and encodes population variability and sampling patterns. In `PFF`, $\mathcal{S}$ enters the vector field via attention-based mechanisms.
2. *Individual-level conditioning* on an optional prefix of measurements for a new individual. When a prefix is available, `PFF` performs forecasting by transporting uncertainty only in the unobserved continuation. When no prefix is available, `PFF` reduces to population synthesis.

Accordingly, we define a context space $\mathscr{C} = \mathscr{S} \times \mathscr{O}$, where $\mathscr{S}$ denotes the set of studies, and $\mathscr{O}$ collects the information available for a new individual. In *population synthesis* we take $\mathscr{O} = \{\varnothing\}$ (no individual measurements). In *forecasting* we use a nontrivial element $o \in \mathscr{O}$ defined below.

A new individual and prefix split. We distinguish the observed study $\mathcal{S}$ from a new *virtual* individual $\mathcal{D}^{\mathrm{n}} = (\mathbf{u}^{\mathrm{n}}, \boldsymbol{\tau}^{\mathrm{n}}, \mathbf{y}^{\mathrm{n}})$, for which we may observe a short prefix of measurements. We introduce a split index $p \in \{0, \dots, P\}$, which partitions the observations into a revealed *past* and an unobserved *future*. Writing $\mathbf{y}^{\mathrm{n}} = (\mathbf{y}^{\mathrm{n}}_{\mathtt{P}}, \mathbf{y}^{\mathrm{n}}_{\mathtt{F}})$ and $\boldsymbol{\tau}^{\mathrm{n}} = (\boldsymbol{\tau}^{\mathrm{n}}_{\mathtt{P}}, \boldsymbol{\tau}^{\mathrm{n}}_{\mathtt{F}})$, we define

$$\mathbf{y}^{\mathrm{n}}_{\mathtt{P}} = (y^{\mathrm{n}}_1, \dots, y^{\mathrm{n}}_p), \quad \mathbf{y}^{\mathrm{n}}_{\mathtt{F}} = (y^{\mathrm{n}}_{p+1}, \dots, y^{\mathrm{n}}_{T_{\mathrm{n}}}),$$



(a) Midazolam (b) Theophylline (c) Memantine

Figure 3: Predictive plots for three representative compounds. Conditioned on the population study context (green), the model uses the partial observations (blue) of a new patient to sample forecasts (red) from the predictive distribution, with the predictive mean (black) shown for reference.

and similarly for $\boldsymbol{\tau}_{\mathrm{P}}^{\mathrm{n}}, \boldsymbol{\tau}_{\mathrm{F}}^{\mathrm{n}}$. Setting $p = 0$ corresponds to population synthesis, so $o = \varnothing$ and $c = (\mathcal{S}, \varnothing)$. For $p > 0$, we perform *forecasting*, with $o = (\mathbf{u}^{\mathrm{n}}, \boldsymbol{\tau}_{\mathrm{P}}^{\mathrm{n}}, \mathbf{y}_{\mathrm{P}}^{\mathrm{n}}, p) \in \mathscr{O}$, and $c = (\mathcal{S}, o) \in \mathscr{C}$. In what follows, we drop the superscript n for new individuals whenever the meaning is clear from context.

Past-future decomposition in function space. Let $\mathcal{T} = [0, T]$ and let $\mathcal{F} = C([0, T])$ be a Hilbert space of continuous concentration trajectories. This choice reflects the physical fact that PK concentration trajectories are continuous in time, and it allows us to speak meaningfully about values at a split time. For a given prefix length p, we define the split time τ_p and the corresponding intervals $\mathcal{T}_{\mathrm{P}} = [0, \tau_p]$, and $\mathcal{T}_{\mathrm{F}} = [\tau_p, T]$. We represent the full trajectory as a pair

$$f \equiv (f_{\mathrm{P}}, f_{\mathrm{F}}), \qquad f_{\mathrm{P}} : \mathcal{T}_{\mathrm{P}} \to \mathbb{R}, \qquad f_{\mathrm{F}} : \mathcal{T}_{\mathcal{F}} \to \mathbb{R},$$

and we identify the full function space with the product $\mathcal{F} = \mathcal{F}_{\mathrm{P}} \times \mathcal{F}_{\mathrm{F}}$, where $\mathcal{F}_{\mathrm{P}}$ and $\mathcal{F}_{\mathrm{F}}$ are Hilbert spaces of functions on $\mathcal{T}_{\mathrm{P}}$ and $\mathcal{T}_{\mathrm{F}}$, respectively. To ensure the pair represents a single continuous trajectory, we impose compatibility at intersection, i.e., $f_{\mathrm{P}}(\tau_p) = f_{\mathrm{F}}(\tau_p)$. Intuitively, f_{P} is the revealed prefix segment and f_{F} is the unobserved continuation.

What is more, we define $\tau_0 = 0$ so that, when $p = 0$, the "past" interval collapses, and the "future" interval corresponds to the full domain. This makes population synthesis a special case of the same construction.

Simulator-induced target measure.

Our simulator P^{sim} first samples latent kinetic parameters (physics) and study designs (dosing, covariates, sampling times). Based on these, it generates continuous concentration curves f and compiles them into discrete studies $\mathcal{S}$ (which contain discrete observations of multiple curves). Crucially, we cannot query the simulator to generate curves for a specific, pre-defined study $\mathcal{S}$; we are restricted to the studies it stochastically generates. Our learning task is therefore to use generated joint pairs $(\mathcal{S}, f)$ to approximate the conditional measure $\nu^{\mathrm{sim}}(\cdot \mid \mathcal{S})$, enabling us to generalize to arbitrary studies later. To extend this framework to forecasting, where an individual is partially observed, we decompose the trajectory as $f = (f_{\mathrm{P}}, f_{\mathrm{F}})$. This joint law defines the *past marginal* $\nu_{\mathrm{P}}^{\mathrm{sim}}(\cdot \mid \mathcal{S}) \in \mathscr{P}(\mathcal{F}_{\mathrm{P}})$, and a corresponding *future conditional* $\nu_{\mathrm{F}}^{\mathrm{sim}}(\cdot \mid \mathcal{S}, f_{\mathrm{P}}) \in \mathscr{P}(\mathcal{F}_{\mathrm{F}})$. Informally, $\nu_{\mathrm{F}}^{\mathrm{sim}}$ describes plausible continuations compatible with a given prefix f_{P}. In practice, accessing f_{P} directly is impossible, and the individual-level context enters through the observations $o = (\boldsymbol{\tau}_{\mathrm{P}}^{n}, \mathbf{y}_{\mathrm{P}}^{n}, \mathbf{u}^{n})$. The ojective is to learn to sample from these target conditionals, i.e. $\nu^{\mathrm{sim}}(\cdot \mid \mathcal{S})$ for synthesis and $\nu_{\mathrm{F}}^{\mathrm{sim}}(\cdot \mid \mathcal{S}, o)$ for forecasting, for *any* observed context.

Reference measure (unified synthesis and forecasting). To train a single model that covers both modes, we use a reference measure on $\mathcal{F}_{\mathrm{F}}$ that depends on whether a prefix is available. Let $\mathrm{GP}_{\mathcal{T}_{\mathrm{F}}}(\cdot)$ denote a Gaussian-process prior on $\mathcal{T}_{\mathrm{F}}$, and let $\mathrm{GP}_{\mathcal{T}_{\mathrm{F}}}(\cdot \mid \mathbf{u}, \mathbf{y}_{\mathrm{P}})$ denote the corresponding GP posterior restricted to $\mathcal{T}_{\mathrm{F}}$. We define

$$\eta_{\mathrm{F}}(\cdot \mid o, p) := \begin{cases} \mathrm{GP}_{\mathcal{T}_{\mathrm{F}}}\big(\cdot \mid \mathbf{u}, \mathbf{y}_{\mathrm{P}}\big) & \text{if } \ p > 0, \\ \mathrm{GP}_{\mathcal{T}_{\mathrm{F}}}\big(\cdot\big) & \text{if } \ p = 0, \end{cases} \tag{12}$$

such that $\eta_{\mathrm{F}}(\cdot \mid o, p) \in \mathscr{P}(\mathcal{F}_{\mathrm{F}})$. Thus, in population synthesis ($p = 0$) the flow starts from a GP prior over full trajectories (since $\mathcal{F}_{\mathrm{F}} \equiv \mathcal{F}$), whereas in forecasting ($p > 0$) it starts from a GP posterior that extrapolates from the observed prefix. In both cases, we use the same trainable vector field conditioned on the study $\mathcal{S}$. The only additional change is whether the input includes an individual prefix, which alters the reference measure and the conditioning context.

Conditional flow matching for PK. The data generation process of Section 4.1 yields samples $\mathcal{D}^{\mathrm{n}}, \mathcal{S} \sim P^{\mathrm{sim}}$. Let us now sample $p \sim \mathrm{Uniform}(0, \ldots, P)$ and introduce the *finite-dimensional* flow variables $\mathbf{z}_0 = (\mathbf{y}_{\mathrm{P}}^{\mathrm{n}}, \mathbf{x}_{\mathrm{F}})$ and $\mathbf{z}_1 = (\mathbf{y}_{\mathrm{P}}^{\mathrm{n}}, \mathbf{y}_{\mathrm{F}}^{\mathrm{n}})$. Here $\mathbf{x}_{\mathrm{F}} = \big(f_{\mathrm{F}}(\tau_{p+1}^{\mathrm{n}}), \ldots, f_{\mathrm{F}}(\tau_{T_{\mathrm{n}}}^{\mathrm{n}})\big)$, where $f_{\mathrm{F}} \sim \eta_{\mathrm{F}}$ is sampled from the reference measure (Eq. 12).

Algorithm 1: Training of Prior-Fitted Functional Flows

Input : Simulator P^{sim}, Network v_θ, Step size η.
Return : Trained parameters θ

while *not converged* **do**
- $(\mathcal{S}, \mathbf{y}^{\text{n}}, \boldsymbol{\tau}^{\text{n}}) \sim P^{\text{sim}}$ // Sample from Simulator
- $p \sim \mathcal{U}\{0, \dots, |\boldsymbol{\tau}^{\text{n}}| - 1\}$ // Sample Split
- $\mathbf{y}^{\text{n}} \to (\mathbf{y}_{\text{P}}, \mathbf{y}_{\text{F}})$ and $\boldsymbol{\tau}^{\text{n}} \to (\boldsymbol{\tau}_{\text{P}}, \boldsymbol{\tau}_{\text{F}})$ // Partition Data
- $c = (\mathscr{S}, \mathbf{u}^{\text{n}}, \boldsymbol{\tau}_{\text{P}}, \mathbf{y}_{\text{P}})$ // Set Context
- $\mathbf{x}_{\text{F}} \sim \text{GP}(\boldsymbol{\tau}_{\text{F}} \mid \mathbf{y}_{\text{P}}, \boldsymbol{\tau}_{\text{P}})$ // Sample Reference
- $\mathbf{z}_0 = (\mathbf{y}_{\text{P}}, \mathbf{x}_{\text{F}}), \mathbf{z}_1 = (\mathbf{y}_{\text{P}}, \mathbf{y}_{\text{F}})$ // Set Start and End
- $t \sim \mathcal{U}(0, 1), \mathbf{z}_t \sim \mathcal{N}(t\mathbf{z}_1 + (1-t)\mathbf{z}_0, \sigma^2_{\min} I)$
- $u_t = (\mathbf{0}, \mathbf{y}_{\text{F}} - \mathbf{x}_{\text{F}})$ // Target Velocity
- $\mathcal{L}(\theta) = \|\mathbf{M}_{\text{F}} \odot (v_\theta(\mathbf{z}_t, t, \mathcal{S}) - u_t)\|^2$ // Loss
- $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$

Similar to Eq. 4, we now introduce the (finite-dimensional) Gaussian conditional path

$$\mathbf{z}_t \sim p_t(\mathbf{z} \mid \mathbf{z}_0, \mathbf{z}_1) = \mathcal{N}\big(t\,\mathbf{z}_1 + (1-t)\mathbf{z}_0,\ \sigma^2_{\min} I\big), \quad (13)$$

and the associated target, triangular vector field $v_t(\mathbf{z} \mid \mathbf{x}_{\text{F}}, \mathbf{y}^{\text{n}}_{\text{F}}) = (\mathbf{0}, \mathbf{y}^{\text{n}}_{\text{F}} - \mathbf{x}_{\text{F}})$. Using this construction, we train a neural operator vector field $v^\theta_t(\mathbf{z}, c)$ by minimizing the conditional flow matching loss

$$\mathcal{L}(\theta) = \mathbb{E}_{t,p,\mathbf{x}_{\text{F}},\mathbf{y}^{\text{n}}_{\text{F}}}\left[\left\|\mathbf{M}_p \odot v^\theta_t(\mathbf{z}_t, c) - (\mathbf{y}^{\text{n}}_{\text{F}} - \mathbf{x}_{\text{F}})\right\|^2\right],$$

where $\mathbf{M}_p$ is a prefix-dependent mask that zeros out the "past" of the learnable vector field.

Vector-field parameterization. We parameterize the transport through a time-dependent triangular velocity field $v^\theta : [0,1] \times \mathscr{S} \times \mathcal{F}_{\text{F}} \to \mathcal{F}_{\text{F}}$. The model is an *operator encoder–decoder transformer* that mirrors the standard sequence-to-sequence architecture of [Vaswani et al., 2017]. A *context encoder* maps the context $\mathscr{C}$ to a set of subject-level representations using multi-head self-attention, with block-diagonal masking to prevent information flow across subjects. A *target decoder* takes the flow-interpolated target individual $\mathbf{z}_t$ and applies non-causal self-attention over its past and future representations, followed by cross-attention blocks over the study-encoder outputs. The observation times are treated as positional encoding. One key departure from the standard transformer architecture is that both encoder and decoder operate on discretized functions over observation times that are subject-specific and not aligned to a common regular time grid. All attention operations are therefore replaced by their *continuum operator* counterparts [Calvello et al., 2024], which correct for the grid irregularities via a trapezoidal quadrature approximation. For further details on the architecture see appendix B.

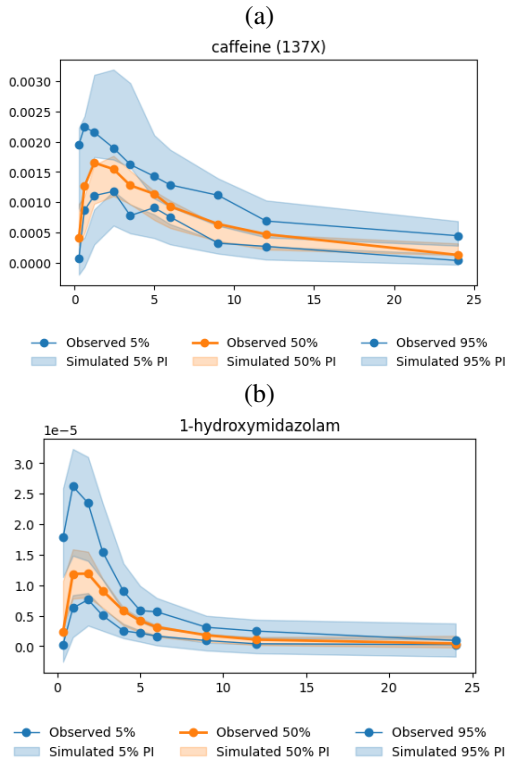


Figure 4: Visual predictive checks (VPCs) for (a) caffeine and (b) 1-hydroxy-midazolam. Each panel compares observed concentrations to simulation-derived prediction intervals.

5 RESULTS

In this section, we describe the empirical evaluation datasets, baselines, ablations, and metrics, and then discuss the main numerical results.

5.1 DATASETS

Evaluation data were obtained from the open-access PK-DB repository, which aggregates pharmacokinetic measurements from clinical and pre-clinical studies [Grzegorzewski et al., 2021]. From PK-DB, we curated plasma concentration–time measurements from a Phase I study in healthy volunteers investigating the safety and dosing of a combination of drugs and their metabolites [Lenuzza et al., 2016]. This dataset comprises 18 compounds, including parent drugs and primary metabolites, with 2019 valid concentration–time observations. Sampling frequencies vary substantially both within and across compounds, resulting in sparse and irregularly sampled time series; each compound is measured in 10 individuals, with up to 15 observations per individual. We also include two pharmacokinetic datasets on indometacin and theophylline (`Indometh` and `Theoph` from the R `datasets` package) [R Core Team, 2025]. To-

Table 1: **Comparison of Log-RMSE Across Models.** For each compound, we report mean log-RMSE with standard deviation in parentheses. The best-performing model for each compound is highlighted in **bold**.

Compound	NLME	Transf.	NODE	AICMET	PFF
caffeine	0.294 ± 0.294	0.451 ± 0.258	0.432 ± 0.209	0.399 ± 0.205	$\mathbf{0.174 \pm 0.086}$
dextromethorphan	0.752 ± 0.752	$\mathbf{0.395 \pm 0.163}$	0.605 ± 0.248	0.595 ± 0.364	0.963 ± 0.426
digoxin	$\mathbf{0.300 \pm 0.300}$	0.683 ± 0.242	0.661 ± 0.315	0.691 ± 0.464	0.375 ± 0.216
indometacin	0.582 ± 0.582	0.429 ± 0.328	0.350 ± 0.291	0.536 ± 0.203	$\mathbf{0.245 \pm 0.170}$
memantine	$\mathbf{0.343 \pm 0.343}$	0.513 ± 0.335	0.554 ± 0.386	0.475 ± 0.341	0.404 ± 0.388
midazolam	0.596 ± 0.596	0.288 ± 0.173	0.352 ± 0.224	$\mathbf{0.264 \pm 0.151}$	0.295 ± 0.300
omeprazole	1.300 ± 1.300	0.842 ± 0.400	1.003 ± 0.618	0.936 ± 0.942	$\mathbf{0.593 \pm 0.161}$
paracetamol	$\mathbf{0.299 \pm 0.299}$	0.447 ± 0.274	0.582 ± 0.302	0.352 ± 0.187	0.309 ± 0.220
repaglinide	0.479 ± 0.479	0.709 ± 0.268	0.693 ± 0.290	$\mathbf{0.339 \pm 0.311}$	0.681 ± 0.155
rosuvastatin	$\mathbf{0.383 \pm 0.383}$	0.502 ± 0.205	0.535 ± 0.303	0.558 ± 0.286	0.495 ± 0.336
theophylline	0.732 ± 0.732	0.356 ± 0.141	0.327 ± 0.144	0.274 ± 0.109	$\mathbf{0.184 \pm 0.110}$
tolbutamide	0.723 ± 0.723	0.678 ± 0.339	0.578 ± 0.263	0.517 ± 0.254	$\mathbf{0.421 \pm 0.214}$
1-hydroxy-midazolam		0.567 ± 0.327	0.737 ± 0.381	0.491 ± 0.309	$\mathbf{0.422 \pm 0.205}$
4-hydroxy-tolbutamide		0.354 ± 0.174	0.353 ± 0.182	0.335 ± 0.164	$\mathbf{0.293 \pm 0.153}$
5-hydroxy-omeprazole		1.286 ± 0.472	1.474 ± 0.745	1.188 ± 0.451	$\mathbf{1.026 \pm 0.356}$
dextrorphan		0.733 ± 0.445	0.834 ± 0.284	$\mathbf{0.563 \pm 0.287}$	0.585 ± 0.487
hydroxy-repaglinide		0.069 ± 0.217	$\mathbf{0.049 \pm 0.156}$	0.076 ± 0.240	0.070 ± 0.221
omeprazole sulfone		$\mathbf{1.054 \pm 0.297}$	1.390 ± 0.488	1.082 ± 0.509	1.156 ± 0.355
paracetamol glucuronide		0.357 ± 0.192	0.297 ± 0.169	0.307 ± 0.184	$\mathbf{0.265 \pm 0.114}$
paraxanthine		0.413 ± 0.181	0.287 ± 0.135	0.312 ± 0.138	$\mathbf{0.273 \pm 0.143}$

gether, these 20 empirical compound-level datasets reflect the small-cohort and sparsely sampled regime targeted by PFF.

5.2 BASELINES AND ABLATIONS

We compare Prior-Fitted Functional Flows (PFF) against one classical pharmacometric baseline and three neural baselines: nonlinear mixed-effects modeling (NLME), NODE-PK, Transformer, and `AICMET`. The NLME baseline is fitted study-by-study for each parent compound using one- and two-compartment models with first-order absorption, inter-individual variability, and combined residual error models, selecting the final model by AIC. Metabolites are excluded from this comparison because they lack explicit dosing inputs. All neural models are pretrained on synthetic PK studies and evaluated in context on empirical studies without dataset-specific fine-tuning. NODE-PK extends latent neural ODEs [Rubanova et al., 2019] with explicit dosing information through an RNN encoder and neural ODE dynamics [Lu et al., 2021], but treats individuals independently and does not aggregate study-level information. The Transformer baseline replaces NODE-PK's neural ODE dynamics with a functional-attention decoder, but does not use the full hierarchical mixed-effects representation of `AICMET`. `AICMET` [Ojeda et al., 2026] is a hierarchical neural-process baseline that encodes study-level and individual-level effects through latent mixed/random-effect representations and decodes concentration functions with functional attention. Relative to `AICMET`, PFF learns a direct transport between measures over concentration functions and parameterizes the transport vector field with continuum-attention neural operators rather than a functional-attention decoder.

We additionally evaluate two PFF ablations. `PFF-white-noise` replaces the GP posterior reference measure used for partially observed individuals with a white-noise reference, isolating the effect of prefix-conditioned functional initialization. `PFF-std-attn` replaces the continuum-attention neural operator in the vector-field architecture by the standard functional attention mechanism used in `AICMET` while keeping the flow-matching objective fixed.

5.3 METRICS

Forecasting evaluation. Forecasting performance is assessed in a leave-one-individual-out setting. For subject i, predictions are conditioned on the study context $\mathcal{S}$, dosing inputs $\mathbf{u}^i$, and the first four PK samples $\mathcal{D}^i_{\mathrm{C}}$. The resulting posterior predictive trajectories are evaluated on the remaining samples $\mathcal{D}^i_{\mathrm{T}}$ using log-RMSE.

Generative evaluation. We assess sample quality in empirical and synthetic settings. For empirical data, we use leave-one-individual-out evaluation across all real studies, where the model is conditioned on the remaining individuals and generates one trajectory for the held-out individual, yielding a generated set matched in size to the empirical dataset. For the synthetic evaluation, no leave-one-out construction is needed. Instead, each synthetic study provides a study context together with a reference population of target individuals sampled directly from the simulator. Conditioned on the same context, the model generates a matching synthetic population, and the generated trajectories are compared directly against the corresponding oracle trajectories. After constructing these empirical and synthetic comparison pairs, we quantify discrepancy using two complemen-

Table 2: **Generative sample fidelity and ablations.** AUC values closer to 0.5 and MMD values closer to zero indicate better agreement between generated and reference trajectories. Values are reported as mean ± standard error over seeds when available.

Model	AUC	MMD
Synthetic simulator-controlled		
AICMET	0.54 ± 0.01	0.002 ± 0.001
PFF	$\mathbf{0.52 \pm 0.01}$	$\mathbf{0.001 \pm 0.001}$
PFF-std-attn	0.61 ± 0.03	0.004 ± 0.004
PFF-white-noise	0.59 ± 0.05	0.004 ± 0.001
Empirical leave-one-individual-out		
AICMET	0.58 ± 0.01	–
PFF	$\mathbf{0.57 \pm 0.02}$	–

tary two-sample metrics: classifier AUC from a two-sample test, where a value of 0.5 corresponds to perfect agreement, and signature-kernel MMD [Király and Oberhauser, 2019], where values closer to zero indicate better agreement. Since we use an unbiased finite-sample estimator, small negative MMD estimates can occur. We also report visual predictive checks (VPCs), comparing simulation percentiles against observed concentrations [Holford, 2005]; implementation details are provided in the Supplementary Material.

5.4 DISCUSSION OF NUMERICAL RESULTS

Table 1 summarizes forecasting performance. PFF achieves the lowest or competitive log-RMSE across the evaluated compounds in the majority of cases, indicating that the learned posterior predictive operator transfers effectively from simulated pretraining studies to sparse empirical PK datasets. Figure 3 illustrates this behavior for partially observed individuals: the model does not merely interpolate observed prefixes, but uses the cohort-level context and individual prefix to infer a distribution over plausible future concentration functions.

Table 2 evaluates zero-shot population synthesis and the main PFF ablations. Compared with AICMET, PFF yields AUC values closer to the ideal value of 0.5 in both empirical and synthetic settings, indicating that its generated trajectories are harder to distinguish from reference trajectories. The synthetic comparison confirms the same trend under simulator-controlled ground truth, while the empirical leave-one-individual-out protocol shows that the improvement transfers to real PK studies. Figure 4 provides the corresponding visual predictive checks. The simulated percentiles align closely with the empirical concentration distributions across study cohorts.

The ablations clarify the source of this improvement. Replacing the GP posterior reference with an unconditional white-noise reference worsens MMD, suggesting that prefix-conditioned reference measures provide a useful initialization before transport. Replacing continuum operator attention with standard attention has a larger effect on AUC, making PFF-std-attn nearly indistinguishable from AICMET. Thus, the empirical gain should not be attributed to function-space flow matching alone, but to its combination with GP posterior references and continuum operator attention for irregular observation grids.

Finally, PFF shifts computational cost from repeated study-specific optimization to one up-front pretraining stage. At deployment time, the pretrained model can be applied off-the-shelf to new studies without retraining, whereas classical NLME workflows often require iterative model development, model comparison, and manual diagnostic inspection. Although PFF generates by integrating a flow ODE ($\sim$100 steps) rather than in a single amortized pass, inference remains cheap in absolute terms. Sampling 100 virtual individuals for a study takes approximately 3.5 s on a single A100 GPU (around 11$\times$ the AICMET single-pass baseline), a one-off cost per query that is negligible beside the repeated per-study optimization it replaces. This makes PFF particularly relevant in early-development settings where datasets are small, irregular, and repeatedly encountered across compounds. A detailed account of hyperparameters, inference times, calibration diagnostics, and failure cases is provided in the Supplementary Material.

6 CONCLUSION

We introduced Prior-Fitted Functional Flows (PFF), a generative foundation model for pharmacokinetics that enables zero-shot population synthesis and individual forecasting without manual parameter tuning. By learning functional vector fields conditioned on entire study populations, PFF captures complex, multi-modal distributions that classical methods often miss. Crucial to our methodology is the construction of a large-scale, open-access literature benchmark, which we used to rigorously validate the physiological realism of our synthetic priors.

Limitations. While PFF demonstrates state-of-the-art accuracy, it currently relies on purely synthetic pre-training (though informed by our systematic literature-mined corpus). Incorporating real-world clinical data into the pretraining would further enhance its ability to capture the pathological variability of specific disease populations. Furthermore, our current validation focuses on single-dose studies; extending the framework to handle complex, irregular multi-dosing remains a key direction for future work. Finally, as with all generative models, there is a risk of generating plausible but incorrect trajectories in extremely sparse data regimes, necessitating careful uncertainty quantification in clinical applications.

ACKNOWLEDGEMENTS

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG) — Project-ID 318763901 — SFB1294 and by the DFG under Germany´s Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, EXC-2046/2, project ID: 390685689); 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. DAF is supported by DOE grant DOE-SC0010008. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 using NERSC award HEPERCAP0027491

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.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

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.

Sathya Kamesh Bhethanabhotla, Omar Swelam, Julien Siems, David Salinas, and Frank Hutter. Mamba4cast: Efficient zero-shot time series forecasting with state space models. *arXiv preprint arXiv:2410.09385*, 2024.

Marin Biloš, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, and Stephan Günnemann. Modeling temporal data as continuous functions with stochastic process diffusion. In *International Conference on Machine Learning*, pages 2452–2470. PMLR, 2023.

Edoardo Calvello, Nikola B Kovachki, Matthew E Levine, and Andrew M Stuart. Continuum attention for neural operators. *arXiv preprint arXiv:2406.06486*, 2024.

Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 03 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp163. URL https://doi.org/10.1093/bioinformatics/btp163.

Samuel Colvin, Eric Jolibois, Hasan Ramezani, Adrian Garcia Badaracco, Terrence Dorsey, David Montague, Serge Matveenko, Marcelo Trylesinski, Sydney Runkle, David Hewitt, Alex Hall, and Victorien Plot. Pydantic validation. https://github.com/pydantic/pydantic, January 2024. URL https://docs.pydantic.dev/latest/.

Giuseppe Da Prato and Jerzy Zabczyk. *Stochastic equations in infinite dimensions*, volume 152. Cambridge university press, 2014.

Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha V Naidu, and Colin White. Forecastpfn: Synthetically-trained zero-shot forecasting. *Advances in Neural Information Processing Systems*, 36, 2024.

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.

Milo Gibaldi and Donald Perrier. *Pharmacokinetics*. crc Press, 1982.

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.

Henrik Häggström, Sebastian Persson, Marija Cvijovic, and Umberto Picchini. Simulation-based inference for stochastic nonlinear mixed-effects models with applications in systems biology. *arXiv preprint arXiv:2504.11279*, 2025.

Christoph Jürgen Hemmer and Daniel Durstewitz. True zero-shot inference of dynamical systems preserving long-term statistics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=RE97LT26w8.

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. In *The Eleventh International Conference on Learning Representations*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637 (8045):319–326, 2025.

Johannes R. Hübers, Maximilian Mauel, David Berghaus, Patrick Seifner, and Ramses J Sanchez. Foundation inference models for ordinary differential equations. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ZBHDZDaG0h.

T Jahn, J Chemseddine, P Hagemann, C Wald, and G Steidl. Trajectory generator matching for time series. *arXiv preprint arXiv:2505.23215*, 2025.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional flow matching. *arXiv preprint arXiv:2305.17209*, 2023.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Dynamic conditional optimal transport through simulation-free flows. *Advances in Neural Information Processing Systems*, 37:93602–93642, 2024.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

Alex Kipnis, Marcel Binz, and Eric Schulz. metabeta–a fast neural model for bayesian mixed-effects regression. *arXiv preprint arXiv:2510.07473*, 2025.

Franz J Király and Harald Oberhauser. Kernels for sequentially ordered data. *Journal of Machine Learning Research*, 20(31):1–45, 2019.

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow matching with gaussian process priors for probabilistic time series forecasting. *arXiv preprint arXiv:2410.03024*, 2024.

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.

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.

Naomi Most. Metapub: A python library for accessing biomedical literature and databases, 2014. URL https://metapub.org. Available at: https://pypi.org/project/metapub/.

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.

National Center for Biotechnology Information. Pubmed central (PMC). https://www.ncbi.nlm.nih.gov/pmc/, 2000. Accessed: 2025-11-25.

Cesar Ojeda, Ramses J Sanchez, Wilhelm Huisinga, and Niklas Hartung. Amortized in-context mixed effect transformer models: A zero-shot approach for pharmacokinetics. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. URL https://openreview.net/forum?id=pdamyf7o8W.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.

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/.

Leonard Richardson. Beautiful soup (python package), November 2025. URL https://pypi.org/project/beautifulsoup4/.

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.

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.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Lajoie, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

U.S. Food and Drug Administration. Bioequivalence studies with pharmacokinetic endpoints for drugs submitted under an abbreviated new drug application. Draft Guidance for Industry FDA-2013-D-1464, U.S. Department of Health and Human Services, Food and Drug Administration, Silver Spring, MD, August 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Qi Wang, Marco Federici, and Herke van Hoof. Bridge the inference gaps of neural processes via expectation maximization. *arXiv preprint arXiv:2501.03264*, 2025.

Prior Fitted Functional Flows (Supplementary Material)

César Ojeda[1] **Niklas Hartung**[1] **Purity Kamene Kavwele**[1] **Tim Jahn**[2] **Piyush Kumar**[1] **Marian Klose**[3]

Wilhelm Huisinga[1] **Ramsés J. Sánchez**[4] **Darius A. Faroughy**[5]
[1]University of Potsdam [2]Technical University Berlin [3]Freie Universität Berlin
[4]Lamarr Institute & University of Bonn [5]NHETC, Rutgers University, USA

A GENERATIVE DATA MODEL AND EMPIRICAL VALIDATION

In this section, we introduce a hierarchical data generation framework that combines mechanistic compartmental pharmacokinetic (PK) systems with stochastic, time-varying parameters. We utilize this model to generate a large-scale synthetic training dataset. We build upon the probabilistic generative framework introduced in the AICMET model Ojeda et al. [2026], described in Section A.1. While the original framework relied on internal meta-analyses for calibration, a core contribution of this work is the rigorous validation of the generator against a large-scale dataset of pharmacokinetic parameters automatically mined from clinical literature (Section A.2).

A.1 MECHANISTIC GENERATIVE PROCESS

The data generation process is hierarchical: we first sample a study configuration (population priors), then sample individual subjects, and finally generate their continuous drug concentration trajectories .

Compartmental Pharmacokinetic Systems. Each simulated study 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

$$\mathbf{X}(t) = \Big(X_{\text{gut}},\, X_c,\, X_{\text{per},1}, \ldots, X_{\text{per},P}\Big)^\top. \tag{14}$$

which solves the ODE system

$$\begin{aligned}
\dot{X}_{\text{gut}} &= -k_a\, X_{\text{gut}}, \\
\dot{X}_c &= k_a X_{\text{gut}} - \Big(k_e + \sum_{j=1}^{P} k_j^+\Big) X_c + \sum_{j=1}^{P} k_j^- X_{\text{per},j}, \\
\dot{X}_{\text{per},j} &= k_j^+ X_c - k_j^- X_{\text{per},j},
\end{aligned} \tag{15}$$

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.

Stochastic parameter model. To capture realistic inter- and intra-individual variability beyond standard parametric families, we model the log-kinetic parameters $\theta_i(t) = \log(V^i, k_a^i, k_e^i, , k_1^{i+}, k_1^{i-}, ..., k_P^{i+}, k_P^{i-})$ as independent Ornstein-Uhlenbeck (OU) processes :

$$d\theta_{i,k}(t) = -\lambda_k(\theta_{i,k}(t) - \mu_{i,k})\; dt + \sigma_k\, dW_{i,k}(t), \tag{16}$$

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

The hyperparameters governing these processes, namely the long-term means $\mu_{i,k}$, reversion speeds λ_k, and volatilities σ_k, are drawn hierarchically from broad uniform priors calibrated to cover physiologically plausible ranges, see Section A.2. Finally, to mimic clinical trial conditions, sparse observations are generated by sampling the continuous trajectories at irregular intervals derived from characteristic timescales $T_{\texttt{peak}}$ and $T_{\texttt{half}}$ and applying proportional measurement noise, see Ojeda et al. [2026].

A.2 AUTOMATED EXTRACTION OF PHARMACOKINETIC PARAMETERS FROM THE LITERATURE

To validate that the generative model produces physiologically realistic concentration-time profiles, we constructed a reference dataset of real-world pharmacokinetic (PK) parameters. Since raw subject-level data is rarely available, we utilized a Large Language Model (LLM) pipeline to mine reported summary statistics from *bioequivalence (BE) studies* published in the open-access biomedical literature. BE studies typically compare a *test* formulation (e.g., a generic) against a *reference* formulation (e.g., an originator) under defined dosing conditions to enable abbreviated regulatory approval if predefined PK criteria relative to the reference product are met. These studies are a particularly well-suited reference because they follow standardized protocols, typically enroll healthy volunteers under controlled conditions, and are required by regulatory agencies to report a consistent set of PK summary statistics. Furthermore, dense sampling ensures that individual PK profiles are well characterized, allowing for the derivation of reliable PK metrics [U.S. Food and Drug Administration, 2021].

A.2.1 Data Acquisition Pipeline

Step 1: Literature Retrieval

We first constructed a candidate corpus ($N = 200$) of BE-related publications from PubMed Central (PMC) [National Center for Biotechnology Information, 2000] by querying the web interface using the search term `"bioequivalence"[title]`. The resulting records contained bibliographic metadata, including PubMed identifiers (PMIDs) and PMC identifiers (PMCIDs). For each of the top 200 retrieved articles, we used the `PubMedFetcher` function from `metapub` 0.6.4 [Most, 2014] to obtain the structured abstract via the PMID and the `Entrez.efetch` function from `Biopython` 1.86 [Cock et al., 2009] to retrieve the complete full-text XML from PMC via the PMCID. Full-text XML was parsed with `BeautifulSoup` 4.13.4 [Richardson, 2025] to obtain plain text.

Step 2: LLM-Based Triage

Not all retrieved papers reported primary PK trial data; many were reviews, commentaries, meta-analyses, or methodological discussions about BE aspects. To exclude these from the database in a time- and cost-efficient manner, we employed an LLM agent (`gpt-5-nano`) via OpenAI's `Responses API` in Python 3.12.6 receiving the title and abstract of the paper as user message. The agent was instructed via a detailed system message to classify papers as *Suitable* (True) only if they reported a single, distinct bioequivalence trial. Machine-readable outputs were enforced using a `Pydantic` 2.10.5 [Colvin et al., 2024] class in combination with the `structured outputs` feature of the `Responses API`, which guarantees that the model returns a Boolean flag rather than free text. This step discarded 86 papers and reduced the corpus to a set of 114 clinical BE trials suitable for further extraction.

Step 3: Structured Data Extraction

For each article that passed triage, we prompted the complete full-body text to the LLM with a different system prompt, instructing it to extract the following information:

- **Study-level metadata:** Compound name, number of subjects enrolled, and year of publication.
- **Dosing Protocol:** Route of administration (e.g., oral), formulation type (e.g., tablet, capsule), and amount and unit of administered dose.
- **Sampling design:** Time of the last blood sample collected (t_{last}).
- **PK characteristics:** Central tendency and dispersion of the *test* formulation for:
  - $C_{\max}$ (Maximum concentration)
  - $T_{\max}$ (Time to maximum concentration)
  - AUC_{0-t} (Area under the concentration–time curve up to the last measured time point)

To ensure machine-readable consistency, we again utilized `structured outputs` enforced via nested `Pydantic` models, yielding one JSON record for each paper. Where both fasting and fed-state results were reported, the LLM was

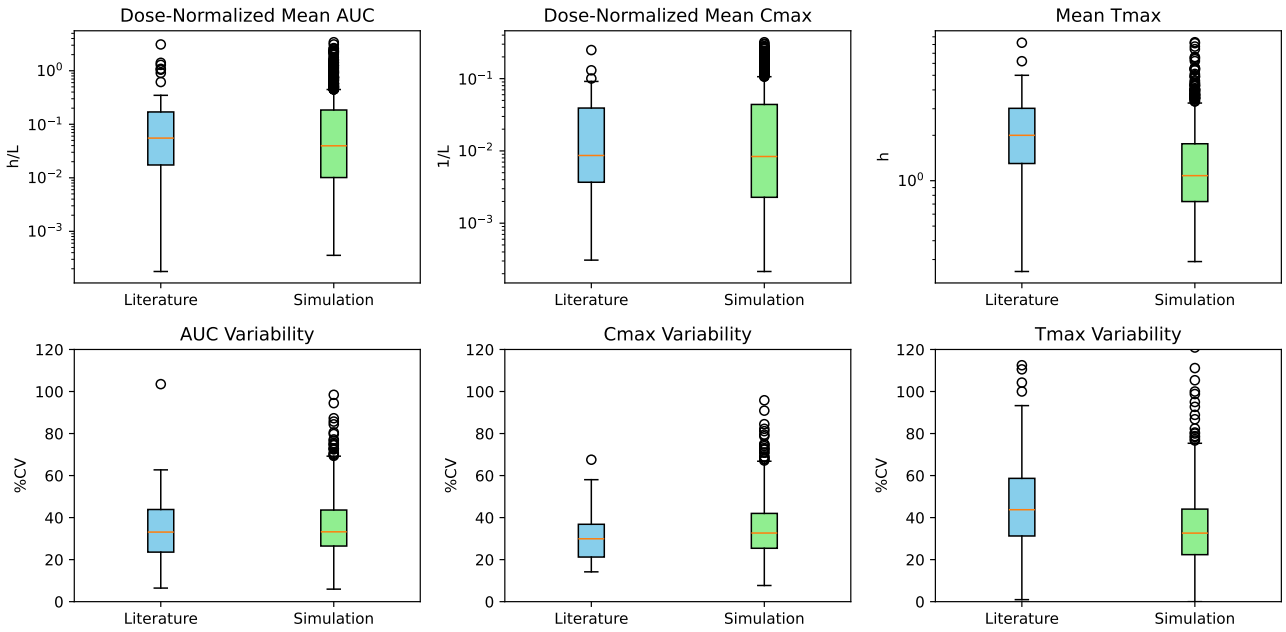


Figure 5: Comparison of three key PK characteristics (Mean/%CV) between literature and 1000 simulated datasets.

instructed to only take fasting values into account. The total API costs for this LLM-based workflow across all 200 screened and 114 classified papers was $0.31 USD.

A.3 POST-PROCESSING

The extracted JSON records were post-processed into a final analysis literature dataset via the following steps:

- Amongst the 114 BE trials, we kept only those 85 corresponding to oral administration[1];
- Units were harmonized across studies and records with incomplete or implausible values (e.g., $T_{\max} > t_{\text{last}}$ or missing units) were excluded;
- AUC_{0-t} and $C_{\max}$ were dose-normalized;
- The dispersion metrics were converted to coefficients of variation (%CV), which was either directly used if reported or calculated from central tendency and absolute dispersion metrics (using mean/median interchangeably and assuming range=4 SD).

The resulting dataset provided empirical reference distributions for dose-normalized AUC_{0-t}, dose-normalized $C_{\max}$ and $T_{\max}$ across a diverse set of compounds and doses, which we used to benchmark the marginal distributions produced by our stochastic generative model.

A.3.1 Benchmark of data generative model against literature

The parameters of the stochastic data-generating model were matched to the PK metrics obtained from the literature, as shown in Fig. 5.

B OPERATOR ENCODER-DECODER TRANSFORMER

In the following we describe in detail the model architecture. First we introduce the Continuum operator attention underlying our transformer architecture, then we describe the model's encoder and decoder.

[1]While our framework is capable of simulating intravenous administration as well, a focus on oral dosing allowed for a more meaningful comparison of $C_{\max}$ and $T_{\max}$

B.1 CONTINUUM OPERATOR ATTENTION

The standard attention mechanism implicitly treats all key positions as equally spaced, effectively applying a uniform Riemann-sum approximation to an underlying integral operator. On an irregular time grid this is a biased estimator. Following Calvello et al. [2024], we replace uniform averaging with a trapezoidal quadrature correction, turning each attention module into a consistent discretization of a continuum integral operator.

Quadrature weights. Given a sorted observation time grid $\tau_1 \le \tau_2 \le \cdots \le \tau_M$ of a subject (either in the context study or the target, depending on the module), define increments $\Delta\tau_k = \tau_k - \tau_{k-1}$ with $\Delta\tau_1 = 0$, and the trapezoidal weights

$$w_k = \tfrac{1}{2}\left(\Delta\tau_k + \Delta\tau_{k+1}\right), \quad k = 1, \ldots, M, \tag{17}$$

setting $\Delta\tau_{M+1} = 0$. Padding positions are assigned weight zero and the remaining weights are normalized to unit sum before being passed to the attention modules.

Operator attention. Let $\boldsymbol{Q} \in \mathbb{R}^{N_q \times d_A}$, $\boldsymbol{K} \in \mathbb{R}^{M \times d_A}$, $\boldsymbol{V} \in \mathbb{R}^{M \times d_A}$ be the query, key, and value matrices for a single attention head, where $d_A = d/H$ is the per-head dimension, with H the number of and d is the hidden dimension. The scaled dot-product attention is defined as

$$\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{K}^\top / \sqrt{d_A} + \boldsymbol{M} \tag{18}$$

where $\boldsymbol{M} \in \mathbb{R}^{N_q \times M}$ is the attention masking matrix. The operator attention output at query position i is given by:

$$\Big[\texttt{OpAttn}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}; \boldsymbol{M}, \boldsymbol{\Delta\tau})\Big]_i \;=\; \frac{\displaystyle\sum_{k=1}^{M} e^{S_{ik}} \cdot w_k \cdot V_k}{\displaystyle\frac{1}{2}\sum_{k=2}^{M}\left(e^{S_{ik}} + e^{S_{i,k-1}}\right)\Delta\tau_k}, \tag{19}$$

where the numerator is a midpoint-rule approximation of the operator $\int e^{q(\tau)\cdot k(\tau)}\, v(\tau)\, d\tau$ and the denominator is a trapezoidal approximation of the partition function $\int e^{q(\tau)\cdot k(\tau)}\, d\tau$. As the grid spacing vanishes uniformly, Eq. (19) converges to the normalized integral operator of Calvello et al. [2024]. In the limit of a uniform grid with spacing $1/M$, both reduce to standard softmax attention.

For *operator self-attention*, $(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$ are linear projections of the same input observations, and $\boldsymbol{\Delta\tau}$ are the quadrature weights of the observation's time grid. For *operator cross-attention*, $\boldsymbol{Q}$ is projected from the query observations, while $(\boldsymbol{K}, \boldsymbol{V})$ and $\boldsymbol{\Delta\tau}$ are projected from, and computed over, the key-value observations and their corresponding time grids. Explicitly:

$$\texttt{SelfOpAttn}(\boldsymbol{x}, \boldsymbol{\Delta\tau}, \boldsymbol{M}) = \texttt{OpAttn}(\boldsymbol{W_Q} \cdot \boldsymbol{x}, \boldsymbol{W_K} \cdot \boldsymbol{x}, \boldsymbol{W_V} \cdot \boldsymbol{x}\,; \boldsymbol{M}, \boldsymbol{\Delta\tau}), \tag{20}$$

$$\texttt{CrossOpAttn}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\Delta\tau}, \boldsymbol{M}) = \texttt{OpAttn}(\boldsymbol{W_Q} \cdot \boldsymbol{x}, \boldsymbol{W_K} \cdot \boldsymbol{y}, \boldsymbol{W_V} \cdot \boldsymbol{y}\,; \boldsymbol{M}, \boldsymbol{\Delta\tau}), \tag{21}$$

where $\boldsymbol{W_{Q,K,V}}$ are the corresponding (learnable) linear transformations to the matrices $(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$, respectively.

B.2 FLOW-TIME CONDITIONING AND TRIANGULAR PAST–FUTURE MASKING

Flow-time embedding. The ODE time $t \in [0, 1]$ is encoded as a d-dimensional sinusoidal Fourier feature vector with log-spaced frequencies,

$$t_{\text{emb}} = \Big[\sin(t\,\omega_1), \ldots, \sin(t\,\omega_{d/2}),\ \cos(t\,\omega_1), \ldots, \cos(t\,\omega_{d/2})\Big] \;\in\; \mathbb{R}^d, \tag{22}$$

where $\omega_k = f_{\max}^{-(k-1)/(d/2-1)}$ for $k = 1, \ldots, d/2$, giving frequencies log-spaced between 1 and $1/f_{\max}$. The embedding t_{emb} is added residually to every feature embedding in both encoder and decoder after each attention block, ensuring the velocity field is a smooth function of the ODE time at every network layer. Note that t is the *flow* ODE time and plays no role in the observation time axis τ; the two are kept strictly separate throughout.

Reference measure and GP posterior. As formalised in Eq. (14) of the main text, the reference measure $\eta_F(\cdot \mid o, p)$ depends on whether a past prefix is available.

- In *population synthesis* ($p = 0$), the flow's source is sampled from a zero-mean GP prior $\mathbf{z}_0 \sim \mathrm{GP}(0,\, k)|_{\boldsymbol{\tau}}$ with the standard RBF kernel evaluated at the target's time grid:

$$k(\tau, \tau') = \nu^2 \exp\!\big(-\frac{(\tau - \tau')^2}{4\ell^2}\big)\,. \tag{23}$$

  Here, the variance ν^2 and the length scale ℓ are model hyperparameters. Since the (normalized) concentrations take values in $\mathbb{R}^+$ we apply a softplus transformation $\mathbf{z}_0 \to \log(1 + \exp(\mathbf{z}_0))$ to the output of the GP that guarantees positive concentrations values along the flow trajectories.

- In *forecasting* ($p > 0$), the flow's source $\mathbf{z}_0 = (\mathbf{y}_\mathtt{P}, \mathbf{x}_\mathtt{F})$ is sampled from the GP *posterior* conditioned on the observed past $(\boldsymbol{\tau}_\mathtt{P}, \mathbf{y}_\mathtt{P})$:

$$\mathbf{x}_\mathtt{F} \sim \mathrm{GP}\big(\boldsymbol{\mu}_*(\boldsymbol{\tau}_\mathtt{F}),\ \boldsymbol{\Sigma}_*(\boldsymbol{\tau}_\mathtt{F}, \boldsymbol{\tau}_\mathtt{F})\big), \tag{24}$$

  where $\boldsymbol{\mu}_*$ and $\boldsymbol{\Sigma}_*$ are the posterior mean and covariance obtained via GP regression with (23). Furthermore, we apply a softplus transformation to the future $\mathbf{x}_\mathtt{F}$, leaving the past unchanged. The posterior already encodes continuity and monotonic behaviour compatible with early-phase PK profiles, providing a warm start for the flow that substantially reduces the variance of the flow-matching objective compared to an uninformed prior. In both cases, past-slot values are fixed to the observed concentrations in both $\mathbf{z}_0 = (\mathbf{y}_\mathtt{P}, \mathbf{x}_\mathtt{F})$ and $\mathbf{z}_1 = (\mathbf{y}_\mathtt{P}, \mathbf{y}_\mathtt{F})$ (see Eq. (15)), so the interpolation is trivial for past observations.

Triangular past–future masking. The prefix-dependent mask $\mathbf{M}_p$ in the flow-matching loss Eq. (16) zeros out the velocity field at all past observations:

$$v^\theta(t, \mathbf{z}_t, \mathcal{S}) \ \leftarrow\ v^\theta(t, \mathbf{z}_t, \mathcal{S}) \odot \mathbf{M}_p, \qquad \mathbf{M}_p = \mathbf{1}\big[\tau > \tau_p\big]. \tag{25}$$

This is the discrete implementation of the triangular map structure from Definition 3.1: no probability mass is transported at past time points, and their values therefore remain exactly equal to the observed prefix irrespective of the ODE time t. Combined with the GP-posterior source (24), the flow learns to move probability mass only in the unobserved future, while the past acts as a fixed conditioning signal entering through the decoder transformer (B.4).

B.3 CONTEXT STUDY ENCODER

A study $\mathcal{S} = \{\mathcal{D}^1, \ldots, \mathcal{D}^I\}$ contains I individuals. After normalization (described below), individual i contributes C time slots (zero-padded where fewer observations exist), yielding a total number of observations $N = I \cdot C$. Each observation in the context study is described by $\mathcal{D}^i_j = (\tau^i_j,\ y^i_j,\ a^i,\ r^i) \in \mathbb{R}^4$, $i \in [I]$, $j \in [C]$, where τ^i_j and y^i_j are observation time and concentration, and (a^i, r^i) are the dose amount and administration route. Concentrations are normalized per-study by the maximum concentration observed across all context individuals and time points; times are normalized by the maximum observation time in the study. Both scale factors are estimated from the context set $\mathcal{S}$ and applied consistently to all context and target quantities. The varying number of observation times per individual are tracked by a binary mask $\boldsymbol{m}^i \in \{0, 1\}^C$ distinguishing true observations from zero-padded entries.

Input embedding Each study observation $\mathcal{D}^i_j$ is lifted to the model dimension via a three-layer MLP $\Phi_\mathcal{S} : \mathbb{R}^4 \to \mathbb{R}^d$ interleaved with GELU activations and combined with a flow-time embedding (Section B.2):

$$\boldsymbol{h}^{i\,(0)}_j = \mathtt{LN}\Big[\Phi_\mathcal{S}\big(\mathcal{D}^i_j\big) + t_{\mathrm{emb}}\Big] \ \in\ \mathbb{R}^d, \tag{26}$$

where $\mathtt{LN}(\cdot)$ stands for layer norm.

Encoder The study encoder applies L_E stacked operator self-attention blocks $\Psi^\ell_\mathcal{S} : \boldsymbol{h}^{(\ell-1)} \to \boldsymbol{h}^{(\ell)}$, where we denote by $\boldsymbol{h}^{(\ell)} \in \mathbb{R}^{N \times d}$ the hidden state after block $\ell \in L$:

$$\boldsymbol{h} \ =\ \boldsymbol{h}^{(\ell-1)} + \mathtt{SelfOpAttn}\big(\mathtt{LN}(\boldsymbol{h}^{(\ell-1)}),\ \boldsymbol{\Delta\tau},\ \boldsymbol{M}\big), \tag{27}$$

$$\boldsymbol{h}^{(\ell)} \ =\ \boldsymbol{h} + \Phi^\ell\big(\mathtt{LN}(\boldsymbol{h})\big) + t_{\mathrm{emb}}\,\mathbf{1}_N^\top\,. \tag{28}$$

Here, Φ^ℓ is a two-layer GELU MLP with expansion ratio $4\times$ and $\boldsymbol{\Delta}\tau$ denotes the subject-specific time increments used to form the trapezoidal quadrature weights (Eq. (17)). To make the study-encoding depend explicitly on the functional flow-matching time t, we add after each block the time embedding t_{emb} broadcast to each observation in the study. Since subjects in the study are exchangeable but observations between subjects should not mix in the encoder, we use an attention mask $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ that enforces *subject-wise* attention only: we apply a block-diagonal pattern so that tokens from subject j may attend only to tokens from the same subject j, preventing information flow across subjects within the encoder self-attention. Finally, the output of the encoder yields the hidden study representation:

$$\boldsymbol{h}^{\mathcal{S}} = \texttt{LN}\big[\Psi_{\mathcal{S}}^{L_E-1}(\boldsymbol{h}^{(L_E-1)}) \circ \cdots \circ \Psi_{\mathcal{S}}^{0}(\boldsymbol{h}^{(0)})\big]\,. \tag{29}$$

B.4 TARGET INDIVIDUAL DECODER

For the observations of a new individual $\mathcal{D}_j = (\tau_j, y_j, a, r) \in \mathbb{R}^4$, the decoder operates over an irregular grid of T time points formed by concatenating the observed prefix times $\boldsymbol{\tau}_{\text{P}}$ and the future query times $\boldsymbol{\tau}_{\text{F}}$, sorted chronologically. At flow time t, the flow interpolated concentration can be written as

$$\mathbf{z}_t \;=\; (\mathbf{y}_{\text{P}}, \mathbf{y}_{t,\text{F}}) \;=\; \big[t\,(\mathbf{y}_{\text{P}}, \mathbf{y}_{\text{F}}) \;+\; (1-t)\,(\mathbf{y}_{\text{P}}, \mathbf{x}_{\text{F}}) \;+\; \sigma_{\min}\,\boldsymbol{\varepsilon}\big] \odot \mathbf{M}_{\mathbf{p}} \tag{30}$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{1})$ and $\sigma_{\min}$ is a small hyperparameter that regularizes the conditional Dirac paths. The irregularity of the observation time grid for target individuals is kept track by a binary mask $\boldsymbol{m} \in \{0,1\}^T$ distinguishing true observations from zero-padded entries.

Input embedding. Each target individual observation $\mathcal{D}_j \in \mathbb{R}^4$ is embedded by a separate three-layer GELU MLP interleaved with GELU activations and added to the flow-time embedding:

$$\boldsymbol{g}_j^{(0)} = \texttt{LN}\Big[\Phi_{\mathcal{T}}(\mathcal{D}_j) \;+\; t_{\text{emb}}\Big] \;\in\; \mathbb{R}^d, \tag{31}$$

where $\texttt{LN}(\cdot)$ stands for layer norm.

Decoder The decoder applies L_D blocks, each containing two attention components with non-causal structure: an intra-individual operator self-attention layer followed by a operator cross-attention layer over the context study $\mathcal{S}$. Writing $\boldsymbol{g}^{(\ell)} \in \mathbb{R}^{T\times d}$ for the hidden state after block $\ell \in L_D$, the decoder block $\Psi_{\mathcal{T}}^{\ell} : \boldsymbol{g}^{(\ell-1)} \to \boldsymbol{g}^{(\ell)}$ is defined by

$$\boldsymbol{g} \;=\; \boldsymbol{g}^{(\ell-1)} \;+\; \texttt{SelfOpAttn}\big(\texttt{LN}(\boldsymbol{g}^{(\ell-1)}),\; (\boldsymbol{\Delta\tau}_{\text{P}}, \boldsymbol{\Delta\tau}_{\text{F}}),\; \boldsymbol{M}_1\big), \tag{32}$$

$$\boldsymbol{g}' \;=\; \boldsymbol{g} \;+\; \Phi_1^{\ell}\big(\texttt{LN}(\boldsymbol{g})\big)\,, \tag{33}$$

$$\boldsymbol{g}'' \;=\; \boldsymbol{g}' \;+\; \texttt{CrossOpAttn}\big(\texttt{LN}(\boldsymbol{g}')\,,\; \boldsymbol{h}^{\mathcal{S}},\; \boldsymbol{\Delta\tau},\; \boldsymbol{M}_2\big) \tag{34}$$

$$\boldsymbol{g}^{(\ell)} \;=\; \boldsymbol{g}'' \;+\; \Phi_2^{\ell}\big(\texttt{LN}(\boldsymbol{g}'')\big) \;+\; t_{\text{emb}}\,\mathbf{1}_T^{\top}, \tag{35}$$

where $(\boldsymbol{\Delta\tau}_{\text{P}}, \boldsymbol{\Delta\tau}_{\text{F}})$ and $\boldsymbol{\Delta\tau}$ are the irregular times increments of the (past, future) target individual and study context, respectively, $\Phi_{1,2}$ are 2-layered MLPs and $\boldsymbol{h}^{\mathcal{S}}$ is the study encoder output used for the cross-attention key and values. The attention mask matrices, $\boldsymbol{M}_1 \in \mathbb{R}^{T\times T}$ and $\boldsymbol{M}_2 \in \mathbb{R}^{T\times N}$, are defined through their respective binary observation masks, preventing queries to attend to invalid zero-padded entries. This setup allows for information to flow freely between the target's future observations $\mathbf{y}_{\text{F}}$ and the past contextual observations $\mathbf{y}_{\text{P}}$ during self-attention, while subsequently cross-attending to all study observations at all possible times. The output of the target individual decoder reads:

$$\boldsymbol{g}^{\mathcal{T}} = \texttt{LN}\big[\Psi_{\mathcal{T}}^{L_D-1}(\boldsymbol{g}^{(L_D-1)}) \circ \cdots \circ \Psi_{\mathcal{T}}^{0}(\boldsymbol{g}^{(0)})\big]\,. \tag{36}$$

Output head. Finally, the decoder's output feeds into the head of the architecture parametrized by a 3-layered MLP Φ_{head}:

$$\big[v_t^{\theta}(\mathbf{z}_t, \mathcal{S})\big]_j = \Phi_{\text{head}}\big(\boldsymbol{g}_j^{\mathcal{T}}\big)\,. \tag{37}$$

This yields the functional velocity field evaluated at all query times.

B.5 B.7 HYPERPARAMETERS AND TRAINING DETAILS

Table 4 summarizes the architectural and training hyperparameters used for the PFF results reported in Section 5.

Table 3: Meta-study configuration parameters used for synthetic PK meta-study generation.

Parameter	**Value / Range**
Study setup	
`drug_id_options`	Drug_A, Drug_B, Drug_C
`num_individuals_range`	[5, 10]
`num_peripherals_range`	[1, 3]
`solver_method`	rk4
Time grid	
`time_start`	0.0
`time_stop`	24.0
`time_num_steps`	100
Absorption parameters (k_a)	
`log_k_a_mean_range`	[-1, 2]
`log_k_a_std_range`	[0.15, 0.45]
`k_a_tmag_range`	[0.01, 0.1]
`k_a_tscl_range`	[1, 5]
Elimination parameters (k_e)	
`log_k_e_mean_range`	[-5, 0]
`log_k_e_std_range`	[0.15, 0.45]
`k_e_tmag_range`	[0.01, 0.1]
`k_e_tscl_range`	[1, 5]
Volume parameters (V)	
`log_V_mean_range`	[1, 7]
`log_V_std_range`	[0.15, 0.45]
`V_tmag_range`	[0.001, 0.01]
`V_tscl_range`	[1, 5]
Peripheral exchange (k_{1p})	
`log_k_1p_mean_range`	[-4, 0]
`log_k_1p_std_range`	[0.15, 0.45]
`k_1p_tmag_range`	[0.01, 0.1]
`k_1p_tscl_range`	[1, 5]
Peripheral return (k_{p1})	
`log_k_p1_mean_range`	[-4, -1]
`log_k_p1_std_range`	[0.15, 0.45]
`k_p1_tmag_range`	[0.01, 0.1]
`k_p1_tscl_range`	[1, 5]
Observation noise	
`rel_ruv_range`	[0.01, 0.1]

Table 4: Hyperparameters of PFF.

Parameter	**Value**
Architecture	
Hidden dimension d	256
Encoder depth L_E	4
Decoder depth L_D	4
Attention heads (encoder / decoder) H	4 / 4
FFN expansion ratio	$4\times$
Dropout	0.1
Fourier max. frequency $f_{\max}$	256
Reference measure (GP regression)	
Kernel variance ν^2	1×10^{-4}
Length scale ℓ	1.7×10^{-3}
Cholesky Jitter ϵ	1×10^{-7}
Output transform	Softplus
Flow matching	
Path noise $\sigma_{\min}$	10^{-4}
ODE integration steps (inference)	100
Optimisation	
Optimizer	AdamW
$(\beta_1,\ \beta_2)$	$(0.9,\ 0.999)$
Learning rate	1×10^{-5}
Gradient clip (global ℓ_2 norm)	0.5
Batch size	64
Epochs	300
Warm-up epochs	5