An amortized approach to non-linear mixed-effects modeling based on neural posterior estimation

Jonas Arruda [1] Yannik Schälte [1 2] Clemens Peiter [1] Olga Teplytska [3] Ulrich Jaehde [3] Jan Hasenauer [1 2]

Abstract

Non-linear mixed-effects models are a powerful tool for studying heterogeneous populations in various fields, including biology, medicine, economics, and engineering. Here, the aim is to find a distribution over the parameters that describe the whole population using a model that can generate simulations for an individual of that population. However, fitting these distributions to data is computationally challenging if the description of individuals is complex and the population is large. To address this issue, we propose a novel machine learning-based approach: We exploit neural density estimation based on conditional normalizing flows to approximate individual-specific posterior distributions in an amortized fashion, thereby allowing for efficient inference of population parameters. Applying this approach to problems from cell biology and pharmacology, we demonstrate its unseen flexibility and scalability to large data sets compared to established methods.

1. Introduction

Heterogeneity within populations is a common phenomenon in various fields, including epidemiology, pharmacology, ecology, and economics. It is, for instance, well-established that the human immune system exhibits substantial variability among individuals (Liston et al., 2021; Brodin & Davis, 2017), that individual patients respond differently to treatments (Claret et al., 2009; Ribba et al., 2014; Groenland et al., 2019), that genetically identical cells develop pronounced cell-to-cell variability (Spencer et al., 2009; Swain et al., 2002), but also that individual students show a broad spectrum of abilities (Goldstein, 1987). This heterogeneity can be described and analyzed using *non-linear mixed-effects (NLME)* models, a powerful class of statistical tools. NLME models can account for similarities and differences between individuals using fixed effects, random effects, and covariates. This allows for a high degree of flexibility and interpretability. These models are widely used for statistical analysis (Yu et al., 2022; Llamosi et al., 2016), hypothesis testing (Bortz & Nelson, 2006), and predictions (Claret et al., 2009; Ribba et al., 2014).

NLME models depend on unknown parameters, such as reaction rates and initial concentrations, which often need to be estimated from data. Estimating these parameters – often also called *parameter inference* – provides key insights about the data and the underlying processes. The main challenge in inferring these parameters lies in the likelihood formulation at the individual level. For this, there is generally no closed-form solution (Pinheiro, 1994). Particularly for large populations, this becomes a problem, as the required marginalization must be performed for all individuals.

Here, we present a new approach based on invertible neural networks to estimate the parameters of NLME models. We use simulation-based neural posterior estimation, which has been developed to address general parameter estimation problems (Cranmer et al., 2020). We train a mapping – a conditional normalizing flow – from a latent distribution to individual-specific posteriors conditioned on observed individual-level data. During the training of this neural posterior estimator, only simulations of a generative model and no real data are used. In the following inference phase, the trained estimator can be applied highly efficiently to any similar data set with different distributions of individuals in the population without any further simulations, facilitating the estimation of NLME model parameters in an amortized fashion. On problems from cell biology and pharmacology, we compare our method with state-of-the-art and widely used techniques in the field of NLME models: the stochastic approximation expectation maximization algorithm (SAEM) (Kuhn & Lavielle, 2005) implemented in `Monolix` (Lixoft SAS, 2023) and the first-order conditional estimation with interaction (FOCEI) (Wang, 2008) implemented in `NONMEM` (Beal & Sheiner, 1980).

[1] Life and Medical Sciences Institute, University of Bonn, 53115 Bonn, Germany [2] Computational Health Center, Helmholtz Zentrum München, 85764 Neuherberg, Germany [3] Pharmaceutical Institute, University of Bonn, 53121 Bonn, Germany. Correspondence to: Jan Hasenauer <jan.hasenauer@uni-bonn.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

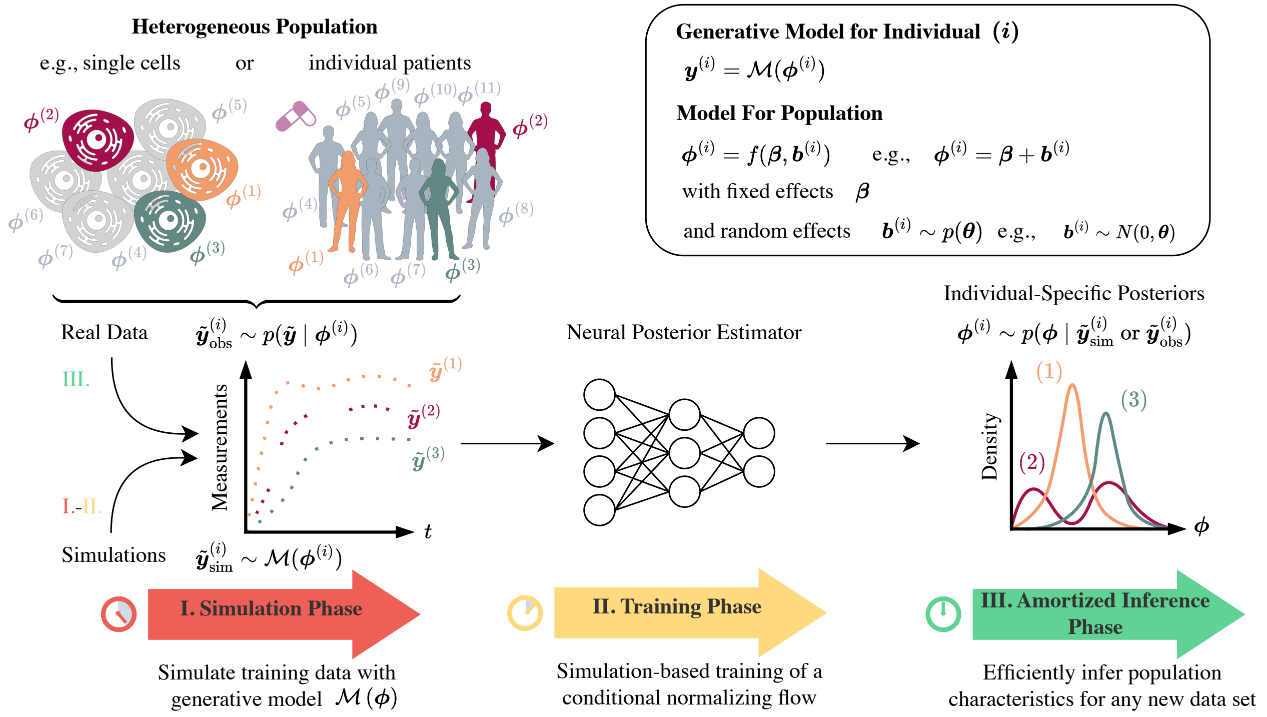

Heterogeneous Population
e.g., single cells
or
individual patients
$\phi^{(1)}$ $\phi^{(2)}$ $\phi^{(3)}$ $\phi^{(4)}$ $\phi^{(5)}$ $\phi^{(6)}$ $\phi^{(7)}$ $\phi^{(8)}$ $\phi^{(9)}$ $\phi^{(10)}$ $\phi^{(11)}$
Generative Model for Individual (i)
$\boldsymbol{y}^{(i)} = \mathcal{M}(\boldsymbol{\phi}^{(i)})$
Model For Population
$\boldsymbol{\phi}^{(i)} = f(\boldsymbol{\beta}, \boldsymbol{b}^{(i)})$ e.g., $\boldsymbol{\phi}^{(i)} = \boldsymbol{\beta} + \boldsymbol{b}^{(i)}$
with fixed effects $\boldsymbol{\beta}$
and random effects $\boldsymbol{b}^{(i)} \sim p(\boldsymbol{\theta})$ e.g., $\boldsymbol{b}^{(i)} \sim N(0, \boldsymbol{\theta})$
Real Data $\tilde{\boldsymbol{y}}^{(i)}_{\text{obs}} \sim p(\tilde{\boldsymbol{y}} \mid \boldsymbol{\phi}^{(i)})$
III.
I.-II.
Simulations $\tilde{\boldsymbol{y}}^{(i)}_{\text{sim}} \sim \mathcal{M}(\boldsymbol{\phi}^{(i)})$
Measurements
t
$\tilde{\boldsymbol{y}}^{(1)}$ $\tilde{\boldsymbol{y}}^{(2)}$ $\tilde{\boldsymbol{y}}^{(3)}$
Neural Posterior Estimator
Individual-Specific Posteriors
$\boldsymbol{\phi}^{(i)} \sim p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}^{(i)}_{\text{sim}} \text{ or } \tilde{\boldsymbol{y}}^{(i)}_{\text{obs}})$
Density
(1) (2) (3)
ϕ
I. Simulation Phase
Simulate training data with generative model $\mathcal{M}(\boldsymbol{\phi})$
II. Training Phase
Simulation-based training of a conditional normalizing flow
III. Amortized Inference Phase
Efficiently infer population characteristics for any new data set

Figure 1. Three phases of the amortized approach. (**I.**) The simulation phase, where we generate data from the model $\mathcal{M}(\phi)$, (**II.**) the training phase, where we train the neural posterior estimator to predict individual-specific posteriors based on the simulations, and (**III.**) the amortized inference phase, where we infer the population parameters of the non-linear mixed-effects model given observed data.

2. Method

Our novel approach to parameter inference for non-linear mixed-effects (NLME) models consists of three phases (summarized in Figure 1):

(I) In the simulation phase, samples from a prior are generated to produce a set of simulations and individual-specific parameters using a generative model. Each simulation belongs to a synthetic individual in a population. Simulations can also be performed online during the next phase.

(II) In the training phase, a global approximation of the posterior distribution is learned for any pair of simulations and parameters. This neural posterior estimator can then predict individual-specific posteriors.

(III) In the amortized inference phase, a population model is assumed and population-level characteristics are inferred using an efficient approximation to the population likelihood based on samples of every individual in the population from the neural posterior estimator.

2.1. Basic Definitions

We consider a population of individuals $i \in \{1, \dots, N\}$, $N \in \mathbb{N}$, for example, a group of people or an ensemble of single cells. For each of them, we have $n \in \mathbb{N}$ noisy measurements $\tilde{y}_j^{(i)} \in \mathbb{R}$ at time points $t_j \in \mathbb{R}_{\geq 0}$ for $j \in \{1, \dots, n\}$. To account for errors introduced during the measurement process, a noise model with i.i.d. measurement noise is assumed. In the simplest case this is $\tilde{y} = y + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is the measurement error with variance σ^2. Then, our observed data is the set $\mathcal{D} = \{\tilde{\boldsymbol{y}}^{(i)}\}_{i=1}^{N}$. We can extend this framework to account for multiple or censored measurements at each time point and different time points for each individual.

The generative model $\mathcal{M}(\boldsymbol{\phi})$ can generate simulations $\tilde{\boldsymbol{y}}^{(i)}_{\text{sim}} \in \mathbb{R}^n$ for a given set of parameters $\boldsymbol{\phi} \in \Omega \subseteq \mathbb{R}^k$ and time points $\boldsymbol{t}$. As a generative model, we understand any parametric model, such as linear models, the solution of (stochastic) differential equations, or Markov jump processes, which can produce simulations for an individual i given some parameters $\boldsymbol{\phi}$ and time points $\boldsymbol{t}$. In our case, this will be the (numerical) solution of a differential equation, and we assume that the noise model is part of $\mathcal{M}$.

The non-linear mixed-effects (NLME) model is a popular way to describe observations of the entire population using the generative model $\mathcal{M}$ and individual-specific parameters $\boldsymbol{\phi}^{(i)} \in \Omega \subseteq \mathbb{R}^k$. Therefore, in NLME models, it is assumed that the population can be described by unknown fixed effects $\boldsymbol{\beta}$, and the distribution of unknown random effects $\boldsymbol{b}^{(i)}$ specific to each individual i (Pinheiro, 1994).

Commonly, fixed and random effects are linked as a linear combination to the individual-specific parameters $\boldsymbol{\phi}^{(i)}$. Here, we link these effects to individual-specific parameters using a *population model* f, such that $\boldsymbol{\phi}^{(i)} = f(\boldsymbol{\beta}, \boldsymbol{b}^{(i)})$, where the function f can be any function injective in each variable. It is possible to extend the population model to include covariates, that is, additional observed information for each individual i.

For ease of notation, we consider a single vector of population parameters $\boldsymbol{\theta}$ that fully characterize the distribution of individual-specific parameters $\boldsymbol{\phi}^{(i)}$. For example, we can represent $\boldsymbol{\theta}$ as a pair of fixed effects $\boldsymbol{\beta}$ and the covariance matrix $\boldsymbol{\Psi}$ of random effects if $\boldsymbol{b}^{(i)} \sim \mathcal{N}(0, \boldsymbol{\Psi})$ and $\boldsymbol{\phi}^{(i)} = f(\boldsymbol{\beta}, \boldsymbol{b}^{(i)})$. In general, we allow the random effects to follow any probability distribution that admits a density, such as (log)-normal distributions, Cauchy distributions, or mixture distributions.

2.2. Joint Likelihood

Our objective is to find the optimal population parameters $\boldsymbol{\theta}^*$ that maximize the joint likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ of all individuals in the population, which is

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(\tilde{\boldsymbol{y}}^{(i)} \mid \boldsymbol{\theta}) \tag{1}$$

since we assumed that measurement noise is i.i.d.

Based on the population model and the distribution of random effects, we induce a population distribution $p_{\text{pop}}(\boldsymbol{\phi} \mid \boldsymbol{\theta})$. For a linear population model $\boldsymbol{\phi}^{(i)} = \boldsymbol{\beta} + \boldsymbol{b}^{(i)}$ and $\boldsymbol{b}^{(i)} \sim \mathcal{N}(0, \boldsymbol{\Psi})$, p_{pop} is the density of a normal random variable with mean $\boldsymbol{\beta}$ and covariance matrix $\boldsymbol{\Psi}$. Using the population distribution, we can thus express the likelihood for every individual i by the law of total probability as

$$p(\tilde{\boldsymbol{y}}^{(i)} \mid \boldsymbol{\theta}) = \int p(\tilde{\boldsymbol{y}}^{(i)} \mid \boldsymbol{\phi}) p_{\text{pop}}(\boldsymbol{\phi} \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\phi}. \tag{2}$$

The marginal likelihood $p(\tilde{\boldsymbol{y}}^{(i)} \mid \boldsymbol{\phi})$ is implicitly induced by $\mathcal{M}$ including the respective noise model. Usually, the likelihood is not fully tractable if the generative model is stochastic.

Since $\boldsymbol{\phi}$ is unknown, we need to marginalize the individual-specific parameters out and solve the integral in (2) for every individual separately. In general, this integral has no closed-form solution, hence solving this marginalization efficiently is the main challenge in parameter inference in non-linear mixed-effects models (Pinheiro, 1994).

2.3. Individual-specific neural posterior estimator

In the following, we develop an approach to efficiently maximize $p(\mathcal{D} \mid \boldsymbol{\theta})$ if we can sample from the individual specific posterior distributions $p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}^{(i)})$. In general, individual measurements are not sufficiently informative to obtain reliable population estimates, and only joint information is reliable (Pinheiro, 1994). However, using a Bayesian approach to describe individuals, we encode all available information on a specific individual i in the posterior of the parameters $\boldsymbol{\phi}^{(i)}$ and then combine samples from this posterior to infer the characteristics of the population. Following this dogma of Bayesian statistics, all parameters – also those which are considered fixed across the population – will first be treated as random variables.

For that, let $p(\boldsymbol{\phi})$ be a prior density that encodes prior knowledge for individual specific parameters. Then, using the product rule, we get the relation of likelihood and posterior density $p(\boldsymbol{\phi})p(\tilde{\boldsymbol{y}} \mid \boldsymbol{\phi}) = p(\tilde{\boldsymbol{y}}, \boldsymbol{\phi}) = p(\tilde{\boldsymbol{y}})p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}})$.

Conditional normalizing flows can transform a complicated conditional density, such as a posterior probability, into a simpler density from which we know how to sample. This method allows for efficient and accurate sampling and density evaluation (Rezende & Mohamed, 2015; Papamakarios et al., 2021).

We introduce latent variables $\boldsymbol{z}$ described by a multivariate normal distribution $p(\boldsymbol{z})$. The parameters $\boldsymbol{\phi}$ are mapped to these latent variables conditional on measurements $\tilde{\boldsymbol{y}}$ by a conditional normalizing flow $h_{\boldsymbol{\psi}}(\boldsymbol{\phi}, \tilde{\boldsymbol{y}}) = \boldsymbol{z}$. This invertible transformation is parameterized by $\boldsymbol{\psi}$ and has a tractable Jacobian by construction. The approximation $q_{\boldsymbol{\psi}}$ to the target density $p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}})$ is given by the change-of-variables formula

$$q_{\boldsymbol{\psi}}(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}) = p(\boldsymbol{z} = h_{\boldsymbol{\psi}}(\boldsymbol{\phi}, \tilde{\boldsymbol{y}})) \left| \det J_{h_{\boldsymbol{\psi}}}(\boldsymbol{\phi}, \tilde{\boldsymbol{y}}) \right|. \tag{3}$$

If we know $h_{\boldsymbol{\psi}}$, we can sample from the posterior by sampling $\boldsymbol{z} \sim p(\boldsymbol{z})$ and applying $h_{\boldsymbol{\psi}}^{-1}(\boldsymbol{z}, \tilde{\boldsymbol{y}}) = \boldsymbol{\phi}$. We call q_{ψ} a *neural posterior estimator*.

To train the conditional normalizing flow $h_{\boldsymbol{\psi}}$, we minimize the Kullback-Leibler divergence between the true and approximate posterior distributions as in (Papamakarios et al., 2017; Le et al., 2017):

$$\begin{aligned}
\boldsymbol{\psi}^* &= \arg\min_{\boldsymbol{\psi}} \mathbb{E}_{p(\tilde{\boldsymbol{y}})} \left[\mathrm{KL}(p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}) \,||\, q_{\boldsymbol{\psi}}(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}})) \right] \\
&= \arg\min_{\boldsymbol{\psi}} \iint -p(\tilde{\boldsymbol{y}}, \boldsymbol{\phi}) \log q_{\boldsymbol{\psi}}(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}) \, \mathrm{d}\tilde{\boldsymbol{y}} \, \mathrm{d}\boldsymbol{\phi} \\
&\approx \arg\min_{\boldsymbol{\psi}} \frac{1}{S} \sum_{s=1}^{S} - \log q_{\boldsymbol{\psi}}(\boldsymbol{\phi}^{(s)} \mid \tilde{\boldsymbol{y}}).
\end{aligned} \tag{4}$$

To estimate the integral, we need samples from the joint distribution $p(\tilde{\boldsymbol{y}}, \boldsymbol{\phi})$. Sampling from the prior distribution $\boldsymbol{\phi} \sim p(\boldsymbol{\phi})$ and simulating using $\mathcal{M}(\boldsymbol{\phi})$ corresponds to sampling from this joint density. Using the transformation in (3), the approximation in (4) can be efficiently evaluated.

By minimizing (4), we train a global approximation of the posterior distribution $p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}})$ for any parameters and data

Algorithm 1 Amortized Inference for NLME Models

```
Phase (I)
  Input: generative model $\mathcal{M}$, individual prior $p(\boldsymbol{\phi})$
  for $i = 1$ to $n_{\text{sim}}$ do
    sample from prior $\boldsymbol{\phi}^{(i)} \sim p(\boldsymbol{\phi})$
    generate simulations $\tilde{\boldsymbol{y}}^{(i)}_{\text{sim}} \sim \mathcal{M}(\boldsymbol{\phi}^{(i)})$
  end for
Phase (II)
  Input: pairs $\{(\boldsymbol{\phi}^{(i)}, \tilde{\boldsymbol{y}}^{(i)}_{\text{sim}})\}_{i=1}^{n_{\text{sim}}}$, normalizing flow $h_{\boldsymbol{\psi}}$
  pass simulations and parameters through $h_{\boldsymbol{\psi}}$
  compute loss according to (4) and find optimal $\boldsymbol{\psi}^*$
Phase (III)
  Input: data $\mathcal{D} = \{\tilde{\boldsymbol{y}}^{(i)}\}_{i=1}^{n}$, population model $p_{\text{pop}}$, trained normalizing flow $h_{\boldsymbol{\psi}^*}$
  for $i = 1$ to $N$ do
    sample $M$ times from posterior $\boldsymbol{\phi}^{(i)} \sim q_{\boldsymbol{\psi}^*}(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}^{(i)})$
  end for
  compute loss according to (5) to find $\boldsymbol{\theta}^*$
```

$(\boldsymbol{\phi}, \tilde{\boldsymbol{y}})$. In particular, we parameterize the conditional normalizing flow by an invertible neural network using neural spline flows (Durkan et al., 2019) and learned summary statistics of $\tilde{\boldsymbol{y}}$.

Summary statistics are a low-dimensional representation of $\tilde{\boldsymbol{y}}$ and learned by a flexible summary network that is trained together with the conditional normalizing flow. In (Papamakarios et al., 2017; Le et al., 2017), it was shown that the samples transformed backward from $p(\boldsymbol{z})$ will follow the true posterior if the conditional normalizing flow and the summary network are expressive enough. In particular, we use long short-term memory neural networks (LSTMs) for time trajectories (Hochreiter & Schmidhuber, 1997). This ensures that regardless of the number of observations, we get a fixed length vector of summary statistics.

2.4. Problem reformulation allows use of posterior density

We note that the likelihood in (2) can be written as a conditional expectation over individual-specific posteriors

$$p(\tilde{\boldsymbol{y}}^{(i)} \mid \boldsymbol{\theta}) = \int \frac{p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}^{(i)}) p(\tilde{\boldsymbol{y}}^{(i)})}{p(\boldsymbol{\phi})} p_{\text{pop}}(\boldsymbol{\phi} \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\phi} = p(\tilde{\boldsymbol{y}}^{(i)}) \mathbb{E}_{\boldsymbol{\phi} \sim p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}^{(i)})} \left[\frac{p_{\text{pop}}(\boldsymbol{\phi} \mid \boldsymbol{\theta})}{p(\boldsymbol{\phi})} \right],$$

given that the prior $p(\boldsymbol{\phi})$ is non-zero in the support of $\boldsymbol{\phi}$. If we can sample from the posterior distribution $p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}^{(i)})$, we can construct a Monte Carlo estimator for the likelihood

$$p(\tilde{\boldsymbol{y}}^{(i)} \mid \boldsymbol{\theta}) \approx p(\boldsymbol{y}^{(i)}) \left(\frac{1}{M} \sum_{j=1}^{M} \left[\frac{p_{\text{pop}}(\boldsymbol{\phi}_j^{(i)} \mid \boldsymbol{\theta})}{p(\boldsymbol{\phi}_j^{(i)})} \right] \right),$$

with $\boldsymbol{\phi}_j^{(i)} \sim p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}^{(i)})$ i.i.d. for $j = 1, \dots, M$ for each individual i.

Taking the logarithm of (1), we can drop the additive term $p(\boldsymbol{y})$ which is independent of $\boldsymbol{\theta}$ and find the optimal population parameters $\boldsymbol{\theta}^*$ by solving the maximization problem

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta}) \approx \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \log \left(\frac{1}{M} \sum_{j=1}^{M} \frac{p_{\text{pop}}(\boldsymbol{\phi}_j^{(i)} \mid \boldsymbol{\theta})}{p(\boldsymbol{\phi}_j^{(i)})} \right). \quad (5)$$

In general, the Monte Carlo approximation to an integral is unbiased, and the error rate of the approximation $\sigma_{\text{MC}}/\sqrt{M}$ depends on the sample size M and the variance σ^2_{MC} of the ratio $p_{\text{pop}}(\boldsymbol{\phi} \mid \boldsymbol{\theta})/p(\boldsymbol{\phi})$ (Robert & Casella, 2004). From this, we directly get the following proposition.

Proposition 2.1. *Assume that $\boldsymbol{\phi}^{(i)} \sim p_{pop}(\boldsymbol{\phi} \mid \boldsymbol{\theta})$, that the prior $p(\boldsymbol{\phi})$ is non-zero in the support of $\boldsymbol{\phi}$ and that we can sample from the true posteriors $p(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}^{(i)})$ for every individual i. Then, $\boldsymbol{\theta}$ converges to the true maximum likelihood estimate $\boldsymbol{\theta}^*$ as the sample size $M \to \infty$.*

The variance σ^2_{MC} depends only on the ratio of $p_{\text{pop}}(\boldsymbol{\phi} \mid \boldsymbol{\theta})$ and $p(\boldsymbol{\phi})$. Therefore, the prior has the role of an importance weight and should be selected to have a shape similar to the population distribution. This decreases the number of samples M we need to get a sufficiently good approximation of the likelihood (see Supplement A.4.1 for further discussion).

The maximization problem (5) can be solved by numerical optimization using samples from the neural posterior estimator $q_{\boldsymbol{\psi}^*}(\boldsymbol{\phi} \mid \tilde{\boldsymbol{y}}^{(i)})$. The optimization is computationally efficient and simple, as no numerical simulations of the underlying model are required. Hence, the computational costs of inferring population parameters are negligible.

We show our novel three-phase procedure for the inference of NLME models in Algorithm 1. By repeating phase (III) using different population models or data sets, we amortize the computational cost of phases (I) and (II). Repeated inference can be desirable due to hypothesis testing of various population models or repeated experiments.

3. Results

The proposed approach to fitting the NLME models is based on the approximation of the individual-specific posterior distributions with conditional normalizing flows. As the accuracy of these approximations is critical, we assessed in a first step the approximation quality (see Supplement A.3). We considered two published ODE-based NLME models of mRNA transfection with measured single-cell data (Fröhlich et al., 2018). These ODE models describe the transfection process (Figure 2A) – which is at the core of modern

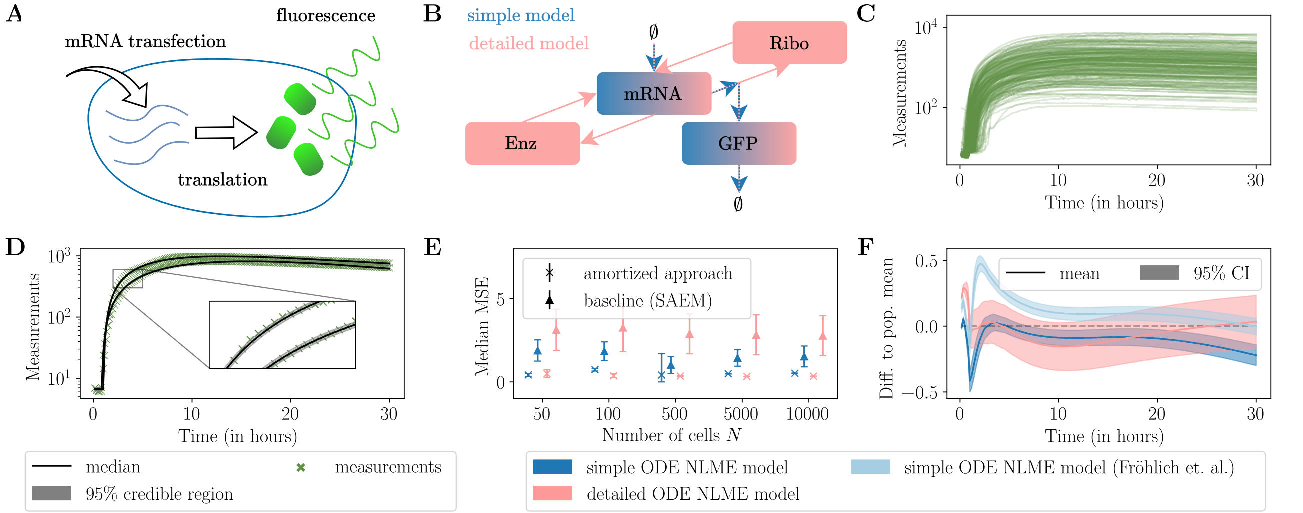


Figure 2. Validation of the amortized approach on single-cell NLME models. (**A**) Schematic representation of GFP translation after mRNA transfection in a single cell. (**B**) Visualization of the simple and detailed single-cell ODE models, where the color indicates the states included in the respective model (see Supplement A.1 for details on the models). (**C**) Fluorescence intensity time courses of 200 single cells out of 5488. (**D**) Credible regions of trajectories (simple single-cell ODE model) estimated by the neural posterior estimator for two real cells. (**E**) Median of the mean squared error (MSE) of the estimated compared to the true parameters of the synthetic data for both single-cell NLME models is shown for different numbers of cells ($M = 100$ posterior samples used per cell, median of 100 runs $\pm$ one standard deviation). (**F**) The difference in the population mean estimated from real trajectories and simulations generated with the estimated population parameters is shown with a 95% confidence interval (CI). In addition to the models fitted with the amortized approach, the best fit of Fröhlich et al. (2018) for the simple ODE model is shown (Fröhlich et al., 2018).

mRNA vaccines (Pardi et al., 2018) – at the single-cell level. The solution to these ODEs corresponds to our generative model $\mathcal{M}$. The models possess, respectively, six and eleven parameters $\boldsymbol{\phi}$ that describe two and four hidden state variables (Figure 2B, see Supplement A.1 for details on the models). The measurements consist of dense temporally resolved fluorescence intensities of different single-cells, which were transfected with mRNA coding for a green fluorescent protein (GFP), and measured for 30 hours using micropatterned protein arrays and time-lapse microscopy (Figure 2C). For each ODE model, we simulate data and train a neural posterior estimator accordingly with phase (I) and (II) from our proposed Algorithm 1. Subsequently, we examine a second application from pharmacokinetics, which is a major application area for NLME models.

We consider two scenarios to validate our method using the same trained neural posterior estimator: a) using synthetic data where we want to recover the true sample parameters, and b) using real data $\mathcal{D} = \mathcal{D}_{\text{eGFP}} \cup \mathcal{D}_{\text{d2eGFP}}$, with two distinct variants of GFP, namely eGFP and d2eGFP, that differ in their protein lifetime (Fröhlich et al., 2018). For the synthetic data, we assume a log-normal distribution with diagonal covariance matrix $\boldsymbol{\Psi}$ for all parameters, that is, $\boldsymbol{\phi} \sim \log\mathcal{N}(\boldsymbol{\beta}, \boldsymbol{\Psi})$ and $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\Psi})$. For the real data and both variants of GFP, we can use the same generative model $\mathcal{M}$ since we assume that they only differ in the parameter that describes GFP degeneration (indexed by γ). We encode the variant as a binary covariate $c^{(i)} \in \{0, 1\}$. Hence, we have $\boldsymbol{\phi}^{(i)} \sim \log\mathcal{N}(\boldsymbol{\beta}, \boldsymbol{\Psi}) \mid c^{(i)}$, where the elements $\boldsymbol{\beta}_\gamma$ and $\boldsymbol{\Psi}_{\gamma\gamma}$ depend on the respective variant of the single-cell i. Then, for each element $\boldsymbol{\phi}_k^{(i)}$ the population model is

$$f(\boldsymbol{\beta}_k, \boldsymbol{b}^{(i)}, c^{(i)}) = \begin{cases} e^{\boldsymbol{\beta}_{\gamma,\text{eGFP}} + \boldsymbol{b}_\gamma^{(i)}} & \text{if } k = \gamma \text{ and } c^{(i)} = 0 \\ e^{\boldsymbol{\beta}_{\gamma,\text{d2eGFP}} + \boldsymbol{b}_\gamma^{(i)}} & \text{if } k = \gamma \text{ and } c^{(i)} = 1 \\ e^{\boldsymbol{\beta}_k + \boldsymbol{b}_k^{(i)}} & \text{else,} \end{cases}$$

where $\boldsymbol{b}^{(i)} \sim \mathcal{N}(0, \boldsymbol{\Psi}) \mid c^{(i)}$.

We implemented conditional normalizing flows using the `BayesFlow` tool (Radev et al., 2023). We assessed the quality of the neural posteriors by comparing them to approximations using MCMC. For synthetic data, we got nearly ideal agreement while on real data agreement up to model misspecification (see Supplement A.3). To estimate population parameters, we implemented the optimization problem (5) as an objective function in the `pyPESTO` toolbox (Schälte et al., 2023) and used the local optimization method L-BFGS (Liu & Nocedal, 1989) implemented in `SciPy` (Virtanen et al., 2020). For further details on the neural network architecture, we refer to the Supplement A.3. We compare our method to the `Monolix` (Lixoft SAS, 2023) implementation of the state-of-the-art method SAEM, which is an unbiased algorithm and converges under very general conditions (Kuhn & Lavielle, 2005).

3.1. Machine learning-based approach provides accurate estimates of population parameters

Given the accurate approximation of posteriors on an individual-specific level (Figure 2D and Supplement A.3), we can use the pre-trained densities to estimate the NLME population parameters $\boldsymbol{\theta}$. To assess the accuracy of our approach, we generated synthetic data using the two NLME models of mRNA transfection (see Supplement A.1.1), and compared the mean squared distance of the true parameters to the estimated parameters of our approach and to the estimated parameters of the state-of-the-art method SAEM (Kuhn & Lavielle, 2005). Moreover, we compared our results with those published in (Fröhlich et al., 2018), which used a Laplacian approximation to the population likelihood to reduce the computational costs compared to SAEM.

Our comparisons show that, for different data set sizes and models, our method was able to recover the true parameters with a lower recovery error than SAEM and a smaller variance in the estimates (Figure 2E). For each ODE model, we trained only one neural posterior estimator, which could be used for inference on all different single-cell data sets, while SAEM needed a full restart for each data set. In addition, the estimated population means for both models of our amortized approach show trajectories similar to those published in (Fröhlich et al., 2018) for the real data (Figure 2F), with only a minor shift observed in the case of the simple model. Furthermore, we can confirm the result of (Fröhlich et al., 2018) that the detailed model describes the initial fluorescence activity more accurately (Supplement A.4).

In summary, our approach based on amortized neural posterior estimation was able to provide accurate estimates of population parameters for synthetic and real data, while we used the same neural posterior estimator for each model.

3.2. Amortization for large populations, new data sets and changing population models achieved

Data simulation and training of the neutral posterior estimator are the most computationally demanding phases. For both phases, the detailed NLME model required twice as much computation time compared to the simple model. However, the subsequent inference phase of population parameters is highly efficient for a new data set.

Our method requires two orders of magnitude less computation time compared to SAEM and has a slower increase with respect to the number of individuals in the population (Figure 3A). In particular, if the population was large ($10,000$ cells in this case), we already amortized the training time cost compared to SAEM for a single data set. In total, for all data sets and models together, SAEM had an overall computation time 112 times longer than our amortized approach including all three phases.

As described above, the parameters in the single-cell NLME models were assumed to be independently distributed. However, cross-correlations between parameters are essential to explain population behavior (Llamosi et al., 2016) but were not captured in (Fröhlich et al., 2018) due to computational costs. Indeed, for the detailed mRNA transfection model, the individual-specific posteriors of the respective parameters show a clear correlation (Figure 3B).

Therefore, we changed the population model to allow for a full covariance matrix of the random effects within each individual and repeated the amortized inference phase without further training of the neural posterior estimator. Including these correlations substantially improved the fit of the population variance (Figure 3C), which confirms the findings on the importance of incorporating cross-correlations between parameters in (Llamosi et al., 2016).

Beyond point estimates, we explored the possibility of performing accurate uncertainty quantification using a profile likelihood analysis, given the computational efficiency of the inference phase in our approach (Figure 3D and further details in Supplement A.4.4).

In summary, our analyses showed that our approach scales to large populations and allows for the reuse of the trained neural posterior estimator on different data sets and for different population models at almost no additional computational cost, rendering it substantially more scalable than SAEM.

3.3. Flexible generative model makes stochastic mixed-effects models easily tractable

As our approach based on neural posterior estimation proved to be valuable for deterministic models, we assessed its capability to cope with stochastic models, which can provide a more adequate description of the underlying process (Wilkinson, 2009; Stumpf et al., 2017). Ignoring the inherent stochastic nature of reactions at single-cell level can bias parameter estimates (Wiqvist et al., 2021), and pooling measurements from several cells is indispensable for reliable estimates (Zechner et al., 2014). Yet, the likelihood function is often unavailable for such stochastic models, which requires computationally demanding techniques such as approximate Bayesian computation or a Metropolis-within-Gibbs algorithm, which can handle the unavailable likelihood via correlated particle filters (Wiqvist et al., 2021; Botha et al., 2021; Sisson et al., 2018). However, our purely simulation-based approach does not need the likelihood function but only a generative model for simulation.

Here, we again considered the processes of mRNA transfection, but described by a stochastic differential equation

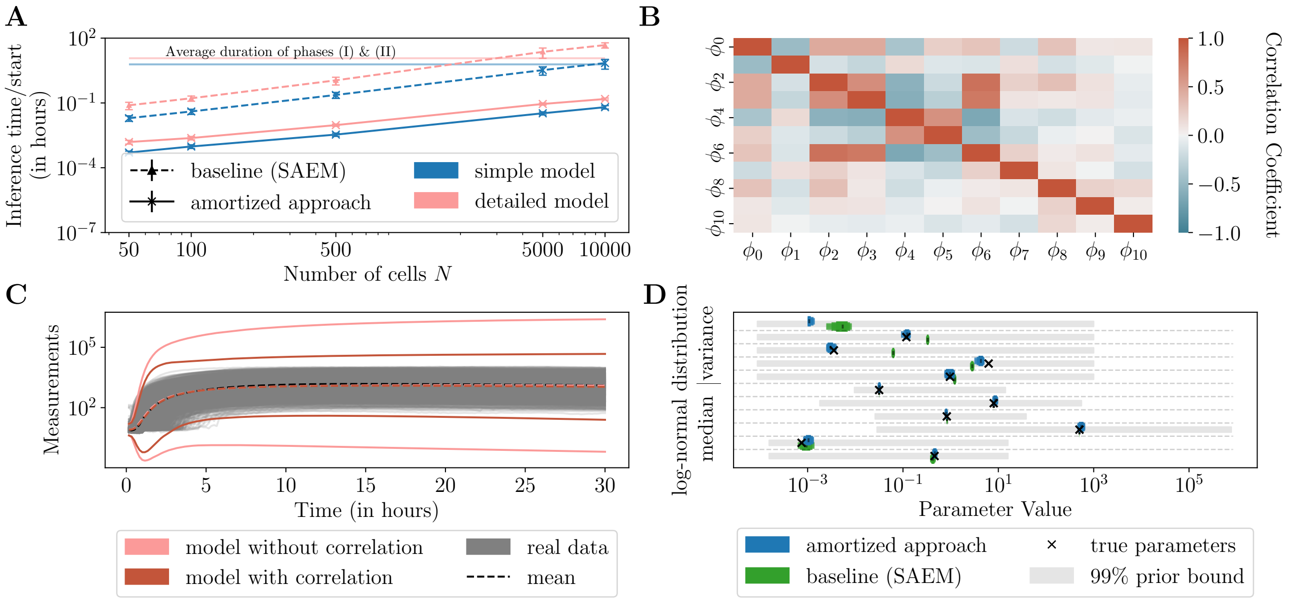


Figure 3. Flexibility and scalability of the amortized approach on the single-cell NLME models. (**A**) Computation time of the amortized inference phase (median ± one standard deviation of a single run out of 100 multi-starts) for the single-cell NLME models compared to the baseline using parallelization. (**B**) Correlation of the posterior medians of the individual-specific parameters ϕ in the detailed single-cell model on real data. (**C**) Mean and 99% confidence intervals of the simulations for the detailed NLME model, where the population parameters are assumed to be log-normally distributed with and without correlations between parameters. (**D**) 80%, 90%, and 95% confidence intervals (CIs) for the simple single-cell NLME model (see Supplement Figure S15 for the other models) using synthetic data with known true parameters.

(SDE) as proposed by (Pieschner et al., 2022) (detailed model specification in Supplement A.1). This model is superior for the description of individual cells and improves parameter identifiability (Pieschner et al., 2022), but has not been used so far in an NLME modeling framework.

The evaluation using the SDE NLME model on synthetic data revealed that our machine learning-based approach was indeed able to accurately recover the stochastic NLME model parameters (Supplement S10). Further analysis of synthetic data generated by the SDE NLME model showed that the simple ODE NLME model estimated parameters such that the variance of the population was three times greater than the true variance while for the stochastic NLME model, the variance is only 1.3 times greater (Supplement S17). Therefore, the stochastic model is capable of capturing the data more accurately. This underlines that a deterministic model can give erroneous results if it inadequately captures the underlying processes. The overall training time (7.5 hours) is comparable to the simple ODE model (6 hours), and the amortized inference phase remains highly efficient.

The simple ODE model of the mRNA transfection processes consists partly of a product of parameters $k{\cdot}m_0{\cdot}scale$, where the individual parameters were not structurally identifiable, which means that not all parameters can be determined from the data (Fröhlich et al., 2018). However, the SDE model offers a more detailed representation encompassing the individual parameters k, m_0 and $scale$. Only using our amortizing NLME framework, we were able to identify all parameters (Supplement S16).

In summary, our amortized approach enables the use of either a deterministic or a stochastic NLME model, whichever is more appropriate. This not only enables a more profound understanding of the actual mechanism, but can also improve model identifiability.

3.4. Amortization allows for full Bayesian analysis and more complex models

So far, we have considered inference problems with densely observed time points. Here, we turn to a model from pharmacokinetics introduced in (Diekstra et al., 2017; Yu et al., 2015) that describes the distribution of an angiogenesis inhibitor, a drug that inhibits the growth of new blood vessels, and its metabolite in a multi-compartmental model. Using data from 47 patients, including covariates such as weight, sex, and drug dosage regimes, we analyzed an ODE model with five states and 13 parameters. This model shows oscillatory behavior due to multiple dosing events (see Figure 4A). For comparison, we also explored FOCEI (Wang, 2008), which is arguably the most commonly used infer-

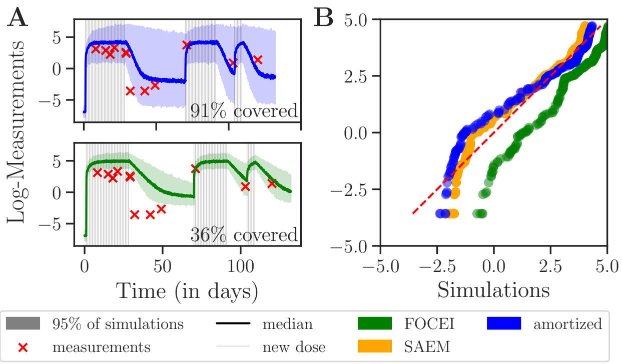


Figure 4. Baseline underestimates measurements. (**A**) Trajectory of sunitinib plasma measurements for one exemplary patient. Simulating samples from the population likelihood convoluted with the noise model using the covariates of this patient based on the estimated parameters of FOCEI and our amortized approach. (**B**) Ordered measurements (Sunitinib plasma) of all individuals against the median of 100 simulations per time point and individual.

ence method in pharmacokinetic modeling, implemented in `NONMEM` (Beal & Sheiner, 1980). Given FOCEI's known tendency to converge to local optima, we performed 300 repeated parameter inference experiments (Pinheiro, 1994).

Inferring population parameters yielded similar results for our amortized approach and SAEM. However, FOCEI underestimates the first observable consistently (see Figure 4B). Moreover, estimated variances with FOCEI are consistently smaller than those from our amortized approach or SAEM (see Supplement A.5). Simulating samples from the population likelihood convoluted with the noise model using estimated parameters from FOCEI and our approach, we found that only 72% (83%) of measurements fell within the region described by 95% (99%) of FOCEI's estimated population variance. In contrast, our amortized approach covered 97% (99%) of measurements within the estimated variance (see also Figure 4A). Consequently, FOCEI systematically underestimates population variance – a known issue (Ge et al., 2004; Jönsson et al., 2004) – which is not encountered for the proposed approach.

Moreover, efficient evaluation of the population likelihood enables a full Bayesian analysis. By defining priors on population parameters, we can easily employ a Markov chain Monte Carlo method. Here, we used a Metropolis-Hastings sampler with adaptive proposal covariance as implemented in `pyPESTO` (Schälte et al., 2023). Generating $100,000$ samples takes only a few minutes, allowing us to analyze uncertainty in variance estimates. This analysis shows that FOCEI's variance estimates do not fall within the 95% credible regions produced by the amortized approach, whereas the estimates of SAEM do. Yet, SAEM provides only a point estimate and approximate confidence intervals.

To show the applicability of our approach, we report the run times of the different methods. Running FOCEI repeatedly took 28 hours, while SAEM needed 37 hours. In contrast, our amortized approach, including repeating phase (III) 200 times and generating $100,000$ samples from the full population posterior, completed in 27 hours.

This demonstrates the feasibility of a full Bayesian analysis and highlights our method's capability to handle complex models with individual-specific dosing regimes. Notably, we observed a less biased population fit compared to FOCEI.

4. Related Work

The inference methods for NLME models most commonly used today are deterministic, starting from the first inference method based on a first-order approximation of the model function around the expected value of random effects (Beal & Sheiner, 1980) and later on conditional modes (Lindstrom & Bates, 1990). The first-order approximation was used, among others, to analyze clinical patient data (Sheiner & Beal, 1980). Pinheiro & Bates reviewed more accurate methods based on the marginal likelihood approximation using Laplace methods or quadrature rules, obtaining potentially higher accuracy at higher computational cost (Pinheiro & Bates, 1995). Today, first-order conditional estimation with interaction (FOCEI) (Wang, 2008) is arguably the most common inference method used in pharmacokinetic modeling. Yet, the aforementioned methods have two main statistical drawbacks. First, they do not necessarily converge to the maximum likelihood estimates, and second, the estimates can be substantially biased when the variability of random effects is large (Ge et al., 2004; Jönsson et al., 2004).

For unbiased results, the stochastic expectation maximization algorithm (SAEM) was introduced, which converges under very general conditions (Kuhn & Lavielle, 2005). This method was applied, for example, to model the response of yeast cells to repeated hyperosmotic shocks (Llamosi et al., 2016). Yet, the algorithm can be computationally demanding, depending on the number of random effects and the structural complexity of the model.

However, all the methods mentioned do not apply to stochastic models for individuals, such as stochastic differential equations (SDEs). So far only Bayesian methods can provide exact inference and inherently facilitate uncertainty quantification for SDEs, but are even more computationally demanding (Wiqvist et al., 2021; Botha et al., 2021).

A simulation-based approach has been proposed to accelerate Bayesian inference in NLME models by approximating the likelihood of the population with simulated measurements and hand-crafted filters (Augustin et al., 2023). However, choosing an effective filter requires experimenting with different filters and numbers of simulations, while the efficiency depends on the availability of the gradient of the underlying simulator, which ours does not. Generalizing this to stochastic systems adds further complexity due to increased noise in the filter likelihood.

Further, one could directly learn the likelihoods based on neural likelihood estimation (Durkan et al., 2018). A different approach would be to directly tackle the full population likelihood or posterior. For example, Radev et al. (2020) used permutation embedding networks that can amortize over a given number of fixed i.i.d. trials, which could potentially be used in a setting with smaller data sets.

In general, computational costs make it difficult to fit NLME models to large data sets and to obtain reliable estimates of model parameters (Fröhlich et al., 2018; Augustin et al., 2023; Persson et al., 2022).

5. Discussion

We developed a novel approach for parameter inference in non-linear mixed-effects models based on amortized neural posterior estimation. The proposed method offers several advantages, such as scalability, flexibility, and accurate uncertainty quantification over established approaches, as we demonstrated on single-cell and pharmacology models.

One of the most important benefits of the method is its scalability. The efficient amortizing inference phase allows to scale to large numbers of individuals and can be applied to previously unseen data. The main computational bottleneck, the simulation and training phases, can be tackled by more extensive parallelization on a high-performance infrastructure since all simulations are independent. Further, the method can be applied to various population models and new data sets with low computational costs using the same trained neural posterior estimator, allowing efficient model selection. In contrast, state-of-the-art methods require a full restart for each population model.

Our machine learning-based approach is purely simulation-based; that is, it does not require the evaluation of likelihoods, but only a generative model to simulate synthetic data. Therefore, it can be used even for complex stochastic models, which established approaches fall short of, as we demonstrated on an SDE-based NLME model of mRNA transfection. Our approach can be easily extended to Markov jump processes, e.g., simulated with the Gillespie algorithm (Gillespie, 1977). This generality with respect to the generative model is unique in the NLME context, as special frameworks were needed to cope with stochastic differential equations (Wiqvist et al., 2021) or Markov jump processes (Zechner et al., 2014).

Lastly, the efficient neural posterior estimator facilitates more accurate and systematic methods to assess parameter uncertainty, here demonstrated by combining our approach with profile likelihoods and Bayesian inference. Conceptually, other uncertainty analysis methods, such as bootstrapping, could also be applied efficiently.

The study raises the question of how amortization can be best used in a hierarchical setup, in which, for example, the order in which problems need to be solved can be influenced. Furthermore, we consider it interesting to develop methods that assess on-the-flight whether problem-specific approximations, which are sequentially updated or an amortized approach is more beneficial for overall efficiency. This becomes relevant for small data sets and if no population model selection or accurate uncertainty quantification is needed since the computation time of the amortized approach will be higher compared to established methods.

Additionally, the proposed method may produce erroneous parameter estimates if the prior is too narrow or if the underlying model is misspecified (Schmitt et al., 2022), or use non-conservative posterior estimates (Hermans et al., 2022). However, misspecification of the model is a general problem for parametric methods. A solution might be to extend the loss function during training to include a misspecification measure (Schmitt et al., 2022). Furthermore, the accuracy of the approximated posteriors can be checked after training, e.g., by simulation-based calibration (Talts et al., 2020), or individual posterior checks by Markov chain Monte Carlo (MCMC) or approximate Bayesian computation (ABC) (Sisson et al., 2018). However, checking individual posteriors introduces an additional computationally expensive step.

Imperfect approximations of true posteriors can occur if the conditional normalizing flow, which might be the case for multimodal distributions (Hagemann et al., 2023). Then, the approximations could be improved by relaxing the constraints of the architecture imposed by invertible neural networks using generalized normalizing flows (Hagemann et al., 2023) or flow- and score-matching methods (Sharrock et al., 2022; Geffner et al., 2023; Dax et al., 2023). Nevertheless, we did not encounter such difficulties here, as we could approximate even the multimodal distributions in the simple ODE model. Thus, we are confident that our approach based on conditional normalizing flows can provide good estimates for the parameters in an NLME model.

In conclusion, the amortized approach we presented in this study offers a powerful solution for non-linear mixed-effects modeling. The approach enables researchers to flexibly use models for individuals and the population while performing accurate parameter estimation and uncertainty analysis in a more scalable manner than state-of-the-art methods.

Impact Statement

This paper presents a more efficient approach to parameter inference in non-linear mixed-effects models, which promises especially advances in medicine, drug development, and pharmacokinetics. While ethical considerations such as data privacy and equitable access should be prioritized for responsible implementation, increased efficiency contributes to resource conservation and highlights the potential for positive societal and environmental impacts.

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF) (EMUNE/031L0293C), the German Research Foundation (DFG) under Germany's Excellence Strategy (EXC 2047-390685813 and EXC 2151-390873048) and the project ID 458597554 - SEPAN, the Schlegel Professorship for J.H., and the EU Horizon 2020 program (ORCHESTRA; 101016167). Y.S. acknowledges financial support by the Joachim Herz Stiftung.

References

Augustin, D., Lambert, B., Wang, K., Walz, A.-C., Robinson, M., and Gavaghan, D. Filter inference: A scalable nonlinear mixed effects inference approach for snapshot time series data. *PLOS Computational Biology*, 19(5), 2023. doi: 10.1371/journal.pcbi.1011135.

Beal, S. and Sheiner, L. The NONMEM system. *Am. Stat.*, 34(2):118–119, 1980. ISSN 00031305.

Blanchard, P., Higham, D. J., and Higham, N. J. Accurately computing the log-sum-exp and softmax functions. *IMA Journal of Numerical Analysis*, 41(4):2311–2330, 2021.

Bortz, D. M. and Nelson, P. W. Model selection and mixed-effects modeling of HIV infection dynamics. *Bulletin of Mathematical Biology*, 68(8):2005–2025, 2006. doi: 10.1007/s11538-006-9084-x.

Botha, I., Kohn, R., and Drovandi, C. Particle methods for stochastic differential equation mixed effects models. *Bayesian Analysis*, 16(2):575 – 609, 2021. doi: 10.1214/20-BA1216.

Boyd, S. and Vandenberghe, L. *Convex Optimisation*. Cambridge University Press, UK, 2004.

Brodin, P. and Davis, M. M. Human immune system variation. *Nat Rev Immunol*, 17(1):21–29, 1 2017. doi: 10.1038/nri.2016.125.

Claret, L., Girard, P., Hoff, P. M., Van Cutsem, E., Zuideveld, K. P., Jorga, K., Fagerberg, J., and Bruno, R. Model-based prediction of phase III overall survival in colorectal cancer on the basis of phase II tumor dynamics. *Journal of Clinical Oncology*, 27(25):4103–4108, 2009.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117.

Dax, M., Wildberger, J., Buchholz, S., Green, S. R., Macke, J. H., and Schölkopf, B. Flow matching for scalable simulation-based inference. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, volume 36. Curran Associates, Inc., 2023.

Diekstra, M., Fritsch, A., Kanefendt, F., Swen, J., Moes, D., Sörgel, F., Kinzig, M., Stelzer, C., Schindele, D., Gauler, T., Hauser, S., Houtsma, D., Roessler, M., Moritz, B., Mross, K., Bergmann, L., Oosterwijk, E., Kiemeney, L., Guchelaar, H., and Jaehde, U. Population modeling integrating pharmacokinetics, pharmacodynamics, pharmacogenetics, and clinical outcome in patients with sunitinib-treated cancer. *CPT: Pharmacometrics & Systems Pharmacology*, 6(9):604–613, 2017. doi: 10.1002/psp4.12210.

Durkan, C., Papamakarios, G., and Murray, I. Sequential neural methods for likelihood-free inference. *arXiv preprint arXiv:1811.08723*, 2018.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Fröhlich, F., Theis, F. J., and Hasenauer, J. Uncertainty analysis for non-identifiable dynamical systems: Profile likelihoods, bootstrapping and more. In Mendes, P., Dada, J. O., and Smallbone, K. O. (eds.), *Proc. 12th Int. Conf. Comp. Meth. Syst. Biol.*, Lecture Notes in Bioinformatics, pp. 61–72. Springer International Publishing Switzerland, 11 2014.

Fröhlich, F., Reiser, A., Fink, L., Woschée, D., Ligon, T., Theis, F. J., Rädler, J. O., and Hasenauer, J. Multi-experiment nonlinear mixed effect modeling of single-cell translation kinetics after transfection. *npj Systems Biology and Applications*, 5(1):1, 2018. doi: 10.1038/s41540-018-0079-7.

Ge, Z., Bickel, P. J., and Rice, J. A. An approximate likelihood approach to nonlinear mixed effects models via spline approximation. *Computational Statistics & Data Analysis*, 46(4):747–776, 2004. doi: 10.1016/j.csda.2003.10.011.

Geffner, T., Papamakarios, G., and Mnih, A. Compositional score modeling for simulation-based inference. In *International Conference on Machine Learning*, pp. 11098–11116. PMLR, 2023.

Gillespie, D. T. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 12 1977. doi: 10.1021/j100540a008.

Goldstein, H. *Multilevel models in education and social research*. Oxford University Press, 1987.

Groenland, S. L., Mathijssen, R. H., Beijnen, J. H., Huitema, A. D., and Steeghs, N. Individualized dosing of oral targeted therapies in oncology is crucial in the era of precision medicine. *European Journal of Clinical Pharmacology*, 75:1309–1318, 2019.

Hagemann, P. L., Hertrich, J., and Steidl, G. *Generalized Normalizing Flows via Markov Chains*. Elements in Non-local Data Interactions: Foundations and Applications. Cambridge University Press, 2023. doi: 10.1017/9781009331012.

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., Begy, V., and Louppe, G. A trust crisis in simulation-based inference? your posterior approximations can be unfaithful. *stat.*, 1050, 2022.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jönsson, S., Kjellsson, M. C., and Karlsson, M. O. Estimating bias in population parameters for some models for repeated measures ordinal data using NONMEM and NLMIXED. *Journal of pharmacokinetics and pharmacodynamics*, 31:299–320, 2004. doi: 10.1023/B:JOPA.0000042738.06821.61.

Kreutz, C., Raue, A., Kaschek, D., and Timmer, J. Profile likelihood in systems biology. *FEBS J*, 280(11):2564–2571, 2013.

Kuhn, E. and Lavielle, M. Maximum likelihood estimation in nonlinear mixed effects models. *Comput. Stat. Data. Anal.*, 49(4):1020 – 1038, 2005. doi: http://dx.doi.org/10.1016/j.csda.2004.07.002.

Lam, S. K., Pitrou, A., and Seibert, S. Numba: A LLVM-based Python JIT compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM '15, New York, NY, USA, 2015. Association for Computing Machinery. doi: 10.1145/2833157.2833162.

Le, T. A., Baydin, A. G., Zinkov, R., and Wood, F. Using synthetic data to train neural networks is model-based reasoning. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3514–3521, 2017. doi: 10.1109/IJCNN.2017.7966298.

Lindstrom, M. and Bates, D. Nonlinear mixed effects models for repeated measures data. *Biometrics*, 9 1990.

Liston, A., Humblet-Baron, S., Duffy, D., and Goris, A. Human immune diversity: from evolution to modernity. *Nature Immunology*, 22(12):1479–1489, 2021. doi: 10.1038/s41590-021-01058-1.

Liu, D. C. and Nocedal, J. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45 (1):503–528, 1989. doi: 10.1007/bf01589116.

Lixoft SAS, a. S. P. c. Monolix 2023r1, 2023.

Llamosi, A., Gonzalez-Vargas, A. M., Versari, C., Cinquemani, E., Ferrari-Trecate, G., Hersen, P., and Batt, G. What population reveals about individual cell identity: single-cell parameter estimation of models of gene expression in yeast. *PLoS Comput. Biol*, 12(2):1–18, 02 2016. doi: 10.1371/journal.pcbi.1004706.

Maier, C., Hartung, N., de Wiljes, J., Kloft, C., and Huisinga, W. Bayesian data assimilation to support informed decision making in individualized chemotherapy. *CPT: pharmacometrics & systems pharmacology*, 9(3):153–164, 2020.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Pardi, N., Hogan, M. J., Porter, F. W., and Weissman, D. mrna vaccines – a new era in vaccinology. *Nature Reviews Drug Discovery*, 17(4):261–279, 2018. doi: 10.1038/nrd.2017.243.

Persson, S., Welkenhuysen, N., Shashkova, S., Wiqvist, S., Reith, P., Schmidt, G. W., Picchini, U., and Cvijovic, M. Scalable and flexible inference framework for stochastic dynamic single-cell models. *PLoS Computational Biology*, 18(5):e1010082, 2022.

Pieschner, S., Hasenauer, J., and Fuchs, C. Identifiability analysis for models of the translation kinetics after mrna transfection. *Journal of Mathematical Biology*, 84(7):56, 2022. doi: 10.1101/2021.05.18.444633.

Pinheiro, J. C. *Topics in mixed effects models*. Ph.d. thesis, University of Wisconsin, Madison, Madison, USA, 1994.

Pinheiro, J. C. and Bates, D. M. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics*, 4(1): 12–35, 1995. doi: 10.2307/1390625.

Rackauckas, C. and Nie, Q. Differentialequations.jl – a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of open research software*, 5(1), 2017.

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., and Köthe, U. Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

Radev, S. T., Schmitt, M., Schumacher, L., Elsemüller, L., Pratz, V., Schälte, Y., Köthe, U., and Bürkner, P.-C. Bayesflow: Amortized bayesian workflows with neural networks. *arXiv preprint arXiv:2306.16015*, 2023.

Raue, A., Schilling, M., Bachmann, J., Matteson, A., Schelke, M., Kaschek, D., Hug, S., Kreutz, C., Harms, B. D., Theis, F. J., Klingmüller, U., and Timmer, J. Lessons learned from quantitative dynamical modeling in systems biology. *PLoS ONE*, 8(9):e74335, 9 2013.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Ribba, B., Holford, N. H., Magni, P., Trocóniz, I., Gueorguieva, I., Girard, P., Sarr, C., Elishmereni, M., Kloft, C., and Friberg, L. E. A review of mixed-effects models of tumor growth and effects of anticancer drug treatment used in population analysis. *CPT: pharmacometrics & systems pharmacology*, 3(5):1–10, 2014.

Robert, C. P. and Casella, G. *Monte Carlo Statistical Methods*. Springer, 2004.

Schälte, Y., Fröhlich, F., Jost, P. J., Vanhoefer, J., Pathirana, D., Stapor, P., Lakrisenko, P., Wang, D., Raimúndez, E., Merkt, S., Schmiester, L., Städter, P., Grein, S., Dudkin, E., Doresic, D., Weindl, D., and Hasenauer, J. pyPESTO: A modular and scalable tool for parameter estimation for dynamic models. *Bioinformatics*, 39(11):btad711, 11 2023. doi: 10.1093/bioinformatics/btad711.

Schmitt, M., Bürkner, P.-C., Köthe, U., and Radev, S. T. Detecting model misspecification in amortized Bayesian inference with neural networks. *arXiv preprint arXiv:2112.08866*, 2022.

Sharrock, L., Simons, J., Liu, S., and Beaumont, M. Sequential neural score estimation: Likelihood-free inference with conditional score based diffusion models. *arXiv preprint arXiv:2210.04872*, 2022.

Sheiner, L. B. and Beal, S. L. Evaluation of methods for estimating population pharmacokinetic parameters. I. Michaelis-Menten model: Routine clinical pharmacokinetic data. *Journal of Pharmacokinetics and Biopharmaceutics*, 8(6):553–571, 1980. doi: 10.1007/BF01060053.

Sisson, S. A., Fan, Y., and Beaumont, M. *Handbook of approximate Bayesian computation*. Chapman and Hall/CRC, 2018.

Spencer, S. L., Gaudet, S., Albeck, J. G., Burke, J. M., and Sorger, P. K. Non-genetic origins of cell-to-cell variability in TRAIL-induced apoptosis. *Nat.*, 459(7245):428–433, 5 2009.

Stumpf, P. S., Smith, R. C., Lenz, M., Schuppert, A., Müller, F.-J., Babtie, A., Chan, T. E., Stumpf, M. P., Please, C. P., Howison, S. D., et al. Stem cell differentiation as a non-markov stochastic process. *Cell Systems*, 5(4):268–282, 2017.

Swain, P. S., Elowitz, M. B., and Siggia, E. D. Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proc. Natl. Acad. Sci. USA*, 99(20):12795–12800, 10 2002. doi: 10.1073/pnas.162041399.

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., and Gelman, A. Validating Bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788*, 2020. doi: 10.48550/arXiv.1804.06788.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Wang, Y. Derivation of various NONMEM estimation methods. *Journal of Pharmacokinetics and Pharmacodynamics*, 35(2):249–249, 2008. doi: 10.1007/s10928-008-9083-7.

Wang, Z., Hasenauer, J., and Schälte, Y. Missing data in amortized simulation-based neural posterior estimation. *bioRxiv*, 2023. doi: 10.1101/2023.01.09.523219.

Wilkinson, D. J. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nat. Rev. Genet.*, 10(2):122–133, 2 2009.

Wiqvist, S., Golightly, A., McLean, A. T., and Picchini, U. Efficient inference for stochastic differential equation mixed-effects models using correlated particle pseudo-marginal algorithms. *Computational Statistics & Data Analysis*, 157:107151, 2021. doi: 10.1016/j.csda.2020.107151.

Yu, H., Steeghs, N., Kloth, J. S., De Wit, D., van Hasselt, J. C., van Erp, N. P., Beijnen, J. H., Schellens, J. H., Mathijssen, R. H., and Huitema, A. D. Integrated semi-physiological pharmacokinetic model for both sunitinib and its active metabolite su 12662. *British Journal of Clinical Pharmacology*, 79(5):809–819, 2015.

Yu, Z., Guindani, M., Grieco, S. F., Chen, L., Holmes, T. C., and Xu, X. Beyond t test and ANOVA: applications of mixed-effects models for more rigorous statistical analysis in neuroscience research. *Neuron*, 110(1):21–35, 2022. doi: 10.1016/j.neuron.2021.10.030.

Zechner, C., Unger, M., Pelet, S., Peter, M., and Koeppl, H. Scalable inference of heterogeneous reaction kinetics from pooled single-cell recordings. *Nat. Methods*, 11: 197–202, 1 2014. doi: 10.1038/nmeth.2794.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, 1 2021. doi: 10.1145/3446776.

A. Supplementary Information

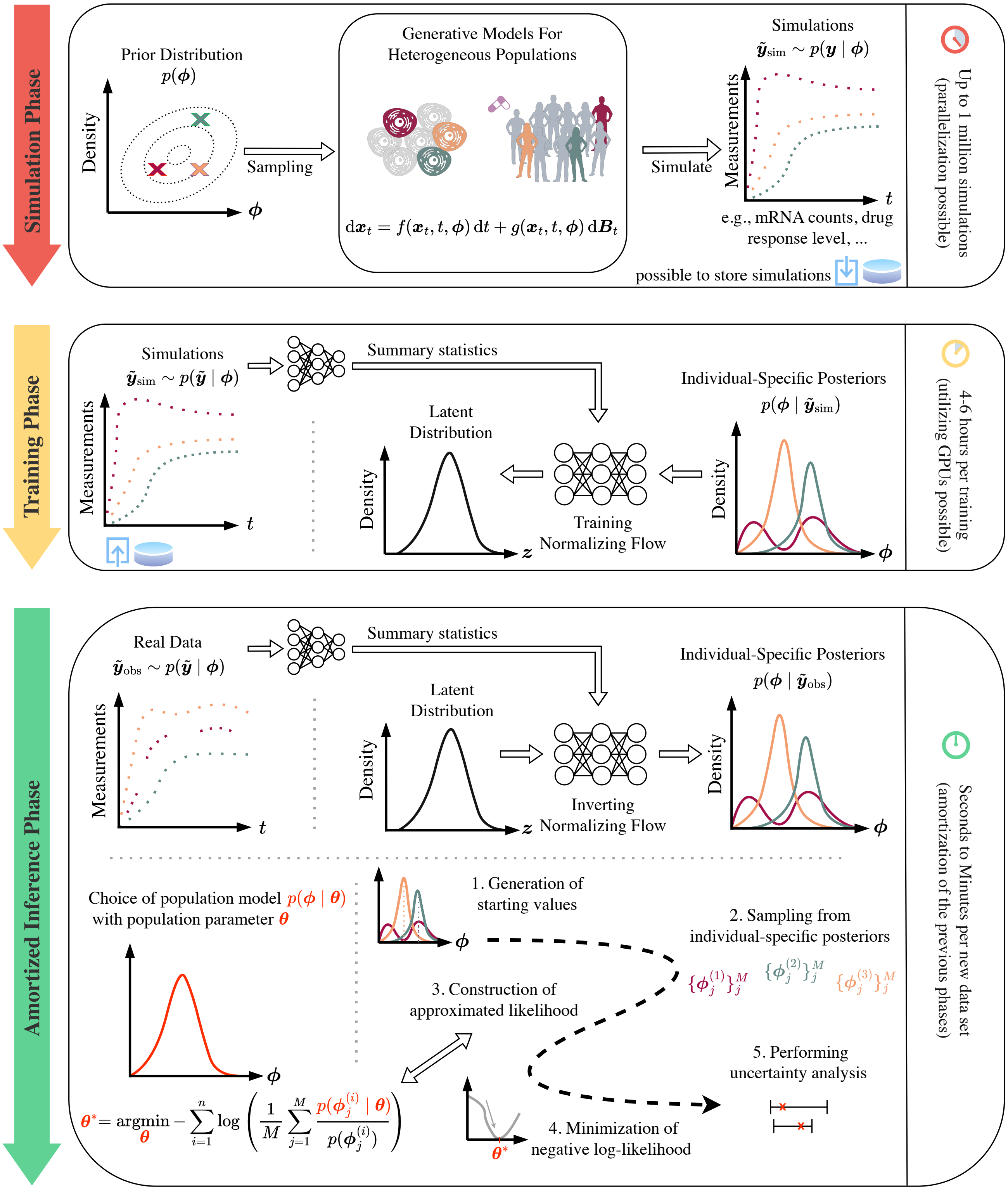

Simulation Phase
Prior Distribution
$p(\phi)$
Density
ϕ
Sampling
Generative Models For
Heterogeneous Populations
$\mathrm{d}x_t = f(x_t, t, \phi)\,\mathrm{d}t + g(x_t, t, \phi)\,\mathrm{d}B_t$
Simulate
Simulations
$\tilde{y}_{\mathrm{sim}} \sim p(y \mid \phi)$
Measurements
t
e.g., mRNA counts, drug response level, ...
possible to store simulations
Up to 1 million simulations (parallelization possible)
Training Phase
Simulations
$\tilde{y}_{\mathrm{sim}} \sim p(\tilde{y} \mid \phi)$
Measurements
t
Summary statistics
Latent
Distribution
Density
z
Training
Normalizing Flow
Individual-Specific Posteriors
$p(\phi \mid \tilde{y}_{\mathrm{sim}})$
Density
ϕ
4-6 hours per training (utilizing GPUs possible)
Amortized Inference Phase
Real Data
$\tilde{y}_{\mathrm{obs}} \sim p(\tilde{y} \mid \phi)$
Measurements
t
Summary statistics
Latent
Distribution
Density
z
Inverting
Normalizing Flow
Individual-Specific Posteriors
$p(\phi \mid \tilde{y}_{\mathrm{obs}})$
Density
ϕ
Choice of population model $p(\phi \mid \theta)$
with population parameter θ
ϕ
$\theta^* = \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^{n} \log \left(\frac{1}{M} \sum_{j=1}^{M} \frac{p(\phi_j^{(i)} \mid \theta)}{p(\phi_j^{(i)})} \right)$
ϕ
1. Generation of
starting values
2. Sampling from
individual-specific posteriors
$\{\phi_j^{(1)}\}_j^M$
$\{\phi_j^{(2)}\}_j^M$
$\{\phi_j^{(3)}\}_j^M$
3. Construction of
approximated likelihood
4. Minimization of
negative log-likelihood
θ^*
5. Performing
uncertainty analysis
Seconds to Minutes per new data set (amortization of the previous phases)

Figure S1. Detailed concept visualization of the neural posterior estimation based amortized approach to NLME model inference.

Code Availability

The code and a guide, aimed at assisting users in applying their own non-linear mixed-effects models, can be found at https://github.com/arrjon/Amortized-NLME-Models/tree/ICML2024. A snapshot of the code and the results underlying this study can be found at https://zenodo.org/record/8245785. The single-cell data has been made publicly available by Fröhlich et al. (2018).

A.1. Specification of the single-cell models

Living cells exhibit variability at the single cell level due to various factors such as mole processes, cell cycle state, environmental differences, and individual cell history (Fröhlich et al., 2018). Fröhlich et al. were interested in the dynamics of protein expression and transfected single cells with enhanced green fluorescent protein (eGFP)-encoding mRNA. The expression of the eGFP reporter gene was recorded every ten minutes for a period of 30 hours using a scanning time-lapse microscope setup. From these data, the authors estimated the parameters of the translation process using ordinary differential equation (ODE) models in an NLME framework.

In this work, we focus on two models termed the “simple” and “detailed” models from (Fröhlich et al., 2018). We denote the abundance of mRNA as m, proteins as p, ribosomes as r, and enzymes as e. For both models, we assume additive normal measurement noise, that is, the measurements $\tilde{y}$ follow $\tilde{y} \sim N(y, \sigma^2)$ and our assumed prior distribution for σ is $\log N(-1, 2)$.

Simple ODE model

The ODE system has variables $\boldsymbol{\phi} = (\delta, \gamma, km_0\, scale, t_0, \mathit{offset}, \sigma)$ and is defined as

$$
\begin{aligned}
\frac{dm}{dt} &= -\delta \cdot m && m(t_0) = 1 \\
\frac{dp}{dt} &= km_0\, scale \cdot m - \gamma \cdot p && p(0) = 0 \\
y &= \log(p + \mathit{offset}),
\end{aligned}
$$

where the priors assumed for the variables are

- mRNA degradation rate $\delta \sim \log \mathcal{N}(-3, 5)$,
- protein degradation rate $\gamma \sim \log \mathcal{N}(-3, 5)$,
- combined parameters $km_0\, scale \sim \log \mathcal{N}(5, 11)$,
- mRNA entering the cell time point $t_0 \sim \log \mathcal{N}(0, 2)$,
- and $\mathit{offset} \sim \log \mathcal{N}(0, 6)$.

The parameters k, m_0, $scale$ can only be identified as a product to improve identifiability (Fröhlich et al., 2018). This ODE system has an analytical solution, which we use to perform simulations in Python.

Detailed ODE model

The ODE system has variables $\boldsymbol{\phi} = (\delta_1 m_0, \delta_2, e_0 m_0, k_2 m_0\ \mathit{scale}, k_2, k_1 m_0, r_0 m_0, \gamma, t_0, \mathit{offset}, \sigma)$ and is defined as

$$\begin{aligned}
\frac{dm}{dt} &= -\delta_1 m_0 \cdot m \cdot e - k_1 m_0 \cdot m \cdot r + k_2 \cdot \left(\frac{r_0}{m_o} - r\right) & m(t_0) &= 1 \\
\frac{de}{dt} &= -\delta_1 m_0 \cdot m \cdot e + \delta_2 \cdot \left(\frac{e_0}{m_0} - e\right) & e(0) &= \frac{e_0}{m_0} \\
\frac{dr}{dt} &= k_2 \cdot \left(\frac{r_0}{m_0} - r\right) - k_1 m_0 \cdot m \cdot r & r(0) &= \frac{r_0}{m_0} \\
\frac{dp}{dt} &= k_2 m_0\ \mathit{scale} \cdot \left(\frac{r_0}{m_0} - r\right) - \gamma \cdot p & p(0) &= 0 \\
y &= \log(p + \mathit{offset}).
\end{aligned}$$

For a detailed description of the parameters, we refer to (Fröhlich et al., 2018). The priors assumed for the variables are

$$\begin{aligned}
\delta_1 m_0 &\sim \log\mathcal{N}(-1, 5), & k_1 m_0 &\sim \log\mathcal{N}(1, 2) \\
\delta_2 &\sim \log\mathcal{N}(-1, 5), & \frac{r_0}{m_0} &\sim \log\mathcal{N}(-1, 2), \\
\frac{e_0}{m_0} &\sim \log\mathcal{N}(-1, 2), & \gamma &\sim \log\mathcal{N}(-6, 5), \\
k_2 m_0\ \mathit{scale} &\sim \log\mathcal{N}(12, 2), & t_0 &\sim \log\mathcal{N}(0, 2), \\
k_2 &\sim \log\mathcal{N}(-1, 2), & \mathit{offset} &\sim \log\mathcal{N}(0, 5).
\end{aligned}$$

This ODE system is simulated using the Rodas5P solver implemented in the Julia package `DifferentialEquations.jl` (Rackauckas & Nie, 2017).

SDE model

The simple ODE model can be easily extended to the SDE model

$$d\begin{pmatrix} m \\ p \end{pmatrix} t = \begin{pmatrix} -\delta \cdot m(t) \\ k \cdot m(t) - \gamma \cdot p(t) \end{pmatrix} dt + \begin{pmatrix} \sqrt{\delta m(t)} & 0 \\ 0 & \sqrt{k \cdot m(t) + \gamma \cdot p(t)} \end{pmatrix} dB_t$$

from (Pieschner et al., 2022) with $\boldsymbol{\phi} = (\delta, \gamma, k, m_0, \mathit{scale}, \mathit{offset}, \sigma)$, where B_t is a two-dimensional standard Brownian motion, $m(t_0) = 1$ and $p(0) = 0$. To compare the model to the previous one we take as observable mapping

$$y = \log(\mathit{scale} \cdot p + \mathit{offset}).$$

The priors assumed for the variables are

$$\begin{aligned}
\delta &\sim \log\mathcal{N}(-3, 5), & \mathit{scale} &\sim \log\mathcal{N}(0, 5), \\
\gamma &\sim \log\mathcal{N}(-3, 5), & t_0 &\sim \log\mathcal{N}(0, 2), \\
k &\sim \log\mathcal{N}(-1, 5), & \mathit{offset} &\sim \log\mathcal{N}(0, 5). \\
m_0 &\sim \log\mathcal{N}(5, 5).
\end{aligned}$$

This SDE system is simulated based on an Euler-Maruyama scheme with a step size of 0.01 and using just in time compilation from `numba` (Lam et al., 2015).

A.1.1. Synthetic data

The synthetic data set is generated by setting the population parameters to reasonable values based on the results in (Fröhlich et al., 2018) (see Table 1, 2 and 3) and then sampling random effects from a log-normal distribution until the desired number of synthetic cells is generated. Since we know all cell-specific parameters, we can compute the sample mean and covariance of the parameters, which are the optimal values that we would like to recover. Fixed effects are modeled as random effects with 0 variance.

Table 1. Population parameters of log-normal distribution for synthetic data of simple single-cell NLME model.

parameter	δ	γ	km_0 *scale*	t_0	*offset*	σ
mean	-0.694	-7.014	6.217	-0.164	2.079	-3.454
variance	0.941	7.014	0.004	0.116	0	0

Table 2. Population parameters of log-normal distribution for synthetic data of detailed single-cell NLME model.

parameter	$\delta_1 m_0$	δ_2	$e_0 m_0$	$k_2 m_0$ *scale*	k_2	$k_1 m_0$	$r_0 m_0$	γ	t_0	*offset*	σ
mean	-0.10144	-0.88443	-0.42549	13.81551	0.42143	0.97477	-3.50153	-6.91273	-0.34573	2.07944	-3.45388
variance	0.56752	0.74721	0.52594	0	0.44084	1.45996	2.3979	4.61512	0.48075	0	0

Table 3. Population parameters of log-normal distribution for synthetic data of the SDE single-cell NLME model.

parameter	δ	γ	k	m_0	*scale*	t_0	*offset*	σ
mean	-0.694	-7.014	0.027	5.704	0.751	-0.164	2.079	-3.454
variance	0.941	7.014	0.675	$6 \cdot 10^{-5}$	0	0.116	0	0

A.2. Implementation

We implemented the individual-specific posterior approximation using the `BayesFlow` tool (Radev et al., 2023). To estimate the population parameters, we implemented the optimization problem (5) as an objective function in the `pyPESTO` toolbox (Schälte et al., 2023). There, we used the local optimization method L-BFGS (Liu & Nocedal, 1989) implemented in `SciPy` (Virtanen et al., 2020) embedded in a multi-start framework with uniformly sampled starting points in the 99% range of the prior. In our applications, usually ten starts were already enough to reliably obtain the same optimum several times (see Figure S14). Parameters that are shared between individuals, that is, parameters that do not consist of a random effect, can be approximated in the given approach by fixing their variance to a small value.

As invertible neural networks, we used neural spline layers (Durkan et al., 2019) with variable depth of seven to eight layers and two to three coupling layers. Since all models describe trajectories over time, we chose a long short-term memory (LSTM) network (Hochreiter & Schmidhuber, 1997) with 2^d units as our summary network. We choose d such that the number of units is larger than the number of observations given by the model.

For each model, multiple neural posterior estimators were trained. We varied the number of invertible layers from six to eight, added a $1d$-convolutional layer on top of the LSTMs and a dense layer at the end. Training consists of several epochs, and in each one we generated 1000 batches of 128 simulations. Simulations can be generated before or during training. Depending on the simulation time of the model, pre-simulation or online training is more efficient. We used online training for the simple ODE model, while we generated simulations beforehand for the other models. For all models, we set a maximum of 500 epochs and training was stopped earlier if the loss calculated on a validation set did not improve for five epochs, which reduced training time and is assumed to improve the generalization capacity of neural networks (Zhang et al., 2021). The error calculated on a validation set during training suggested convergence for all models (Figure S2).

For each start in `Monolix` (Lixoft SAS, 2023), we increased the iteration limit for each task (SAEM, standard error, and likelihood estimation) to ensure convergence at each step. We set the maximum number of iterations in the two phases (exploratory and smoothing) of SAEM to 10,000 and 1,000, respectively. `Monolix`' auto-stop criteria usually stopped the algorithm before it could perform that many iterations. The iteration limit for estimation of the standard errors was set to 1,000 and the Monte Carlo size for likelihood estimation was set to 50,000. All other settings were left at their default values. The starting points were sampled from the prior.

For comparison, we also explored a multi-start approach using FOCEI (Wang, 2008), implemented in `NONMEM` (Beal & Sheiner, 1980), which is arguably the most widely used inference method in pharmacokinetic modeling. We performed 300 starts, where we sampled the starting values from the prior.

We ran all analyses on a computing cluster using eight CPU cores for parallelization and one GPU for training the neural networks. The computing cluster uses an AMD EPYC 7F72 CPU with a core clock speed of up to 3.2 GHz and 20 GB of RAM per available core. The neural network training was performed on a cluster node with an NVIDIA A100 graphics card with 40 GB of VRAM.

The simulations of the generative model, multi-starts in pyPESTO and a single start in Monolix used all available cores for parallelization. Moreover, the contribution of each individual could also be evaluated in parallel, giving the option of further parallelizing the calculation of the objective in a single start.

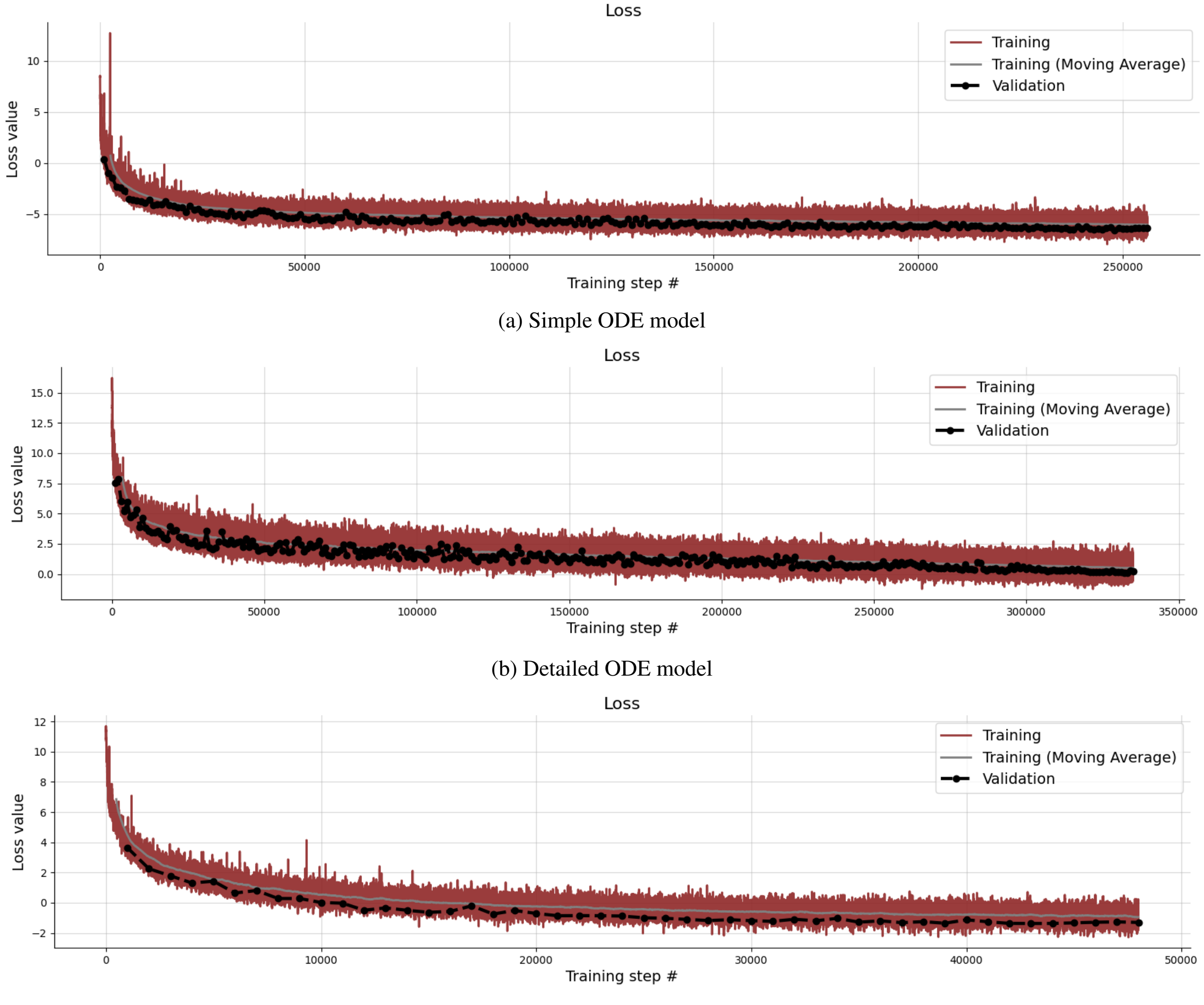
Table 4. Detailed runtime analysis. We report duration of the simulation phase, the average training time, and the number of simulations of the underlying ODE or SDE model. For SAEM we report the average number of ODE model simulations (assuming one simulation per individual per iteration) for the data set with $10,000$ cells and for all data sets and repeated inference runs combined. For any new data set, there are no additional simulations in the amortized approach. SAEM is not capable to perform inference for the SDE NLME model.

model	(I) simulation phase	(II) training phase	epochs	total simulations	# simulations in one SAEM run	total number of simulations in SAEM
simple	-	6.11 h	416	53 Mio.	34.8 Mio.	5 030 Mio.
detailed	3.74 h	8.00 h	355	64 Mio.	99.8 Mio.	9 640 Mio.
SDE	2.16 h	5.12 h	417	64 Mio.	-	-

A.3. Conditional normalizing flows provide accurate and efficient approximation of individual-specific posteriors

We checked the convergence of neural posterior estimators based on their calibration plots, a diagnostic tool that comes with BayesFlow. Simulation-based calibration is a method to detect systematic biases in any Bayesian posterior sampling method (Talts et al., 2020). Incorrect calibration can be seen by deviations from uniformity. None of our estimators showed systematic biases (Figure S3). Furthermore, for the best estimators, we assessed the validity of the individual-specific posteriors of the real data by comparing them with the posterior approximations given by an MCMC approximation with adaptive parallel tempering implemented in pyPESTO (Schälte et al., 2023). In particular, the bimodal distributions of the parameters δ and γ in the simple ODE model are nicely recovered (Figure S4). For the detailed model, MCMC showed poor convergence behavior over repeated runs with a small effective sample size.

An assessment of computation time revealed that the employed MCMC sampler required approximately 1 million samples and 10 chains with an effective sample size of 195, which took around 20 minutes of computation time for a single cell. In comparison, the trained neural posterior estimator only required a few seconds for the same effective sample size and on the same set-up (see details on the implementation in Methods A.2). Thus, in this case, the training time of the neural networks to obtain individual-specific posteriors, ~ 6.5 hours, would be amortized after around 20 cells, or even after an individual cell if a sufficiently high sample size is required. This demonstrates the efficiency of neural posterior estimation for parameter estimation also outside a mixed-effects context.

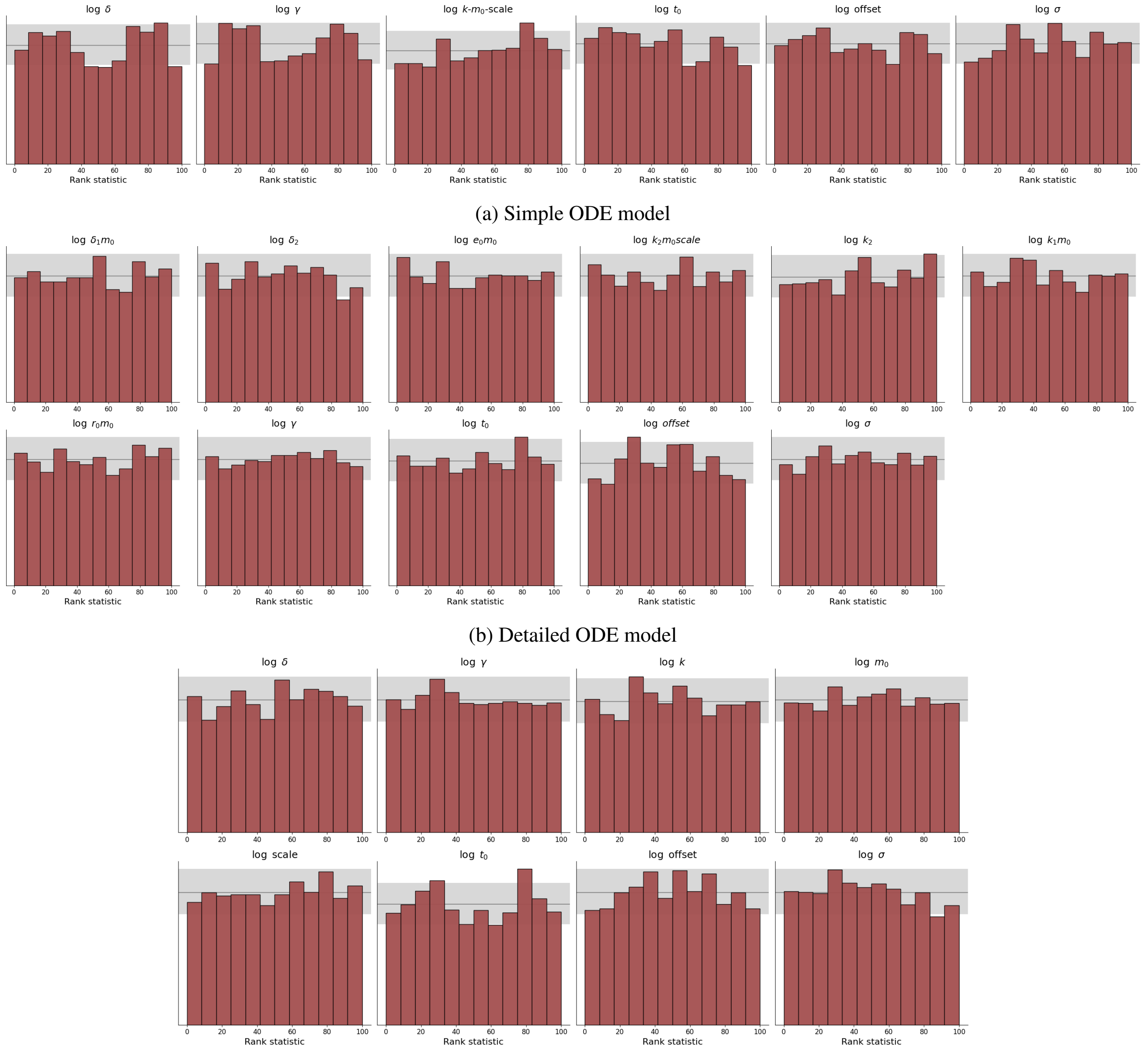


(a) Simple ODE model

(b) Detailed ODE model

(c) SDE model

Figure S2. Exemplary loss during training.

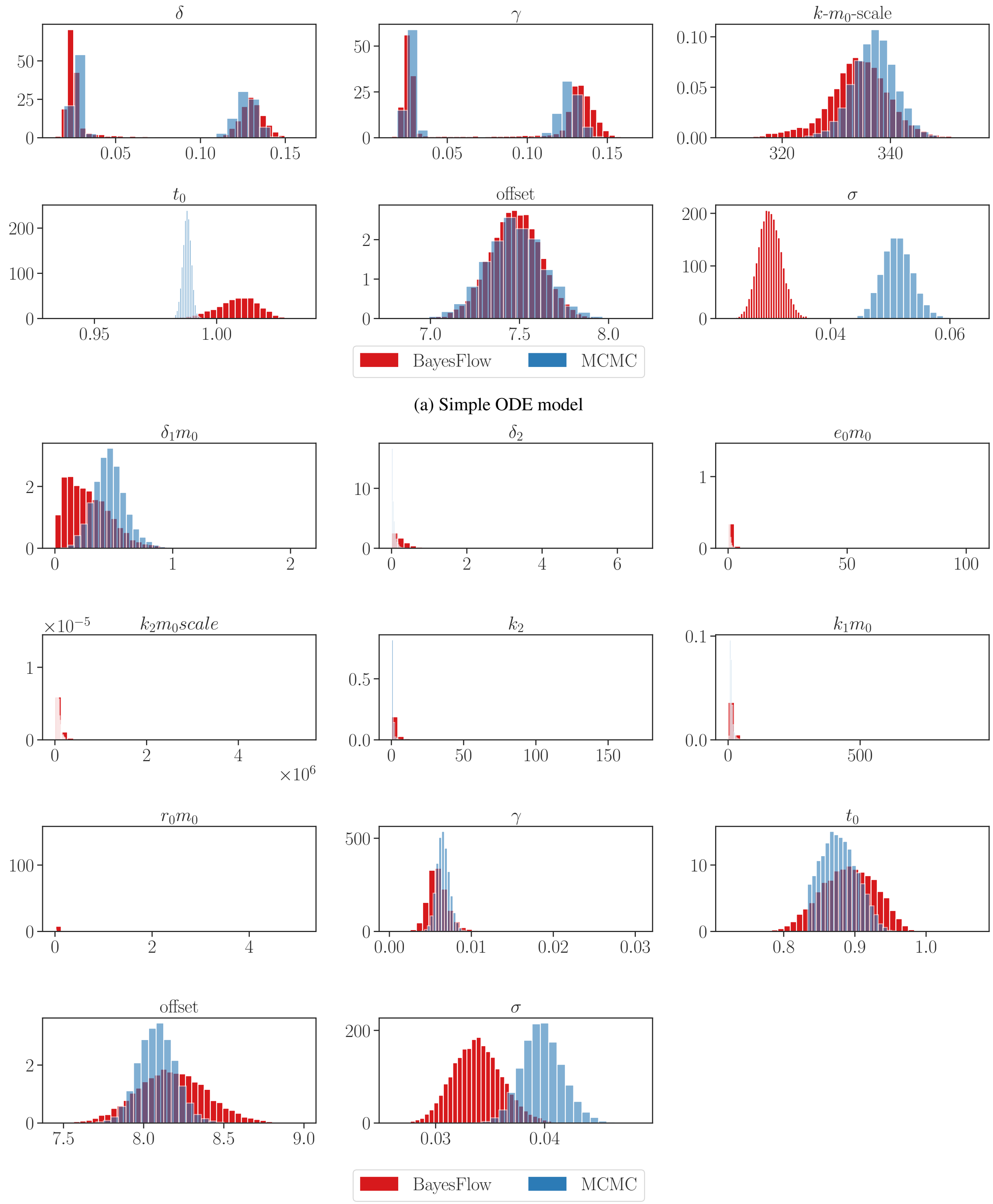


(a) Simple ODE model

(b) Detailed ODE model

(c) SDE model

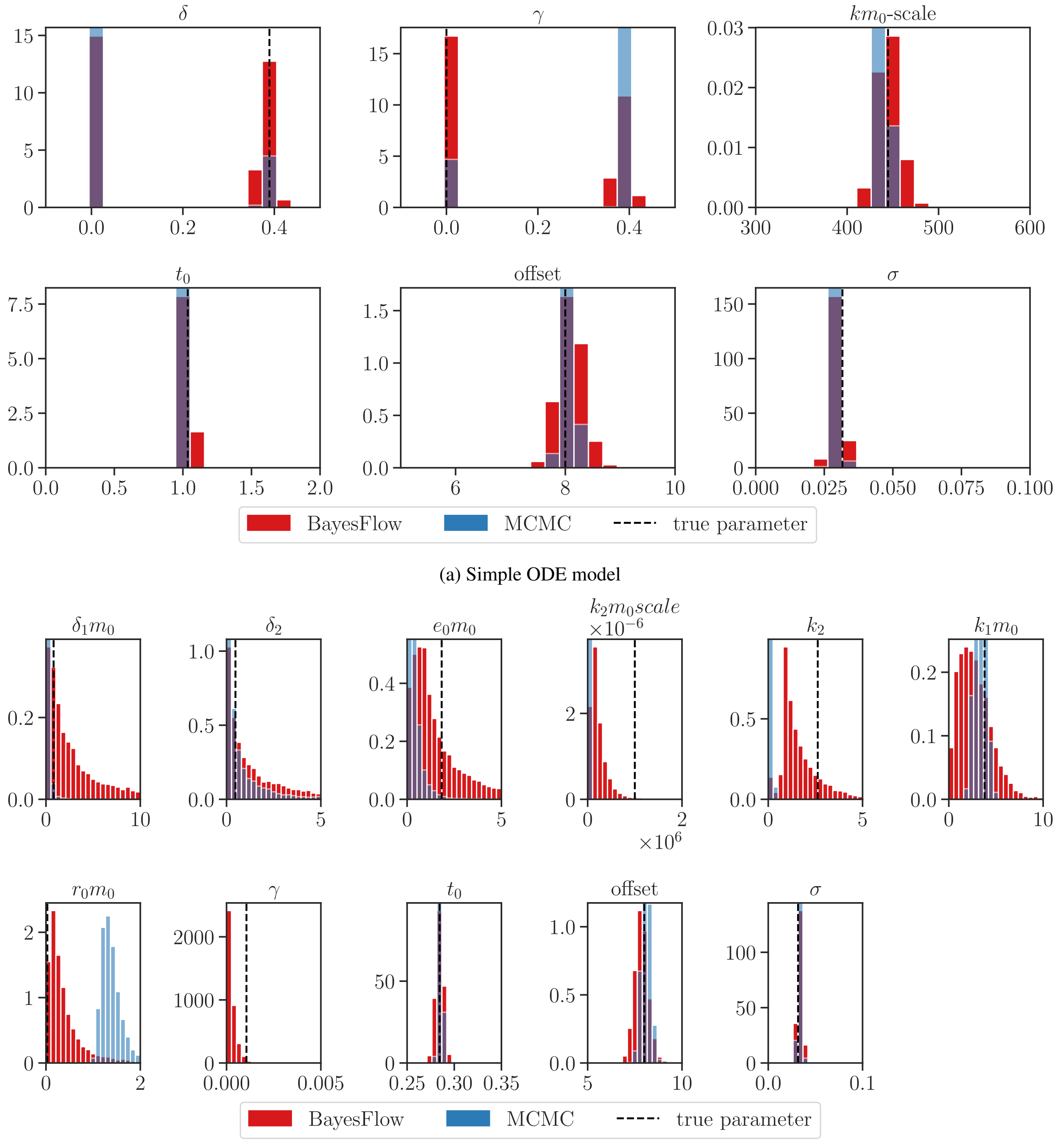
Figure S3. Simulation-based calibration plots of the individual posteriors for the (a) simple ODE, (b) detailed ODE and (c) SDE models. Incorrect calibration can be seen by deviations from uniformity (bars outside the gray area).



(a) Simple ODE model

(b) Detailed ODE model

Figure S4. Comparing individual-specific posteriors from an MCMC approximation and the neural posterior estimator for a single real cell in the (a) simple and the (b) detailed ODE model.



(a) Simple ODE model

(b) Detailed ODE model

Figure S5. Comparing individual-specific posteriors from an MCMC approximation and the neural posterior estimator for a single synthetic cell in the (a) simple and the (b) detailed ODE model.

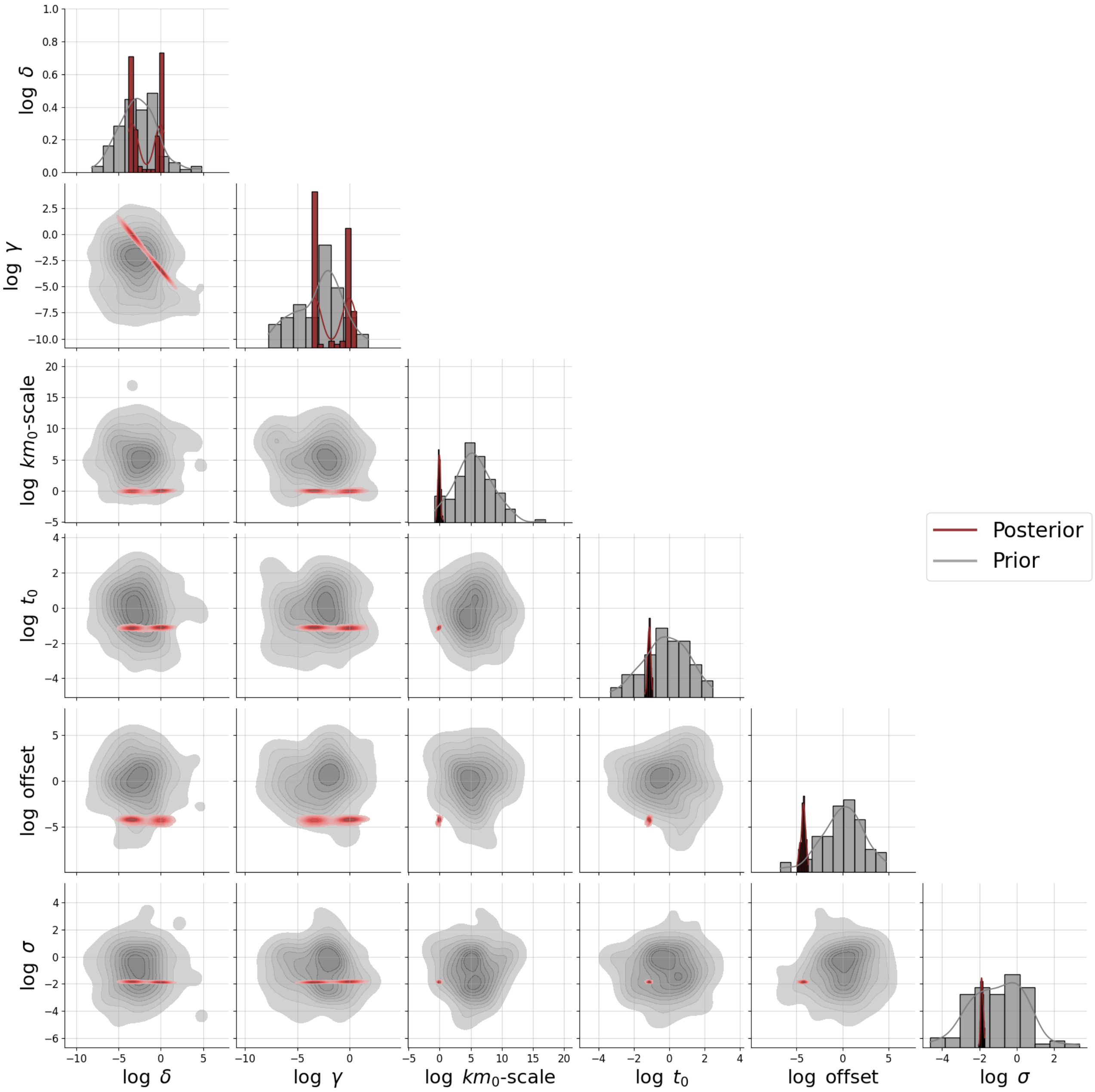


Figure S6. Full posterior for the simple ODE model and an exemplary real single cell

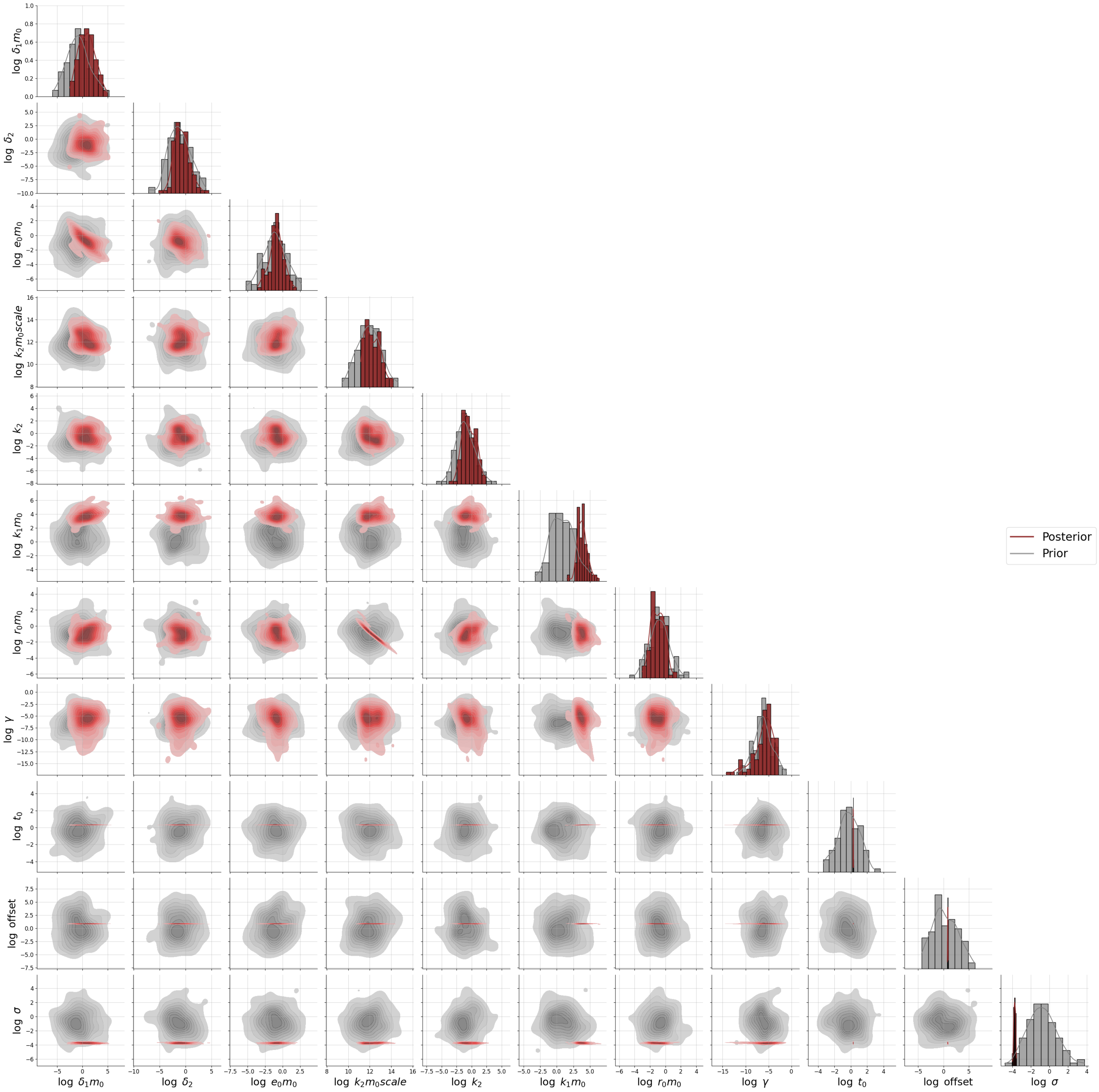


Figure S7. Full posterior for the detailed ODE model and an exemplary real single cell

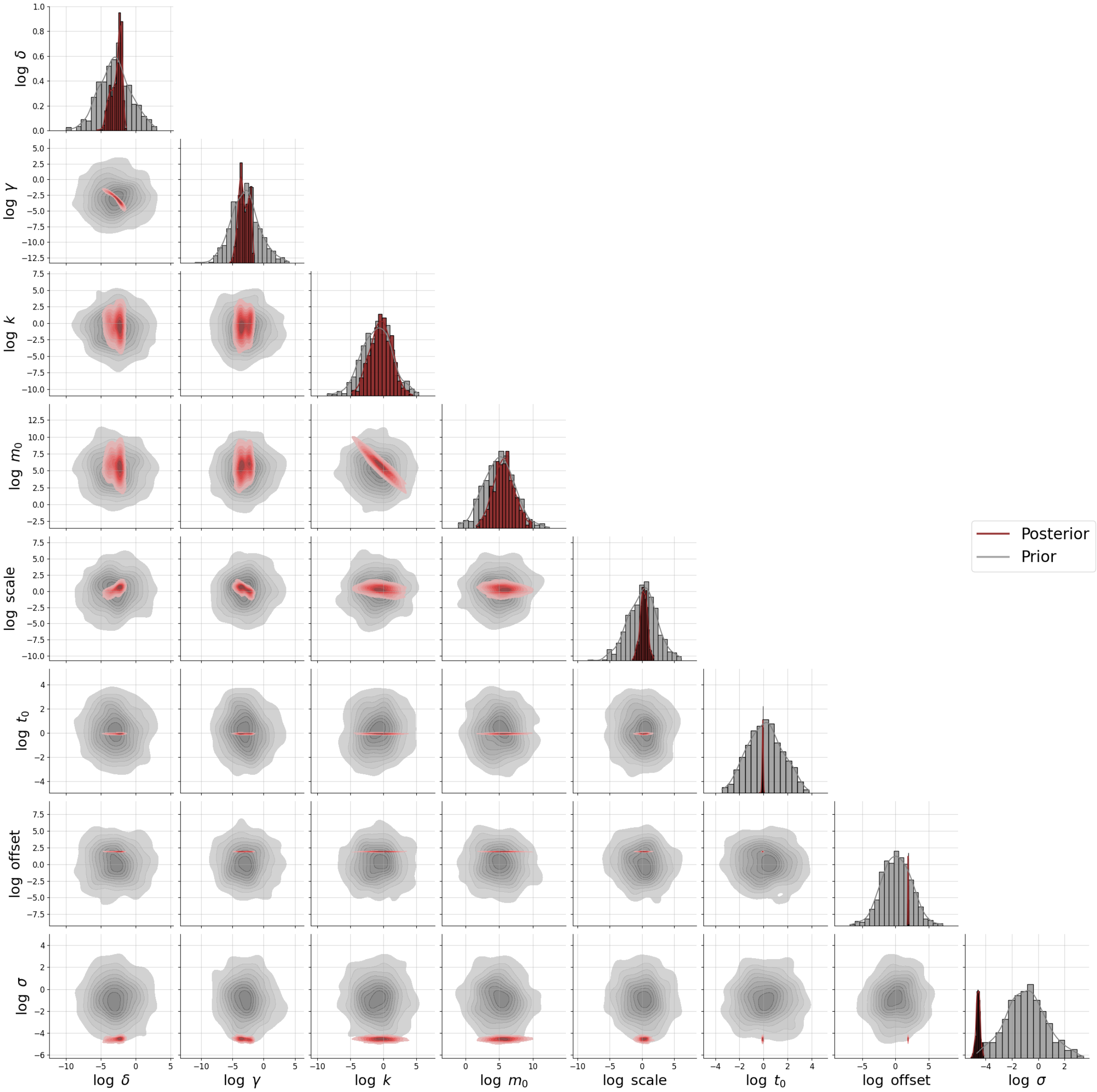


Figure S8. Full posterior for the SDE model and an exemplary real single cell

A.4. Further analysis of the single-cell NLME models

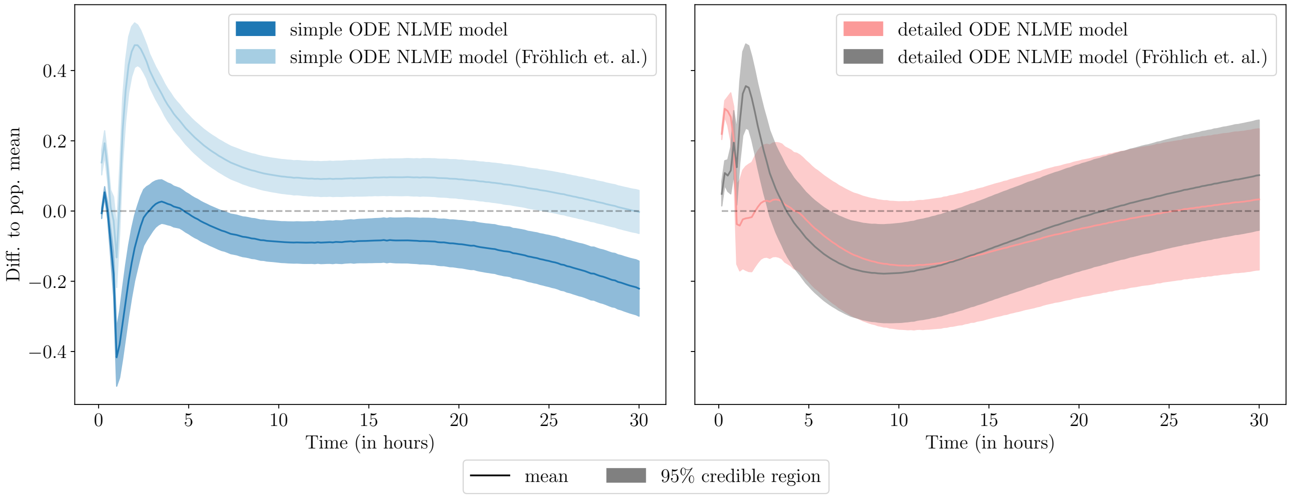


Figure S9. The difference in the population mean estimated from real trajectories and simulations generated with the estimated population parameters is shown with a 95% confidence interval (CI). The detailed model captures the population mean after 0.82 hours and then shows no significant deviation from the population mean, while the simple model needs 1.97 hours to recover the population mean and then differs significantly from the population mean. In addition to the models fitted with the amortized approach, the best fit of Fröhlich et al. (2018) for the simple and detailed ODE NLME model is shown (Fröhlich et al., 2018).

A.4.1. Influence of the number of posterior samples

We also checked the impact of the number of posterior samples M on the estimated population parameters. In Figure S10a, we see a small decrease in the mean squared error for a larger number of posterior samples across all ODE NLME models and data sets. Monte Carlo integration theory suggests that the error rate ($\sigma_{\text{MC}}/\sqrt{M}$) should be 5 times lower when increasing the sample size from $M = 10$ samples to $M = 250$. However, the median error decreases only by a factor of around 1.5. We assume that the lower rate comes from the fact that we already have a good approximation with a small number of samples due to the following observation: In the cases shown here, the population density p_{pop} and the prior $p(\boldsymbol{\phi})$ come from the exponential family. Hence, the loss function consists of logarithms of sums of exponentials. For example, if both are normal distributions $\mathcal{N}(\boldsymbol{\beta}, \boldsymbol{\Psi})$, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ respectively, then we can define $x_j := \frac{1}{2}(\boldsymbol{\phi}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi}_j - \boldsymbol{\mu}) - \frac{1}{2}(\boldsymbol{\phi}_j - \boldsymbol{\beta})^T \boldsymbol{\Psi}^{-1}(\boldsymbol{\phi}_j - \boldsymbol{\beta})$. We can bound the logarithm of sums of exponentials, by the maximum function (Boyd & Vandenberghe, 2004) through

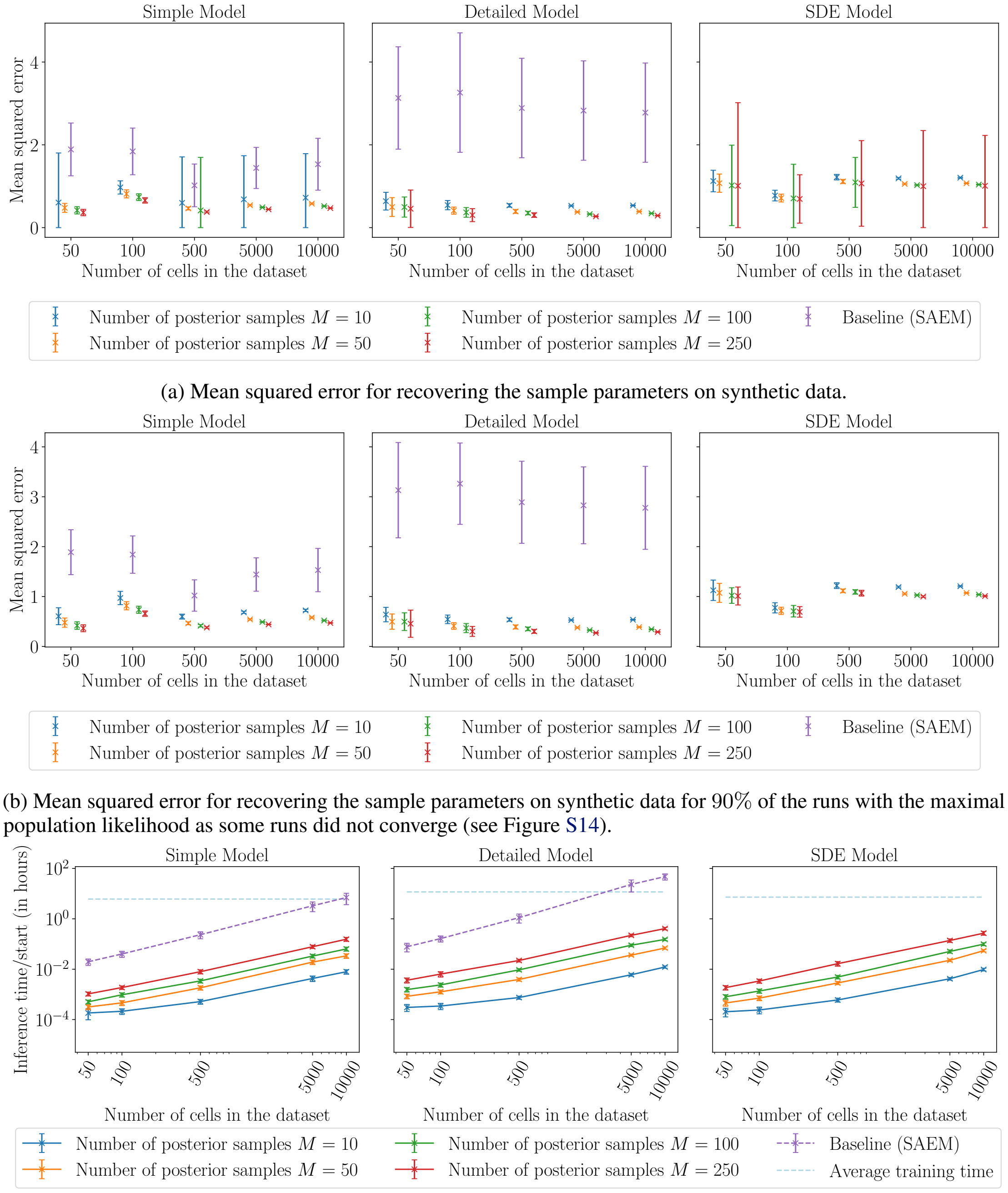
$$\max(x_1, \dots, x_M) \leq \text{logsumexp}(x_1, \dots, x_M) \leq \max(x_1, \dots, x_M) + \log M.$$

This is true since $\max(x_1, \dots, x_M) = \log(\max(\exp(x_1, \dots, x_M))) < \log\left(\sum_{j=1}^{M} \exp(x_j)\right)$ and $\sum_{j=1}^{M} \exp(x_j) \leq M \max(\exp(x_1), \dots, \exp(x_M))$. The latter inequality is an equality if and only if all x_j are equal. The log-sum-exp function is even convex (Boyd & Vandenberghe, 2004) and can be numerically stable evaluated by using the log-sum-exp-trick (Blanchard et al., 2021).

Therefore, the main contribution to the loss function comes from "good" posterior samples $\boldsymbol{\phi}_j$ that optimally balance the population distribution and individual priors (a maximal x_j). If we add a "bad" sample to a set of already "good" samples, only the upper bound will change by $\log((M + 1)/M)$. Intuitively, this explains why we only need a small number of "good" samples to get a reasonable approximation of the population likelihood.

In Figure S10b, we see that the inference time increases when a larger sample size is used for the posterior, but that inference is still faster than the baseline method. Moreover, for each model, we reused the same neural posterior estimator on all data sets, whereas for the baseline method, we needed to restart the whole optimization. Therefore, to get estimates for 100 runs on multiple data sets, we saved 128 times of computational resources using the amortized approach compared to SAEM.

In Figure S11–S13, we see that for the amortized approach the variance of the parameter estimates of a multi-start optimization decreases with increasing posterior sample size or size of the data set. However, in general, for increasing size of the data set, more multi-starts were needed for our approach and SAEM (Figure S14). The amortized approach shows that for a posterior sample size of at least $M = 50$, most runs reach a similar likelihood value, in particular for the smaller data sets. Furthermore, the amortized approach is able to recover the multi-modality in the first two parameters of the simple ODE NLME model (Figure S11). This modality comes from the fact, that we can swap δ and γ in the simple ODE model without changing the solution of the ODE.



(a) Mean squared error for recovering the sample parameters on synthetic data.

(b) Mean squared error for recovering the sample parameters on synthetic data for 90% of the runs with the maximal population likelihood as some runs did not converge (see Figure S14).

(c) Inference time for multiple new synthetic data sets.

Figure S10. Accuracy and inference time for multiple new synthetic data sets for the single-cell NLME models. The median of the mean squared error for 100 runs is shown with one standard deviation. Different numbers of posterior samples were used to estimate population parameters. For each model, we reused the same neural posterior estimator on all data sets.

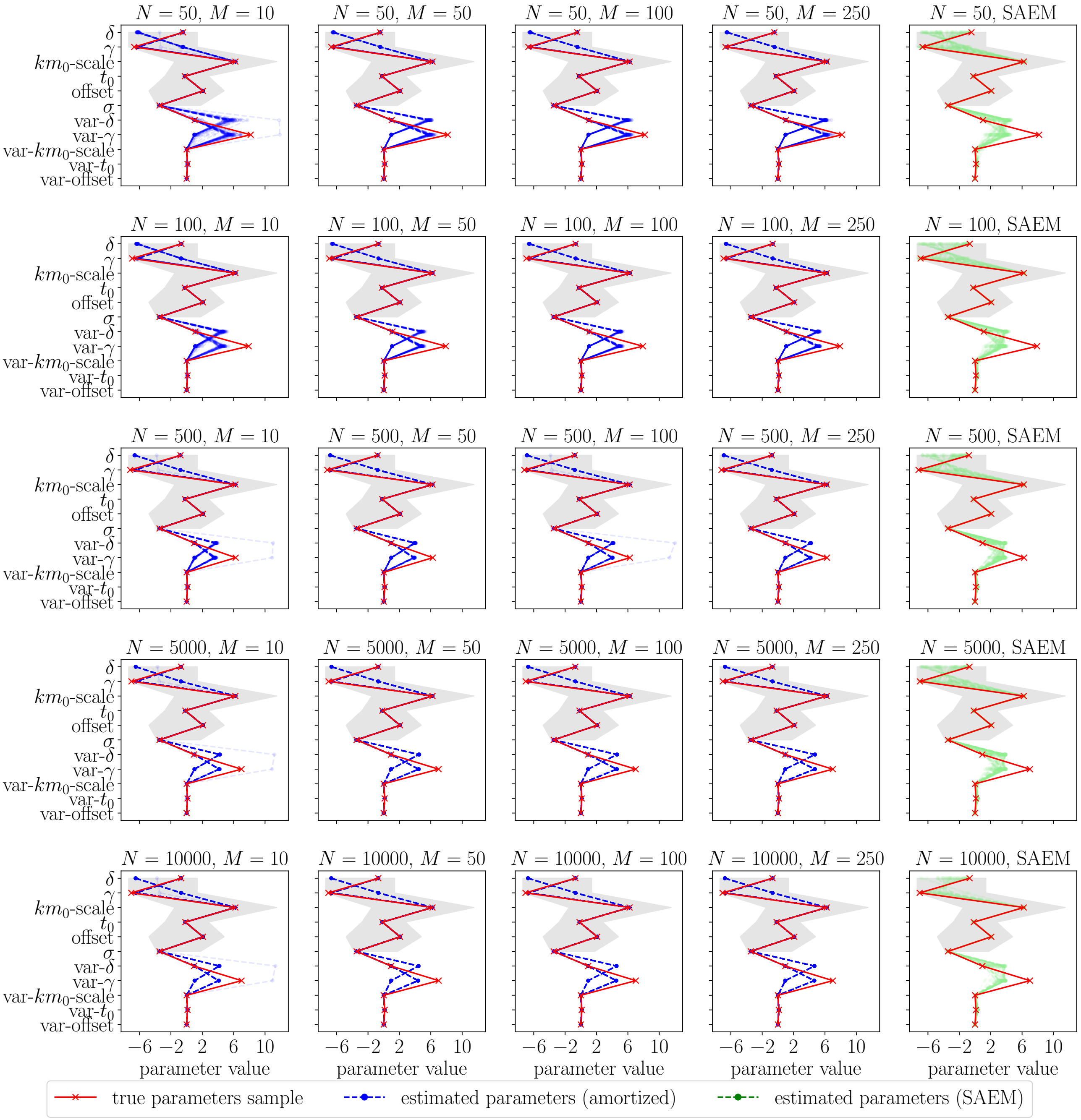


Figure S11. Parameter estimates on synthetic data sets for the simple single-cell NLME model for 100 runs. Different numbers of posterior samples M were used to estimate population parameters. For each data set of size N, we reused the same neural posterior estimator.

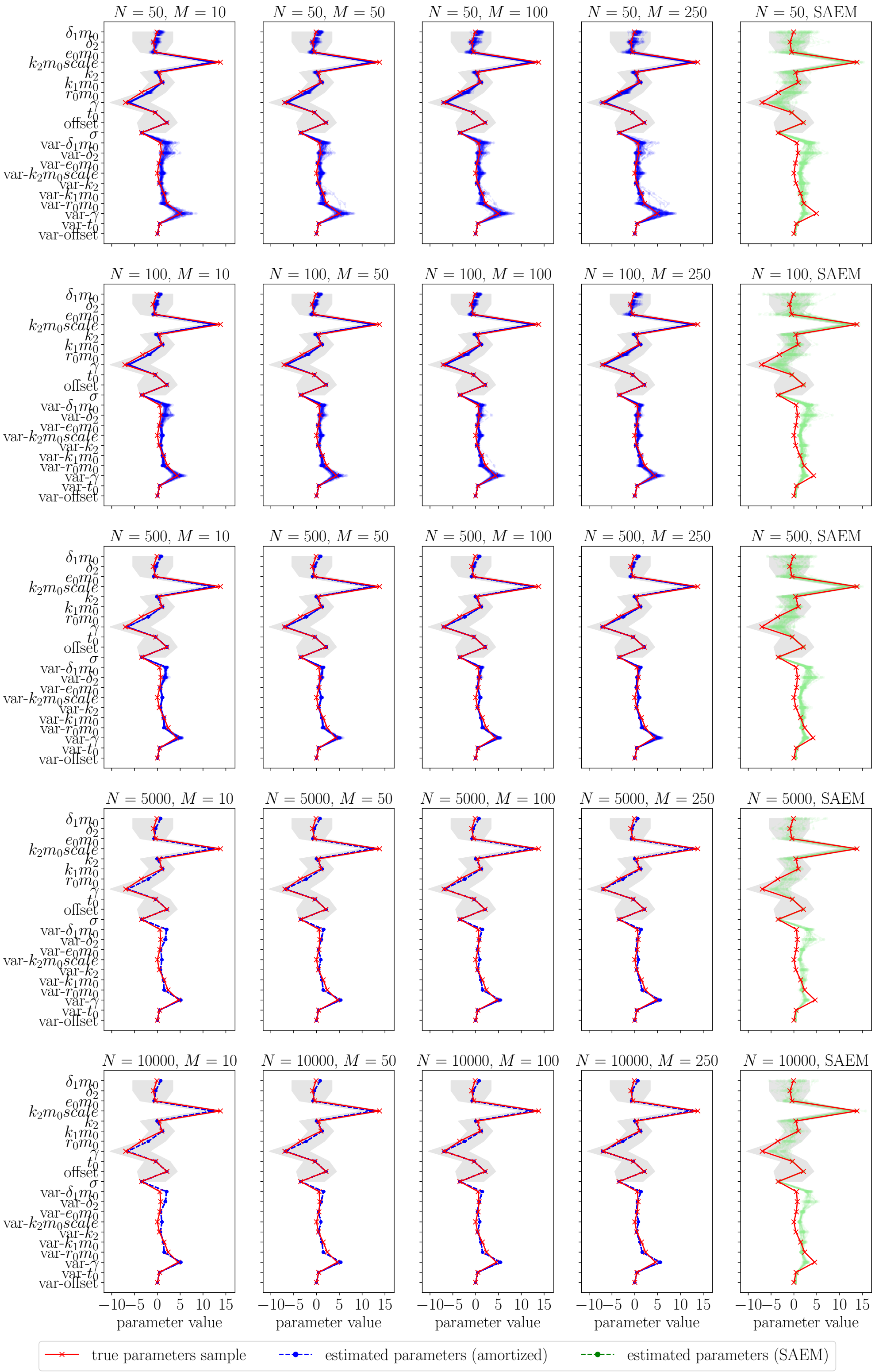


Figure S12. Parameter estimates on synthetic data sets for the detailed single-cell NLME model for 100 runs. Different numbers of posterior samples M were used to estimate population parameters. For each data set of size N, we reused the same neural posterior estimator.

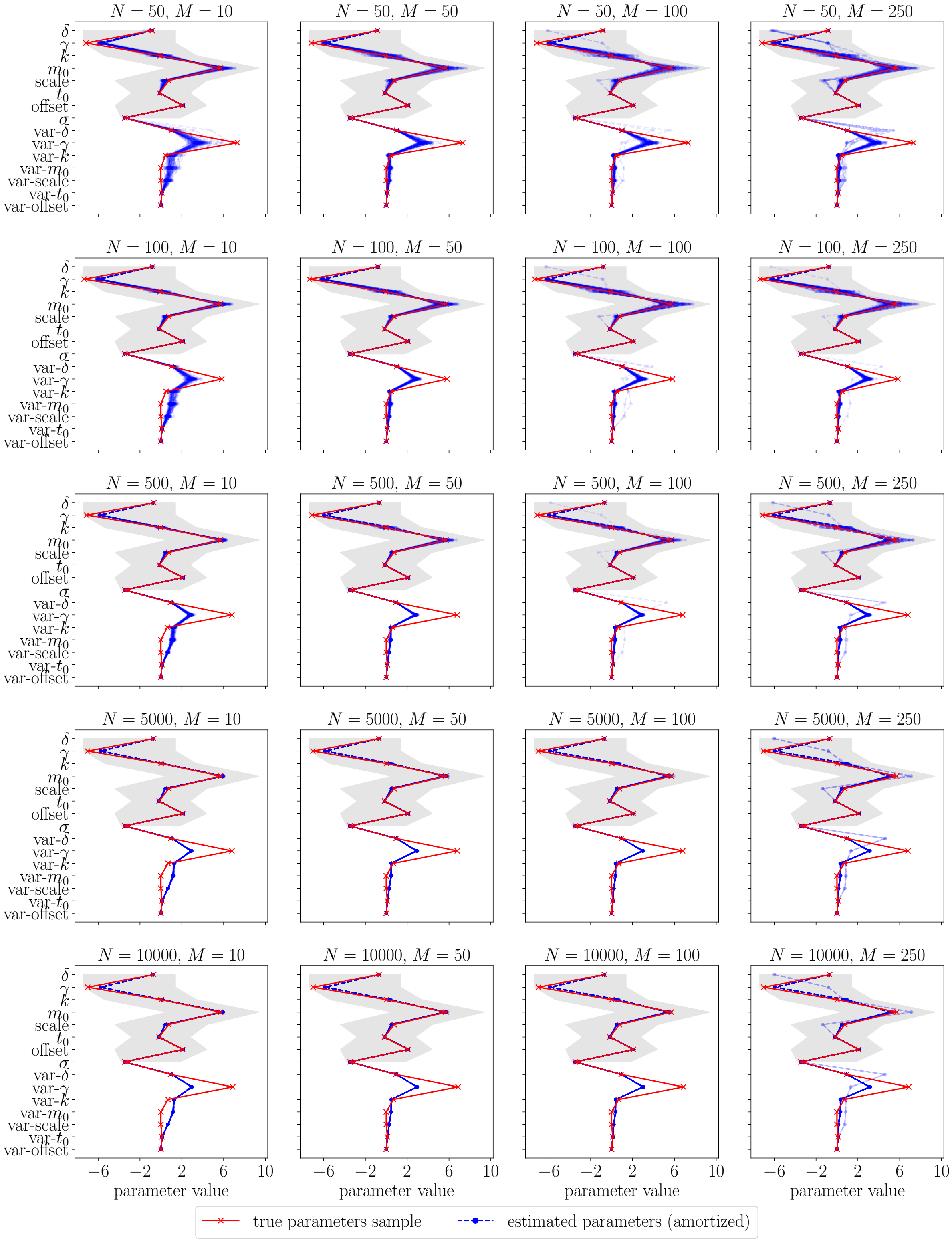
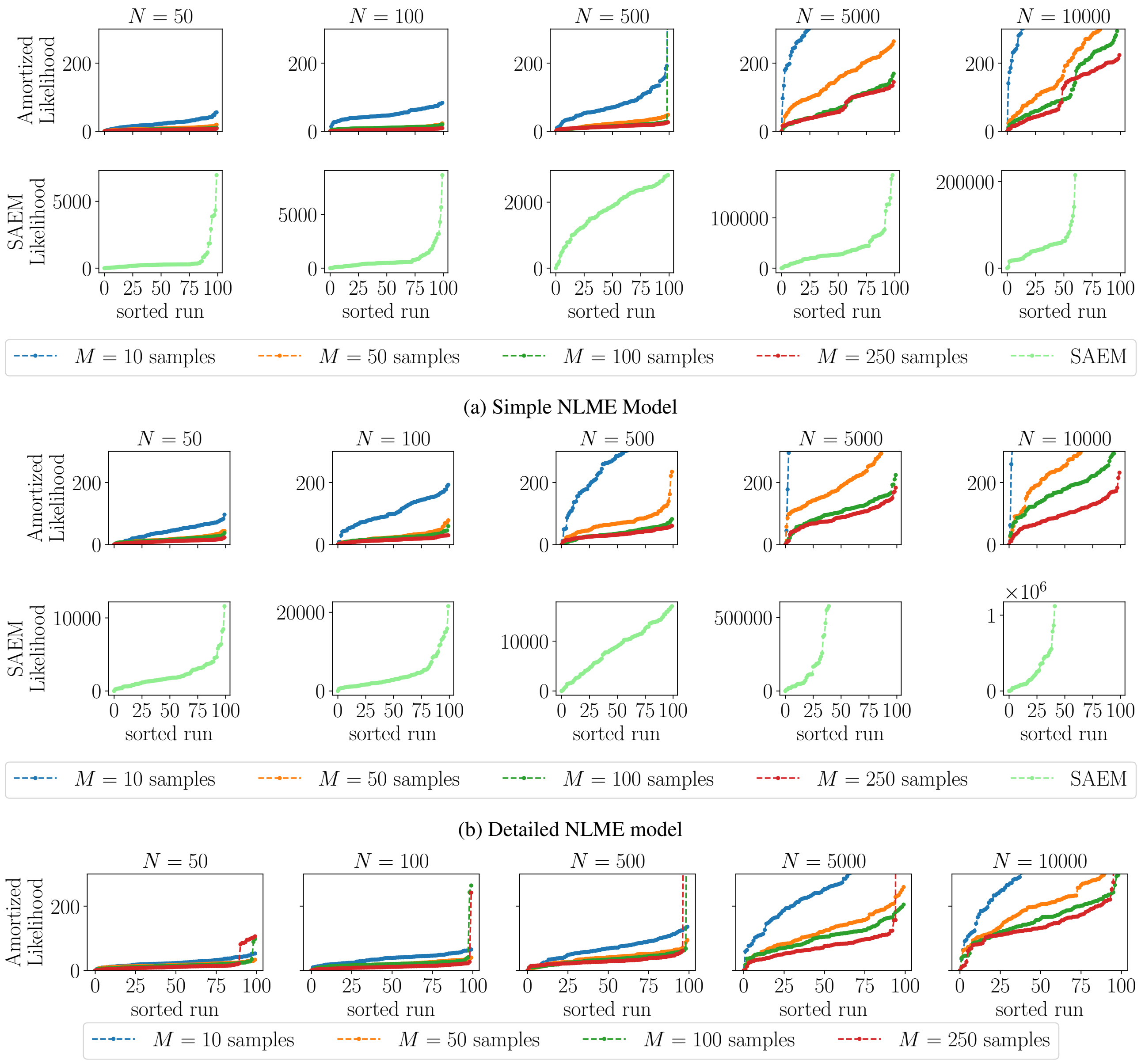


Figure S13. Parameter estimates on synthetic data sets for the SDE single-cell NLME model for 100 runs. Different numbers of posterior samples M were used to estimate population parameters. For each data set of size N, we reused the same neural posterior estimator.



(a) Simple NLME Model

(b) Detailed NLME model

(c) SDE NLME model

Figure S14. Approximated negative log-likelihood values (with an offset of the minimum value) on synthetic data sets for the single-cell NLME models for 100 runs. Different numbers of posterior samples M were used to estimate population parameters. For each data set of size N, we reused the same neural posterior estimator.

A.4.2. Uncertainty Quantification

A.4.3. Efficient inference of the population parameters enables robust uncertainty analysis

Our approach based on amortized neural posterior estimation allows efficient construction of point estimates. Beyond point estimates, in many applications, it is important to assess the uncertainty of the parameters to determine the identifiability of the parameters, draw reliable conclusions, and perform representative predictions (Raue et al., 2013; Maier et al., 2020). The implementation of SAEM in `Monolix` allows standard errors to be obtained through linearization of the likelihood or by a stochastic approximation of the Fisher information matrix, which yields asymptotically correct results under the assumption of normally distributed errors and a large amount of data. Using these standard errors, the confidence intervals are calculated using the Wald statistic (Lixoft SAS, 2023).

However, to ensure the validity of the confidence intervals, it is often advisable to use bootstrapping or non-local approaches such as profile likelihoods, as these are more accurate when the above assumptions are not met. This can, for example, allow for non-symmetric confidence intervals (Fröhlich et al., 2014). Such tests are infeasible when the computational time is high, as it is often the case with SAEM.

We explored the possibility of performing accurate uncertainty quantification, given the computational efficiency of the inference phase in our approach. Specifically, we applied profile likelihood analysis (further details can be found in Supplement A.4.4), as it is a widely used non-local frequentist approach to uncertainty quantification in systems biology (Kreutz et al., 2013). The computation of profile likelihoods took seconds, whereas SAEM took on the order of minutes. On the synthetic data, the confidence intervals based on profile likelihoods were comparable to those based on linearization using SAEM for most parameters. Yet, for three variance parameters, the 80% CIs computed with SAEM actually did not cover the true parameter (for all 100 runs of the multi-start), while the CIs computed with profiles from the amortized approach did (Figure 3D).

In conclusion, our amortized approach allows for an efficient and robust uncertainty quantification by computing profile likelihoods. The cheap amortized inference phase is a key advantage, as other frequentist methods do not allow for robust uncertainty analysis due to substantially higher computational costs. Moreover, the efficient evaluation of the population likelihood allows us to perform a full Bayesian analysis as well as we demonstrated in Section 3.4.

A.4.4. Profile likelihoods

We show that we can use our approximated population likelihood (5) for uncertainty quantification using the profile likelihood method. To compute confidence intervals from profiles we need to compute the profile likelihood ratio

$$R_i(c) = \exp\left(\min_{\boldsymbol{\theta}_{j\neq i}} \log p(\mathcal{D} \mid \boldsymbol{\theta}) - \log p(\mathcal{D} \mid \hat{\boldsymbol{\theta}})\right) \qquad \text{s.t. } \boldsymbol{\theta}_i = c$$

as discussed in (Fröhlich et al., 2014). In our case, we need to compute

$$\begin{aligned} R_i(c) = \exp\Bigg(&\min_{\boldsymbol{\theta}_{j\neq i}} \sum_{i=1}^{N} \left(\log p(\tilde{\boldsymbol{y}}^{(i)}) + \log \mathbb{E}_{\boldsymbol{\phi}\sim p(\boldsymbol{\phi}\mid\tilde{\boldsymbol{y}}^{(i)})}\left[\frac{p_{\text{pop}}(\boldsymbol{\phi}\mid\boldsymbol{\theta})}{p(\boldsymbol{\phi})}\right]\right) \\ &- \sum_{i=1}^{N} \left(\log p(\tilde{\boldsymbol{y}}^{(i)}) + \log \mathbb{E}_{\boldsymbol{\phi}\sim p(\boldsymbol{\phi}\mid\tilde{\boldsymbol{y}}^{(i)})}\left[\frac{p_{\text{pop}}(\boldsymbol{\phi}\mid\hat{\boldsymbol{\theta}})}{p(\boldsymbol{\phi})}\right]\right)\Bigg) \\ = \exp\Bigg(&\min_{\boldsymbol{\theta}_{j\neq i}} \sum_{i=1}^{N} \left(\log \mathbb{E}_{\boldsymbol{\phi}\sim p(\boldsymbol{\phi}\mid\tilde{\boldsymbol{y}}^{(i)})}\left[\frac{p_{\text{pop}}(\boldsymbol{\phi}\mid\boldsymbol{\theta})}{p(\boldsymbol{\phi})}\right] - \log \mathbb{E}_{\boldsymbol{\phi}\sim p(\boldsymbol{\phi}\mid\tilde{\boldsymbol{y}}^{(i)})}\left[\frac{p_{\text{pop}}(\boldsymbol{\phi}\mid\hat{\boldsymbol{\theta}})}{p(\boldsymbol{\phi})}\right]\right)\Bigg). \end{aligned}$$

Therefore, we can use the approximation of the population likelihood even though we do not know $p(\tilde{\boldsymbol{y}})$. Since the evaluation of this approximation is fast, we can efficiently compute profiles and confidence intervals (see Figure S15). We compute confidence intervals using the implementation in `pyPESTO` (Schälte et al., 2023).

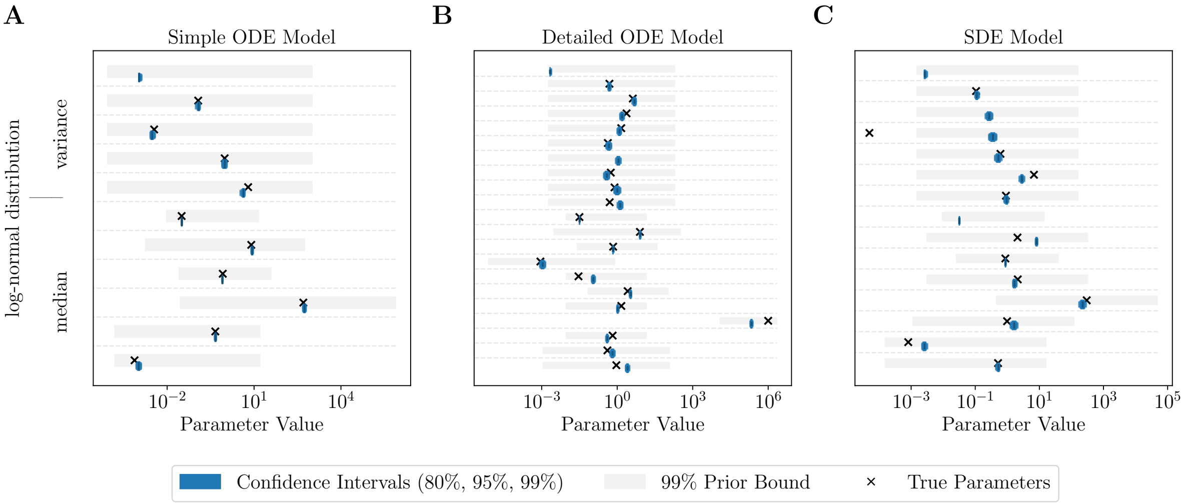


Figure S15. Confidence intervals for the single-cell models on synthetic data. Data was generated by (**A**) the simple ODE model, (**B**) the detailed ODE model and, (**C**) the SDE model. The parameters (median and variance of the log-normal distribution) and CIs (based on profile likelihoods) were then estimated using the amortized approach to NLME models. True parameters, which are 0, are not shown.

A.4.5. ODE vs. SDE NLME model

The simple ODE model of the mRNA transfection processes possessed structural non-identifiabilities, meaning that not all the parameters can be determined from the data. Consequently, the ODE model encompasses only the product $k \cdot m_0 \cdot scale$, while the SDE model encompasses the individual parameters k, m_0 and $scale$, offering a more detailed representation. Indeed, using our amortizing NLME framework, we were able to identify all parameters of the stochastic NLME model (see Figure S16B).

Further analysis on synthetic data generated by the SDE NLME model showed that the simple ODE NLME model estimated parameters such that the variance of the population was 3 times larger than the true variance, while for the stochastic NLME model the variance is only 1.3 times larger and hence capable of capturing the data more accurately (Figure S17). This, in particular, underlines that a deterministic model can give erroneous results if it inadequately captures the underlying processes.

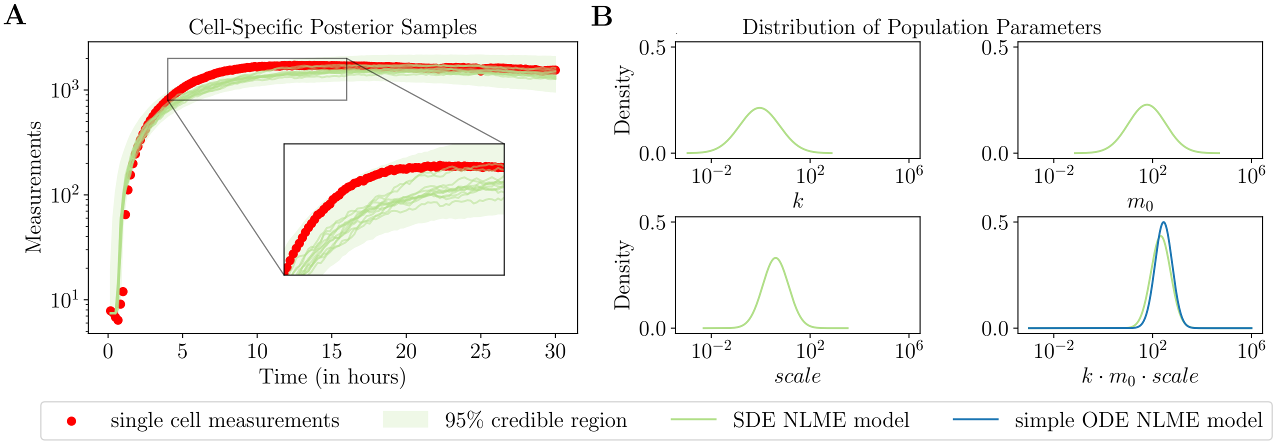


Figure S16. Stochastic NLME model improves identifiability compared to deterministic counterpart. (**A**) Credible regions of a trajectory of the SDE single-cell model estimated by the neural posterior estimator for a real cell. The estimated median of the posterior was simulated 10 times. (**B**) Estimated population distributions for the parameters k, m_0 and $scale$ for the SDE NLME model and their product in the simple ODE NLME model.

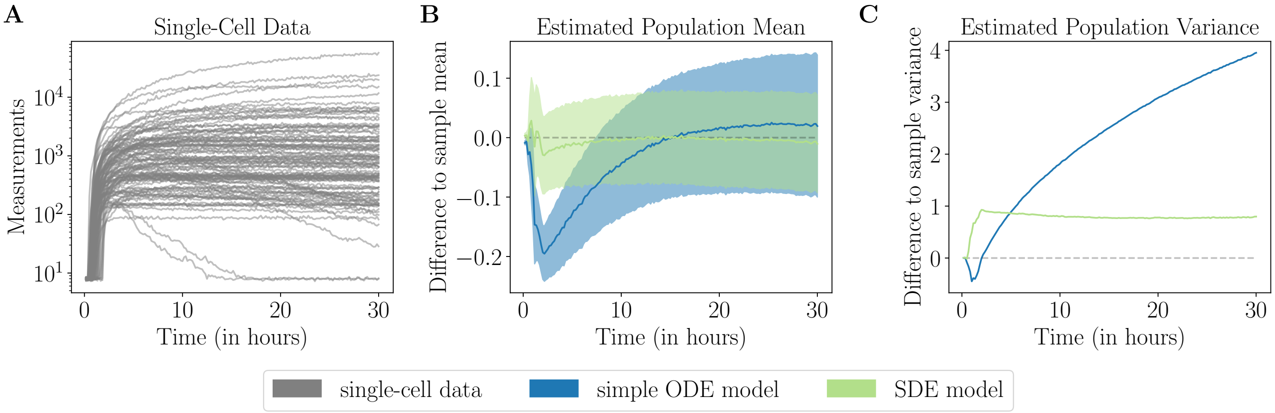


Figure S17. Fit for SDE NLME model on synthetic data. (**A**) Synthetic data describing single-cell translation kinetics after mRNA transfection generated by the SDE NLME model. (**B–C**) Difference of estimated population mean (**B**) and variance (**C**) over time of the SDE and ODE NLME model on synthetic data generated by the SDE model.

A.5. Specification of the pharmacokinetic model

Over the past two decades, many oral targeted therapies have been developed in the field of oncology, many of which target the angiogenesis of neoplasms, which plays an important role in tumor growth. However, angiogenesis inhibitors generally show high variability between patients, leading to significant differences in exposure (Groenland et al., 2019). Therefore, pharmacokinetic (PK) modeling is required to develop targeted dosing strategies for sub-populations or even in a personalized manner, discover concentration thresholds for toxicity, investigate potential interactions, and guide study planning, among other purposes. Sunitinib, an angiogenesis inhibitor, which belongs to the class of tyrosine kinase inhibitors, was the subject of the population pharmacokinetic model, which is described in more detail below. In the model developed by Diekstra et al., the distribution of sunitinib is described by a single compartment model, while for its metabolite SU12662, a two compartment model was used. Presystemic metabolization in (Diekstra et al., 2017) was described according to the model by Yu et al. by a hypothetical enzyme compartment. The hypothetical compartment was parameterized as follows, with Q_H being the calculated concentration:

$$CLIV = \frac{k_a \cdot A_D + Q_H \cdot \frac{A_{c,\text{sunitinib}}}{V_{c,\text{sunitinib}}}}{Q_H + CL_{\text{sunitinib}}}.$$

k_a denotes for the absorption rate constant, while A_D and $A_{c,\text{sunitinib}}$ represent the amounts in the dosing or central compartment, respectively. $CL_{\text{sunitinib}}$ and $V_{c,\text{sunitinib}}$ denote the clearance and volume of distribution of the central compartment of sunitinib in this equation.

The model includes the sex and weight of the patients as covariates. Each patient i received a personal medication (DOS_i) and was measured over a different period of time and at varying time points. In the following, we present the model for each individual; therefore, the index i is removed. The patient's weight is normalized as follows

$$wt := \begin{cases} 83 & \text{if weight is missing and sex} = 1 \\ 75 & \text{if weight is missing and sex} = 0 \\ \text{weight} & \text{else} \end{cases}, \qquad ASCL := \left(\frac{wt}{70}\right)^{0.75}, \qquad ASV := \frac{wt}{70}.$$

The parameters we want to estimate are $\theta \in \mathbb{R}^{13}_{\geq 0}$, and $\eta \in \mathbb{R}^{4}_{\geq 0}$, which are incorporated in the ODE model as follows:

$$\begin{aligned} k_a &= \theta_1 & Q_{34} &= \theta_7 \cdot ASCL \\ V_2 = V_{c,\text{sunitinib}} &= \theta_2 \cdot ASV \cdot \eta_1 & V_4 = V_{p,\text{SU12662}} &= \theta_8 \cdot ASV \\ Q_H &= \theta_3 \cdot ASCL & f_m &= \theta_9 \cdot \eta_4 \\ CL_{\text{sunitinib}} &= \theta_4 \cdot ASCL \cdot \eta_3 & Q_{25} &= \theta_{10} \cdot ASCL \\ CL_{\text{SU12662}} &= \theta_5 \cdot ASCL & V_5 = V_{p,\text{sunitinib}} &= \theta_{11} \cdot ASV \\ V_3 = V_{c,\text{SU12662}} &= \theta_6 \cdot ASV \cdot \eta_2 \end{aligned}$$

and

$$\begin{aligned} \frac{dA_D}{dt} &= \frac{dA_1}{dt} = -k_a A_1 & A_1(0) &= 0 \\ \frac{dA_{c,sunitinib}}{dt} &= \frac{dA_2}{dt} = Q_H \cdot CLIV - \frac{Q_H}{V_2} A_2 - \frac{Q_{25}}{V_2} A_2 + \frac{Q_{25}}{V_5} A_5 & A_2(0) &= 0 \\ \frac{dA_{c,\text{SU12662}}}{dt} &= \frac{dA_3}{dt} = f_m \cdot CL_{\text{sunitinib}} \cdot CLIV - \frac{CLM}{V_3} A_3 - \frac{Q_{34}}{V_3} A_3 + \frac{Q_{34}}{V_4} A_4 & A_3(0) &= 0 \\ \frac{dA_{p,\text{SU12662}}}{dt} &= \frac{dA_4}{dt} = \frac{Q_{34}}{V_3} A_3 - \frac{Q_{34}}{V_4} A_4 & A_4(0) &= 0 \\ \frac{dA_{p,\text{sunitinib}}}{dt} &= \frac{dA_5}{dt} = \frac{Q_{25}}{V_2} A_2 - \frac{Q_{25}}{V_5} A_5 & A_5(0) &= 0. \end{aligned}$$

As in the baseline (Diekstra et al., 2017), we fix $\theta_3 = 80$, $\theta_9 = 0.21$, and $\theta_{11} = 588$ to get comparable results.

Furthermore, whenever a patient takes medication (at t_j^{DOS}), we have

$$A_1(t_j^{DOS}) = \lim_{t \to t_j^{DOS}} A_1(t) + DOS\,.$$

In the noise model we apply a censoring from below by

$$y_1 = \theta_{12} \cdot \epsilon_1 + \begin{cases} \log(0.001) & \text{if } A_2 < 0.001 \\ \log(A_2) & \text{else,} \end{cases}$$

$$y_2 = \theta_{13} \cdot \epsilon_2 + \begin{cases} \log(0.001) & \text{if } A_3 < 0.001 \\ \log(A_3) & \text{else,} \end{cases}$$

where $\epsilon_1, \epsilon_2 \sim \mathcal{N}(0, \sigma^2)$. In (Diekstra et al., 2017), $\sigma^2 = 1$ was fixed, therefore we fix it as well. This ODE system is simulated using the Rodas5P solver implemented in the Julia package `DifferentialEquations.jl` (Rackauckas & Nie, 2017).

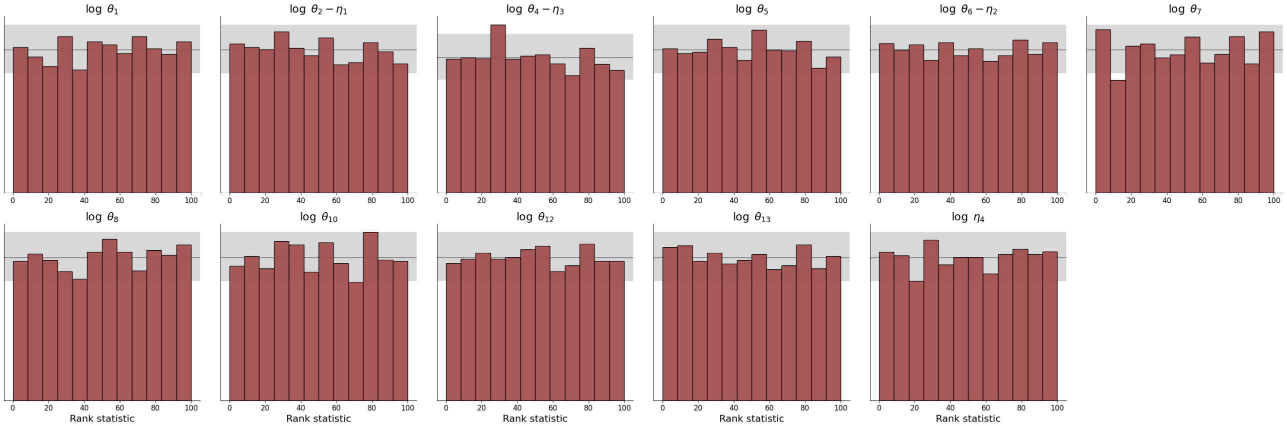


Figure S18. Simulation-based calibration plots of the individual posteriors for the pharmacokinetic model. Incorrect calibration can be seen by deviations from uniformity (bars outside the gray area)

A.5.1. Dosing events

In our amortizing framework, covariates such as sex and weight can be treated as part of the population model. If they are instead part of the model $\mathcal{M}$, then they need to be synthetically generated during the simulation phase. This is the case with dosing regimes, which refer to the prescribed schedules and dosages of the medications that are administered to patients. Therefore, we encoded the dosing events as part of the observations, which are given to the summary network together with the simulated measurements. Hence, the observation at each time point j consists of a vector $(y_2^{(j)}, y_3^{(j)}, DOS_j, t_j, DOS\text{-}Indicator_j)$, where $DOS\text{-}Indicator_j$ is a binary indicator of a dosing event following the ideas on encoding missing data and time points in (Wang et al., 2023). If a dosing event occurs, the variables are $y_2^{(j)}$ and $y_3^{(j)}$ are set to 0, otherwise DOS_j is set to 0. We trained two LSTMs using the split summary network architecture provided in `BayesFlow` (Radev et al., 2023), where each summary network got the input depending on the binary variable. During the simulation phase, we sampled the dosing events and observed time points from the time points and events in the data set because we were only interested in this particular data set. However, one could also generate events and observation time points from a reasonable distribution to be able to amortize over multiple different data sets.

A.5.2. Comparison between different estimation methods

We estimated the population parameters using FOCEI, SAEM and our amortized approach. We report the results in Figure S19. Furthermore, we show the measurement trajectories for three different patients to show the underestimation of the population variance of FOCEI (see Figure S20).

Our amortized approach, including all phases, repeating phase (III) 200 times and generating $100,000$ samples from the full population posteriorly, was completed in 27 hours. For an optimization run, we need on average 0.34 minutes. Sampling takes 5.1 minutes.

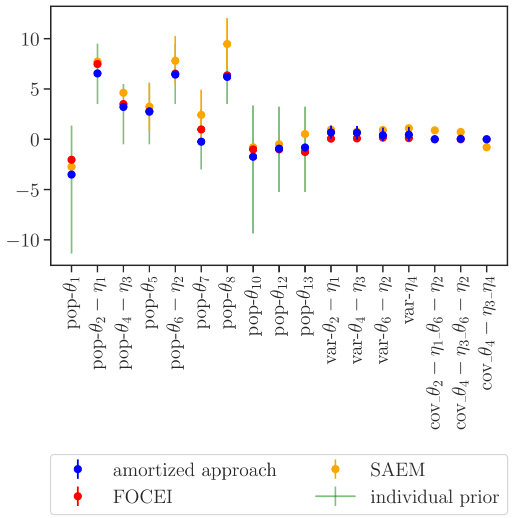


Figure S19. Population parameter estimates for the pharmacokinetic model.

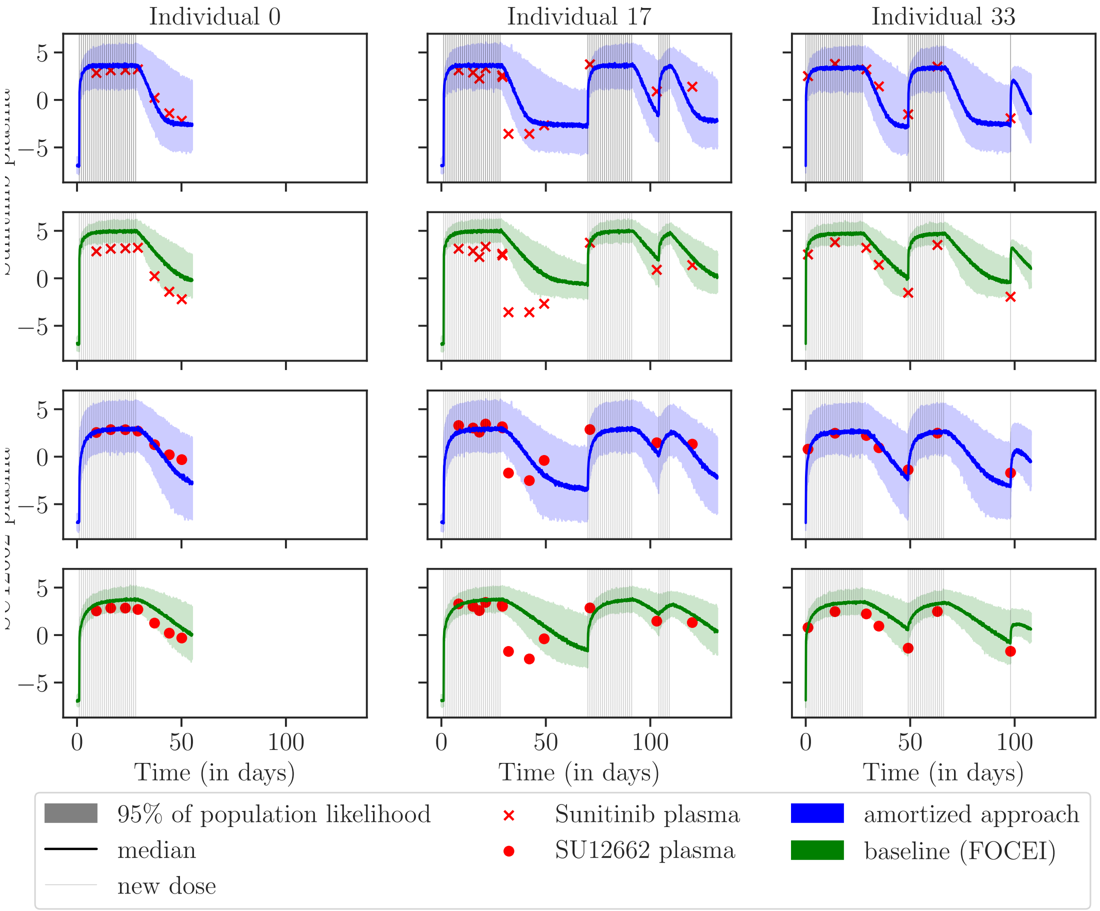


Figure S20. Baseline underestimates variance of measurements. Trajectories of sunitinib plasma and SU12662 plasma measurements for three patients. Simulating samples from the population likelihood convoluted with the noise model using the covariates of this patient based on the estimated parameters of FOCEI and our amortized approach.