

Inferring Optical Tissue Properties from Photoplethysmography using Hybrid Amortized Inference

Jens Behrmann^{*†§}, Maria R. Cervera^{*†§}, Antoine Wehenkel^{*†§}, Andrew C. Miller[†], Albert Cerussi[†], Pranay Jain[†], Vivek Venugopal[†], Shijie Yan[†], Guillermo Sapiro[†], Luca Pegolotti[‡], Jörn-Henrik Jacobsen[‡]

Abstract

Smart wearables enable continuous tracking of established biomarkers such as heart rate, heart rate variability, and blood oxygen saturation via photoplethysmography (PPG). Beyond these metrics, PPG waveforms contain richer physiological information, as recent deep learning (DL) studies demonstrate. However, DL models often rely on features with unclear physiological meaning, creating a tension between predictive power, clinical interpretability, and sensor design. We address this gap by introducing PPGen, a biophysical model that relates PPG signals to interpretable physiological and optical parameters. Building on PPGen, we propose hybrid amortized inference (HAI), enabling fast, robust, and scalable estimation of relevant physiological parameters from PPG signals while correcting for model misspecification. In extensive in-silico experiments, we show that HAI can accurately infer physiological parameters under diverse noise and sensor conditions. Our results illustrate a path toward PPG models that retain the fidelity needed for DL-based features while supporting clinical interpretation and informed hardware design.

Data and Code Availability We use in-silico datasets generated by the proposed biophysical model (PPGen). Generating this data relies on an internally developed light transport software, which prevents us from sharing the data. Due to our inability to share the in-silico dataset, we do not make our code available.

1. Introduction

Wearable photoplethysmogram (PPG) sensors—small devices that emit light into tissue and measure the transmitted or reflected light—have become central to personal health and fitness monitoring (Charlton et al., 2023; Kyriacou and Allen, 2021). From the outset, they have enabled the extraction of clinically meaningful biomarkers such as blood oxygen saturation, heart rate, and heart rate variability, all of which rest on well-understood biophysical principles.

More recently, deep learning (DL) has reshaped how PPG signals are analyzed. Beyond traditional biomarkers, DL models have shown that PPG waveform dynamics contain information useful for detecting a wide range of medical conditions (Abbaspourazad et al., 2024). While these advances highlight the rich information encoded in PPG signals, DL models often exploit features that lack clear physiological interpretation. This lack of interpretability not only complicates clinical adoption but also hinders the design of next-generation PPG sensors, which must guarantee that such features are captured with sufficient fidelity to be reliably useful.

To align our theoretical understanding of PPG with the potential of DL-based predictive models, we propose a hybrid modeling approach that combines first-principles PPG models with deep generative models. This approach is designed to infer underlying physiological parameters while remaining robust to model misspecification, thereby overcoming the limitations of first-principles models without sacrificing predictive power or interpretability.

Our contributions are threefold. **1.** We introduce PPGen, a novel PPG Pulse Generator that explicitly relates waveform characteristics to both physiological parameters and sensor architecture, using physically grounded light-transport simulations. **2.** We develop hybrid amortized inference (HAI), a learn-

^{*} Equal contribution, ordered alphabetically.

[†] Apple.

[‡] Work done while at Apple.

[§] {j.behrmann, m.cervera, a.wehenkel}@apple.com

ing algorithm that enables fast, robust, and scalable parameter inference while mitigating the impact of model misspecification. **3.** We systematically evaluate the model and inference framework in-silico across diverse configurations, demonstrating their ability to accurately estimate relevant physiological parameters from PPG measurements.

2. Methods

We restrict temporal signals to single PPG pulses, i.e. characteristic waveforms representing the cyclical change in blood volume in a peripheral artery with each heartbeat. We denote an *observed* real-world PPG pulse as $\mathbf{x}_o \in \mathbb{R}^{R \times N \times T}$, where R, N, T respectively denote the number of light receivers, light sources, and timesteps in the considered pulse.

From a given PPG pulse \mathbf{x}_o , our goal is to infer a set of underlying biophysical parameters of interest $\theta \in \mathbb{R}^{9+2T}$ (see appendix [Table 3](#) for details), including 9 static parameters (e.g. melanin concentration or arterial oxygenation) and two dynamic parameters that represent blood volume changes over the course of a PPG pulse. Formally, our objective is to learn an estimator that describes the posterior distribution of parameters θ given any PPG signal \mathbf{x}_o , i.e., $p(\theta | \mathbf{x}_o)$.

Since in-vivo measurement of biophysical tissue properties θ is invasive, we cannot rely on labeled data to solve this task. To mitigate this problem, we distill expert knowledge into a model $\pi(\theta)p(\mathbf{x}_s | \theta)$ that describes a joint generative process of biophysical parameters θ and *synthetic* PPG pulses $\mathbf{x}_s \in \mathbb{R}^{R \times N \times T}$. This model provides a good approximation of the *true* relationship $p(\mathbf{x}_o | \theta)$. However, some of its simplifying assumptions, e.g., ignoring movement artifacts and assuming a fixed geometry of skin layers, reduce its ability to accurately describe real-world PPG pulses \mathbf{x}_o .

In summary, our goal of estimating $p(\theta | \mathbf{x}_o)$ is challenged by **1.** Data scarcity: we only have unlabeled real-world data $\mathcal{D} := \{\mathbf{x}_o\}_{i=1}^N$; and **2.** Model misspecification: because the underlying mechanisms affecting PPG measurements are numerous and complex, first-principle models are imperfect. In [subsection 2.1](#), we introduce PPG Pulse Generator (PPGen), a differentiable biophysical model $p(\mathbf{x}_s | \theta)\pi(\theta)$ that helps us address data scarcity.

With PPGen defined, the main methodological obstacle remaining for estimating $p(\theta | \mathbf{x}_o)$ is the misspecification gap between \mathbf{x}_s and \mathbf{x}_o . To address it, we introduce in [subsection 2.2](#) hybrid amortized in-

ference (HAI), our solution that learns jointly from unlabeled real-world data \mathcal{D} and from labeled synthetic measurements $(\theta, \mathbf{x}_s) \sim p(\mathbf{x}_s | \theta)\pi(\theta)$. In [subsection 2.3](#) we discuss the main modeling choices we followed to conduct our experiments.

2.1. PPG Pulse Generator (PPGen)

We model skin as a multi-layered material composed of $l = 3$ layers. We can then parameterize the optical skin properties with some biophysical parameters $\theta \in \Theta$ that describe the main biophysical factors impacting the propagation of light in skin. In particular, these parameters can be decomposed into $\theta = [\theta_s, \theta_d]$. The static parameters, denoted $\theta_s \in \mathbb{R}^9$, are fixed over the measurement period considered, whereas the dynamic parameters $\theta_d \in \mathbb{R}^{2T}$ drive the time-dependence of the PPG signal. The likelihood can thus be decomposed into $p(\mathbf{x}_s | \theta) = \prod_{t=1}^T p(\mathbf{x}_s^t | \theta_s, \theta_d^t)$, where $\theta_d^t \in \mathbb{R}^2$ are the two dynamic parameters at timestep t and $\mathbf{x}_s^t \in \mathbb{R}^{R \times N}$. For readability, we denote $\theta^t \in \mathbb{R}^{11}$ the vector composed of θ_s and θ_d^t .

We decompose the time-independent joint distribution $p(\theta^t, \mathbf{x}_s^t)$ into four main components: **1.** an informed prior $\pi(\theta)$ over the biophysical parameters; **2.** a function f_b that maps the instantaneous parameters θ^t to two vectors $\mu_a \in \mathbb{R}^l$ and $\mu_s \in \mathbb{R}^l$ describing light absorption and scattering in the l considered skin layers; **3.** a neural network based surrogate of a light propagation simulator in layered materials, denoted \hat{f}_{LT} , which maps optical properties (μ_a, μ_s) to the instantaneous PPG the sensor would measure were it noiseless; and **4.** a noise model of the sensor.

$\pi(\theta)$ – Informed prior over biophysical parameters. For the set of static parameters θ_s , we sample from a uniform distribution over a range of plausible values obtained from the literature. For the two dynamic parameters θ_d consisting of blood volume changes over time, we leverage mechanistic models of the cardiovascular system ([Charlton et al., 2019](#); [Melis, 2018](#)) and sample over plausible ranges of cardiac and vascular properties to obtain a variety of blood volume waveforms. For additional details on $\theta \sim \pi(\theta)$ please refer to [subsection B.1](#).

f_b – Modeling the relationship between biophysical parameters and optical skin properties. The function $f_b(\theta^t, \lambda) : \Theta^t \times \Lambda \rightarrow \mathbb{R}^l \times \mathbb{R}^l$ relates biophysical parameters θ to the tissue's optical properties, $\mu_a \in \mathbb{R}^l$ and $\mu_s \in \mathbb{R}^l$, for a given wavelength $\lambda \in \Lambda$, where the $l = 3$ skin layers cor-

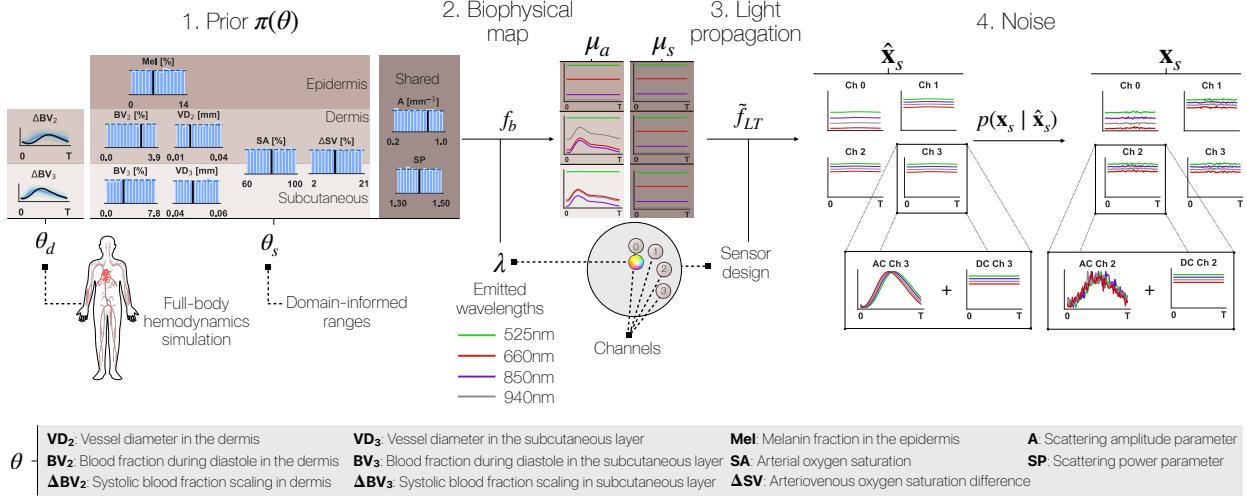


Figure 1: Overview of PPGGen, our biophysical forward model for generating synthetic PPG signals. **1.** The process starts by sampling a set of biophysical parameters θ from a prior distribution $\pi(\theta)$. This prior combines dynamic parameters (θ_d), such as blood volume waveforms from a hemodynamics simulator, with static parameters (θ_s), such as melanin fraction and vessel diameters, sampled from literature-informed ranges. **2.** These parameters, along with a specific light wavelength (λ), are given as inputs to a biophysical mapping function, f_b , which calculates the optical absorption (μ_a) and scattering (μ_s) coefficients for each skin layer. **3.** Subsequently, a light transport model, \hat{f}_{LT} (a neural network surrogate for Monte Carlo simulations), uses these optical coefficients and a given sensor architecture to predict the clean, noiseless PPG signal, $\hat{\mathbf{x}}_s$. **4.** In the final step, a realistic noise model, $p(\mathbf{x}_s | \hat{\mathbf{x}}_s)$, adds shot and electronic noise to produce the final synthetic raw sensor reading, \mathbf{x}_s . [Algorithm 2](#) in [section B](#) describes sampling and density evaluation. Appendix [section B](#) further details each building block of PPGGen.

respond to the epidermis, dermis, and subcutaneous. The absorption coefficient μ_a is a function of the concentration of absorbing components in skin, such as hemoglobin or melanin (the complete list is described in [Figure 1](#)). By contrast, μ_s describes the trajectories that light rays follow in skin, and is described in our model by the scattering parameters A and SP.

Crucially, because both of these properties depend on wavelength λ , f_b takes λ as an additional input. For additional details on how μ_a and μ_s are computed, please refer to [subsection B.2](#).

\hat{f}_{LT} – Modeling light propagation in multi-layered tissue via an efficient and differentiable surrogate. The propagation of light within layered materials is well understood and can be simulated with GPU-accelerated Monte Carlo modeling of light transport in multi-layered tissues (MCML, [Wang et al., 1995](#); [Alerstam et al., 2008b](#)). Simulations depend on the layer-dependent optical tissue properties $\mu_a \in \mathbb{R}^l$ and $\mu_s \in \mathbb{R}^l$, the sensor architecture (e.g., the emitter and detector locations) and the layers’ geometry (e.g., layer thicknesses). The two latter are considered fixed in our experiments, unless

stated otherwise. For a given pair of optical parameters (μ_a, μ_s) , the output of the MCML simulation $f_{LT}(\mu_a, \mu_s) \in \mathbb{R}^R$ is an estimator of detected versus emitted light at each of the R light detectors. In our experiments, we use a light transport tool that was extensively validated against MCML and consider sensors with four receiving photo-diodes and one shared emitter location, hence $R = 4$.

While faithful to the physics, light transport simulators are computationally intensive and pose a major bottleneck in the generative process. We address this limitation by replacing f_{LT} with a fast neural network, \hat{f}_{LT} , that is trained via supervised learning on a large-scale dataset of optical-property-to-measurement pairs, $\left\{ \left(\mu_a^{(i)}, \mu_s^{(i)} \right), \mathbf{x}_s^{(i)} \right\}_{i=1}^{N_{sim}}$, obtained by running the light transport simulator on a wide range of inputs (for details refer to [subsection B.3](#)). The resulting light transport simulator surrogate $\hat{f}_{LT} \approx f_{LT}$ can generate instantaneous and noiseless PPG signals $\hat{\mathbf{x}}_s^t \in \mathbb{R}^{R \times N}$ via $\hat{\mathbf{x}}_s^t = \hat{f}_{LT}(\mu_a, \mu_s)$. Because this surrogate has the added benefit of being differentiable, combining it with the analytical biophysical map f_b re-

sults in a fully differentiable forward model $f_t(\theta^t) := \left[\hat{f}_{\text{LT}} \circ f_b(\theta^t, \lambda_1), \dots, \hat{f}_{\text{LT}} \circ f_b(\theta^t, \lambda_N) \right]$, which concatenates along the wavelength dimension. To describe a complete pulse, we concatenate outputs of f_t along time t such that $f : \Theta \mapsto \mathbb{R}^{R \times N \times T}$.

Noise sensor model. We further improve the faithfulness of the model by adding measurement noise in the form of two additive, time-independent noise sources: *shot noise*—a zero-mean Gaussian noise with signal-dependent variance—, and *white noise*. While shot noise arises from the particle nature of light and models the random arrival of photons at the receiver, white noise captures sensor-internal effects, such as thermal and chip noise. We denote this combined noise model as $p(\mathbf{x}_s | \hat{\mathbf{x}}_s)$ and provide details in [subsection B.4](#).

[Figure 1](#) depicts the whole PPGen, which provides a differentiable way of generating PPG pulses and evaluating the corresponding likelihood.

2.2. Hybrid Amortized Inference (HAI)

This section introduces Hybrid Amortized Inference (HAI), our algorithm to obtain an amortized estimator of the posterior $p(\theta | \mathbf{x}_o)$, given the prescribed model $p(\theta, \mathbf{x}_s)$ and unsupervised data $\mathcal{D} = \{\mathbf{x}_o^i\}_{i=1}^M$.

Misspecification model. Because labels θ are not available for real measurements \mathbf{x}_o , we aim to learn a projection from \mathbf{x}_o into a physical signal \mathbf{x}_s , from which parameters θ can be inferred. This projection, a misspecification model that corrects the gap between real and synthetic data, therefore needs to be independent of the real parameter θ . This results in the independence assumption

$$\mathbf{x}_o \perp \theta | \mathbf{x}_s, \quad (1)$$

which essentially states that a real-world measurement \mathbf{x}_o contains no more information about θ than the corresponding synthetic \mathbf{x}_s . As leveraged in previous work ([Wehenkel et al., 2025](#)), this assumption has the attractive property of resulting in the following decomposition:

$$p(\theta | \mathbf{x}_o) = \int p(\theta | \mathbf{x}_s) p(\mathbf{x}_s | \mathbf{x}_o) d\mathbf{x}_s. \quad (2)$$

Our original objective of estimating $p(\theta | \mathbf{x}_o)$ can thus be disentangled into learning independent estimators for $p(\theta | \mathbf{x}_s)$ and $p(\mathbf{x}_s | \mathbf{x}_o)$.

Algorithm 1 Training and inference with Hybrid Amortized Inference (HAI).

```

PretrainNPE( $\pi(\theta)$ ,  $p(\mathbf{x}_s | \theta)$ )
repeat
  Sample  $\theta \sim \pi(\theta)$ ,  $\mathbf{x}_s \sim p(\mathbf{x}_s | \theta)$ 
   $\mathcal{L}_s(\phi) \leftarrow -\log q_\phi(\theta | \mathbf{x}_s)$ 
   $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_s(\phi)$ 
until convergence
return  $q_{\phi^*}(\theta | \mathbf{x}_s)$ 

LearnMisspec( $q_{\phi^*}$ ,  $\mathcal{D} = \{\mathbf{x}_o\}$ )
repeat
  foreach  $\mathbf{x}_o \in \text{minibatch } \mathcal{D}$  do
     $\mathbf{x}_s \sim q_\psi(\mathbf{x}_s | \mathbf{x}_o)$ 
     $\theta \sim q_{\phi^*}(\theta | \mathbf{x}_s)$ 
     $J(\psi, \omega) \leftarrow \log q_\omega(\mathbf{x}_o | \mathbf{x}_s) + \log p(\mathbf{x}_s | \theta)$ 
     $(\psi, \omega) \leftarrow (\psi, \omega) + \eta \nabla_{(\psi, \omega)} J(\psi, \omega)$ 
  end
until convergence
return  $q_{\psi^*}(\mathbf{x}_s | \mathbf{x}_o)$ 

Infer( $\mathbf{x}_o, N_{samples}$ )
for  $i = 1, \dots, N_{samples}$  do
   $\mathbf{x}_s^{(i)} \sim q_{\psi^*}(\mathbf{x}_s | \mathbf{x}_o)$ 
   $\theta^{(i)} \sim q_{\phi^*}(\theta | \mathbf{x}_s^{(i)})$ 
end
return  $\{\theta^{(i)}\}_{i=1}^{N_{samples}}$ 

```

Neural posterior estimation of $p(\theta | \mathbf{x}_s)$. Following the neural posterior estimation (NPE) algorithm introduced by [Lueckmann et al. \(2017\)](#), we train a neural conditional density estimator on pairs of parameters and simulated observations $(\theta, \mathbf{x}_s) \sim p(\mathbf{x}_s | \theta)\pi(\theta)$ directly. Formally, we consider a family of approximators parameterized by ϕ denoted, $q_\phi(\theta | \mathbf{x}_s)$, and perform stochastic gradient descent on ϕ to minimize

$$\mathcal{L}_s(\phi) := -\frac{1}{L} \sum_{i=1}^L \log q_\phi(\theta^i | \mathbf{x}_s^i) \quad (3)$$

with $(\theta^i, \mathbf{x}_s^i) \sim p(\mathbf{x}_s | \theta)\pi(\theta)$.

Provided sufficient capacity and proper optimization, the model obtained, denoted $q_{\phi^*}(\theta | \mathbf{x}_s)$, can be made arbitrarily close to the “true” posterior $p(\theta | \mathbf{x}_s)$ on synthetic data.

Maximum likelihood estimation of the misspecification model $p(\mathbf{x}_s | \mathbf{x}_o)$. We consider a

family of misspecification approximators $q_\psi(\mathbf{x}_s \mid \mathbf{x}_o)$, parameterized by a vector $\psi \in \mathbb{R}^{|\psi|}$. Then, the likelihood of the parameter ψ given a dataset of observation \mathcal{D} becomes

$$\log p(\mathcal{D} \mid \psi) = \sum_{\mathbf{x}_o \in \mathcal{D}} \log p(\mathbf{x}_o \mid \psi) \quad (4)$$

$$\approx \sum_{\mathbf{x}_o \in \mathcal{D}} \mathbb{E}_{q_\psi(\mathbf{x}_s \mid \mathbf{x}_o)} [q_\omega(\mathbf{x}_o \mid \mathbf{x}_s) p(\mathbf{x}_s \mid \theta)], \quad (5)$$

where $q_\omega(\mathbf{x}_o \mid \mathbf{x}_s)$ approximates $p(\mathbf{x}_o \mid \mathbf{x}_s) = \frac{q_\psi(\mathbf{x}_s \mid \mathbf{x}_o)p(\mathbf{x}_o)}{p(\mathbf{x}_s)}$, the inverse of $q_\psi(\mathbf{x}_s \mid \mathbf{x}_o)$. We can directly convert Equation 5 into a loss function if the approximators $q_\omega(\mathbf{x}_o \mid \mathbf{x}_s)$ and $q_{\theta^*}(\theta \mid \mathbf{x}_s)$ allow differentiable sampling, and if $q_\psi(\mathbf{x}_s \mid \mathbf{x}_o)$ allows differentiable density evaluation.

In summary, HAI combines NPE with maximum likelihood estimation to learn a model of the misspecification, while leveraging the independence assumption in Equation 2. The training and inference algorithms are described in Algorithm 1 (more details in subsection C.1).

2.3. Variational families and implementation details

As highlighted in the previous section, the HAI framework heavily relies on differentiable parameterization of conditional probability distributions. We now provide details on the neural network architectures used to model these various functions.

Parameterization of q_ϕ . In our experiments we use a simple variational family that assumes conditional independence between all elements of θ given the observation \mathbf{x}_s so that we can re-express the posterior as $q_\phi(\theta \mid \mathbf{x}_s) = \prod_{i=1}^{9+2T} q_\phi^i(\theta_i \mid \mathbf{x}_s)$. The parametric density for q_ϕ^i is a fixed-variance Gaussian whose mean is determined by the output of a neural network parameterized by ϕ , $q_\phi^i(\theta_i \mid \mathbf{x}_s) := \mathcal{N}(f_\phi^i(\mathbf{x}_s) - \mu_i, \sigma_i^2)$, where μ_i and σ_i respectively denote the mean and variance of the chosen prior for θ_i . In our experiments, we use the posterior mean as the point estimate for θ . We use a shared U-net architecture to jointly model $f_\phi^i(\mathbf{x}_s)$ for all $i \in [1, 9 + 2T]$. Importantly, we also engineer the input to the U-net and explicitly decompose the signal into a pulsatile (AC) and a quasi-static (DC) baseline components, which have typically very different scales that would be hard for the neural network to pick up. More details about the neural network architecture are provided in subsection C.2. Alternatives to this simple model include

conditional normalizing flows, which would enable to model uncertainty on θ jointly and highlight dependencies among parameters.

Misspecification model $q_\psi(\mathbf{x}_s \mid \mathbf{x}_o)$. To demonstrate the importance of carefully designing the inductive bias of $q_\psi(\mathbf{x}_s \mid \mathbf{x}_o)$, our experiments rely on a simple additive correction model, $q_\psi(\mathbf{x}_s \mid \mathbf{x}_o, \theta) := \delta_{\mathbf{x}_o}(\mathbf{x}_s + \beta(\psi))$, where δ denotes the Dirac delta distribution and $\beta : \mathbb{R}^{R \times N} \rightarrow \mathbb{R}^{R \times N \times T}$ denote an offset shared over time and computed from $\psi \in \mathbb{R}^{R \times N}$. A perfect estimator of the inverse misspecification $p(\mathbf{x}_o \mid \mathbf{x}_s) = \frac{q_\psi(\mathbf{x}_s \mid \mathbf{x}_o)p(\mathbf{x}_o)}{p(\mathbf{x}_s)}$ can be derived analytically as $q_\psi(\mathbf{x}_o \mid \mathbf{x}_s) := \delta_{\mathbf{x}_s}(\mathbf{x}_o - \beta(\psi))$. In practice, one could consider more expressive misspecification models, e.g., parameterized with a neural network. However, this may require careful regularization to ensure the learnt misspecification model does not overrule the prescribed model $p(\mathbf{x}_s \mid \theta)$. Our experiments will further discuss the trade-offs between expressivity and generalization of the learned misspecification model.

3. Related work

Mechanistic modeling of PPG. Similar to our work, both Boonya-Ananta et al. (2020) and Tang et al. (2020) capture the dynamic nature of PPG using light transport simulations. While Boonya-Ananta et al. (2020) introduce a dynamic 3D model of blood vessels, Tang et al. (2020) approximate PPG pulses with waveform templates. In contrast, our model leverages the full-body hemodynamics simulator of Charlton et al. (2019) to describe pulse dynamics. It further incorporates Windkessel models (Frasch et al., 1996) to capture microvascular effects, as in Tanaka (2022) for PPG waveform modeling or Doostdar and Khalilzadeh (2014) for studying aging. Our model combines these hemodynamic waveforms with Monte Carlo simulations of light propagation in multilayered tissue (Jacques and Wang, 1995; Wang et al., 1995). Recent work has proposed important links between optical properties and skin physiology: e.g. Chatterjee et al. (2020) investigated the origin of PPGs across multiple wavelengths; Haque et al. (2022) and Reiser et al. (2022) modeled the impact of blood vessels; and Al-Halawani et al. (2024) analyzed pigmentation via melanin absorption. Building on this, our simulator, PPGen, integrates mechanistic models of skin optics and pulsatile blood flow, enabling it to accurately

generate PPG pulse waveforms while preserving the underlying links to sensor physics and physiology.

Inference of biophysical parameters on PPG.

Existing methods for inferring tissue properties, whether simulation lookup tables (e.g. Zhong et al. (2014) and Das et al. (2021)) or recent machine learning models (see Table 1 in Scarbrough et al. (2024) for an overview), limited their scope to inferring static optical parameters. Contrarily, our work extends the inference to biophysical parameters and dynamic waveforms crucial for PPG analysis. Furthermore, these two approaches have distinct shortcomings: classical methods rely on slow, iterative optimization that scales poorly (Fredriksson et al., 2012), while modern ML methods use data augmentation for robustness (Scarbrough et al., 2024), which fails against errors unseen during training. We address these gaps using hybrid amortized inference (HAI) which allows scalable, dynamic waveform estimation and robust adaptation to misspecification.

SBI/ Hybrid inference under misspecification.

A key challenge in Simulation-Based Inference (Cranmer et al., 2020, SBI), is its sensitivity to model misspecification (Cannon et al., 2022), a limitation that has spurred the development of robust methods (Wehenkel et al., 2025; Ward et al., 2022; Huang et al., 2023; Kelly et al., 2023). Among these, HAI is closely related to RoPE (Wehenkel et al., 2025), as both leverage the independence assumption from Equation 1 to improve robustness. However, unlike RoPE, HAI requires a differentiable forward model to explicitly learn a misspecification model. This explicit learning of misspecification connects HAI to Generalized Bayesian Inference (Bissiri et al., 2016; Jewson et al., 2018; Chérief-Abdellatif and Alquier, 2020; Matsubara et al., 2022; Dellaporta et al., 2022, GBI), which replaces the initial likelihood with a generalized loss-based likelihood. The primary novelty of HAI compared to previous GBI approaches is its use of a generative model to learn the misspecification, allowing it to benefit from advances in deep generative modeling. HAI is further connected to physics-integrated autoencoders (Takeishi and Kalousis, 2021), which offer a hybrid learning strategy for inference under misspecification. Compared to this generic approach, HAI simplifies the training procedure into two distinct steps using the independence assumption from Equation 1.

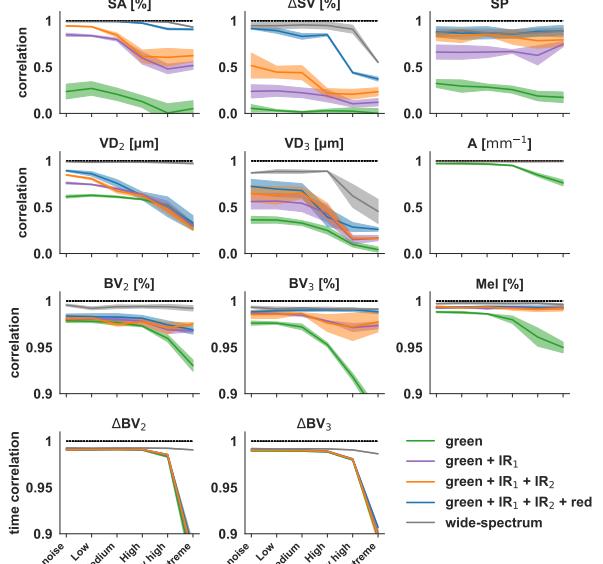


Figure 2: Identifiability analysis of biophysical parameters under varying noise levels and wavelength configurations. We report the Pearson correlation coefficient between the groundtruths and the values inferred from \mathbf{x}_s . For dynamic properties, we report averages of across-time correlation coefficients. We use three random seeds to estimate the variation across different training runs.

4. Results

Through a series of in-silico experiments, we now demonstrate the potential of combining PPGen and HAI to infer interpretable biophysical parameters from real-world PPG measurements.

4.1. Inference of biophysical properties under multiple PPG wavelength configurations

In Figure 2, we analyze the identifiability of the biophysical parameters θ as a function of emitter wavelengths and noise levels in \mathbf{x}_s . The noise levels, shown in Figure 5, range from optimistic to pessimistic hardware noise specifications. The wavelength configurations (subsets of green, infrared, red) mimic standard sensor configurations and are compared to a wide-spectrum setup covering 500–1000nm at 1nm resolution (see subsection B.4 for details on sensor and noise settings). For this analysis, we use only the NPE model q_ϕ described in section 2, trained on matching simulations (details in section D).

Focusing on the 9 static parameters in Figure 2, we find that the wide-spectrum sensor consistently out-

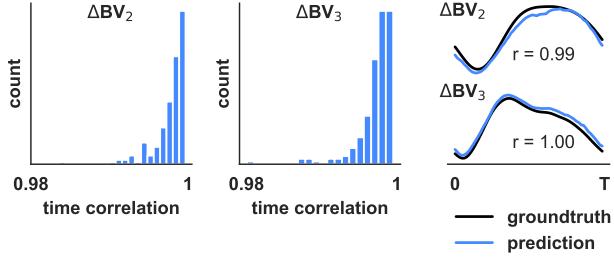


Figure 3: Inferences of blood volume waveforms with a four-wavelength PPG sensor without misspecification at medium noise level. Histograms show the distribution of across-time correlation coefficients between groundtruths and predictions, and the right hand side shows an example waveform pair.

performs all other configurations across noise levels and parameters (see Figure 8 for a breakdown of results at a medium level of noise). Overall these results show the ability of NPE to learn an effective inference procedure, where identifiability slowly declines with increasing noise levels. Focusing on common sensor configurations with few wavelengths we observe that some parameters, particularly those modeling vessel diameters, are challenging to estimate even under optimal conditions. Interestingly, we observe that oxygen saturation parameters (SA, ΔSV) are only robustly identified by multi-wavelength sensors. This well-known biophysical fact (Tamura, 2019) provides evidence that PPGnet faithfully represents some underlying causes of PPGs.

Analyzing the identifiability of dynamic parameters (ΔBV_2 , ΔBV_3) in Figure 2, we observe high reconstruction quality across wavelength configurations up to reasonably high noise levels. As shown in Figure 3, no reconstruction falls below 98% time-correlation under medium noise. This interesting in-silico result suggests that even green-only sensors can disentangle a PPG waveform into two blood volume components and improve reconstruction quality, likely helped by strong priors on waveform structure. Since these inferred waveforms can be mechanistically linked to fundamental cardiovascular parameters (see subsubsection B.1.2), they may open new directions for monitoring cardiovascular health.

4.2. Robustness to misspecification

In Table 1, we assess the ability of HAI to overcome model misspecification, which is expected to be present in real-world data. We consider the following misspecification settings: **None**: no model misspeci-

Table 1: Parameter inference under several misspecifications, as measured by Pearson correlation and mean absolute percentage error (MAPE) averaged over all biophysical parameters. For each result we report \pm the standard deviation over multiple random seeds.

Misspec.	Method	Correlation	MAPE
4-wavelength sensor			
None	HAI	0.93 ± 0.01	5.9 ± 0.6
	Sim-only	0.93 ± 0.01	5.9 ± 0.6
	Real-only	0.74 ± 0.01	14.6 ± 0.6
Noise	HAI	0.83 ± 0.01	10.5 ± 0.4
	Sim-only	0.81 ± 0.01	11.8 ± 1.0
	Real-only	0.73 ± 0.01	16.9 ± 0.6
Sensor	HAI	0.96 ± 0.00	5.2 ± 0.2
	Sim-only	0.82 ± 0.00	11.9 ± 0.7
	Real-only	0.72 ± 0.01	17.1 ± 0.5
Skin	HAI	0.71 ± 0.01	18.2 ± 0.6
	Sim-only	0.70 ± 0.02	18.0 ± 0.8
	Real-only	0.67 ± 0.02	19.1 ± 1.4
Combined	HAI	0.67 ± 0.01	22.6 ± 0.4
	Sim-only	0.65 ± 0.01	24.1 ± 0.8
	Real-only	0.65 ± 0.00	22.1 ± 1.0
wide-spectrum sensor			
Combined	HAI	0.84 ± 0.00	12.6 ± 0.3
	Sim-only	0.84 ± 0.00	11.2 ± 1.4
	Real-only	0.71 ± 0.00	15.2 ± 0.2

fication; **Noise**: forward model underestimates noise in observed data; **Sensor**: geometry of the sensor is imperfectly captured, which we model as constant per channel shift in observed data; **Skin**: imperfect modeling of skin, which we model by using a different skin layer thicknesses; **Combined**: combination of *Noise*, *Sensor*, and *Skin* misspecification. A visualization of these effects on PPG signals is provided in Figure 6 and details are provided in subsection B.4. We compare HAI with two baselines. **Sim-only** directly applies NPE, without any learned correction and therefore lacks the ability to handle potential misspecification. **Real-only** optimizes jointly the encoder q_ϕ and misspecification ψ to maximize Equation 5, and therefore lacks physical inductive biases introduced via NPE pretraining. Experimental details are provided in section D.

We first observe a performance decline under nearly all types of misspecification compared to the well-specified case (*None*). Across all settings, HAI

outperforms or matches the two baselines, with its advantage being most pronounced under *Sensor* misspecification—where the inductive bias of the misspecification model aligns best with the actual gap between real and synthetic data. These contrasted results show that HAI can mitigate model misspecification, while also underscoring the need for careful validation to identify effective misspecification models in practice.

Finally, wide-spectrum sensor results in [Table 1](#) suggest that more informative sensors can improve parameter estimation even under misspecification (see also extended results in [subsection D.3](#)). A parameter-specific breakdown in [Figure 9](#) and [Figure 10](#) reveals two points: first, HAI consistently achieves the best estimates for arterial oxygen saturation, and second, the same parameters that were difficult to estimate under noisy settings are difficult to infer under misspecification.

4.3. Scalability of HAI

The results presented highlight the scalability of combining PPGen with HAI for real-world PPG analysis. Indeed, our framework handles wide-spectrum sensors that increase data volume per sample by over $100\times$, as illustrated in [Figure 2](#) and [Table 1](#). These results also suggest that wide-spectrum sensors could serve as reference non-invasive devices to obtain approximate labels for biophysical parameters in human studies and help better understand the origins of PPG signals. Importantly, while training the neural networks ($\hat{f}_{LT}, q_\phi, q_\psi$) incurs a significant upfront cost, inference is nearly instantaneous and fully parallelizable on modern GPUs. In contrast, iterative inversion of the underlying light transport simulations requires repeated calls to the computationally expensive simulator, making large-scale analysis infeasible. By amortizing both simulations and inference, our framework enables the analysis of millions of PPG pulses efficiently, opening the door to population-level studies of biophysical parameter distributions.

5. Discussion

The clinical utility of PPG is often limited by a lack of annotations and the signal’s poor interpretability. In this paper, we address these challenges by adopting a hybrid ML philosophy. We first propose PPGen, an efficient PPG pulse simulator that is deeply rooted in biophysics and captures the entire gener-

tive process from cardiac parameters to PPG signals. Leveraging this efficient simulator, we then propose HAI, a framework for inferring biophysical parameters from PPG data. Our in-silico experiments show that HAI can robustly and scalably infer these parameters, even under noise and model misspecification.

A key advantage of our framework is its ability to enhance the interpretability of PPG signals. For example, while PPG waveforms are difficult to interpret, decomposing them into contributions from different parts of the peripheral vascular system (i.e. blood volume waveforms at different skin depths) opens the door to non-invasive monitoring of important cardiovascular parameters that are otherwise difficult to estimate. This interpretability also extends to sensor design, where inferring tissue scattering properties can inform the development of next-generation devices that more precisely target specific skin structures.

Though grounded in established biophysical principles, PPGen only approximates the processes underlying real-world PPG measurements. For example, we neglect important sources of variability such as spatial heterogeneity in scattering properties and inter-subject differences in skin structure. Beyond the simplicity of the skin model, another limitation of our simulator is its focus on single heartbeats, ignoring both slower dynamics (e.g., respiration) and artifacts such as motion. In addition to forward-model misspecification, PPGen’s capabilities are constrained by the informed prior distribution we introduced. By design, we limited its informativeness to bounding static parameters and loosely structuring what realistic blood waveforms may look like. These assumptions, however, may be overly conservative, since we expect at least some parameters to be correlated in real-world measurements. While our experiments investigate the potential impact of some of these simplifying assumptions, future work should validate our findings on real-world data with clearly defined modeling objectives. Developing rigorous validation strategies for such objectives will be essential to further demonstrate the practical value of PPGen and HAI.

Taken together, our results show that HAI offers potential for extracting useful health information from real-world PPG signals, while enabling a degree of clinical interpretability not readily available from either raw signals or traditional black-box ML approaches. While our results are a proof of concept based on in-silico experiments, future validation on

real-world data could establish the unique potential of hybrid-ML tools for leveraging wearable data in health applications.

References

- Salar Abbaspourazad, Oussama Elachqar, Andrew Miller, Saba Emrani, Udhayakumar Nallasamy, and Ian Shapiro. Large-scale training of foundation models for wearable biosignals. In *The Twelfth International Conference on Learning Representations*, 2024.
- Raghda Al-Halawani, Meha Qassem, and Panicos A Kyriacou. Monte carlo simulation of the effect of melanin concentration on light-tissue interactions in transmittance and reflectance finger photoplethysmography. *Scientific Reports*, 14(1), 2024.
- Erik Alerstam, Stefan Andersson-Engels, and Tomas Svensson. White monte carlo for time-resolved photon migration. *Journal of biomedical optics*, 13(4), 2008a.
- Erik Alerstam, Tomas Svensson, and Stefan Andersson-Engels. Parallel computing with graphics processing units for high-speed monte carlo simulation of photon migration. *Journal of biomedical optics*, 13(6), 2008b.
- I Benemerito, A Melis, Antoine Wehenkel, and A Marzo. openbf: an open-source finite volume 1d blood flow solver. *Physiological Measurement*, 45(12), 2024.
- Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5), 2016.
- Frank P Bolin, Luther E Preuss, Roy C Taylor, and Robert J Ference. Refractive index of some mammalian tissues using a fiber optic cladding method. *Applied optics*, 28(12), 1989.
- Tananant Boonya-Ananta, Andres J Rodriguez, Anders K Hansen, Joshua D Hutcheson, and Jessica C Ramella-Roman. Modeling of a photoplethysmographic (ppg) waveform through monte carlo as a method of deriving blood pressure in individuals with obesity. In *Optical Interactions with Tissue and Cells XXXI*, volume 11238. SPIE, 2020.
- Patrick Cannon, Daniel Ward, and Sebastian M Schmon. Investigating the impact of model misspecification in neural simulation-based inference. *arXiv preprint arXiv:2209.01845*, 2022.
- Peter H Charlton, Jorge Mariscal Harana, Samuel Vennin, Ye Li, Phil Chowienczyk, and Jordi Alastruey. Modeling arterial pulse waves in healthy aging: a database for in silico evaluation of hemodynamics and pulse wave indexes. *American Journal of Physiology-Heart and Circulatory Physiology*, 317(5), 2019.
- Peter H Charlton, John Allen, Raquel Bailón, Stephanie Baker, Joachim A Behar, Fei Chen, Gari D Clifford, David A Clifton, Harry J Davies, Cheng Ding, et al. The 2023 wearable photoplethysmography roadmap. *Physiological measurement*, 44(11), 2023.
- S. Chatterjee, K. Budidha, and P. A. Kyriacou. Investigating the origin of photoplethysmography using a multiwavelength Monte Carlo model. *Physiological Measurement*, 41(8), 2020.
- Badr-Eddine Chérif-Abdellatif and Pierre Alquier. Mmd-bayes: Robust bayesian estimation via maximum mean discrepancy. In *Symposium on Advances in Approximate Bayesian Inference*. PMLR, 2020.
- Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48), 2020.
- Kaustav Das, Tomonori Yuasa, Takaaki Maeda, Izumi Nishidate, Hideki Funamizu, and Yoshihisa Aizu. Simple detection of absorption change in skin tissue using simulated spectral reflectance database. *Measurement*, 182, 2021.
- Charita Dellaporta, Jeremias Knoblauch, Theodoros Damoulas, and François-Xavier Briol. Robust bayesian inference for simulator-based models via the mmd posterior bootstrap. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.
- H Doostdar and MA Khalilzadeh. Quantification the effect of ageing on characteristics of the photoplethysmogram using an optimized windkessel model. *Journal of Biomedical Physics & Engineering*, 4(3), 2014.

- Luca Formaggia, Alfio Quarteroni, and Allesandro Veneziani. *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*, volume 1. Springer Science & Business Media, 2010.
- H Frederick Frasch, J Yasha Kresh, and Abraham Noordergraaf. Two-port analysis of microcirculation: an extension of windkessel. *American Journal of Physiology-Heart and Circulatory Physiology*, 270(1), 1996.
- Ingemar Fredriksson, Marcus Larsson, and Tomas Strömberg. Inverse monte carlo method in a multilayered tissue model for diffuse reflectance spectroscopy. *Journal of biomedical optics*, 17(4), 2012.
- George M Hale and Marvin R Querry. Optical constants of water in the 200-nm to 200- μ m wavelength region. *Applied optics*, 12(3), 1973.
- Chowdhury Azimul Haque, Tae-Ho Kwon, and Ki-Doo Kim. Cuffless blood pressure estimation based on monte carlo simulation using photoplethysmography signals. *Sensors*, 22(3), 2022.
- Daolang Huang, Ayush Bharti, Amauri Souza, Luigi Acerbi, and Samuel Kaski. Learning robust statistics for simulation-based inference under model misspecification. *Advances in Neural Information Processing Systems*, 36, 2023.
- Steven L Jacques. Optical properties of biological tissues: a review. *Physics in Medicine & Biology*, 58(11), 2013.
- Steven L Jacques and Lihong Wang. Monte carlo modeling of light transport in tissues. In *Optical-thermal response of laser-irradiated tissue*. Springer, 1995.
- Jack Jewson, Jim Q Smith, and Chris Holmes. Principled bayesian minimum divergence inference. *Entropy*, 20(6), 2018.
- Ryan P Kelly, David J Nott, David T Frazier, David J Warne, and Chris Drovandi. Misspecification-robust sequential neural likelihood for simulation-based inference. *arXiv preprint arXiv:2301.13368*, 2023.
- P.A. Kyriacou and J. Allen. *Photoplethysmography: Technology, Signal Analysis and Applications*. Academic Press, 2021.
- Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. *Advances in neural information processing systems*, 30, 2017.
- Laura Manduchi, Antoine Wehenkel, Jens Behrmann, Luca Pegolotti, Andy C Miller, Ozan Sener, Marco Cuturi, Guillermo Sapiro, and Jörn-Henrik Jacobsen. Leveraging cardiovascular simulations for in-vivo prediction of cardiac biomarkers. *arXiv preprint arXiv:2412.17542*, 2024.
- Takuo Matsubara, Jeremias Knoblauch, François-Xavier Briol, and Chris J Oates. Robust generalised bayesian inference for intractable likelihoods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(3), 2022.
- Igor V Meglinski and SJ Matcher. Computer simulation of the skin reflectance spectra. *Computer methods and programs in biomedicine*, 70(2), 2003.
- A Melis. openbf: Julia software for 1d blood flow modelling. *University of Sheffield, Software*, 2018.
- Scott Prahl et al. Optical absorption of hemoglobin. 1999.
- Narasimhan Rajaram, Ashwini Gopal, Xiaojing Zhang, and James W Tunnell. Experimental validation of the effects of microvasculature pigment packaging on in vivo diffuse reflectance spectroscopy. *Lasers in surgery and medicine*, 42(7), 2010.
- Maximilian Reiser, Andreas Breidenassel, and Oliver Amft. Simulation framework for reflective ppg signal analysis depending on sensor placement and wavelength. In *2022 IEEE-EMBS International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE, 2022.
- Allison Scarbrough, Keke Chen, and Bing Yu. Designing a use-error robust machine learning model for quantitative analysis of diffuse reflectance spectra. *Journal of Biomedical Optics*, 29(1), 2024.

- Sanathana Konugolu Venkata Sekar, Ilaria Bargigia, Alberto Dalla Mora, Paola Taroni, Alessandro Ruggeri, Alberto Tosi, Antonio Pifferi, and Andrea Farina. Diffuse optical characterization of collagen absorption from 500 to 1700 nm. *Journal of biomedical optics*, 22(1), 2017.
- Naoya Takeishi and Alexandros Kalousis. Physics-integrated variational autoencoders for robust and interpretable generative modeling. *Advances in Neural Information Processing Systems*, 34, 2021.
- Toshiyo Tamura. Current progress of photoplethysmography and spo2 for health monitoring. *Biomedical engineering letters*, 9(1):21–36, 2019.
- Akio Tanaka. Analysis of a microcirculatory windkessel model using photoplethysmography with green light: A pilot study. *IEICE Electronics Express*, 19(21), 2022.
- Qunfeng Tang, Zhencheng Chen, Rabab Ward, and Mohamed Elgendi. Synthetic photoplethysmogram generation using two gaussian functions. *Scientific Reports*, 10(1), 2020.
- RLP Van Veen, HJCM Sterenborg, A Pifferi, A Torricelli, and R Cubeddu. Determination of vis-nir absorption coefficients of mammalian fat, with time- and spatially resolved diffuse reflectance and transmission spectroscopy. In *Proceedings of Biomedical Topical Meetings*, 2000.
- RLP Van Veen, W Verkruyse, and HJCM Sterenborg. Diffuse-reflectance spectroscopy from 500 to 1060 nm by correction for inhomogeneously distributed absorbers. *Optics letters*, 27(4), 2002.
- Lihong Wang, Steven L Jacques, and Liqiong Zheng. Mcml—monte carlo modeling of light transport in multi-layered tissues. *Computer methods and programs in biomedicine*, 47(2), 1995.
- Daniel Ward, Patrick Cannon, Mark Beaumont, Matteo Fasiolo, and Sebastian Schmon. Robust neural posterior estimation and statistical model criticism. *Advances in Neural Information Processing Systems*, 35, 2022.
- Antoine Wehenkel, Juan L. Gamella, Ozan Sener, Jens Behrmann, Guillermo Sapiro, Joern-Henrik Jacobsen, and marco cuturi. Addressing misspecification in simulation-based inference through data-driven calibration. In *Forty-second International Conference on Machine Learning*, 2025.
- Xiewei Zhong, Xiang Wen, and Dan Zhu. Lookup-table-based inverse model for human skin reflectance spectroscopy: two-layered monte carlo simulations and experiments. *Optics express*, 22(2), 2014.

Appendix A. Notations

Table 2: Summary of notation.

Symbol	Description
$\mathbf{x}_o \in \mathbb{R}^{R \times N \times T}$	Observed PPG pulse with R receivers, N emitters, T timesteps
$\hat{\mathbf{x}}_s \in \mathbb{R}^{R \times N \times T}$	Noiseless simulated (synthetic) PPG pulse generated by the forward model
$\mathbf{x}_s \in \mathbb{R}^{R \times N \times T}$	Simulated (synthetic) PPG pulse generated by the forward model
$\mathbf{x}_s^t \in \mathbb{R}^{R \times N}$	Simulated PPG at timestep t
$\theta \in \Theta \subset \mathbb{R}^{9+2T}$	Biophysical parameters (static + dynamic)
$\theta_s \in \mathbb{R}^9$	Static parameters (e.g., melanin, blood fractions, vessel diameters, oxygenation)
$\theta_d = (\theta_d^1, \dots, \theta_d^T)$, $\theta_d^t \in \mathbb{R}^2$	Dynamic parameters (blood volume changes per timestep)
$\theta^t = [\theta_s, \theta_d^t] \in \mathbb{R}^{11}$	Biophysical parameters at timestep t
$\lambda \in \mathbb{R}$	Wavelength of emitted light
$\mu_a(\lambda) \in \mathbb{R}^l$	Absorption coefficients at wavelength λ in l skin layers
$\mu_s(\lambda) \in \mathbb{R}^l$	Scattering coefficients at wavelength λ in l skin layers
$f_b(\theta^t, \lambda)$	Mapping from biophysical parameters to optical properties (μ_a, μ_s)
$\hat{f}_{LT}(\mu_a, \mu_s)$	Surrogate light transport simulator: maps optical properties to noiseless PPG
$f(\theta)$	Full differentiable forward model $f = \hat{f}_{LT} \circ f_b$
$p(\mathbf{x}_s \theta)$	Likelihood of simulated PPG given parameters
$p(\mathbf{x}_o \theta)$	Likelihood of observed PPG given parameters
$p(\theta)$ or $\pi(\theta)$	Prior over biophysical parameters
$p(\theta, \mathbf{x}_s) = p(\mathbf{x}_s \theta)\pi(\theta)$	Joint generative model
$q_\psi(\mathbf{x}_s \mathbf{x}_o)$	Misspecification (correction) model, parameterized by ψ
$q_\phi(\theta \mathbf{x}_s)$	Neural posterior estimator of θ given synthetic \mathbf{x}_s
$q_\phi^*(\theta \mathbf{x}_s)$	Optimal neural posterior estimator (limit of training)
$p(\theta \mathbf{x}_o)$	True posterior distribution of parameters given observed data
k_s, σ_w^2	Shot noise and white noise parameters of sensor model
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$\mathbb{E}_{q(\cdot)}[\cdot]$	Expectation under distribution q
\mathcal{L}	Loss function

Appendix B. PPGen: PPG Pulse Generator

B.1. Prior over biophysical parameters

A detailed list of the biophysical parameters $\theta = [\theta_s, \theta_d]$ considered in our model can be found in [Table 3](#).

	Name	Description	Range	Unit	Type
μ_s	A	Scattering amplitude parameter.	[0.25 - 1.0]	mm^{-1}	Static
	SP	Scattering power parameter.	[1.3 - 1.5]	-	Static
Mel		Melanin fraction in the epidermis.	[0.25 - 14]	%	Static
BV_2		Blood fraction during diastole in the dermis. It informs about the tissue perfusion.	[0.1 - 4]	%	Static
BV_3		Blood fraction during diastole in the subcutaneous layer. It informs about the tissue perfusion.	[0.1 - 8]	%	Static
VD_2		Vessel diameter in the dermis. It is an aging biomarker.	[0.01 - 0.04]	mm	Static
VD_3		Vessel diameter in the subcutaneous layer. It is an aging biomarker.	[0.04 - 0.06]	mm	Static
μ_a	SA	Arterial oxygen saturation. It informs on the oxygenation status (e.g., hypoxemia).	[60 - 100]	%	Static
	ΔSV	Arteriovenous oxygen saturation difference. It informs on the oxygen extraction (e.g., exercise, sepsis).	[1 - 20]	%	Static
	ΔBV_2	Systolic blood fraction scaling in dermis. It informs on the cardiovascular system behavior.	[1.0 - 1.02]	-	Dynamic
	ΔBV_3	Systolic blood fraction scaling in subcutaneous layer. It informs on the cardiovascular system behavior.	[1.0 - 1.02]	-	Dynamic

Table 3: List of biophysical parameters considered in our model. While most parameters can be considered static within short time windows (≈ 1 second), blood volume related changes occur in much shorter time-scales and are considered dynamic in our pipeline.

The set of biophysical parameters is sampled from an informed prior $\pi(\theta)$. We describe below the two different strategies used to obtain samples for static parameters θ_s and for dynamic parameters θ_d .

B.1.1. STATIC PARAMETER SAMPLING

Static biophysical parameters θ_s are sampled uniformly from a range of values considered biophysically plausible and extracted from the literature. The ranges considered for each parameter are provided in [Table 3](#).

B.1.2. BLOOD VOLUME WAVEFORM GENERATION

Our priors for dynamic tissue properties θ_d , namely, the blood volume waveforms in the dermis and subcutaneous tissue layers, are based on mechanistic models. The dynamic component of tissue blood volume is a result of pulsatile pressure waves traveling from large underlying arteries through a network of smaller vessels (capillaries, arterioles and small arteries) in the tissue before draining into the venous system. For our purposes, we use synthetic arterial pressure waves simulated at the radial artery of the (left) wrist, and map them to blood volume waveforms proximal to the skin at the same location.

Synthetic arterial pressure waveforms. We consider the dataset of arterial pressure waves introduced in [Manduchi et al. \(2024\)](#). These waveforms were simulated using the finite-volume solver described in [Melis \(2018\)](#); [Benemerito et al. \(2024\)](#) by randomly sampling parameters that influence the cardiovascular system function of virtual subjects (heart rate, stroke volume, peak flow time, reverse flow volume, subject height)

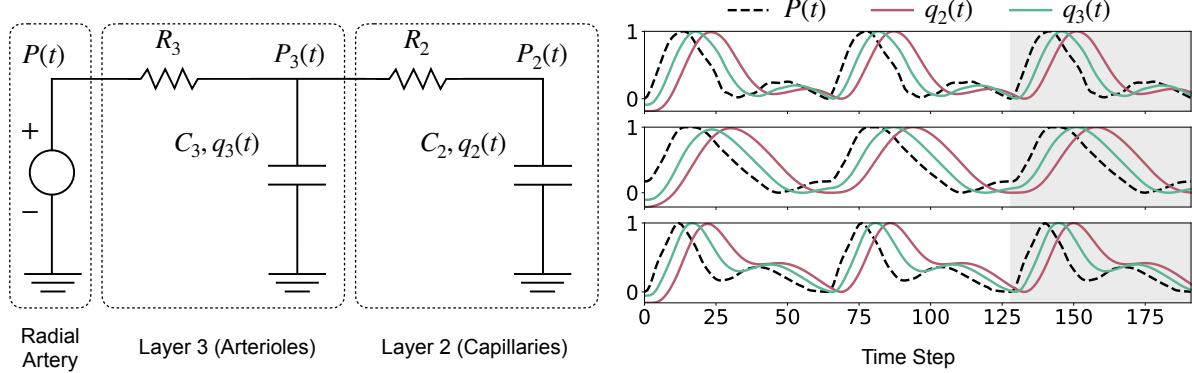


Figure 4: On the left, Windkessel model relating the arterial pressure wave at the radial artery $P(t)$ to blood volumes in the two layers of the skin $q_2(t)$ and $q_3(t)$. On the right, solution of the Windkessel equations for three different virtual subjects over three cardiac cycles. The shaded area represents the last cardiac cycle, which would be used to extract the blood volumes to use in our model.

in physiological ranges. See [Charlton et al. \(2019\)](#) for details on how these virtual subjects were defined. Although the solver allows for the evaluation of pressure waveforms at any point in the human body, we chose to only consider those simulated at a random location of the left radial artery (the vessel supplying blood to the left wrist) because of its relevance in wearable devices such as smartwatches. More specifically, the dataset consists of $N_s = 81'660$ waveforms of the form $[P_{1,j}, \dots, P_{T,j}]$, for $j = 1, \dots, N_s$, where $P_{i,j}$ is the reading of the arterial pressure wave of the virtual subject j at the sample point i . As all waveforms are resampled over a common grid composed of $T = 64$ points, we also keep track of the sampling frequency $f_j = T/M_j$, where M_j is the number of sample points per cardiac cycle in the original waveform (note that, in the original dataset, all waveforms are stored with a common sampling frequency of 500 Hz).

Blood volume waveforms. Given the arterial pressure waveforms described above, we use Windkessel models to compute blood volume waveforms proximal to the skin. For a reference of the application of Windkessel models in the microvasculature see e.g. [Frasch et al. \(1996\)](#). These models simplify the fluid dynamics of blood flow by reducing a cardiovascular system of vessels to an analogous lumped parameter electrical circuit.

In our case, we consider the two-layer model shown in [Figure 4](#) (left), with three compartments that represent the radial artery, narrower vessels in subcutaneous (Layer 3 in our tissue model), and even narrower vessels in dermis (Layer 2). Pressure in the radial artery is equivalent to the potential $P(t)$ at the voltage source on the left. The two layers act as a low-pass RC filters with time constants $\tau_3 = R_3 C_3$ and $\tau_2 = R_2 C_2$ respectively. Note that Layer 1 (not shown in [Figure 4](#)) representing the top epidermis layer is not included in the model due to absence of any significant vessels in its structure. The blood volumes in layers 3 and 2 are equivalent to the charges $q_3(t)$ and $q_2(t)$ on the respective capacitors. Eventually, the mathematical relationships between $P(t)$, $q_3(t)$ and $q_2(t)$ are second-order differential equations of the form

$$\begin{aligned} P(t) &\approx \frac{\tau_2 \tau_3}{C_2} \frac{d^2 q_2}{dt^2} + \frac{\tau_2 + \tau_3}{C_2} \frac{dq_2}{dt} + \frac{1}{C_2} q_2(t), \\ \tau_2 C_3 \frac{dP(t)}{dt} + P(t) &\approx \frac{\tau_3 \tau_2}{C_3} \frac{d^2 q_3(t)}{dt^2} + \frac{\tau_3 + \tau_2}{C_3} \frac{dq_3(t)}{dt} + \frac{1}{C_3} q_3(t). \end{aligned} \quad (6)$$

To create blood volume waveforms, we solve (6) starting from batches of the arterial pressure waves by sampling a set of indices $\mathcal{J} \in \{1, \dots, N_s\}$ with $|\mathcal{J}| = n_{\text{bs}}$ is the batch size. For each batch, we randomly sample system parameters R_2 , C_2 , R_3 , C_3 , while enforcing physiological conditions based on realistic assumptions on the radii in the different compartments. We recall that, if R and C are the resistance and capacitance of

a blood vessel and r is its radius, then $R \propto r^{-4}$ and $C \propto r^3$ (see [Formaggia et al. \(2010\)](#)). Since vessel radii decrease from deeper to shallower tissue, our sampling strategy enforces

$$\begin{aligned} R_2 &> R_3 \\ C_2 &< C_3 \\ \tau_2 = R_2 C_2 &> \tau_3 = R_3 C_3, \end{aligned} \tag{7}$$

where τ_2 and τ_3 are the time constants of the two compartments. In order to solve the continuous-time dynamics we approximate (6) in discrete time using pole-zero matching, and recursively compute the blood volume waveforms at time step size Δt . Given the notation introduced in the previous paragraph, the blood volume waveform $q_{2,i,j}$ at time step i for virtual subject $j \in \mathcal{J}$ can be expressed as

$$\begin{aligned} q_{2,i,j} &= P_{i-2,j} \cdot C_2 \cdot (1 - e^{-\Delta t/\tau_2}) \cdot (1 - e^{-\Delta t/\tau_3}) \\ &\quad + q_{2,i-1,j} \cdot (e^{-\Delta t/\tau_2} + e^{-\Delta t/\tau_3}) \\ &\quad - q_{2,i-2,j} \cdot (e^{-\Delta t/\tau_2} \cdot e^{-\Delta t/\tau_3}), \end{aligned} \tag{8}$$

where $\Delta t = 1/f_j$ is the time step size. Similarly, for Layer 3, the blood volume waveform $q_{3,i,j}$ can be expressed as

$$\begin{aligned} q_{3,i,j} &= (P_{i-1} - e^{-\Delta t/\tau_2} \cdot P_{i-2}) \cdot C_3 \cdot (1 - e^{-\Delta t/\tau_3}) \\ &\quad + q_{3,i-1,j} \cdot (e^{-\Delta t/\tau_2} + e^{-\Delta t/\tau_3}) \\ &\quad - q_{3,i-2,j} \cdot (e^{-\Delta t/\tau_2} \cdot e^{-\Delta t/\tau_3}). \end{aligned} \tag{9}$$

To use these equations in practice, we need to assign initial conditions to $q_{2,i,j}$ and $q_{3,i,j}$. In particular, we choose zero initial condition as

$$q_{2,1,j} = q_{2,2,j} = q_{3,1,j} = q_{3,2,j} = 0,$$

for all $j \in \mathcal{J}$. Due to this strong approximation, we allow the system to reach convergence (i.e., a quasi-periodic behavior over a single heart beat) by simulating multiple cardiac cycles. This is possible because the arterial pressure waves are designed to be approximately periodic, namely $P_{i,j} \approx P_{T-1,j}$ for $j = 1, \dots, N_s$, and the same arterial pressure wave can be used for all cardiac cycles. [Figure 4](#) (right) shows the solution of [Equation 8](#) and [Equation 9](#) for three virtual subjects over three cardiac cycles.

Once $q_2(t)$ and $q_3(t)$ are computed by the solver, they are used as the shape of the blood volume waveforms ΔBV_2 and ΔBV_3 . Afterwards, the minimum and maximum for both waveforms are sampled in $[1.0, 1.02]$, with the constrain that $\max(\Delta BV) - \min(\Delta BV) \geq 1.01$. This constraint establishes a lower bound on the amplitude of a blood volume pulse.

B.2. Optical tissue model

Once the biophysical parameters θ have been sampled from the prior $\pi(\theta)$, we use an optical tissue model to compute the corresponding optical properties. The optical properties of the tissue given a set of parameters θ consist of the layer-specific absorption coefficient μ_a and scattering coefficient μ_s . Note that, while μ_a is layer-specific, we simplify our tissue model and assume μ_s is shared across all tissue layers. The modeling approaches for absorption and scattering differ significantly and are explained below.

B.2.1. ABSORPTION COEFFICIENT μ_a

The absorption coefficient, $\mu_a(\lambda) \in \mathbb{R}$, scales with the molar extinction coefficient, $\varepsilon_k(\lambda) \in \mathbb{R}$, as a weighted sum

$$\mu_a(\lambda) = \sum_{k=1}^K \varepsilon_k(\lambda) c_k.$$

The molar concentrations $c_k \in \mathbb{R}$ of the dominating K light-absorbing molecules (chromophores) in each tissue layer, primarily oxygenated hemoglobin, deoxygenated hemoglobin, and melanin, are drawn from established extinction spectra from the following references:

- Hemoglobin ([Pralh et al., 1999](#))
- Water ([Hale and Querry, 1973](#))
- Lipid ([Van Veen et al., 2000](#))
- Collagen ([Sekar et al., 2017](#))
- Melanin ([Jacques, 2013](#)).

For a visualization of extinction spectra, see for example Figure 1 in [Meglinski and Matcher \(2003\)](#).

We consider different compositions across tissue layers. In particular, we consider the following six chromophores: hemoglobin (oxygenated and de-oxygenated), Water, Lipids, Collagen, and Melanin. See [Table 4](#) for an overview of concentrations across tissue layers.

For hemoglobin (blood) we consider a more detailed model as it is the key chromophore that lets us understand the cardiovascular system. First, we define arterial and venous blood fraction with a fixed static ratio of 1 / 3 (arterial / venous). Using this and the arterial oxygen saturation (SA) and venous oxygen saturation ($SV = SA - \Delta SV$), we define the concentration of oxygenated and de-oxygenated blood as

$$\begin{aligned} c_{oxy} &= \frac{1}{4} \cdot \frac{SA}{100} \cdot BV \cdot \Delta BV + \frac{3}{4} \cdot \frac{SV}{100} \cdot BV \\ c_{deoxy} &= \frac{1}{4} \left(1 - \frac{SA}{100}\right) BV \cdot \Delta BV + \frac{3}{4} \left(1 - \frac{SV}{100}\right) BV, \end{aligned}$$

where the first summand in each equation corresponds to the contribution of arterial and the second to venous blood. The component ΔBV represents the systolic blood scaling factor (blood volume waveforms). Our simulation model for these parameters is described in [subsubsection B.1.2](#).

Furthermore, we employ a vessel diameter (denoted as VD) based correction to include the effects of mean blood vessel size of the absorption of blood as

$$\tilde{\mu}_a^{blood} = \frac{BV \cdot \Delta BV \cdot (1 - e^{-\frac{\mu_a^{blood}}{BV} \cdot \Delta BV \cdot VD})}{VD}.$$

This correction was proposed by [Van Veen et al. \(2002\)](#) and later validated by [Rajaram et al. \(2010\)](#).

Lastly, we share the arterial (SA) and venous oxygen saturation (SV) across the dermis and subcutaneous layer, while blood fraction and blood volume waveforms are considered separately for both layers. For an overview of all physiological parameters see [Table 3](#).

B.2.2. SCATTERING COEFFICIENT μ_s

In contrast to absorption μ_a , the scattering coefficient μ_s is modeled empirically. While scattering is influenced by biological factors like cell density and size, the relationship is less direct. We therefore adopt a standard empirical formulation that effectively models the observed wavelength-dependence of tissue scattering ([Jacques, 2013](#)). More specifically we model the scattering parameter $\mu_s(\lambda)$ as

$$\mu_s(\lambda) = \frac{\mu'_s}{1 - g},$$

where $g \in \mathbb{R}$ is a fixed anisotropy coefficient and $\mu'_s \in \mathbb{R}$ is the reduced scattering coefficient, modeled via a power law as

$$\mu'_s(\lambda) = A \left(\frac{\lambda}{1000} \right)^{-SP},$$

Name	Epidermis	Dermis	Subcutaneous
Hemoglobin (oxygenated)	0	vary	vary
Hemoglobin (de-oxygenated)	0	vary	vary
Water	60	70	10
Lipid	35	5	90
Collagen	0	25	0
Melanin	vary	0	0

Table 4: Overview of considered chromophores for the three considered tissue layers, where each value represents a volume fraction in %. Some concentrations are not fixed to a specific value (denoted by *vary*) and are thus (indirectly) inferred using our parameters θ as described in subsection B.2.

where wavelength λ is represented in nm units with 1000 being the base wavelength, A describes the scattering amplitude and SP the scattering power parameter.

In our tissue model we consider three tissue layers (epidermis, dermis, and subcutaneous layer). While most tissue properties vary from individual to individual, we fixed the following parameters to simplify the model. For the epidermis layer we employ a layer thickness of 0.2 mm, for the dermis 1.5 mm, and for the subcutaneous layer we use 18.3 mm. In terms of scattering, we assume that scattering properties are the same across tissue layers. We use a refractive index of 1.4 for each layer as proposed by [Bolin et al. \(1989\)](#). Lastly, we employ the Henyey-Greenstein phase function with $g = 0.9$ to represent the anisotropy of scattering, which is consistent with reported tissue values ([Jacques, 2013](#)).

B.3. Building a surrogate model of light transport

Simulation settings. We build surrogate models of the computationally expensive light transport simulations with the following key settings for the simulation:

- 10^7 photons for each simulation
- Post-processing from scattering-only base simulation as proposed by [Alerstam et al. \(2008a\)](#)
- Wavelength-independence assumption with average properties of the sensor probe (reflectivity, refractive index).

As stated above, for simplicity we assume that light propagation does not depend on the wavelength and is fully governed by absorption μ_a and scattering μ_s . This assumption may be violated in practice if the material of the sensor has wavelength-dependent reflection properties, but it allows us to reduce the computational burden significantly. In particular, we can build a surrogate model $\hat{f}_{LT} : (\mu_a, \mu_s) \mapsto \mathbf{x}_s^t$, which is independent of the wavelength.

Simulation lookup table. This property is then used to create a training dataset for the surrogate model, which we call *simulation lookup table* (LUT). For this dataset we first define the ranges for μ_a and μ_s per tissue layer by taking the ranges for the tissue parameters (see Table 3) and using the optical tissue model (see subsection B.2) to compute maximal and minimal values. In summary, this step yields min/max values in a 4 dimensional space (absorption for three tissue layers and a shared scattering value for all tissue layers).

To sample this 4D-space, we apply a log-transform to the absorption values, which over-weights lower values during training. This approach is motivated by the extinction properties of the considered chromophores. For instance, hemoglobin's extinction coefficient is very low for wavelengths above 600nm, meaning changes in its concentration produce only minor perturbations in absorption in this range.

We then employ various sampling strategies to form the LUT:

- Sobol Sampling (quasi-random sampling to ensure guaranteed coverage)

- Two independent Latin Hypercube samples (to balance randomness with coverage of the space).

Furthermore, we aim to increase sensitivity in our surrogate model towards small blood-induced perturbations as this is the key chromophore we want to measure with PPG sensors. For this, we add 5 different sets of perturbations in absorption per layer, where we use random multiplier of μ_a with standard deviations from [0.00001, 0.0001, .001, 0.01, 0.1, 0.0].

In summary, the LUT consists of

- 35 different values for μ_s
- 10k samples for each Sobol and Latin Hypercube sample
- 6 μ_a perturbations for each base simulation.

This yields 5.25 million datapoints that can be used to train a surrogate model of the computationally expensive light simulation.

Training settings for the surrogate model. After constructing a simulation lookup table (LUT), we define the surrogate model of light transport as a mapping

$$\hat{f}_{LT} : (\mu_a^1, \mu_a^2, \mu_a^3, \mu_s) \mapsto \mathbf{x}_s^t,$$

where μ_a^l is the absorption in tissue layer l and $\mathbf{x}_s^t \in \mathbb{R}^4$ as we consider 4 distinct receiver channel (paths between single light emitter and 4 detectors). As the mapping may be highly non-linear and due to having access to a large LUT, we employ neural networks for \hat{f}_{LT} with the following workflow:

1. Log-transform of μ_a and \mathbf{x}_s^t
2. Standardization of inputs and outputs based on mean/std computed over 10 batches from the training set
3. Multi-layer perceptron with 3 layers of size [100, 100, 100] and tanh nonlinearity to guarantee smoothness of the predictions.

After defining the architecture, we split the LUT into training and validation dataset by leaving 15% of μ_s values for validation. We split across μ_s , because we have fewer diversity in scattering due to the computationally expensive simulation w.r.t. scattering. We use evenly spaced intervals. This ensures the validation groups are spread out across the entire range of μ_s values, which is useful as μ_s represents a continuous physical parameter value and we want to ensure the validation set tests the model's performance across that entire range.

As training losses, we use mean-absolute-error (MAE) to reduce the effect of outliers from the LUT (which may happen due to the stochastic nature of the Monte-Carlo light transport simulation). We train the surrogate model \hat{f}_{LT} using both a MAE loss of amplitudes as well as a MAE loss on amplitude differences for μ_a base simulations and perturbation cases (see description of dataset in previous paragraph). Both losses are weighted equally after normalizing each loss component with the loss value at from the first batch.

We then train with a batch size of 1000, a learning rate of 1e-4, Adam optimizer for 400 epochs. The model is then selected based on the lowest loss on the validation set.

B.4. Sensor model and noise model

Sensor Model. As described in [section 2.1](#), we consider a common four-wavelength PPG setup and a wide-spectrum spectroscopic device as two test cases with the following characteristics (see [Figure 1](#) for a visual sketch of the sensor):

- Four-wavelength PPG setup with $N = 4$ wavelengths (green, red and two infrared LEDs) and $R = 4$ channels to simulate different spacings between LEDs and photodetectors.

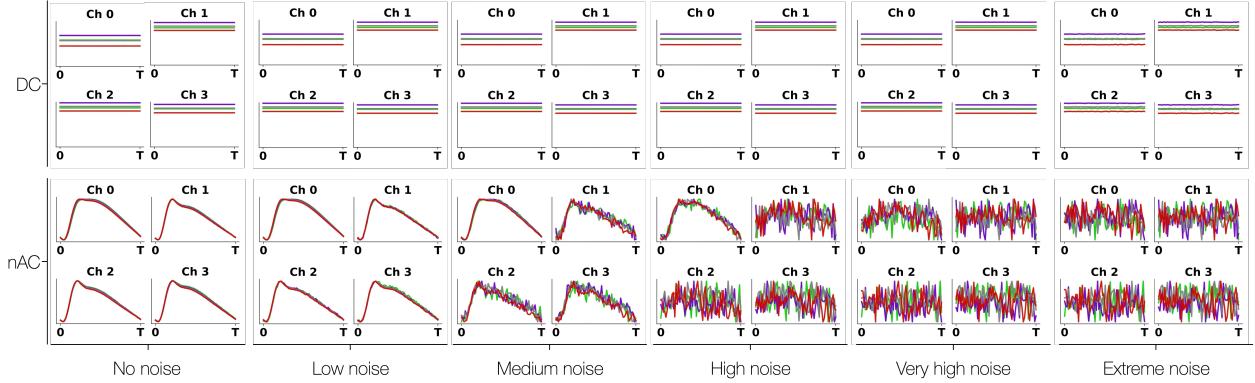


Figure 5: Visualization of the six noise levels considered in our experiments for a randomly selected four-wavelength measurement. The effect of the noise is mostly noticeable on the normalized AC component of PPG measurements. Nevertheless, it challenges parameter inference as shown in Figure 2.

- Wide-spectrum spectroscopic sensor with $N = 531$ (470nm - 1000nm) and $R = 4$ as above.

To make the comparison fair, we consider the same geometrical layout of light source and detector for both cases. In particular, we have $R = 4$ channels that are obtained from a single light source and four distinct spacings of detectors. The spacings are 3/4/5/6 mm, the center illumination source is 3.2 mm and detection fibers are approximately 0.5mm.

To model the four-wavelength PPG sensor, we employ profiles over the wavelength range to mimic the emission of lights from LEDs using in PPG sensors. Since LEDs are rather emitting light over a range of wavelengths, each light source emits a distribution of photons with different wavelengths. Formally, for a PPG signal at timestep t from channel r and color n , we apply a weighted sum

$$\mathbf{x}_s^{r,n,t} = \sum_i \mathbf{x}_s^{r,t}(\lambda_i) L_n(\lambda_i),$$

with LED profiles $\sum_i L_n(\lambda_i) = 1$. In our four-wavelength sensor setup, we considered LED profiles centered at 525 nm (green), 660nm (red), 850 nm (infrared), and 940nm (infrared).

Noise Model. We apply noise independently to each scalar $\hat{\mathbf{x}}_s^{r,n,t}$ in the signal tensor $\hat{\mathbf{x}}_s \in \mathbb{R}^{R \times N \times T}$. First, we represent shot noise as zero-mean Gaussian noise with signal-dependent variance $\eta_{Shot} \sim \mathcal{N}(0, k_s \cdot \hat{\mathbf{x}}_s^{r,n,t})$, where k_s controls the magnitude of shot noise. Second, we model additive Gaussian noise $\eta_w \sim \mathcal{N}(0, \sigma_w^2)$, representing sensor-internal effects such as thermal and chip noise. After sampling these two noise components, we add both components to the clean signal. To cover multiple levels of noise, we explore various values for k_{shot} and σ_w , see Table 5.

Artificial misspecification. We construct multiple misspecification settings as discussed in subsection 4.2. The **Noise** misspecification uses clean signals for the simulated data \mathbf{x}_s and medium noise for the observed data \mathbf{x}_o . For **Sensor** misspecification we add an offset of 0.1 to each receiver channel, before normalization (see details on normalization in section D). The **Skin** misspecification is obtained by training another surrogate model \tilde{f}_{LT} using a simulation lookup table that was generated with a slightly perturbed layer thickness. The layer thickness for this perturbed setting was 0.15 mm for the epidermis (instead of 0.2 mm), 1.5 mm for the dermis layer (as before), and 18.35 mm for the subcutaneous layer (instead of 18.3 mm). The **Combined** misspecification applies the offset shift of 0.1, uses the perturbed surrogate model \tilde{f}_{LT} instead of the original \hat{f}_{LT} for PPGen, and considers medium level noise for \mathbf{x}_s and high noise for \mathbf{x}_o . The effect of these different types of misspecification is visualized in Figure 6.

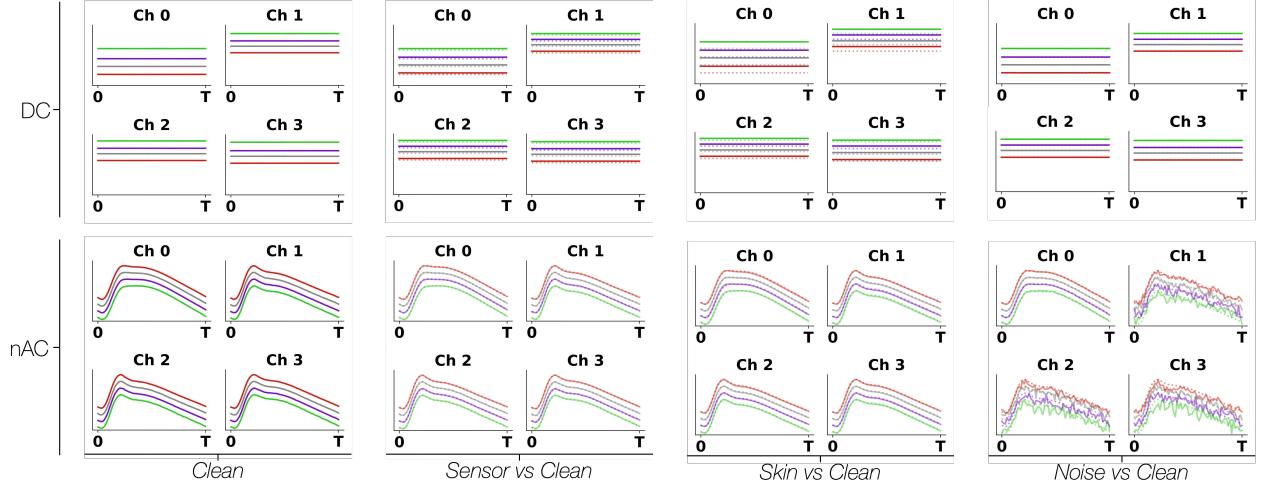


Figure 6: Vizualization of the misspecifications considered in our experiments against the same clean simulation (shown in dotted lines in the three right columns), for a randomly selected four-wavelength measurement. For the normalized AC we introduce an arbitrary vertical offset for the of vizualization.

Level	White Gaussian Noise σ	Shot Noise k_{shot}
No noise	0	0
Low noise	1.0E-06	1.0E-07
Medium noise	1.0E-05	1.0E-06
High noise	1.0E-04	1.0E-05
Very high noise	1.0E-03	1.0E-04
Extreme noise	1.0E-02	1.0E-03

Table 5: Overview of considered noise levels. See [Figure 5](#) for a visualization of noise.

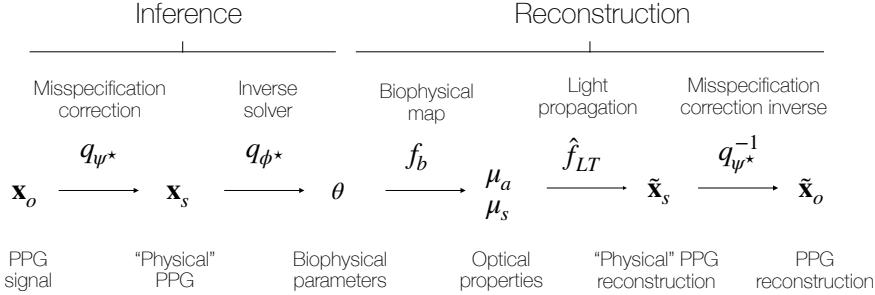


Figure 7: Schematic of the Hybrid Amortized Inference (HAI) framework.

Appendix C. HAI – Algorithms and implementation

C.1. Algorithms

We provide a schematic of the HAI framework in Figure 7 and the sampling algorithm in Algorithm 2.

Algorithm 2 Differentiable density evaluation and sampling for the PPG forward model.

```

EvaluateDensity( $\theta, \mathbf{x}_s$ ) Decompose  $\theta$  into  $\theta_s$  and  $\{\theta_d^t\}_{t=1}^T$ 
 $\ell \leftarrow 0$ 
for  $t = 1, \dots, T$  in parallel do
  foreach  $\lambda \in \Lambda$  in parallel do
    |  $(\mu_a^{(t,\lambda)}, \mu_s^{(\lambda)}) \leftarrow f_b([\theta_s, \theta_d^t], \lambda)$ 
  end
   $\hat{\mathbf{x}}_s^t \leftarrow \hat{f}_{LT}(\{\mu_a^{(t,\lambda)}\}_{\lambda \in \Lambda}, \{\mu_s^{(\lambda)}\}_{\lambda \in \Lambda})$ 
   $\ell \leftarrow \ell + \log p(\mathbf{x}_s^t | \hat{\mathbf{x}}_s^t)$ 
end
return  $\ell$ 

SampleFromModel( $\Lambda, T$ )
(Prior sampling) Static:
 $\theta_s \sim \text{Uniform}(\text{literature ranges})$ 
(Prior sampling) Dynamic:
draw arterial pressure waves from a full-body hemodynamics simulator and map to dermis/subcutis blood-volume waveforms via Windkessel; form  $\{\theta_d^t\}_{t=1}^T$ 
 $\theta \leftarrow [\theta_s, \{\theta_d^t\}_{t=1}^T]$ 
for  $t = 1, \dots, T$  in parallel do
  foreach  $\lambda \in \Lambda$  in parallel do
    |  $(\mu_a^{(t,\lambda)}, \mu_s^{(\lambda)}) \leftarrow f_b([\theta_s, \theta_d^t], \lambda)$ 
  end
   $\hat{\mathbf{x}}_s^t \leftarrow \hat{f}_{LT}(\{\mu_a^{(t,\lambda)}\}_{\lambda \in \Lambda}, \{\mu_s^{(\lambda)}\}_{\lambda \in \Lambda})$ 
end
Assemble  $\hat{\mathbf{x}}_s \leftarrow [\hat{\mathbf{x}}_s^1, \dots, \hat{\mathbf{x}}_s^T]$ 
 $\mathbf{x}_s \sim p(\mathbf{x}_s | \hat{\mathbf{x}}_s)$ 
return  $(\theta, \mathbf{x}_s)$ 

```

C.2. PPG features

The pulsatile (AC) component of a PPG signal, which contains key morphological information, is typically much smaller in magnitude than the quasi-static (DC) baseline. Consequently, a standard reconstruction loss, such as the likelihood introduced in subsection 2.1, can be dominated by the DC component, hindering the accurate reconstruction of the vital AC waveform. To address this, we decompose the observation \mathbf{x}_o into its constituent parts. We define the DC component as the signal’s temporal mean and the AC component as the mean-centered signal. We then compute the reconstruction loss using separate L2 norms on the AC and DC components, which is equivalent to assuming an isotropic Gaussian likelihood over these decomposed features. To further isolate the waveform’s shape, we also compute a normalized AC (nAC) feature. These engineered features (AC, DC, and nAC) are also provided as augmented inputs to the encoder network q_ϕ . Without this approach of augmenting both the loss function and the input features, capturing the subtle morphological changes in the PPG signal—and the physiological parameters that influence them—would be difficult due to the overwhelming amplitude of the DC component.

Appendix D. Experimental details and extended results

Before setting the hyperparameters in our training workflow, we ran a hyperparameter search over most parameters.

D.1. Architectures and training details

Normalization. We standardize the additional AC and normalized AC input features (see subsection C.2) by computing the mean and standard deviation over 10 batches. The DC input features and biophysical parameters are standardized using the moments from the surrogate model training described in subsection B.3.

Mapping parameter to specified ranges. Since the parameters θ are constrained to the ranges given in Table 3, we need to map the real-valued outputs of the encoder to these ranges. For this we employ the cumulative distribution function (CDF) of the Gaussian distribution to map to $[0, 1]$. Afterwards we linearly transform the interval $[0, 1]$ to specified ranges for each parameter.

Neural Posterior Estimator (NPE). The NPE, denoted as q_ϕ , is implemented as a 1D U-Net, with the architecture described in Algorithm 3 for the four-wavelength setting and in Algorithm 4 for the wide-spectrum setting. In the four-wavelength setting, we flatten the receiver and wavelength dimensions (step 9), while we extract embeddings for the wide-spectrum setting (step 9-13). Afterwards the U-Net architecture processes the temporally embedded data and applies the decoding stage to predict dynamic biophysical parameters (two blood volume waveforms, ΔBV_2 and ΔBV_3). The intermediate representations from the U-Net’s encoder are preserved for the subsequent static parameter inference head.

Pretraining details for NPE (used in subsection 4.1). We pretrain the four-wavelength NPE model for 2500 epochs and the wide-spectrum model for 2000 epochs. Each epoch consists of 100 iterations with a batch size of 200 samples. During pretraining we create synthetic data from PPGen on-the-fly, i.e. the model learns on unseen batches at each step of the training procedure. The loss function $\mathcal{L}_s(\phi)$ is implemented using the mean-squared-error (MSE). We use the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 0.0007 for the four-wavelength model and of 0.0002 for the wide-spectrum setting. Furthermore, we employ cosine annealing (Loshchilov and Hutter, 2016) of the learning rate with $T_{max} = 1915$ (1921) and $eta_{min} = 7.8 \times 10^{-5}$ (4.9×10^{-6}) (wide-spectrum setting in parentheses). Lastly, we regularize via weight decay with a coefficient of 2.8×10^{-8} (2.1×10^{-9}). The best pretrained model is selected using the lowest validation loss, where the validation set is drawn randomly from the PPGen simulator.

Misspecification learning details (used in subsection 4.2). We train the misspecification correction layer q_ψ for 500 epochs, while the NPE model is fixed with the pretrained model described above. Each epoch consists of 100 iterations with a batch size of 200 samples (before splitting off 20% for validation).

Algorithm 3 Neural network architecture for four-wavelength setting.

```

1: function CONVBLOCK( $x, C_{in}, C_{out}$ )
2:    $y \leftarrow \text{Conv1d}(x, C_{in}, C_{out}, \text{kernel} = 3, \text{padding} = 1)$ 
3:    $y \leftarrow \text{ReLU}(y)$ 
4:    $y \leftarrow \text{Conv1d}(y, C_{out}, C_{out}, \text{kernel} = 3, \text{padding} = 1)$ 
5:    $y \leftarrow \text{ReLU}(y)$ 
6:   return  $y$ 
7: end function

8: function FORWARD( $x \in \mathbb{R}^{N \times 4 \times 4 \cdot 3 \times 64}$ )
9:    $e_0 \leftarrow \text{Flatten}(x, 1, 2)$                                  $\triangleright$  Flatten wavelength and channel
                                                                $\triangleright$  Encoder path

10:   $e_1 \leftarrow \text{ConvBlock}(e_0, 48, 32)$ 
11:   $e_2 \leftarrow \text{ConvBlock}(\text{MaxPool1d}(e_1), 32, 32)$ 
12:   $e_3 \leftarrow \text{ConvBlock}(\text{MaxPool1d}(e_2), 32, 16)$ 
13:   $e_4 \leftarrow \text{ConvBlock}(\text{MaxPool1d}(e_3), 16, 8)$                                  $\triangleright$  Bottleneck

14:   $b \leftarrow \text{ConvBlock}(\text{MaxPool1d}(e_4), 8, 2 \cdot 8)$                                  $\triangleright$  Waveform decoder path with skip connections

15:   $d_1 \leftarrow \text{ConvTranspose1d}(b, 2 \cdot 8, 16, \text{kernel} = 2, \text{stride} = 2)$ 
16:   $d_1 \leftarrow \text{Concatenate}(d_1, e_4)$                                  $\triangleright$  Skip connection
17:   $d_1 \leftarrow \text{ConvBlock}(d_1, 8 + 16, 16)$ 
18:   $d_2 \leftarrow \text{ConvTranspose1d}(d_1, 16, 32, \text{kernel} = 2, \text{stride} = 2)$ 
19:   $d_2 \leftarrow \text{Concatenate}(d_2, e_3)$                                  $\triangleright$  Skip connection
20:   $d_2 \leftarrow \text{ConvBlock}(d_2, 16 + 32, 32)$ 
21:   $d_3 \leftarrow \text{ConvTranspose1d}(d_2, 32, 32, \text{kernel} = 2, \text{stride} = 2)$ 
22:   $d_3 \leftarrow \text{Concatenate}(d_3, e_2)$                                  $\triangleright$  Skip connection
23:   $d_3 \leftarrow \text{ConvBlock}(d_3, 32 + 32, 32)$ 
24:   $d_4 \leftarrow \text{ConvTranspose1d}(d_3, 32, 32, \text{kernel} = 2, \text{stride} = 2)$ 
25:   $d_4 \leftarrow \text{Concatenate}(d_4, e_1)$                                  $\triangleright$  Skip connection
26:   $d_4 \leftarrow \text{ConvBlock}(d_4, 32 + 32, 32)$ 
27:   $d_5 \leftarrow \text{GaussianSmoothing}(d_4, \text{sigma} = 5.7, \text{kernel} = 19)$                                  $\triangleright$  Inference head for dynamic parameters

28:   $p_1 \leftarrow \text{Conv1d}(d_5, 32, 32, \text{kernel} = 5, \text{pad} = 4, \text{dil} = 2)$ 
29:   $p_2 \leftarrow \text{Conv1d}(p_1, 32, 2, \text{kernel} = 5, \text{pad} = 2)$                                  $\triangleright$  Inference head for static parameters

30:   $s_1 \leftarrow \text{Concatenate}(e_1, e_2, e_3, e_4, b)$ 
31:   $s_2 \leftarrow \text{MeanPool1D}(s_1)$                                  $\triangleright$  Skip connections
32:   $s_3 \leftarrow \text{ReLU}(\text{LinearLayer}(s_2, 104, 600))$ 
33:   $s_4 \leftarrow \text{ReLU}(\text{LinearLayer}(s_3, 600, 250))$ 
34:   $s_5 \leftarrow \text{ReLU}(\text{LinearLayer}(s_4, 250, 125))$ 
35:   $s_6 \leftarrow \text{LinearLayer}(s_5, 125, 9)$                                  $\triangleright$  Time averaging

36: end function

```

Algorithm 4 Neural network architecture for wide-spectrum setting.

```

1: function CONVBLOCK( $x, C_{in}, C_{out}$ )
2:    $y \leftarrow \text{Conv1d}(x, C_{in}, C_{out}, \text{kernel} = 3, \text{padding} = 1)$ 
3:    $y \leftarrow \text{ReLU}(y)$ 
4:    $y \leftarrow \text{Conv1d}(y, C_{out}, C_{out}, \text{kernel} = 3, \text{padding} = 1)$ 
5:    $y \leftarrow \text{ReLU}(y)$ 
6:   return  $y$ 
7: end function

8: function FORWARD( $x \in \mathbb{R}^{N \times 531 \times 4 \cdot 3 \times 64}$ )  

9:    $w_1 \leftarrow \text{Flatten}(x, 1, 2)$                                  $\triangleright$  Obtain embedding of wavelength and channel dimension
10:   $w_2 \leftarrow \text{ReLU}(\text{LinearLayer}(w_1, 531 \cdot 4 \cdot 3, 1000))$      $\triangleright$  Flatten wavelength and channel
11:   $w_3 \leftarrow \text{ReLU}(\text{LinearLayer}(w_2, 1000, 1000))$        $\triangleright$  Linear layer separate across time
12:   $w_4 \leftarrow \text{ReLU}(\text{LinearLayer}(w_3, 1000, 1000))$        $\triangleright$  Linear layer separate across time
13:   $w_5 \leftarrow \text{LinearLayer}(w_3, 1000, 200)$            $\triangleright$  Linear layer separate across time  

14:   $e_1 \leftarrow \text{ConvBlock}(w_5, 200, 32)$                                  $\triangleright$  Encoder path
15:   $e_2 \leftarrow \text{ConvBlock}(\text{MaxPool1d}(e_1), 32, 64)$ 
16:   $e_3 \leftarrow \text{ConvBlock}(\text{MaxPool1d}(e_2), 64, 64)$ 
17:   $e_4 \leftarrow \text{ConvBlock}(\text{MaxPool1d}(e_3), 64, 64)$                                  $\triangleright$  Bottleneck
18:   $b \leftarrow \text{ConvBlock}(\text{MaxPool1d}(e_4), 64, 2 \cdot 64)$   

19:   $d_1 \leftarrow \text{ConvTranspose1d}(b, 2 \cdot 64, 128, \text{kernel} = 2, \text{stride} = 2)$        $\triangleright$  Waveform decoder path with skip connections
20:   $d_1 \leftarrow \text{Concatenate}(d_1, e_4)$                                  $\triangleright$  Skip connection
21:   $d_1 \leftarrow \text{ConvBlock}(d_1, 64 + 128, 128)$ 
22:   $d_2 \leftarrow \text{ConvTranspose1d}(d_1, 128, 64, \text{kernel} = 2, \text{stride} = 2)$ 
23:   $d_2 \leftarrow \text{Concatenate}(d_2, e_3)$                                  $\triangleright$  Skip connection
24:   $d_2 \leftarrow \text{ConvBlock}(d_2, 64 + 64, 64)$ 
25:   $d_3 \leftarrow \text{ConvTranspose1d}(d_2, 64, 64, \text{kernel} = 2, \text{stride} = 2)$ 
26:   $d_3 \leftarrow \text{Concatenate}(d_3, e_2)$                                  $\triangleright$  Skip connection
27:   $d_3 \leftarrow \text{ConvBlock}(d_3, 64 + 64, 64)$ 
28:   $d_4 \leftarrow \text{ConvTranspose1d}(d_3, 64, 32, \text{kernel} = 2, \text{stride} = 2)$ 
29:   $d_4 \leftarrow \text{Concatenate}(d_4, e_1)$                                  $\triangleright$  Skip connection
30:   $d_4 \leftarrow \text{ConvBlock}(d_4, 32 + 32, 32)$                                  $\triangleright$  Inference head for dynamic parameters
31:   $p_1 \leftarrow \text{Conv1d}(d_4, 32, 32, \text{kernel} = 5, \text{pad} = 4, \text{dil} = 2)$ 
32:   $p_2 \leftarrow \text{Conv1d}(p_1, 32, 2, \text{kernel} = 5, \text{pad} = 2)$   

33:   $s_1 \leftarrow \text{Concatenate}(e_1, e_2, e_3, e_4, b)$                                  $\triangleright$  Inference head for static parameters
34:   $s_2 \leftarrow \text{MeanPool1D}(s_1)$                                  $\triangleright$  Skip connections
35:   $s_3 \leftarrow \text{ReLU}(\text{LinearLayer}(s_2, 352, 600))$        $\triangleright$  Time averaging
36:   $s_4 \leftarrow \text{ReLU}(\text{LinearLayer}(s_3, 600, 250))$ 
37:   $s_5 \leftarrow \text{ReLU}(\text{LinearLayer}(s_4, 250, 125))$ 
38:   $s_6 \leftarrow \text{LinearLayer}(s_5, 125, 9)$   

39: end function

```

In contrast to the pretraining stage, we fix the dataset to mimic a fixed sized real-world dataset. The loss function is a combination of a MSE loss on DC features with a coefficient of 1, a MSE loss on AC with a coefficient of 0.1, and a MSE loss on normalized AC features with a coefficient of 0.1 (see subsection C.2 for a discussion of these common PPG features). To balance these different loss components more easily, we normalize each loss component with its value at initialization. This ensures that the coefficients at initialization directly reflect the weighting of each loss component. As before, we use the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 0.0001. We regularize via weight decay with a coefficient of 2.8×10^{-8} (2.1×10^{-9}). The best model is selected using the lowest MAE loss for DC reconstruction on the validation set, with a train/val split of 80/20.

Real-only baseline training details (used in subsection 4.2). For the Real-only baseline, we train both NPE model q_ϕ and misspecification correction model q_ψ jointly using the same losses as during misspecification learning (1.0 for DC, 0.1 for AC, and 0.1 for normalized AC). In the four-wavelength setting we train the model for 2000 epochs and in the wide-spectrum setting for 200 epochs. As during misspecification learning, each epoch consists of 100 iterations with a batch size of 200 samples (before splitting off 20% for validation). The other training hyperparameters are the same as in the pretraining setting for the NPE. The best model is selected using the lowest MAE loss for DC reconstruction on the validation set, with a train/val split of 80/20.

Sim-only baseline training details (used in subsection 4.2). This baseline consists of only training the NPE model. For this baseline, we can directly use the pretrained models.

D.2. A detailed view of inference results using the four-wavelength PPG sensor

In Figure 8 we report inference results on static biophysical parameters using a four-wavelength PPG sensor without model misspecification, at medium noise level.

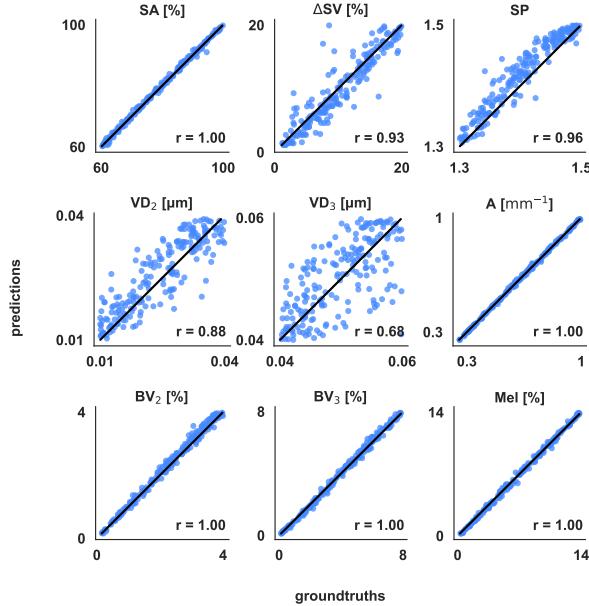


Figure 8: Inference of static biophysical parameters using a four-wavelength PPG sensor without misspecification and under medium noise level.

D.3. Extended results for inference under misspecification

In [Figure 9](#) we provide a breakdown of inference results for each parameter under all misspecification settings for the four-wavelength sensor. In [Figure 10](#) we show the results for the wide-spectrum sensor. The aggregated results for the wide-spectrum sensor can be found in [Table 6](#).

Table 6: Wide-spectrum setting: Parameter inference under several misspecifications, as measured by Pearson correlation and mean absolute percentage error (MAPE) averaged over all biophysical parameters. For each result we report \pm the standard deviation over multiple random seeds.

Misspec.	Method	Correlation	MAPE
<i>None</i>	HAI	0.99 ± 0.00	1.0 ± 0.0
	Sim-only	0.99 ± 0.00	1.0 ± 0.0
	Real-only	0.72 ± 0.01	13.9 ± 0.8
<i>Noise</i>	HAI	0.97 ± 0.00	2.4 ± 0.1
	Sim-only	0.97 ± 0.01	2.2 ± 0.1
	Real-only	0.72 ± 0.01	14.0 ± 0.9
<i>Sensor</i>	HAI	0.99 ± 0.00	1.5 ± 0.1
	Sim-only	0.94 ± 0.00	3.8 ± 0.0
	Real-only	0.71 ± 0.00	15.7 ± 0.8
<i>Skin</i>	HAI	0.86 ± 0.00	9.6 ± 0.3
	Sim-only	0.86 ± 0.00	9.0 ± 1.2
	Real-only	0.72 ± 0.01	16.5 ± 0.8
<i>Combined</i>	HAI	0.84 ± 0.00	12.6 ± 0.3
	Sim-only	0.84 ± 0.00	11.2 ± 1.4
	Real-only	0.71 ± 0.00	15.2 ± 0.2

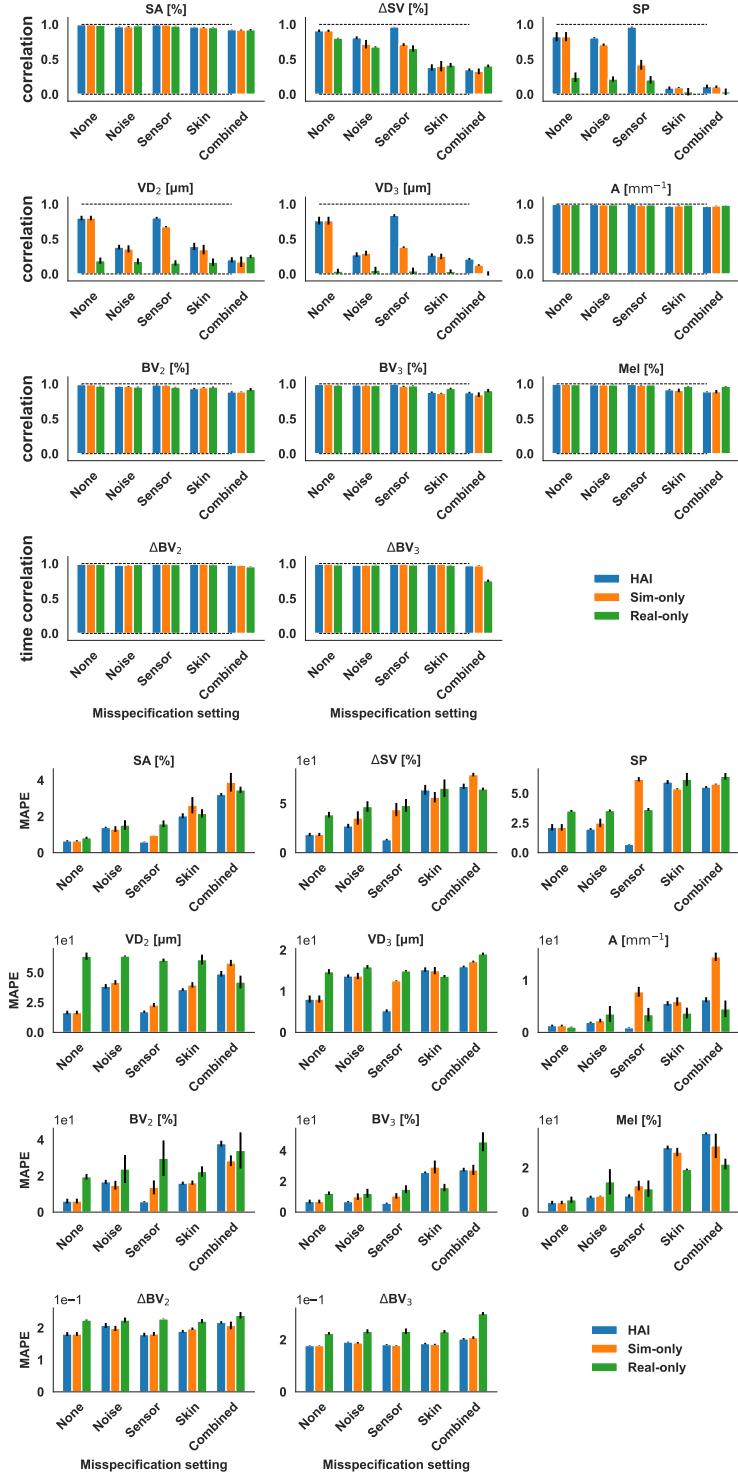


Figure 9: Four-wavelength PPG sensor: Tissue property inference under model misspecification. For each physiological parameter of interest, we report the Pearson correlation coefficient (upper panel) or mean-absolute percentage error (MAPE) (lower panel) between groundtruth and inferred parameters. For dynamic properties like systolic blood fractions, we report averages of across-time correlation coefficients.

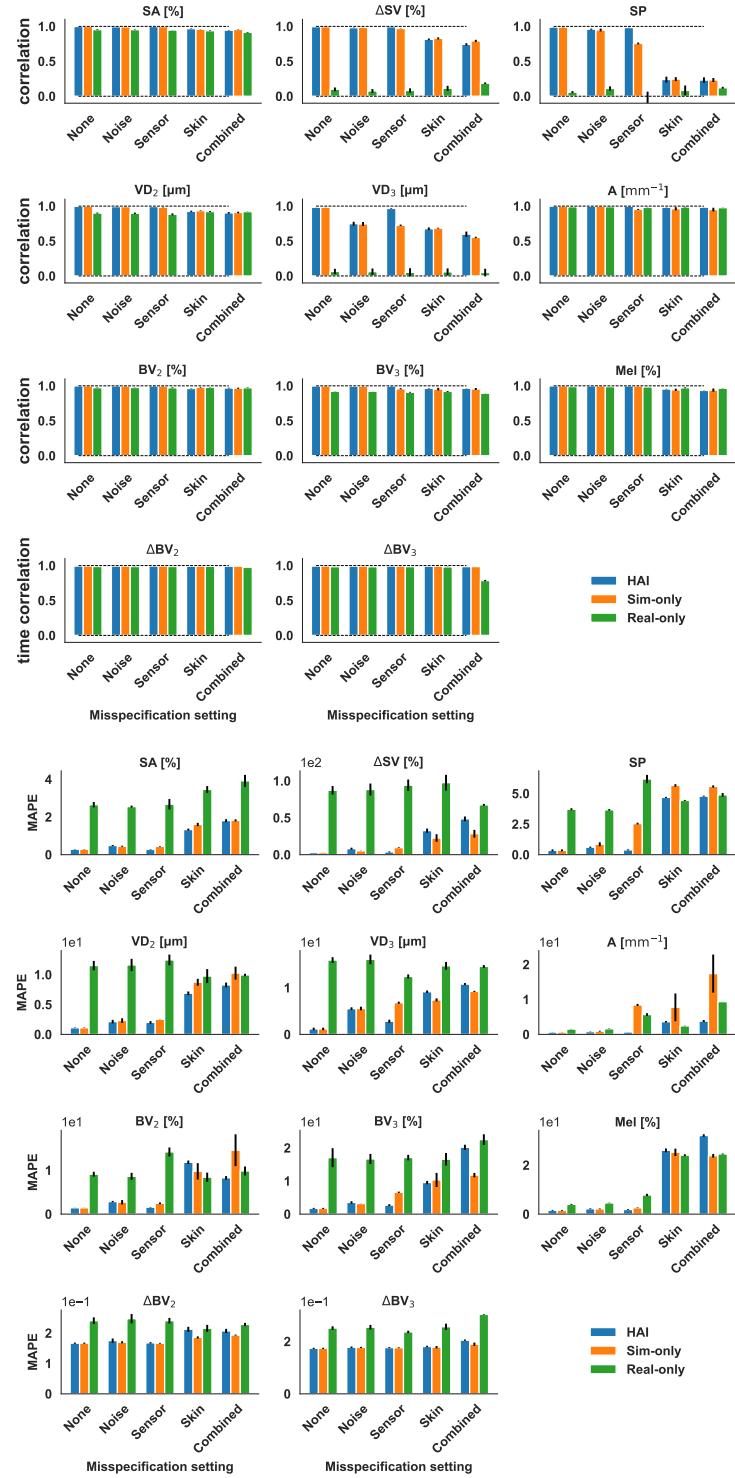


Figure 10: Wide-spectrum setting: Tissue property inference under model misspecification. For each physiological parameter of interest, we report the Pearson correlation coefficient (upper panel) or mean-absolute percentage error (MAPE) (lower panel) between groundtruth and inferred parameters. For dynamic properties like systolic blood fractions, we report averages of across-time correlation coefficients.