Meta-Learning in Biologically Informed Recurrent Neural Networks for Personalized System Identification of Glucose–Insulin Dynamics

Stefano De Carli, Nicola Licini, Francesco Corrini, Davide Previtali, Chiara Toffanin, *Member, IEEE*, Fabio Previdi, *Member, IEEE*, and Antonio Ferramosca, *Senior Member, IEEE*

***Abstract*—The management of type 1 diabetes mellitus (T1DM) represents a major challenge, as it requires continuous regulation of blood glucose (BG) levels. Automated systems, such as the artificial pancreas (AP), address this need by relying on accurate patient-specific models. Linear models are widely adopted in this context due to their simplicity and suitability for control design. However, their linear structure struggles to capture the complex dynamics of the glucose–insulin system, leading to degraded performance. This issue is further accentuated by the significant interpatient variability in glucose–insulin dynamics. A valid alternative to linear models is represented by recurrent neural networks (RNNs), which present higher expressivity while maintaining a level of complexity suitable for control purposes. However, such networks might lead to overfitting whenever the training dataset is scarce. In this work, we tackle this problem by embedding prior physiological knowledge about the glucose–insulin relation in the training procedure of RNNs, following the framework of knowledge-guided learning (KGL). Furthermore, we employ model-agnostic meta-learning (MAML) to deal with interpatient variability. A systematic results analysis is conducted using the UVA/Padova simulator to show that the proposed methodologies increase identification performance while maintaining an affordable level of complexity.**

***Index Terms*—Knowledge-guided machine learning, meta-learning, neural networks, system identification, type 1 diabetes.**

I. Introduction

TYPE 1 diabetes mellitus (T1DM) is a chronic autoimmune disorder that disrupts the ability of the human body to maintain glucose homeostasis. Due to the high prevalence of the disease, affecting an estimated 9.5 million people worldwide in 2025 [1], the artificial pancreas (AP) has become a major research topic in control engineering [2], [3]. An AP system combines devices that continuously monitor the patient's blood glucose (BG) levels and automatically deliver insulin to prevent dangerous conditions such as hyperglycemia (BG > 180 mg/dL) and hypoglycemia (BG < 70 mg/dL) [4]. Its core component is a control algorithm that computes, in real time, the appropriate insulin dose using BG measurements and supplementary information provided by the patient (e.g., clinical parameters and meal-related information). Model-based strategies, particularly model predictive control (MPC), are widely studied in the literature for this purpose [5], [6], [7], [8], [9], [10]. MPC is especially attractive because it can anticipate undesired glucose excursions and compute insulin dosages that satisfy constraints (i.e., nonnegativity and maximum delivery limits). Its effectiveness, however, strongly depends on the accuracy of the underlying model. Although linear physiological models have been proposed [11], motivated by the associated simple control scheme, glucose–insulin dynamics are inherently complex and nonlinear [12], making linear models insufficient for high-quality control. A further challenge is the substantial variability between patients, known as interpatient variability, as individuals exhibit different physiological responses to insulin. This problem is exacerbated by the limited amount of patient-specific data. The common approach is to train a personalized model for each patient using patient-specific data [3], [13], [14]. However, even if each patient's responses differ, they all share similar underlying dynamics, suggesting that leveraging both interpatient and intrapatient data may lead to better identification results. In this article, we will refer to the challenges reported above as follows.

(C1) Training expressive models of glucose–insulin dynamics with a suitable complexity for control purposes.

(C2) Dealing with interpatient variability.

A. Related Works

Concerning Challenge I, advanced identification techniques have been explored, such as transformers [15], bidirectional recurrent neural networks (RNNs) [14], and attention-based deep learning architectures [13]. However, the complexity of

Received 23 December 2025; revised 23 December 2025 and 19 June 2026; accepted 27 June 2026. This work was supported by the National Plan for National Recovery and Resilience Plan (NRRP) Complementary Investments [Piano Nazionale Complementare (PNC), established with the decree-law May 6, 2021, n. 59, converted by law n. 101 of 2021] in the call for the funding of Research Initiatives for Technologies and Innovative Trajectories in the Health and Care Sectors (Directorial Decree n. 931 of 06-06-2022)—AdvaNced Technologies for Human-CentrEd Medicine (ANTHEM) under Project PNC0000003. Recommended by Associate Editor J. Scott. *(Corresponding author: Stefano De Carli.)*

Stefano De Carli, Nicola Licini, Francesco Corrini, Davide Previtali, Fabio Previdi, and Antonio Ferramosca are with the Department of Management, Information and Production Engineering, University of Bergamo, 24044 Dalmine, Italy (e-mail: stefano.decarli@unibg.it; nicola.licini@unibg.it; francesco.corrini@unibg.it; davide.previtali@unibg.it; fabio.previdi@unibg.it; antonio.ferramosca@unibg.it).

Chiara Toffanin is with the Department of Electrical, Computer and Biomedical Engineering, University of Pavia, 27100 Pavia, Italy (e-mail: chiara.toffanin@unipv.it).

Digital Object Identifier 10.1109/TCST.2026.3709082

those models is too high to be implemented in an MPC scheme since the large number of parameters and states would substantially increase the computational complexity of the control optimization problem. A class of models with high expressiveness but complexity suited for control is RNNs, in particular, long short-term memory (LSTM) [16] and gated recurrent unit (GRU) [17] networks. While these networks can approximate any dynamical system, they might overfit the training dataset rather than learning the dynamics of the system itself, especially when available data is scarce [18, Ch. 15]. Although this problem can be tackled using common regularization techniques, in recent years, the framework of knowledge-guided learning (KGL) [19] has emerged. In KGL, physical information of the system is embedded into the model training workflow by designing a specific loss function that embeds physical information, inducing the learned network to behave according to prior knowledge of the system. A notable example of KGL is physics-informed neural networks (PINNs) [20], in which a regularization term in the loss function penalizes the mismatch between the predictions of the network and the output of a simpler physics-based model of the system. In this way, the RNN not only seeks to fit the available data but also incorporates prior knowledge of the system's dynamics.

Concerning Challenge II, meta-learning [21] offers a promising strategy for handling classes of similar dynamical systems. A particular meta-learning framework is the model-agnostic meta-learning (MAML) framework [22], which is similar to a fine-tuning procedure. MAML consists of a two-step approach: first, a so-called *meta-model* is trained using data from several systems of the same class. Then, the meta-model is adapted to each system with its specific data. In our case, each patient can be seen as a specific system of the same class. This allows us to train a population-level meta-model using data from the entire cohort, and then obtain a patient-tailored model through a fine-tuning procedure. This process effectively links population dynamics and individual physiological traits, such as patient-specific insulin sensitivity. MAML has already been applied to model glucose–insulin dynamics for T1DM patients [13], [14]. However, the models proposed in [13] and [14] are too complex or incompatible to be used for control purposes, not fulfilling Challenge I.

Recent studies have also investigated the integration of gated recurrent architectures within optimization-based control frameworks [23], showing that such models can be effectively embedded in MPC schemes. However, practical deployment in this context requires a careful tradeoff between model expressiveness and computational tractability. Motivated by this perspective, the present work addresses this issue at the identification level by focusing on recurrent architectures whose complexity remains compatible with MPC integration, while leaving the explicit controller design for future research.

B. Contributions

Inspired by other works in which PINNs are employed in biological applications [24], [25], we address Challenge I by developing a custom loss function for the training procedure of RNNs. The proposed loss function embeds prior biological information about glucose–insulin dynamics, which is represented by the physiological model in [11]. To directly address Challenge II, we apply MAML to GRU and LSTM networks to identify the glucose–insulin dynamics for T1DM patients within the KGL framework. The meta-learning stage identifies a robust RNN parameter initialization that serves as a physiological prior, which is then rapidly adapted to the individual patient using limited patient-specific data. The proposed methodology is validated across a broad cohort of in-silico patients of the FDA-accepted UVA/Padova T1DM simulator [26], the only simulator accepted as a substitute for preclinical animal trials for insulin therapies. The version employed in this study extends the original single-meal formulation [27] by incorporating circadian variability in insulin sensitivity [28], thereby enabling the simulation of complete 24-h glycemic profiles. We jointly analyze the impact of both the incorporation of biological information and MAML to understand if the proposed methodologies significantly contribute to increasing identification performance.

C. Article Organization

Section II describes the linear biological model used in the loss function and introduces core background on gated RNNs (LSTM and GRU networks). Section III presents the biologically informed RNNs, their training procedure, and MAML applied to RNNs. Section IV details the systematic evaluation and analysis of 40 model configurations across 100 in-silico patients of the UVA/Padova T1DM simulator. Section V summarizes the findings and outlines future research directions.

D. Notation

Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, and $\mathbb{N}$ be the sets of real, nonnegative real, positive real, and natural numbers (including zero), respectively. For a set $\mathcal{S}$, $|\mathcal{S}|$ denotes its cardinality. We denote a subset of $n \leq |\mathcal{S}|$ distinct elements chosen uniformly at random from $\mathcal{S}$ as $\mathcal{S}' = \text{rand}(\mathcal{S}, n)$. Vectors are denoted by boldface. For $n, m \in \mathbb{N}$, $\mathbb{R}^n$ represents the space of n-dimensional real column vectors, and $\mathbb{R}^{n\times m}$ represents the space of $n \times m$ real matrices. For a vector $\boldsymbol{v} \in \mathbb{R}^n$, $\|\boldsymbol{v}\|_2$ denotes its 2-norm. $\mathrm{I}^{n\times n}$ denotes the n-dimensional identity matrix. For a scalar $g \in \mathbb{R}$, $\boldsymbol{g}_n \in \mathbb{R}^n$ denotes the vector with all entries equal to g. The Hadamard (elementwise) product is denoted by $\circ$. For a scalar function $f : \mathbb{R} \to \mathbb{R}$, its bold counterpart $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ denotes the componentwise application, i.e., $\boldsymbol{f}(\boldsymbol{v}) = [f(v_1), \ldots, f(v_n)]^\top$ for any $\boldsymbol{v} = [v_1, \ldots, v_n] \in \mathbb{R}^n$. We define the sigmoid function as $\sigma(x) = (1 + e^{-x})^{-1}$ and the hyperbolic tangent function as $\tanh(x) = (e^x - e^{-x})(e^x + e^{-x})^{-1}$, both for any $x \in \mathbb{R}$. The ceiling function $\lceil x \rceil$ maps $x \in \mathbb{R}$ to the smallest integer greater than or equal to x. Time-dependent variables are defined as follows. A continuous-time signal $s : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is denoted as $s(t)$, where $t \in \mathbb{R}_{\geq 0}$ [s] is the time variable, and $\dot{s}(t)$ represents its time derivative. Discrete-time signals s_k are obtained via sampling as $s_k = s(kT)$, where $T \in \mathbb{R}_{>0}$ [s] is the sampling time, and $k \in \mathbb{N}$ is the discrete time step. Given a signal vector $\boldsymbol{x}_k = [x_{1,k}, \ldots, x_{n,k}]^\top \in \mathbb{R}^n$, we define the subvector $\boldsymbol{x}_{r:p,k} = [x_{r,k}, \ldots, x_{p,k}]^\top \in \mathbb{R}^{p-r+1}$, where $r, p, n \in \mathbb{N}$ and $1 \leq r \leq p \leq n$.

II. Preliminaries

In this section, following the problem statement, we detail the linear biological model that describes glucose–insulin dynamics. This model provides the physiological insights necessary for understanding the biologically informed RNNs (BI-RNNs) in Section III-A. We then present the RNN architectures and their training methodology employed in this study. We consider two types of RNNs: LSTM and GRU networks. Both are introduced as multilayer structures.

A. Problem Statement

We consider the problem of modeling and identifying a discrete-time multiple-input multiple-output (MIMO) dynamical system with inputs $\boldsymbol{u}_k \in \mathbb{R}^{n_u}, n_u \in \mathbb{N}$, and outputs $\boldsymbol{y}_k \in \mathbb{R}^{n_y}, n_y \in \mathbb{N}$. Specifically, we are interested in modeling the glucose–insulin dynamics of the T1DM patient system, which has $n_u = 2$ inputs $\boldsymbol{u}_k = [\iota_k, \mu_k]^\top$, i.e., the exogenous insulin delivery $\iota_k \in \mathbb{R}_{\geq 0}$ [U][1] and carbohydrate intake $\mu_k \in \mathbb{R}_{\geq 0}$ [g], and $n_y = 1$ measurable output $y_k = \gamma_k$, namely the BG concentration $\gamma_k \in \mathbb{R}_{\geq 0}$ [mg/dL].

B. Linear Biological Model

The mathematical model employed for the custom loss function of the RNNs of this work is based on the proposal in [11], formulated as a continuous-time state-space representation

$$\dot{\boldsymbol{s}}(t) = A\boldsymbol{s}(t) + B\boldsymbol{u}(t) + E, \quad \boldsymbol{s}(0) = \boldsymbol{s}_0. \tag{1}$$

The state vector $\boldsymbol{s} = [s_1, s_2, s_3, s_4, s_5]^\top \in \mathbb{R}^5_{\geq 0}$ comprises five physiological variables: BG concentration s_1 [mg/dL] (i.e., the measured output γ), insulin delivery rate in plasma s_2 [U/min], subcutaneous insulin delivery rate s_3 [U/min], rate of carbohydrate absorption from the gut s_4 [g/min], and glucose delivery rate from the stomach s_5 [g/min]. The model is initialized at $\boldsymbol{s}(0) = \boldsymbol{s}_0$, i.e., at the equilibrium state $\boldsymbol{s}_0 = [G_\mathrm{b}, I_\mathrm{b}, I_\mathrm{b}, 0, 0]^\top$ with $G_\mathrm{b} \in \mathbb{R}_{\geq 0}$ and $I_\mathrm{b} \in \mathbb{R}_{\geq 0}$ representing the glucose and insulin values at *fasting* conditions, respectively (i.e., at *basal* condition). The model is parametrized by six physiological parameters $\mathcal{P} = \{p_0, \ldots, p_5\}$, as described in Table I, devoted to modulating the dynamics of the states specifically for each patient. The model in (1) allows the estimation of two clinically relevant metrics

$$\begin{aligned} \mathrm{IOB}(t) &= p_4 (s_2(t) + s_3(t)) \\ \mathrm{Ra}(t) &= p_3 s_4(t) \end{aligned}$$

where IOB is the insulin-on-board [U], i.e., the amount of insulin still active in the body due to previous injections, while Ra is the glucose entry rate in plasma due to meals [mg/(dL·min)]. We discretize the continuous-time model in (1) using the forward Euler method with sampling time $T \in \mathbb{R}_{>0}$, obtaining $A_\mathrm{d} = TA + \mathrm{I}^{5\times 5}$, $B_\mathrm{d} = TB$, and $E_\mathrm{d} = TE$. The resulting discrete-time model is expressed as

$$\boldsymbol{s}_{k+1} = A_\mathrm{d}\boldsymbol{s}_k + B_\mathrm{d}\boldsymbol{u}_k + E_\mathrm{d}. \tag{2}$$

[1]"U" (International Units of Insulin, IU) is an activity-based unit for insulin (standard in insulin pumps/pens). For reference, 1 U ≈ 34.7 μg ≈ 6 nmol of human insulin, although clinical dosing is defined in IU rather than mass.

TABLE I
Description of the Parameters of the Physiological Model

Parameter	Description
p_0 [mg/(dL · min)]	Endogenous glucose production
p_1 [1/min]	Glucose effectiveness
p_2 [mg/(dL · U)]	Insulin sensitivity
p_3 [mg/(dL · g)]	Carbohydrate factor
p_4 [min]	Insulin absorption time constant
p_5 [min]	Meals absorption time constant

We remark that the model in (1) is a simplified linear approximation of human glucose–insulin dynamics, and it was originally proposed by Ruan et al. [11]. In their work, the authors introduce a minimal yet physiologically grounded compartmental structure that preserves the dominant causal pathways of glycemic regulation. Despite its linear form, the model in (1) reflects key mechanisms found in other relevant (nonlinear) models from the literature (e.g., Bergman [12], Hovorka et al. [8], or Dalla Man et al. [29]), namely: 1) carbohydrates transit from the stomach to the gut before appearing as Ra in plasma and 2) exogenous insulin is absorbed through subcutaneous and plasma compartments, contributing to active insulin (IOB). The patient-specific parameters $\mathcal{P}$ modulate absorption rates and insulin effectiveness, enabling this reduced-order model to approximate interpatient variability reasonably well. The study in [11] has shown that such linearized dynamics, although simplified, capture the aggregate behavior of the underlying nonlinear physiology sufficiently well for control and forecasting tasks.

Importantly, in our work, the linear model is *not* intended to replace the expressive capacity of the neural network model proposed in Section III-A. Instead, it serves as a *mechanistic prior* within the training loss, providing a biologically consistent structure that complements data-driven learning. By encoding known causal relationships in the RNN training procedure (i.e., the meal increases Ra and so glucose rises; insulin increases IOB and provokes glucose lowering), the model in (1) regularizes the predictions toward physiologically plausible trajectories without imposing excessive modeling assumptions. This balance leverages the flexibility of deep recurrent networks while preventing physiologically inconsistent or overfitted behaviors, especially in scenarios with limited or noisy data. The choice of a linear model is deliberate: its analytical tractability, stability, and differentiability make it well-suited for integration within gradient-based RNN training pipelines. Although more detailed nonlinear glucose–insulin models (e.g., [29], [30]) could provide a richer physiological description, their adoption would introduce additional latent states, compartments, and patient-specific parameters, substantially increasing the complexity of the training problem. Given the long-term objective of employing the proposed framework in control-oriented applications, we opted for a prior that captures the dominant physiological causal pathways while remaining simple, differentiable, and computationally efficient. Thus, the adopted linear formulation represents an effective compromise for its role as a regularizing prior in the learning architecture that will be presented in Section III-B.

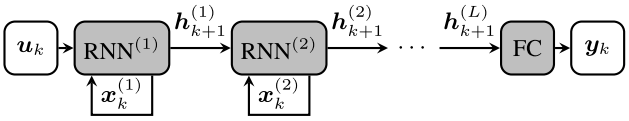

Fig. 1. Multilayer RNN architecture.

C. LSTM Network

LSTM networks, as introduced in [16], are a class of RNNs that employ gating mechanisms to retain long-term dependencies without running into vanishing/exploding gradient problems during training [18, Ch. 15]. They generally consist of $L \in \mathbb{N}$ stacked LSTM layers. Each layer $l \in \mathfrak{L} = \{1, \dots, L\}$ contains $n_h^{(l)} \in \mathbb{N}$ hidden units. The lth layer of an LSTM network amounts to a discrete-time nonlinear dynamical system in state-space form whose state vector $\boldsymbol{x}_k^{(l)} \in \mathbb{R}^{2n_h^{(l)}}$ is divided into two components: the cell state $\boldsymbol{c}_k^{(l)} \in \mathbb{R}^{n_h^{(l)}}$ and the hidden state $\boldsymbol{h}_k^{(l)} \in \mathbb{R}^{n_h^{(l)}}$, i.e., $\boldsymbol{x}_k^{(l)} = [\boldsymbol{c}_k^{(l)^\top}, \boldsymbol{h}_k^{(l)^\top}]^\top$. Four gating mechanisms (forget, input, output, and candidate memory) control the network information flow through each layer; we refer to [16] for the detailed gate equations. The LSTM network can be written in a state-space layered form as

$$\boldsymbol{h}_{k+1}^{(l)} = \text{LSTM}^{(l)}\left(\boldsymbol{x}_k^{(l)}, \tilde{\boldsymbol{u}}_k^{(l)}\right) \quad \forall l \in \mathfrak{L} \tag{3a}$$

$$\boldsymbol{y}_k = \text{FC}\left(\boldsymbol{h}_{k+1}^{(L)}\right) \tag{3b}$$

$$\tilde{\boldsymbol{u}}_k^{(l)} = \begin{cases} \boldsymbol{u}_k, & \text{if } l = 1 \\ \boldsymbol{h}_{k+1}^{(l-1)}, & \text{if } l \in \mathfrak{L} \setminus \{1\} \end{cases} \tag{3c}$$

where $\tilde{\boldsymbol{u}}_k^{(l)}$ is the lth layer input, (3a) represents the LSTM update for each layer, and (3b) is the fully connected (FC) mapping of the predicted output: $\boldsymbol{y}_k = W_y \boldsymbol{h}_{k+1}^{(L)} + \boldsymbol{b}_y$, with $W_y \in \mathbb{R}^{n_y \times n_h^{(L)}}$ and $\boldsymbol{b}_y \in \mathbb{R}^{n_y}$. In (3a), both the cell state $\boldsymbol{c}_k^{(l)}$ and the hidden state $\boldsymbol{h}_k^{(l)}$ remain hidden inside each LSTM layer as the state $\boldsymbol{x}_k^{(l)}$. The LSTM network architecture in (3) is a specific instance of the more general multilayer RNN architecture shown in Fig. 1, where RNNs may include LSTM blocks as components. The LSTM model in (3) relies on a set of tunable parameters θ comprising the input weights, recurrent weights, and bias vectors of each gating mechanism across all L layers (see [16]), together with the FC parameters $\{W_y, \boldsymbol{b}_y\}$. These parameters are optimized during the training procedure detailed in Section II-E. Note that structural choices, such as L and $n_h^{(l)}, l \in \mathfrak{L}$, are hyperparameters requiring appropriate tuning during the training phase.

D. GRU Network

GRU networks, as introduced in [17], are an LSTM network variant designed with fewer gating mechanisms (reset, update, and candidate hidden state); we refer to [17] for the detailed gate equations. Unlike LSTM layers, the state vector of a GRU layer $l \in \mathfrak{L}$ corresponds solely to its hidden state, i.e., $\boldsymbol{x}_k^{(l)} = \boldsymbol{h}_k^{(l)}$, resulting in $n_x = \sum_{l=1}^{L} n_h^{(l)}$ total states, which is half the size of the state vector in an equivalent LSTM network. The GRU network can be written in a layered form as

$$\boldsymbol{h}_{k+1}^{(l)} = \text{GRU}^{(l)}\left(\boldsymbol{x}_k^{(l)}, \tilde{\boldsymbol{u}}_k^{(l)}\right) \quad \forall l \in \mathfrak{L} \tag{4a}$$

$$\boldsymbol{y}_k = \text{FC}\left(\boldsymbol{h}_{k+1}^{(L)}\right) \tag{4b}$$

where $\tilde{\boldsymbol{u}}_k^{(l)}$ and the FC mapping are defined as in Section II-C. Similar to the LSTM model in (3), the GRU model relies on a set of tunable parameters θ comprising the input weights, recurrent weights, and bias vectors of each gating mechanism across all L layers, together with the FC parameters $\{W_y, \boldsymbol{b}_y\}$, estimated as detailed in Section II-E.

Remark 1: In this work, we employ single-layer RNN configurations ($L = 1$). Prior studies on glucose–insulin dynamics modeling [25], [31], [32], [33] suggest that additional layers do not significantly yield performance gains. Furthermore, the regularization inherent in our framework allows a shallow architecture to effectively capture nonlinear dynamics while mitigating the risk of overfitting [18, Ch. 13].

E. RNN Model Training

The RNN training procedure determines the optimal parameter set θ for either the LSTM or GRU architecture. We consider a dataset $\mathcal{D} = \{\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(V)}\}$ of $V \in \mathbb{N}$ input–output sequences $\mathcal{D}^{(v)} = \{(\boldsymbol{u}_k^{(v)}, \boldsymbol{y}_k^{(v)}) : k \in \{0, \dots, N^{(v)} - 1\}\}, v \in \{1, \dots, V\}, N^{(v)} \in \mathbb{N}$, which is partitioned into mutually exclusive training ($\mathcal{D}_{\text{tr}}$), validation ($\mathcal{D}_{\text{val}}$), and test ($\mathcal{D}_{\text{tst}}$) subsets for model estimation, hyperparameter tuning, and final performance evaluation, respectively [18, Ch. 13]. The parameters are estimated by minimizing the mean squared error (MSE) loss function $\mathcal{L}$

$$\begin{aligned} \mathcal{L}(\theta; \mathcal{D}_{\text{tr}}) &= \text{MSE}(\theta; \mathcal{D}_{\text{tr}}) \\ &= \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{\mathcal{D}^{(v)} \in \mathcal{D}_{\text{tr}}} \left[\frac{1}{N^{(v)}} \sum_{k=0}^{N^{(v)}-1} \left\| \boldsymbol{y}_k^{(v)} - \hat{\boldsymbol{y}}_k^{(v)}(\theta) \right\|_2^2 \right] \end{aligned} \tag{5}$$

where $\hat{\boldsymbol{y}}_k^{(v)}(\theta)$ is the model's prediction at the time step k.

The model parameters are initialized as θ_0 and iteratively updated using a gradient-based optimization procedure (e.g., Adam [18, Ch. 8]). Due to the small size of the datasets in this study, we perform full-batch updates, using the entire $\mathcal{D}_{\text{tr}}$ at each iteration. To prevent overfitting and enhance generalization, we incorporate dropout and early stopping [18, Ch. 13]; the latter monitors the validation MSE to select the parameter set $\theta^\star$ that achieves the best performance on $\mathcal{D}_{\text{val}}$.

III. BI-RNNs and Meta-Learning Framework

In this section, we present the novel contributions of this study, describing both the BI-RNN model and its training methodology, as well as the employed meta-learning framework.

A. Biologically Informed RNNs

The proposed BI-RNN architecture integrates domain-specific physiological knowledge directly into the learning process of RNNs [25]. The BI-RNN is designed as an augmented state estimator that jointly predicts both the glucose concentration and the internal states of the linear biological model introduced in Section II-B, leading to the vector to be predicted $\tilde{\boldsymbol{y}}_k = [y_k, \boldsymbol{s}_{2:5,k}^\top]^\top$, where we remind that $\boldsymbol{s}_{2:5,k}$

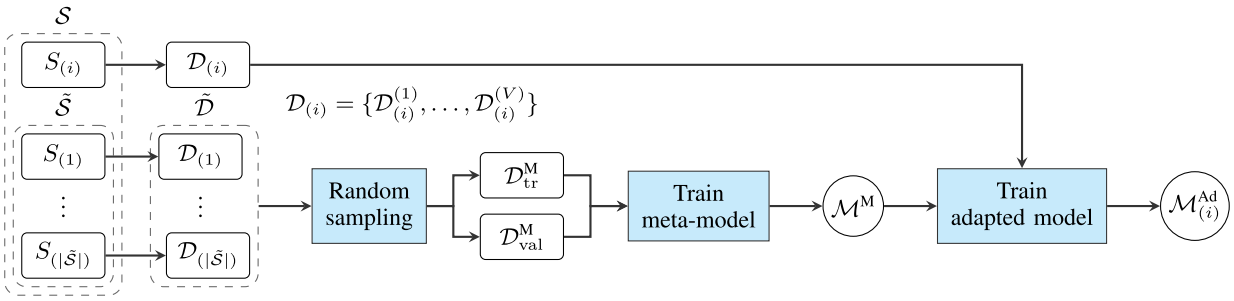

Fig. 2. Schematic of the proposed meta-learning approach for system identification. The full class of dynamical systems $\mathcal{S}$ is shown, highlighting the available finite subset $\tilde{\mathcal{S}}$ composed of $|\tilde{\mathcal{S}}|$ individual systems and their respective datasets. Meta-training ($\mathcal{D}_{\text{tr}}^{\text{M}}$) and meta-validation ($\mathcal{D}_{\text{val}}^{\text{M}}$) datasets are constructed by randomly sampling sequences from the datasets in $\tilde{\mathcal{D}}$, as described in (12), and used to train the meta-model $\mathcal{M}^{\text{M}}$ with parameters θ^{M} (see Section III-D). Once trained, these meta-parameters θ^{M} are used as initialization θ_0 for adapting the meta-model to a specific system $S_{(i)} \in \mathcal{S}$, obtaining the adapted model $\mathcal{M}_{(i)}^{\text{Ad}}$ by further training on its dataset $\mathcal{D}_{(i)}$.

are the states from the biological model in (2), whereas y_k is the measured BG concentration, which may differ from the state s_1. Architecturally, BI-RNNs consist of stacked RNNs, implementing either LSTM or GRU structures, followed by an FC layer mapping as in Fig. 1. To address Challenge I, we explicitly enforce nonnegativity constraints on the predicted physiological quantities. Thus, we write the BI-RNN model in layered form as

$$\boldsymbol{h}_{k+1}^{(l)} = \text{RNN}^{(l)}\left(\boldsymbol{x}_k^{(l)}, \tilde{\boldsymbol{u}}_k^{(l)}\right) \quad \forall l \in \mathfrak{L} \tag{6a}$$

$$\tilde{\boldsymbol{y}}_k = \mathbf{max}\left(\mathbf{0}_5, \text{FC}\left(\boldsymbol{h}_{k+1}^{(L)}\right)\right) \tag{6b}$$

where $\text{RNN}^{(l)}$ can either represent LSTM or GRU layers as in (3a) and (4a), respectively. Predicting the full state vector $\tilde{\boldsymbol{y}}_k$, rather than glucose alone, is a design choice strictly motivated by the BI-RNN training framework. These estimates are utilized to compute the residuals of the discrete-time governing (2), which are then minimized via an augmented loss function to promote dynamic consistency with the underlying biological principles. Consequently, the BI-RNN optimization objective replaces the purely data-driven error term in (5) with a composite loss function $\mathcal{L}(\theta; \mathcal{P}, \mathcal{D}_{\text{tr}})$, formulated as

$$\begin{aligned}\mathcal{L}(\theta; \mathcal{P}, \mathcal{D}_{\text{tr}}) = {} & \alpha_{\text{D}} \mathcal{L}^{\text{D}}(\theta; \mathcal{D}_{\text{tr}}) \\ & + \alpha_{\text{B}} \mathcal{L}^{\text{B}}(\theta; \mathcal{P}, \mathcal{D}_{\text{tr}}) \\ & + \alpha_{\text{Au}} \mathcal{L}^{\text{Au}}(\theta; \mathcal{D}_{\text{tr}})\end{aligned} \tag{7}$$

where the components, better detailed in Section III-B, are defined as follows:

(L1) $\mathcal{L}^{\text{D}}(\theta; \mathcal{D}_{\text{tr}})$ promotes the alignment between predicted and measured glucose regardless of inherent biological constraints. It amounts to the loss in (5).

(L2) $\mathcal{L}^{\text{B}}(\theta; \mathcal{P}, \mathcal{D}_{\text{tr}})$ embeds prior knowledge of the underlying biological dynamics from the model in Section II-B.

(L3) $\mathcal{L}^{\text{Au}}(\theta; \mathcal{D}_{\text{tr}})$ includes penalties designed to enforce boundary conditions and constraints on specific outputs.

In (7), $\alpha_{\text{D}}, \alpha_{\text{B}}, \alpha_{\text{Au}} \in \mathbb{R}_{\geq 0}$ are weighting factors for their respective loss components, treated as tunable hyperparameters of the BI-RNN. The joint minimization of these terms aids the BI-RNN to both accurately predict the observed outputs and behave consistently with the known biological principles of the system, directly complying with Challenge I. Incorporating $\mathcal{L}^{\text{B}}(\theta; \mathcal{P}, \mathcal{D}_{\text{tr}})$ and $\mathcal{L}^{\text{Au}}(\theta; \mathcal{D}_{\text{tr}})$ also serves as an implicit regularization form, enhancing the network's generalization ability.

B. BI-RNN Model Training

BI-RNN training follows the gradient-based optimization procedure described in Section II-E, but uses the augmented loss in (7) and requires an expanded dataset $\mathcal{D}^{(v)} = \{(\boldsymbol{u}_k^{(v)}, y_k^{(v)}, \boldsymbol{s}_k^{(v)}) : k \in \{0, \ldots, N^{(v)} - 1\}, v \in \{1, \ldots, V\}\}$, which includes the linear model physiological states. The data loss $\mathcal{L}^{\text{D}}(\theta; \mathcal{D}_{\text{tr}})$ in a is the MSE between the actual and predicted glucose

$$\begin{aligned}\mathcal{L}^{\text{D}}(\theta; \mathcal{D}_{\text{tr}}) &= \text{MSE}(\theta; \mathcal{D}_{\text{tr}}) \\ &= \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{\mathcal{D}^{(v)} \in \mathcal{D}_{\text{tr}}} \left[\frac{1}{N^{(v)}} \sum_{k=0}^{N^{(v)}-1} \left\| y_k^{(v)} - \hat{y}_k^{(v)}(\theta) \right\|_2^2 \right].\end{aligned} \tag{8}$$

The biological loss $\mathcal{L}^{\text{B}}(\theta; \mathcal{P}, \mathcal{D}_{\text{tr}})$ in b quantifies the one-step state prediction residual when BI-RNN outputs are substituted into the discrete-time linear biological model in (2), averaged across training sequences

$$\begin{aligned}\mathcal{L}^{\text{B}}(\theta; \mathcal{P}, \mathcal{D}_{\text{tr}}) &= \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{\mathcal{D}^{(v)} \in \mathcal{D}_{\text{tr}}} \ell^{\text{B},(v)}(\theta; \mathcal{P}, \mathcal{D}_{\text{tr}}) \\ \ell^{\text{B},(v)}(\theta; \mathcal{P}, \mathcal{D}_{\text{tr}}) &= \frac{1}{N^{(v)}} \sum_{k=0}^{N^{(v)}-2} \\ &\qquad \left\| A_{\text{d}} \hat{\tilde{\boldsymbol{y}}}_k^{(v)}(\theta) + B_{\text{d}} \boldsymbol{u}_k^{(v)} + E_{\text{d}} - \hat{\tilde{\boldsymbol{y}}}_{k+1}^{(v)}(\theta) \right\|_2^2.\end{aligned} \tag{9}$$

The corresponding patient-specific discrete-time state-space matrices A_{d} and B_{d} are obtained from the identified linear model parameters $\mathcal{P}$ described in Section II-B, estimated prior to BI-RNN training as proposed in [9].

Finally, to define the auxiliary loss $\mathcal{L}^{\text{Au}}(\theta; \mathcal{D}_{\text{tr}})$ in c, which enforces boundary conditions and constraints, we consider a subset of the training data indexed by the set

$$\tilde{\mathcal{N}}^{(v)} = \text{rand}\left(\left\{1, \ldots, N^{(v)} - 1\right\}, \left\lceil \xi N^{(v)} \right\rceil\right) \tag{10}$$

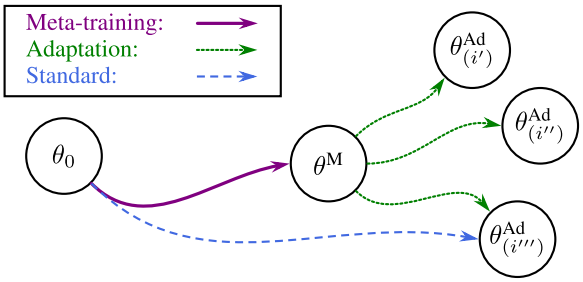

Fig. 3. Illustration of the MAML approach: starting from initial parameters θ_0, a meta-model is first trained through meta-training, obtaining its meta-parameters θ^{M}, and then further adapted for each ith system, learning a respective adapted model with parameters $\theta^{\mathrm{Ad}}_{(i)}$. The standard training approach, as described in Section II-E, is qualitatively depicted for comparison.

where $\xi \in [0, 1)$ is the fraction of time steps included for each sequence. Then, $\mathcal{L}^{\mathrm{Au}}(\theta; \mathcal{D}_{\mathrm{tr}})$ in c amounts to

$$\begin{aligned}
\mathcal{L}^{\mathrm{Au}}(\theta; \mathcal{D}_{\mathrm{tr}}) &= \frac{1}{2|\mathcal{D}_{\mathrm{tr}}|} \sum_{\mathcal{D}^{(v)} \in \mathcal{D}_{\mathrm{tr}}} \\
&\qquad \left(\ell_{\mathrm{s}}^{\mathrm{Au},(v)}(\theta; \mathcal{D}_{\mathrm{tr}}) + \ell_{0}^{\mathrm{Au},(v)}(\theta; \mathcal{D}_{\mathrm{tr}}) \right) \\
\ell_{\mathrm{s}}^{\mathrm{Au},(v)}(\theta; \mathcal{D}_{\mathrm{tr}}) &= \frac{1}{|\tilde{\mathcal{N}}^{(v)}|} \sum_{k \in \tilde{\mathcal{N}}^{(v)}} \left\| \boldsymbol{s}^{(v)}_{2:5,k} - \hat{\boldsymbol{s}}^{(v)}_{2:5,k}(\theta) \right\|_2^2 \\
\ell_{0}^{\mathrm{Au},(v)}(\theta; \mathcal{D}_{\mathrm{tr}}) &= \left\| \boldsymbol{s}_0 - \hat{\boldsymbol{y}}^{(v)}_0(\theta) \right\|_2^2
\end{aligned} \tag{11}$$

where $\ell_{\mathrm{s}}^{\mathrm{Au},(v)}$ quantifies the discrepancy between model states and network outputs over $\tilde{\mathcal{N}}^{(v)}$, and $\ell_{0}^{\mathrm{Au},(v)}$ captures the deviation from the known initial condition. The set $\tilde{\mathcal{N}}^{(v)}$ in (10) is generated at every $\mathcal{L}^{\mathrm{Au}}(\theta; \mathcal{D}_{\mathrm{tr}})$ computation. The stochastic sampling of $\tilde{\mathcal{N}}^{(v)}$ acts as a regularizer, preventing overfitting to the approximate linear model while preserving the capacity to learn data-driven nonlinearities.

C. Meta-Learning for Interpatient Variability

We adopt a meta-learning framework inspired by the MAML approach [22]. In particular, we employ a first-order approximation strategy [34] to address the interpatient variability and data scarcity prevalent in glucose–insulin dynamics modeling [refer Challenge I]. In this paradigm, a meta-model $\mathcal{M}^{\mathrm{M}}$ with parameters θ^{M} represents a class of dynamical systems $\mathcal{S}$ (e.g., glucose–insulin dynamics across different patients) that share underlying behaviors. During meta-training, we estimate θ^{M} using only a finite subset of available systems, $\tilde{\mathcal{S}} \subset \mathcal{S}$, as depicted in Fig. 2, and their collected datasets $\tilde{\mathcal{D}}$ ($\mathcal{D}_{(i)}$ for each system $S_{(i)} \in \tilde{\mathcal{S}}$). The resulting meta-model captures the dominant, shared dynamics present in $\tilde{\mathcal{S}}$, effectively learning the common underlying patient dynamics, approximating the behavior of the system class $\mathcal{S}$. The process of meta-training is further explained in Section III-D. The generalized meta-model $\mathcal{M}^{\mathrm{M}}$ can then be efficiently adapted to a particular system $S_{(i)} \in \mathcal{S}$ through an adaptation phase [22], [35], obtaining an adapted model $\mathcal{M}^{\mathrm{Ad}}_{(i)}$ with parameters $\theta^{\mathrm{Ad}}_{(i)}$. The approach to meta-training and adaptation is depicted in Fig. 2, while the fundamental idea of MAML is shown in Fig. 3. In this work, both $\mathcal{M}^{\mathrm{M}}$ and $\mathcal{M}^{\mathrm{Ad}}_{(i)}$ utilize the RNN and BI-RNN models.

Algorithm 1 Meta-Learning Procedure

Inputs: Available datasets $\tilde{\mathcal{D}}$, dataset $\mathcal{D}_{(i)}$ of the system to adapt to, number of meta-training sequences per system $\tilde{V}_{\mathrm{tr}}$, number of meta-validation sequences per system $\tilde{V}_{\mathrm{val}}$, initial parameters θ_0, training hyperparameters (for meta-training and adaptation).
Output: Best parameters of the adapted model $\theta^{\mathrm{Ad}\star}_{(i)}$.

▷ *Meta-Training Phase*
Initialize $\mathcal{D}^{\mathrm{M}}_{\mathrm{tr}} \leftarrow \emptyset$
Initialize $\mathcal{D}^{\mathrm{M}}_{\mathrm{val}} \leftarrow \emptyset$
for all $\mathcal{D}_{(j)} \in \tilde{\mathcal{D}}$ **do**
  Sample $\mathcal{D}^{\mathrm{M}}_{\mathrm{tr},(j)} \leftarrow \mathrm{rand}\left(\mathcal{D}_{(j)}, \tilde{V}_{\mathrm{tr}}\right)$ and $\mathcal{D}^{\mathrm{M}}_{\mathrm{val},(j)} \leftarrow \mathrm{rand}\left(\mathcal{D}_{(j)}, \tilde{V}_{\mathrm{val}}\right)$ such that $\mathcal{D}^{\mathrm{M}}_{\mathrm{tr},(j)} \cap \mathcal{D}^{\mathrm{M}}_{\mathrm{val},(j)} = \emptyset$ holds
  Join $\mathcal{D}^{\mathrm{M}}_{\mathrm{tr}} \leftarrow \mathcal{D}^{\mathrm{M}}_{\mathrm{tr}} \cup \mathcal{D}^{\mathrm{M}}_{\mathrm{tr},(j)}$ and $\mathcal{D}^{\mathrm{M}}_{\mathrm{val}} \leftarrow \mathcal{D}^{\mathrm{M}}_{\mathrm{val}} \cup \mathcal{D}^{\mathrm{M}}_{\mathrm{val},(j)}$
Train $\mathcal{M}^{\mathrm{M}}$ (Section II-E) with $\mathcal{D}^{\mathrm{M}}_{\mathrm{tr}}$, $\mathcal{D}^{\mathrm{M}}_{\mathrm{val}}$, θ_0 to get $\theta^{\mathrm{M}\star}$
▷ *Adaptation Phase for System* $S_{(i)}$
Get $\mathcal{D}^{(i)}_{\mathrm{tr}}$ and $\mathcal{D}^{(i)}_{\mathrm{val}}$ from $\mathcal{D}_{(i)}$
Adapt $\mathcal{M}^{\mathrm{M}}$ (Section II-E) with $\mathcal{D}^{(i)}_{\mathrm{tr}}$, $\mathcal{D}^{(i)}_{\mathrm{val}}$, and initial parameters $\theta_0 \leftarrow \theta^{\mathrm{M}\star}$ to get $\theta^{\mathrm{Ad}\star}_{(i)}$
return $\theta^{\mathrm{Ad}\star}_{(i)}$

D. Meta-Training and Adaptation

The employed meta-training procedure, as displayed in Fig. 2, follows the standard training procedure described in Section II-E, with the main distinction being the datasets construction. The meta-training ($\mathcal{D}^{\mathrm{M}}_{\mathrm{tr}}$) and meta-validation ($\mathcal{D}^{\mathrm{M}}_{\mathrm{val}}$) datasets are constructed by aggregating sequences randomly sampled from each available system dataset $\mathcal{D}_{(i)}$ (representing a T1DM patient)

$$\mathcal{D}^{\mathrm{M}}_{\mathrm{tr}} = \bigcup_{\mathcal{D}_{(i)} \in \tilde{\mathcal{D}}} \mathrm{rand}\left(\mathcal{D}_{(i)}, \tilde{V}_{\mathrm{tr}}\right) \tag{12a}$$

$$\mathcal{D}^{\mathrm{M}}_{\mathrm{val}} = \bigcup_{\mathcal{D}_{(i)} \in \tilde{\mathcal{D}}} \mathrm{rand}\left(\mathcal{D}_{(i)}, \tilde{V}_{\mathrm{val}}\right) \tag{12b}$$

where $\tilde{V}_{\mathrm{tr}} \in \mathbb{N}$ and $\tilde{V}_{\mathrm{val}} \in \mathbb{N}$ are the number of sequences sampled per system (set to be consistent across all systems), such that $\mathcal{D}^{\mathrm{M}}_{\mathrm{tr}}$ and $\mathcal{D}^{\mathrm{M}}_{\mathrm{val}}$ are mutually exclusive, and $\tilde{V}_{\mathrm{tr}} + \tilde{V}_{\mathrm{val}} < V_{(i)} = |\mathcal{D}_{(i)}|$. The meta-training dataset will then comprise $|\mathcal{D}^{\mathrm{M}}_{\mathrm{tr}}| = |\tilde{\mathcal{S}}| \cdot \tilde{V}_{\mathrm{tr}}$ sequences and the validation dataset of $|\mathcal{D}^{\mathrm{M}}_{\mathrm{val}}| = |\tilde{\mathcal{S}}| \cdot \tilde{V}_{\mathrm{val}}$ sequences. Once the meta-model $\mathcal{M}^{\mathrm{M}}$ is trained, we adapt it into $\mathcal{M}^{\mathrm{Ad}}_{(i)}$ for a particular system $S_{(i)}$ by reapplying the standard training procedure with the system's dataset $\mathcal{D}_{(i)}$, initializing the parameters θ_0 to θ^{M}. Algorithm 1 summarizes the meta-training and adaptation procedures.

Remark 2: By ensuring $\tilde{V}_{\mathrm{tr}} + \tilde{V}_{\mathrm{val}} < V_{(i)}$, we intentionally reserve a portion of data from each system's dataset $\mathcal{D}_{(i)}$. This reserved data is neither used for meta-training nor meta-validation, providing a dedicated set of unseen sequences for testing the later-adapted models.

IV. Results and Discussion

In this section, we detail the experimental design and datasets used to train multiple RNN configurations. We analyze the impact of factors like architecture, number of hidden units, biological information, and meta-learning on model performance. We then detail the training process hyperparameters and the evaluation metrics. Finally, we examine the training convergence of these configurations and perform a model selection analysis, emphasizing the performance of the best-selected model across different patient datasets. A complementary sensitivity analysis on the role of the biological loss weight α_{B} with respect to the amount of available training data, independent of the meta-learning framework, is reported in the Appendix.

A. Experimental Setup

This study utilizes data generated from the UVA/Padova T1DM simulator [26], which provides a cohort of 100 virtual adult patients. To ensure experimental consistency while preserving physiological realism, identical input protocols are applied across all subjects, maintaining the inherent patient-specific variability by including the physiological circadian variations described in [28]. All glucose, insulin, and meal data are sampled at 5-min intervals throughout the simulations. The experimental data comprise a total of six distinct simulation scenarios for each patient.

1) Training Datasets ($\mathcal{D}_{\mathrm{Tr}}$): The training data consists of four distinct simulation scenarios designed to capture diverse patient behaviors and control strategies:

1) Ten-day open-loop simulation implemented using the functional insulin treatment (FIT) protocol [36], which combines continuous basal insulin with meal-related boluses. To simulate real-world conditions, bolus calculations included intentional errors and administration delays (randomly between 5 and 30 min postmeal). The meal schedule consisted of three structured daily meals (40 g of carbohydrates at 7:00, 50 g at 12:00, 60 g at 19:00), with random variations in timing ($\pm$30 min) and portion size ($\pm$20%). Additional snacks were introduced randomly in the morning (25 g at 10:00), in the afternoon (35 g at 16:00), or in the evening (25 g at 21:00).
2) Ten-day closed-loop simulation using automated insulin delivery controlled by the MPC proposed in [37]. The schedule matched the ten-day open-loop protocol with the same structured meals and daily variability parameters.
3) Four-day open-loop simulation following the identification protocol described in [38, Table 2], using the FIT protocol for insulin administration.
4) Four-day closed-loop simulation using the same identification protocol as the previous dataset, but with automated insulin delivery instead of manual boluses.

2) Validation Dataset ($\mathcal{D}_{\mathrm{Val}}$): A three-day open-loop simulation following the test protocol for individualized models outlined in [38, Table 3], using the FIT protocol for insulin administration.

3) Test Dataset ($\mathcal{D}_{\mathrm{Tst}}$): A three-day closed-loop simulation following the test protocol for individualized models outlined in [38, Table 3], with automated insulin delivery.

The datasets comprise $|\mathcal{D}_{\mathrm{tr}}| = 4$, $|\mathcal{D}_{\mathrm{val}}| = 1$, and $|\mathcal{D}_{\mathrm{tst}}| = 1$ sequences per patient, for a total of six sequences per patient and 600 unique sequences for the cohort of 100 simulated patients.

It is important to remark that all the results reported in this section are obtained with open-loop predictions using the full length of the simulated input sequences, effectively corresponding to an *infinite prediction horizon*. This choice allows us to assess the intrinsic identification and generalization capabilities of the proposed models, independently of any specific receding-horizon or closed-loop control implementation.

B. Model Configurations

To assess the influence of architectural design, size, and training methodology, we compare distinct RNN configurations. As specified in Remark 1, we exclusively employ single-layer RNN models ($L = 1$), so the (l) superscripts, introduced in Section II-C, are omitted for simplicity. Each configuration is defined by a combination of the following factors.

1) *RNN Architecture*: LSTM or GRU architecture.
2) *Number of Hidden Units*: The RNN state size, with tested values $n_h \in \{8, 16, 32, 64, 128\}$.
3) *Biologically Informed*: Standard RNN or BI-RNN proposed in Section III-A.
4) *Meta-Learning*: Personalized model $\mathcal{M}^{\mathrm{P}}$, i.e., trained solely on individual patient data, or adapted model $\mathcal{M}^{\mathrm{Ad}}$ obtained from an initial meta-model $\mathcal{M}^{\mathrm{M}}$ within the framework described in Section III-C.

Considering the possible levels of each factor, the total number of configurations is 40. Since each configuration is evaluated across all 100 patients, a total of 4000 neural networks are estimated. For clarity, we adopt the following compact notation for each configuration: (BI-){LSTM,GRU}$^{\{\mathrm{P,Ad}\}}_{n_h}$ (e.g., BI-LSTM$^{\mathrm{Ad}}_{32}$ stands for an adapted BI-RNN employing a LSTM architecture with $n_h = 32$).

C. Model Training

Model training follows a unified pipeline across all configurations, comprising data preprocessing, patient-specific parameter identification, and model optimization. Initially, for each ith patient, parameters $\mathcal{P}_{(i)}$ of the linear glucose–insulin dynamics are identified via regularized least squares on nonstandardized four-day open-loop simulations of $\mathcal{D}_{\mathrm{tr},(i)}$, following the framework in [9]. These are then used both as a baseline for validation and for the regularization in the BI-RNN loss function. Subsequently, for RNN training, all data are standardized by subtracting the mean and dividing by the standard deviation, following common practice [18, Ch. 13]. Each model configuration is trained using its corresponding procedure: personalized models follow standard RNN (see Section II-E) or BI-RNN (see Section III-B) training steps, while adapted models utilize the parameters from a pretrained meta-model (see Section III-D) as initialization for a

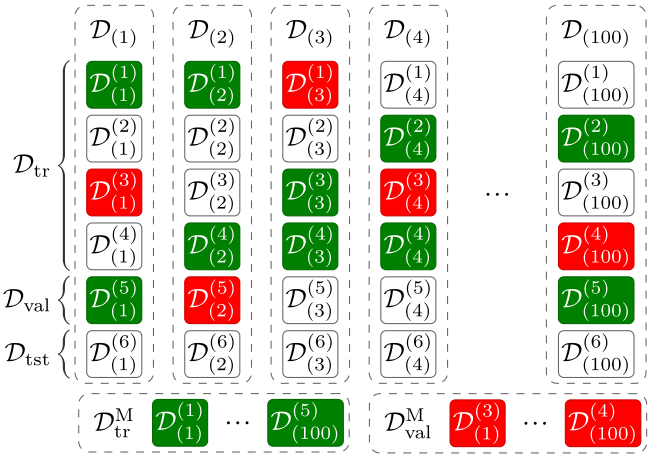

Fig. 4. Depiction of how we randomly sampled the meta-training dataset $\mathcal{D}^{\text{M}}_{\text{tr}}$ (sequences in green) and meta-validation dataset $\mathcal{D}^{\text{M}}_{\text{val}}$ (sequences in red) from the 100 T1DM patient datasets.

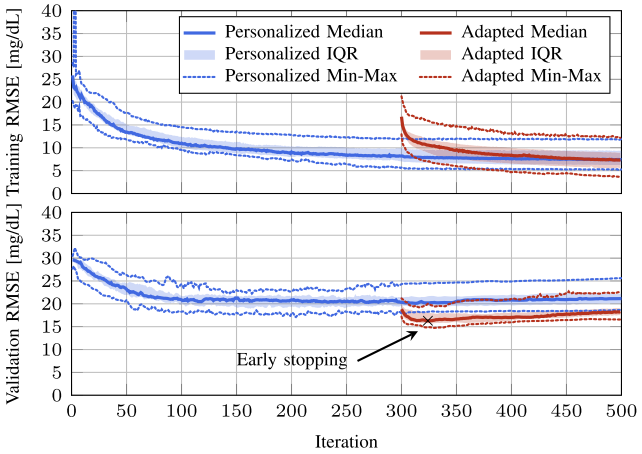

Fig. 5. Training (top) and validation (bottom) RMSE over training iterations for all configurations. Each curve displays, for both personalized (blue) and adapted (red) RNNs, the group-level median (thick line), interquartile range (highlighted area), and minimum–maximum range (dotted lines) computed across all 20 model configurations per group. RMSE values are unstandardized to the glucose observation scale. For the adapted group (red), the origin at iteration 300 represents the start of the adaptation phase (iteration 0) from the meta-model initialization, shifted for visual comparison against the performance of personalized models. The black arrow in the bottom panel roughly highlights the early stopping point for the adapted models.

subsequent adaptation phase. Following [25], the loss weights in (7) used for BI-RNN models are $\alpha_{\text{D}} = 0.5$, $\alpha_{\text{B}} = 0.25$, and $\alpha_{\text{Au}} = 0.25$ to penalize more the loss related to data with respect to the biological and auxiliary ones. All models in this study are trained using the Adam optimizer [18, Ch. 8]. The key hyperparameters for each training procedure are selected according to prior personalized glucose prediction studies [25], [33]. We set the dropout rate to 0.2 and evaluated the validation MSE every two iterations across all training phases for early stopping.

1) Meta-training runs for 300 iterations with a learning rate of 10^{-2}. To simulate real-world scenarios where patient-specific data are scarce, we construct the meta-datasets by sampling only a few sequences from each of the 100 patients, rather than using their entire datasets. Specifically, we sampled $\tilde{V}_{\text{tr}} = 2$ training sequences and $\tilde{V}_{\text{val}} = 1$ validation sequence per patient from $\mathcal{D}_{\text{tr},(i)} \cup \mathcal{D}_{\text{val},(i)}$ (see Fig. 4). This sampling strategy creates a "meta-batch" that aggregates diverse physiological responses from all 100 patients, resulting in $\mathcal{D}^{\text{M}}_{\text{tr}}$ having 200 sequences and $\mathcal{D}^{\text{M}}_{\text{val}}$ having 100 sequences. This approach prevents the meta-model from overfitting to individual physiological traits, favoring instead the learning of a generalized population-level representation that serves as a robust starting point for rapid adaptation. For BI-RNN configurations, the biological loss term (9) is *excluded* ($\alpha_{\text{B}} = 0$) during meta-training as the patient-specific parameters $\mathcal{P}_{(i)}$ are not shared, using solely $\alpha_{\text{D}} = 0.5$ and $\alpha_{\text{Au}} = 0.25$ for the remaining terms.
2) Adapted training proceeds after meta-training for 200 iterations on the entire patient dataset $\mathcal{D}_{(i)}$, using a finer learning rate of 10^{-3} for precise fine-tuning to individual patient dynamics as in [22].
3) Personalized training runs for 500 iterations on $\mathcal{D}_{(i)}$ to ensure a fair comparison. This schedule utilizes a learning rate of 10^{-2} for the initial 300 iterations, followed by a reduced learning rate of 10^{-3} for the final 200 iterations, matching the ones used during meta-training and adaptation.

D. Convergence Analysis

We evaluate the effect of meta-learning on model training by examining the recorded training and validation root-mean-squared error (RMSE) profiles across all training iterations. Specifically, we focus on comparing the adapted models $\mathcal{M}^{\text{Ad}}$ and the personalized models $\mathcal{M}^{\text{P}}$. In total, we analyze 40 unique network configurations (20 per group, as established in Section IV-B), with training performed independently for all 100 patients. To condense the results, we employed a two-stage aggregation process. First, for each specific network configuration, we computed the median RMSE curve across all 100 patients to filter out subject-specific variability. Subsequently, these 40 configuration-specific profiles were grouped by model type to calculate the overall median, interquartile range (IQR), and minimum–maximum range, as depicted in Fig. 5. The upper subplot presents the aggregated training RMSE profiles, revealing that both groups converge to similar training RMSE values. A key finding is the significant difference in training efficiency: adapted models achieve convergence more rapidly, stabilizing within approximately 200 iterations, whereas the personalized models require up to 500 iterations to reach a comparable result. This faster convergence confirms that meta-learning substantially reduces training effort. The lower subplot highlights the main advantage of meta-learning through the aggregated validation RMSE profiles. Adapted models begin their training with substantially lower RMSEs than personalized models due to their meta-training initialization, providing a strong starting point with better generalization capabilities compared to personalized models. Furthermore, adapted models reach their lowest validation RMSEs and attain the best parameters after only 20 – 30 iterations, showcasing that the effective training requires far fewer iterations than allocated,

TABLE II

PATIENT VALIDATION RMSE QUARTILES [MG/DL] BY NETWORK CONFIGURATION (IN BOLD, THE BEST MEDIAN RMSE FOR EACH $n_\hbar$)

$n_\hbar$	Statistics	Standard RNN				BI-RNN			
		GRU		LSTM		GRU		LSTM	
		P	Ad	P	Ad	P	Ad	P	Ad
8	1st qrt	15.2	14.3	14.0	15.0	15.8	14.0	15.5	14.8
	Median	17.9	16.5	17.7	17.2	18.6	**16.1**	19.0	17.0
	3rd qrt	26.9	19.2	25.0	20.4	24.6	19.1	26.0	19.8
16	1st qrt	14.8	13.6	13.6	14.2	13.8	13.5	13.7	13.9
	Median	18.5	16.5	16.4	16.7	16.9	**15.1**	18.4	15.4
	3rd qrt	24.7	19.0	23.3	19.7	25.0	18.6	24.9	18.1
32	1st qrt	14.4	13.0	13.6	13.1	13.5	12.5	12.7	12.8
	Median	18.7	14.8	16.6	14.7	16.8	**14.4**	16.0	15.1
	3rd qrt	26.2	17.5	21.7	17.7	23.7	17.8	22.2	17.7
64	1st qrt	14.8	12.8	13.4	13.4	12.9	12.1	12.8	12.4
	Median	18.5	14.5	17.2	15.7	15.5	**13.9**	16.0	14.2
	3rd qrt	25.6	17.0	23.6	17.7	24.1	17.4	23.6	16.4
128	1st qrt	14.5	13.1	13.9	13.3	13.6	11.9	13.0	12.3
	Median	19.0	14.6	17.4	14.7	16.7	**13.9**	16.2	14.1
	3rd qrt	27.5	17.0	24.6	17.0	24.5	17.1	22.2	16.4

further emphasizing the efficiency of meta-learning in model optimization.

E. Model Selection and Final Performance Assessment

After training the 4000 neural networks, we conducted an analysis to understand which of the 40 considered configurations has the best performance when modeling glucose–insulin dynamics. Table II presents the quartiles of the distribution of patient validation RMSE for each network configuration, showing that meta-learning and biological information lead to an improvement in identification performance in the majority of configurations. We also evaluated alternative performance metrics in addition to the RMSE, namely the mean absolute error (MAE) [14], [15], the mean absolute percentage error (MAPE) [13], the coefficient of determination (R^2), and the FIT index [25], which are commonly reported in the glucose prediction literature for comparison purposes [39]. However, the analysis led to consistent conclusions; therefore, for brevity, we report only the RMSE results in the main text, while the complete set of metrics, included for comparison purposes with other works in the literature, is reported in the Supplementary Material of this article.

Going back to Table II, increasing the number of hidden units $n_\hbar$ over 16 does not have a great impact on the RMSE. To better understand these results, analysis of variance (ANOVA) [40, Ch. 11] techniques are employed to assess which factors reported in Section IV-B significantly affect the validation RMSE, which is the dependent variable. In particular, since all the 40 configurations are tested on each of the 100 patients, the repeated measures ANOVA (RMANOVA) approach [40, Ch. 11] is required due to the correlation between the RMSEs of the same patient. In our setting, there are four factors φ_q, $q \in \{1, \dots, 4\}$, which are the RNN architecture, the number of hidden units $n_\hbar$, the use of biological information, and the employment of meta-learning. The levels that each factor can assume are reported in Section IV-B. Let $\phi_{q,j}$, $j \in \{1, \dots, J_q\}$, $J_q \in \mathbb{N}$, be the jth level of φ_q. Furthermore, let $\lambda_{q,j}$ be the expected value of the dependent variable with respect to all the factors except φ_q, which takes the fixed level $\phi_{q,j}$. For each factor φ_q, the RMANOVA technique tests the null hypothesis that the expected values $\lambda_{q,j}$ are equal for each level $\phi_{q,j}$, i.e.,

$$H_0^q : \lambda_{q,1} = \lambda_{q,2} = \cdots = \lambda_{q,J_q}, \quad q \in \{1, \dots, 4\} \tag{13}$$

with H_1^q being the alternative hypothesis

$$H_1^q : \exists\ j_1, j_2 \in \{1, \dots, J_q\} \ \big|\ \lambda_{q,j_1} \neq \lambda_{q,j_2}.$$

Hence, at least one $\lambda_{q,j}$ is different from the others. When the null hypothesis H_0^q is accepted, the factor φ_q does not influence the dependent variable since varying the level of φ_q does not change the expected value $\lambda_{q,j}$. If H_0^q is rejected, then it is possible to say that φ_q affects the dependent variable.

To apply the RMANOVA approach, the following assumptions must be met [40, Ch. 11].

1) *Randomness*: The dependent variable for different subjects must be uncorrelated.
2) *Normality*: For each combination of factors, the dependent variable must be normally distributed.
3) *Sphericity*: For each factor, the differences of the dependent variable computed on all pairs of levels of the factor must have the same variance. Notice that, if the factor has only two levels, this assumption is automatically met, since there is only one pair of levels, and so only one variance can be computed.

In this work, randomness is automatically met since data are obtained from different patients, so their glucose levels (and their RMSEs) are not correlated. Although the dependent variable is not normally distributed, the second assumption is ensured by applying a logarithmic transformation to the RMSEs. We tested the normality of the log-transformed RMSEs with the Shapiro–Wilk hypothesis test [41]. The third assumption is automatically met for the factors that present only two levels. For the number of hidden units, which has five levels, sphericity does not hold, implying that its corresponding p-value could be underestimated. To overcome this problem, it is possible to correct the p-value using different rationales. In this work, we consider the *lower bound estimate* [40], which is the most restrictive among all corrections.

Applying the RMANOVA approach to the validation RMSEs of the estimated RNNs leads to the results reported in Table III. According to the p-values, the most significant factor that affects the performance of the RNNs is the meta-learning framework proposed in Section III-C. Table II shows that each configuration estimated using meta-learning presents a lower

TABLE III

p-VALUES OF THE HYPOTHESIS TEST IN (13) FOR EACH OF THE FACTORS DESCRIBED IN SECTION IV-B. THE p-VALUE OF THE NUMBER OF HIDDEN UNITS IS ADJUSTED USING THE LOWER BOUND ESTIMATE CORRECTION

Variable	p-value
Network architecture	$1.51 \cdot 10^{-1}$
Number of hidden units	$5.81 \cdot 10^{-8}$
Biologically-informed	$1.27 \cdot 10^{-5}$
Meta-learning	$7.11 \cdot 10^{-27}$ (≈ 0)

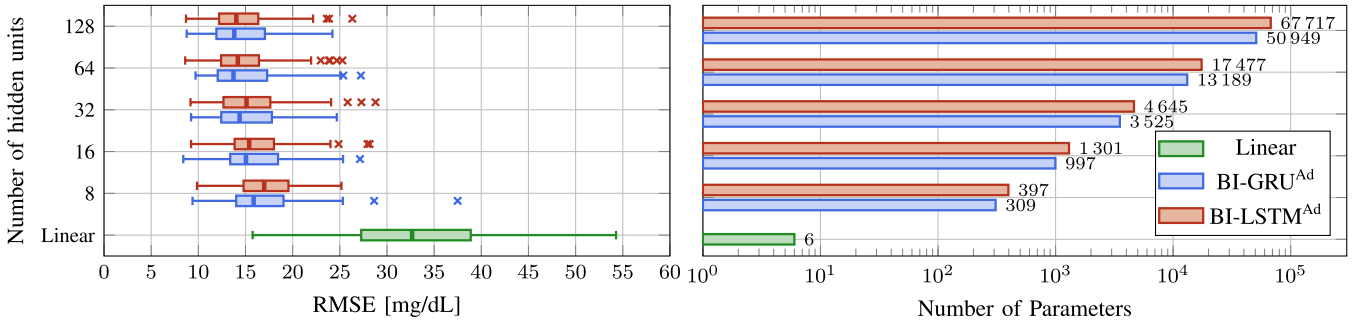

Fig. 6. Performance of BI-GRU$^{\text{Ad}}$, BI-LSTM$^{\text{Ad}}$, and the linear model on all patients. On the left, the box plot of the validation RMSEs, while on the right, their respective number of parameters.

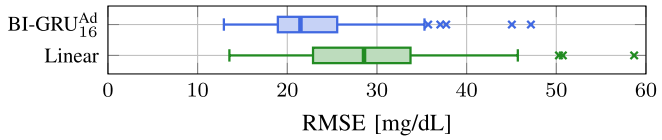

Fig. 7. RMSE on $\mathcal{D}_{\text{tst}}$ comparison between the selected BI-GRU$_{16}^{\text{Ad}}$ and the linear model in (2) on all patients.

median RMSE than the respective configuration without meta-learning. As a result, we can conclude that meta-learning leads to a substantial improvement in performance, confirming the MAML's potential to overcome Challenge I.

Another factor that has an impact on performance is the embedding of biological information presented in Section III-A. According to Table II, most configurations with biological information possess a lower median RMSE. As a result, we can say that biological information significantly improves identification performance. According to the p-values, the network architecture does not significantly influence the RMSE, which means that there is no substantial difference in performance between LSTM and GRU networks. Instead, the number of hidden units appears to have a significant impact on the performance. However, it is not immediate to see the effect of increasing or decreasing the number of hidden units. In fact, when decreasing the number of parameters, the network may suffer from underfitting, while having a high number of parameters can lead to overfitting. In general, the best practice is to select the best-performing model with the fewest number of parameters to avoid overfitting and reduce computational time. Box-plots of patient validation RMSEs in Fig. 6 confirm that LSTM and GRU networks improve the performance over the linear model drastically. Despite this, the median RMSEs of the RNNs' configurations do not decrease monotonically when increasing the number of hidden units. It is possible to see that selecting a number of hidden units higher than 16 does not lead to a significant increase in performance. Due to all these considerations, the best selected configuration to model the relation with an RNN is the BI-GRU$_{16}^{\text{Ad}}$, which has the best tradeoff between model complexity and identification performance. Finally, we evaluated BI-GRU$_{16}^{\text{Ad}}$ on the unseen test set $\mathcal{D}_{\text{tst}}$, obtaining a median RMSE of 21.5 mg/dL; the corresponding values for MAE, MAPE, R^2, and FIT index are reported in the Supplementary Material of this article. In Fig. 7, we report the comparison between the performance of the BI-GRU$_{16}^{\text{Ad}}$ and the linear model in (2) on $\mathcal{D}_{\text{tst}}$, showing that the BI-GRU$_{16}^{\text{Ad}}$ achieves better performance. To assess the statistical validity, we performed a Wilcoxon test [40, Ch. 18] between the RMSEs of BI-GRU$_{16}^{\text{Ad}}$ and the RMSEs of the linear model. The p-value is in the order of 10^{-15}, suggesting that there is a significant difference between the identification performance of the two models.

The aggregated glycemic prediction performance on $\mathcal{D}_{\text{tst}}$ across the entire UVA/Padova adult in-silico cohort is reported in Fig. 8. For each model, the figure displays the median glucose trajectory together with the 25th–75th percentile bands, allowing a population-level comparison between the proposed BI-GRU$_{16}^{\text{Ad}}$ model, the linear baseline, and the ground-truth trajectories generated by the nonlinear time-varying UVA/Padova simulator. These distributional results highlight that the BI-GRU$_{16}^{\text{Ad}}$ model consistently tracks rapid fluctuations and nonlinear patterns in BG levels more accurately than the linear model, especially during postprandial excursions and periods of fast glycemic variability. This demonstrates the benefit of combining recurrent neural architectures with regularization, leading to improved fidelity and reduced prediction error at the cohort level.

To complement the population-wise glycemia analysis, Fig. 9 reports the IOB predictions for a representative adult patient. In this case, the BI-GRU$_{16}^{\text{Ad}}$ model reproduces the IOB profile with an accuracy comparable to that of the linear baseline, demonstrating that the biologically informed regularization effectively constrains the network to maintain physiologically meaningful insulin absorption dynamics. Such behavior is noteworthy, as a purely data-driven RNN would typically replicate only the glycemia trajectory while failing to produce coherent estimates of other physiological quantities. Together, the cohort-level glycemic trends and the patient-specific IOB evaluation indicate that the proposed framework improves predictive accuracy for glucose dynamics compared to a purely linear modeling approach while ensuring that relevant physiological quantities, such as IOB, remain consistent with known biological behavior.

Although Fig. 8 shows that the proposed BI-GRU$_{16}^{\text{Ad}}$ model achieves a markedly lower population-level RMSE compared to the linear baseline, a closer inspection of the predicted trajectories reveals a limitation that is not captured by

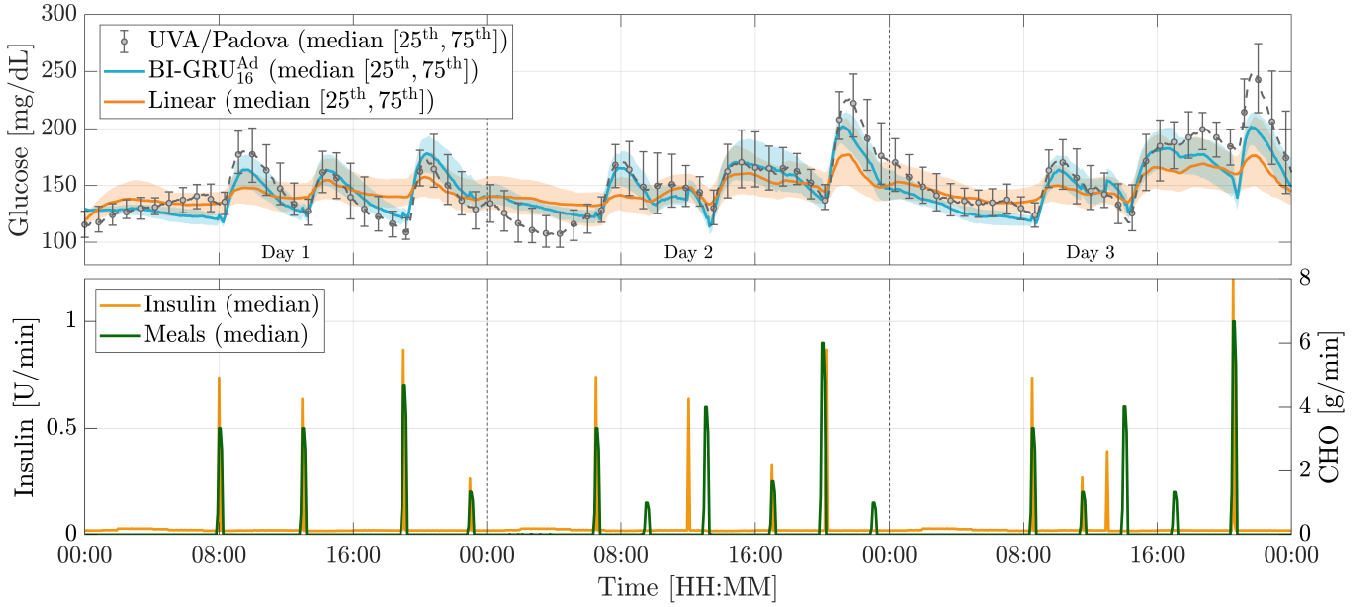

Fig. 8. Testing scenario for all adult patients of the UVA/Padova in-silico cohort. (Top) Compares glucose predictions in term of median and [25th, 75th] percentiles predictions: gray dots and lines indicate the reference trajectory generated by the nonlinear time-variant UVA/Padova model, blue lines correspond to the proposed BI-GRU$_{16}^{\text{Ad}}$, and orange lines to the linear model from [9]. (Bottom) Displays the administered inputs, i.e., insulin (left axis) and carbohydrate intake rate (CHO, right axis).

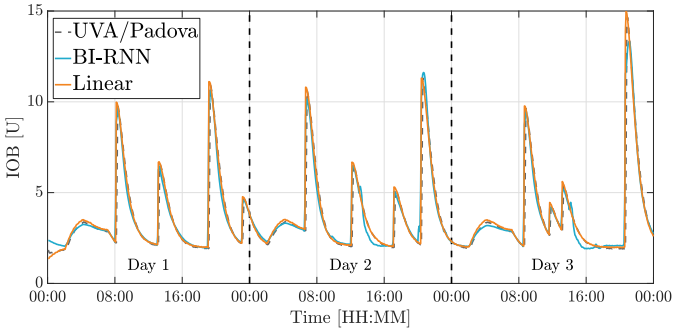

Fig. 9. Testing scenario on a single adult patient for insulin-on-board predictions: the dashed line indicates the reference trajectory generated by the nonlinear time-variant UVA/Padova T1DM simulator, the blue line corresponds to the proposed BI-GRU$_{16}^{\text{Ad}}$, and the orange line to the linear model from [9].

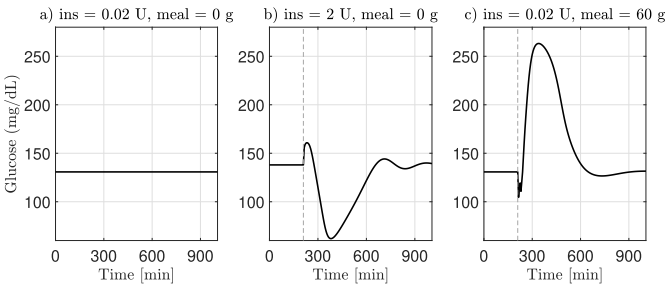

Fig. 10. Physiological validation of BI-GRU$_{16}^{\text{Ad}}$. Each plot illustrates the glucose response to a specific input scenario. (a) Constant basal insulin infusion maintains stable glycemia. (b) Bolus of insulin is administered, resulting in a drop in glucose levels. (c) Meal is introduced, causing an increase in glycemia.

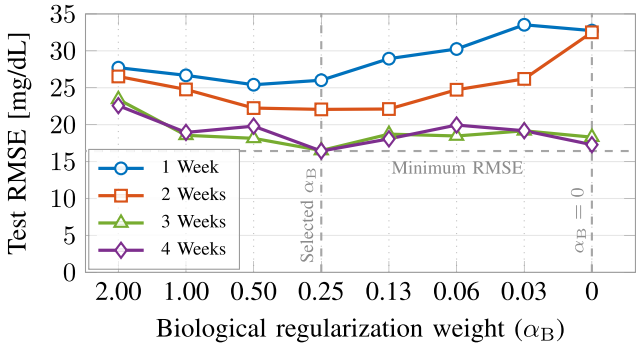

Fig. 11. Sensitivity analysis of test RMSE for a representative patient. The x-axis is inverted so that model complexity increases (and biological regularization emphasis decreases) from left to right. Biological regularization is most effective in low-data regimes (up to two weeks of data), while its impact diminishes and performance saturates as training data increases.

aggregate error metrics. In particular, during short hypoglycemic episodes, the model may exhibit a tendency to overestimate BG levels. From a purely numerical standpoint, these events are relatively rare and short-lived, and therefore, their contribution to global metrics such as RMSE remains limited. However, from an application-oriented perspective, this behavior is critical. Indeed, in closed-loop glucose control or when the model is employed for hypoglycemia alarms, systematic overestimation in low-glycemia regions may lead to inappropriate insulin delivery or delayed corrective actions, potentially resulting in clinically relevant hypoglycemic events. This observation highlights a well-known mismatch between standard identification metrics and control-oriented performance [42], where local accuracy in safety-critical regions is often more important than global prediction error.

To better interpret the long-horizon prediction behaviors from a dynamical standpoint, particularly the localized inaccuracies observed in critical glycemic regions, the response of BI-GRU$_{16}^{\text{Ad}}$ to artificial input sequences was analyzed and is reported in Fig. 10. Overall, the general input–output relationships are correctly captured: glucose levels decrease

following bolus insulin administration, increase in response to carbohydrate intake (i.e., meal intake, modeled as a square pulse corresponding to 60 g of carbohydrate distributed over a 15-min interval) while staying constant during continuous basal insulin injection. This behavior is consistently observed across all virtual patients in the cohort. However, a transient inverse effect can be observed at the onset of the impulse response. In particular, glucose levels rise temporarily after an insulin bolus injection or slightly decrease after carbohydrate intake. This unexpected behavior is likely attributable to the design of the network, where the separation of input effects is not explicitly encoded in its architectural structure, but rather implicitly encouraged through a soft constraint imposed by the augmented loss function during training.

V. Conclusion

In this work, we explored the use of meta-learning in RNNs for glucose–insulin dynamics modeling, particularly inspired by the MAML approach [22], to address the significant challenge of data scarcity in physiological measurements. This method involves learning a meta-representation of the glucose–insulin systems, which we then efficiently adapt to each patient. We also introduced a novel BI-RNN architecture that leverages prior physiological knowledge during training to further enhance standard RNN performance. We tested our methodologies with various configurations on a cohort of 100 in-silico patients, statistically assessing the effects of the network sizes, RNN types, and the aforementioned methods. Our results confirm the superiority of BI-RNN architectures over traditional RNNs while indicating that all design hyperparameters are significant except for the architectural choice between LSTM or GRU layers. Notably, meta-learning proves to be the most impactful factor, both for performance and for substantially lowering training effort. By establishing a meta-model for entire system classes, meta-learning enables quick adaptation to specific patients with only a few training steps. Through analysis, we determined the optimal configuration among the 40 tested to be the adapted biologically informed GRU network with 16 hidden units, offering the best performance-complexity tradeoff. Ultimately, these findings suggest that both the meta-learning and BI-RNN methodologies substantially improve modeling and predictive performance, directly addressing Challenges I and II.

Despite these encouraging results, a limitation of the current training framework is the uniform weighting of prediction errors across the entire glycemic range. Since hypoglycemic and hyperglycemic events are less frequent in training datasets, especially those generated under closed-loop control, standard loss functions may not sufficiently prioritize accuracy in these clinically critical regions. Consequently, errors in the hypoglycemic range might be underrepresented, despite their significant impact on patient safety. A promising research direction would be the adoption of risk-aware or region-weighted data loss functions [43], which penalize deviations in critical zones more heavily. Such an approach could enhance local accuracy in safety-critical conditions without increasing model complexity, better aligning the identification objective with the requirements of automated insulin delivery and hypoglycemia alert systems.

Future work should also focus on validating the proposed framework using real-world patient data. While this study was limited to in-silico experiments, meta-learning itself provides a natural mitigation strategy for this limitation: the ability to learn a robust meta-model solely or partially from in-silico data and then better adapt to scarce real data than traditionally trained personalized models is a potential feature to explore [35], [44]. Other research directions include the inclusion of additional physiological factors, such as physical activity [45], into the BI-RNN model, and the exploration of more advanced MAML variants [46]. Finally, as stated in Section I, a further research direction is the integration of the learned BI-RNN within nonlinear MPC schemes, to assess the tradeoff between closed-loop performance and online computational burden.

Appendix
Sensitivity Analysis on α_B and Data Availability

To assess the role of the biological loss term α_B in (7) under varying data availability, we conducted an additional analysis independent of the meta-learning framework presented in Section III-C, i.e., by training the BI-RNN model from scratch on individual patient data only (personalized training, as in Section IV-C). This isolation allows us to evaluate the standalone contribution of the biological regularization, decoupled from the population-level prior provided by meta-training. For a representative patient, we generated six weeks of data following the ten-day open-loop FIT protocol described in Section IV-A, partitioned into four weeks for training ($\mathcal{D}_{tr}$), one week for validation ($\mathcal{D}_{val}$), and one week for testing ($\mathcal{D}_{tst}$). We trained the BI-RNN model on progressively larger subsets of $\mathcal{D}_{tr}$ (1, 2, 3, and 4 weeks), keeping the hyperparameters identical to those in Section IV-C, and evaluating a range of biological loss weights $\alpha_B \in \{0, 0.03, 0.06, 0.12, 0.25, 0.50, 1.00, 2.00\}$, with $\alpha_B = 0.25$ corresponding to the nominal value adopted in Section IV-C. The remaining loss weights were kept fixed ($\alpha_D = 0.5$, $\alpha_{Au} = 0.25$) for a fair comparison. Fig. 11 reports the test RMSE as a function of α_B for each data regime. The results show that the biological loss is most beneficial when training data is scarce (one to two weeks), where an intermediate α_B achieves the best performance, while both excessively low and high values degrade accuracy. As the amount of training data increases, the sensitivity to α_B progressively diminishes, with performance saturating already at three to four weeks of data. This confirms that the proposed biologically informed regularization specifically addresses the data scarcity aspect of Challenge I, complementing the population-level prior provided by meta-learning (see Section III-C), which instead targets interpatient variability [see Challenge I].

Acknowledgment

This work reflects only the authors' views and opinions; neither the Ministry for University and Research nor European Commission can be considered responsible for them-CUP B53C22006700001.

REFERENCES

[1] G. D. Ogle et al., "Global type 1 diabetes prevalence, incidence, and mortality estimates 2025," *Diabetes Res. Clin. Pract.*, vol. 225, Jul. 2025, Art. no. 112277.

[2] C. Cobelli, E. Renard, and B. Kovatchev, "Artificial pancreas: Past, present, future," *Diabetes*, vol. 60, no. 11, pp. 2672–2682, Nov. 2011.

[3] M. Messori, C. Cobelli, and L. Magni, "Artificial pancreas: From in-silico to in-vivo," *IFAC-PapersOnLine*, vol. 48, pp. 1300–1308, Jan. 2015.

[4] A. Katsarou et al., "Type 1 diabetes mellitus," *Nat. Rev. Dis. Primers*, vol. 3, Mar. 2017, Art. no. 17016.

[5] R. Gondhalekar, E. Dassau, and F. J. Doyle, "Velocity-weighting & velocity-penalty MPC of an artificial pancreas," *Automatica*, vol. 91, pp. 105–117, May 2018.

[6] D. Boiroux et al., "Overnight glucose control in people with type 1 diabetes," *Biomed. Signal Process. Control*, vol. 39, pp. 503–512, Jan. 2018.

[7] I. Hajizadeh et al., "Adaptive personalized multivariable artificial pancreas using plasma insulin estimates," *J. Process Control*, vol. 80, pp. 26–40, Aug. 2019.

[8] R. Hovorka et al., "Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes," *Physiolog. Meas.*, vol. 25, no. 4, pp. 905–920, Aug. 2004.

[9] P. Abuin, P. Rivadeneira, A. Ferramosca, and A. González, "Artificial pancreas under stable pulsatile MPC," *J. Process Control*, vol. 92, pp. 246–260, Aug. 2020.

[10] B. Sonzogni, J. M. Manzano, M. Polver, F. Previdi, and A. Ferramosca, "CHoKI-based MPC for blood glucose regulation in artificial pancreas," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 9672–9677, 2023.

[11] Y. Ruan, M. E. Wilinska, H. Thabit, and R. Hovorka, "Modeling day-to-day variability of glucose–insulin regulation over 12-week home use of closed-loop insulin delivery," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 6, pp. 1412–1419, Jun. 2017.

[12] R. N. Bergman, "Toward physiological understanding of glucose tolerance: Minimal-model approach," *Diabetes*, vol. 38, no. 12, pp. 1512–1527, Dec. 1989.

[13] S. Langarica, M. Rodriguez-Fernandez, F. Núñez, and F. J. Doyle, "A meta-learning approach to personalized blood glucose prediction in type 1 diabetes," *Control Eng. Pract.*, vol. 135, Jun. 2023, Art. no. 105498.

[14] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Personalized blood glucose prediction for type 1 diabetes using evidential deep learning and meta-learning," *IEEE Trans. Biomed. Eng.*, vol. 70, no. 1, pp. 193–204, Jan. 2023.

[15] S. Rancati et al., "Exploration of foundational models for blood glucose forecasting in type-1 diabetes pediatric patients," *Diabetology*, vol. 5, no. 6, pp. 584–599, Nov. 2024.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[17] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[18] K. P. Murphy, *Probabilistic Machine Learning*. Cambridge, MA, USA: MIT Press, 2022.

[19] A. Karpatne, R. Kannan, and V. Kumar, *Knowledge-Guided Machine Learning*, 1st ed., Boca Raton, FL, USA: Chapman and Hall/CRC, 2022.

[20] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.

[21] J. Schmidhuber, "Evolutionary principles in self-referential learning, or on learning how to learn," Diploma thesis, Dept. Inform., Technische Universität München, Munich, Germany, 1987.

[22] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," 2017, *arXiv.1703.03400*.

[23] F. Bonassi, A. La Bella, M. Farina, and R. Scattolini, "Nonlinear MPC design for incrementally ISS systems with application to GRU networks," *Automatica*, vol. 159, Jan. 2024, Art. no. 111381.

[24] A. Yazdani, L. Lu, M. Raissi, and G. E. Karniadakis, "Systems biology informed deep learning for inferring parameters and hidden dynamics," *PLOS Comput. Biol.*, vol. 16, no. 11, Nov. 2020, Art. no. e1007575.

[25] S. De Carli, N. Licini, D. Previtali, F. Previdi, and A. Ferramosca, "Integrating biological-informed recurrent neural networks for glucose-insulin dynamics modeling," *IFAC-PapersOnLine*, vol. 59, no. 2, pp. 91–96, 2025.

[26] R. Visentin et al., "The UVA/Padova Type 1 Diabetes simulator goes from single meal to single day," *J. Diabetes Sci. Technol.*, vol. 12, no. 2, pp. 273–281, 2018.

[27] C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, and C. Cobelli, "The UVA/PADOVA type 1 diabetes simulator," *J. Diabetes Sci. Technol.*, vol. 8, no. 1, pp. 26–34, Jan. 2014.

[28] R. Visentin, C. D. Man, Y. C. Kudva, A. Basu, and C. Cobelli, "Circadian variability of insulin sensitivity," *Diabetes Technol. Ther.*, vol. 17, pp. 1–7, Jan. 2015.

[29] C. Dalla Man, M. Camilleri, and C. Cobelli, "A system model of oral glucose absorption," *IEEE Trans. Biomed. Eng.*, vol. 53, pp. 2472–2478, 2006.

[30] R. N. Bergman, G. Bortolan, C. Cobelli, and G. Toffolo, "Identification of a minimal model of glucose disappearance for estimating insulin sensitivity," *IFAC Proc. Volumes*, vol. 12, no. 8, pp. 883–890, Sep. 1979.

[31] P. A. Mongini, M. Drecogna, and C. Toffanin, "New physics-LSTM hybrid models for control-oriented glucose prediction in type 1 diabetes," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2025, pp. 662–667.

[32] M. Drecogna, C. Cobelli, and C. Toffanin, "Personalized LSTM-based alarm systems for hypoglycemia prevention in an intraperitoneal artificial pancreas," in *Proc. 32nd Medit. Conf. Control Autom. (MED)*, Jun. 2024, pp. 233–238.

[33] F. Iacono, L. Magni, and C. Toffanin, "Personalized LSTM-based alarm systems for hypoglycemia and hyperglycemia prevention," *Biomed. Signal Process. Control*, vol. 86, Sep. 2023, Art. no. 105167.

[34] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*.

[35] D. Piga, F. Pura, and M. Forgione, "On the adaptation of in-context learners for system identification," *IFAC-PapersOnLine*, vol. 58, no. 15, pp. 277–282, 2024.

[36] K. Howorka, *Functional Insulin Treatment*. Cham, Switzerland: Springer, 2012.

[37] C. Toffanin and L. Magni, "Constrained versus unconstrained model predictive control for artificial pancreas," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 5, pp. 2288–2299, Sep. 2023.

[38] M. Messori, C. Toffanin, S. D. Favero, G. De Nicolao, C. Cobelli, and L. Magni, "Model individualization for artificial pancreas," *Comput. Methods Programs Biomed.*, vol. 171, pp. 133–140, Apr. 2019.

[39] P. G. Jacobs et al., "Artificial intelligence and machine learning for improving glycemic control in diabetes: Best practices, pitfalls, and opportunities," *IEEE Rev. Biomed. Eng.*, vol. 17, pp. 19–41, 2024.

[40] D. Howell, *Statistical Methods for Psychology*. Boston, MA, USA: Thomson Wadsworth, 2007.

[41] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, Dec. 1965.

[42] C. Mosquera-Lopez, R. Dodier, N. Tyler, N. Resalat, and P. Jacobs, "Leveraging a big dataset to develop a recurrent neural network to predict adverse glycemic events in type 1 diabetes," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 9, pp. 2527–2536, Sep. 2020.

[43] S. D. Favero, A. Facchinetti, and C. Cobelli, "A glucose-specific metric to assess predictors and identify models," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 5, pp. 1281–1290, May 2012.

[44] A. Rafiei, R. Moore, S. Jahromi, F. Hajati, and R. Kamaleswaran, "Meta-learning in Healthcare," *Social Netw. Comput. Sci.*, vol. 5, p. 791, Aug. 2024.

[45] N. Licini, B. Sonzogni, P. Abuin, F. Previdi, A. H. González, and A. Ferramosca, "Artificial pancreas under stable pulsatile model predictive control: Including the physical activity effect," *IFAC J. Syst. Control*, vol. 36, Jun. 2026, Art. no. 100414.

[46] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Meta-learning in neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, Sep. 2021.