Article

Not peer-reviewed version

Fine-Tuning LLMs for Real-Time Fuzzy Insulin Control in Type I Diabetes

Jordan Kralev *

Posted Date: 20 April 2026

doi: 10.20944/preprints202604.1316.v1

Keywords: fine-tuning; fuzzy insulin control; type 1 diabetes; large language models; artificial pancreas

 doi:10.20944/preprints202604.1316.v1

Article

Fine-Tuning LLMs for Real-Time Fuzzy Insulin Control in Type I Diabetes

Jordan Kralev

Department of Systems and Control, Technical University of Sofia, 1756 Sofia, Bulgaria; jkralev@ieee.org

Abstract

The paper propose a novel fine-tuning framework that adapts a large language model (LLM) to real-time fuzzy insulin control for type 1 diabetes mellitus (T1DM). The method combines a Hovorka-based glucose-insulin model, fuzzy membership encoding of glucose and insulin states, and a causal language model trained through low-rank adaptation to map recent physiological history to insulin dosing decisions. The central idea is to represent glucose and insulin variables through linguistic fuzzy terms, such as hypoglycemia, target range, hyperglycemia, and zero, low, or high insulin dose, and to embed these terms directly into the language model's token space. This enables the model to act as a sequence-aware fuzzy controller while preserving interpretability through membership functions and defuzzification of the output logits into a crisp insulin dose. The proposed controller is trained in closed loop using virtual patients generated from the Hovorka model, with the objective of minimizing glucose deviation from a clinically relevant target. Additional validation in the UVa/Padova simulator demonstrates that the learned policy transfers to a standardized benchmark environment and achieves strong time-in-range performance with low hypoglycemic risk. The study shows that a fine-tuned language model can be repurposed as a real-time biomedical decision-support component when its inputs and outputs are structured through fuzzy logic. This hybrid framework offers a promising direction for interpretable and adaptive artificial pancreas control, combining physiological modeling, linguistic reasoning, and modern language-model adaptation in a unified closed-loop system.

Keywords: fine-tuning; fuzzy insulin control; type 1 diabetes; large language models; artificial pancreas

1. Introduction

Type 1 diabetes (T1D) is a chronic autoimmune disease, characterized by the T-cell-mediated destruction of insulin-producing β cells in pancreatic islets, that results in insulin deficiency [1–3]. Therefore, insulin substitution therapy is required for T1D subjects by subcutaneously or intravenously administration of insulin substitutes as short acting Lispro [4]. Healthy blood glucose range is between 90 and 180 mg/dL, with optimal target of 105 mg/dL. If the glucose is above 300 mg/dL a hyperglycemic episode is onset which when extended or frequently repeated can lead to diabetic keto-acidosis or hyperosmolar hyperglycemic state. If the glucose is below 70 mg/dL a hypoglycemic episode is onset, which if not treated by carbohydrate ingestion can lead to unconsciousness or seizures. Chronic consequence of unregulated glucose can be retinopathy, nephropathy, and neuropathy, ardiovascular disease, cerebrovascular disease, and peripheral vascular disease. In T1D the glucagon secretion from insulin activated pancreatic alpha cells is also impacted [5,6]. This additional metabolic disturbance manifests as delayed or absent onset of glucagon secretion during hypoglycemia, or as counter-regulatory glucagon secretion during onset of hyperglycemic period.

With advances of portable, battery powered wearable subcutaneous glucose sensors [7,8] and insulin pumps [9], many automatic insulin delivery systems, called artificial pancreas (AP), are under development - OpenAPS [10], MiniMed [11], Nightscout [12], INCA [13]. While intravenous insulin delivery would guarantee minimal glucose deviations [14], the subcutaneous pathway is more inert [15] and posing significant control challenge [16–18]. A key requirement for AP system is aperiodic

 doi:10.20944/preprints202604.1316.v1

behaviour because once administered, insulin cannot be removed. That's why, a hyperglycemic peak is usually followed by a hypoglycemic period due to over regulation. The American Food and Drug Administration (FDA) accepted the UVa/Padova large-scale metabolic simulator [5] for T1D treatment as a substitute for pre-clinical animal model studies, where novel control algorithms can be benchmarked in standardized way. The simulator offers a population of ten adult, adolescent and children virtual patient models, as well as, capability to design meal administration scenarios. Once a T1D controller is submitted to FDA for approval, an internal testing with more than hundred virtual patient population is carried out.

There are many control approaches for AP system with announced or unannounced meals [19–21]. Authors in [22] are using temporal cost function with discount factors reflecting inter-subject variability. In [23], authors propose event-triggered model predictive controller, also zone oriented model predictive control can be seen in [24]. We can find also multi-model PID control tuned with a genetic algorithm [25] with fuzzy gain scheduling strategy. Machine learning (ML) in AP systems [26] could detect anomalous patterns. Robust μ-synthesis technique is applied in [27], also H_∞ control by [28,29] or multi-objective H_2/H_∞ design by [30].

Fuzzy logic methods in control theory have long history of success [31]. The design of a fuzzy controller aims to incorporate expert linguistic knowledge for system behaviour, which is defined as logical expressions between propositions with continuous degree of validity between true and false alternatives. The fuzzy inference systems are built after establishment of membership functions for input and output variables, fuzzy rule declarations and defuzzifying algorithm [32,33]. Numerous attempts for control of blood glucose with fuzzy non-linear controllers can be found varying from very basic approaches [34] to modifications like introduction of personalization factor in [35], and more advanced type-2 fuzzy control with the aim to manage the uncertainty from inter-subject variability [36].

The connection between fuzzy set theory and natural language processing (NLP) was surveyed by [37]. Classically, common NLP tasks are text classification, question answering, translation, summarization, text generation, fill-mask, etc. [38]. The [39,40] examine fuzzy theoretic interpretation of similarity score between word embedding vectors for fuzzy control, also [41] use fuzzy scores for top-k selection during retrieval augmented translation (RAT). Fuzzy reasoning can be applied as a mechanism for sentient classification by [42]. Fuzzy decision making is employed for test-to-sql generator models [43]. However, fuzzy logic alone is not able to model the full complexity of the natural language or even to pretend to be reasonable model to capture semantics without further development as [44] shows.

There are two groups of methods in NLP. First group analyse morphology, syntax and semantics using conventional pattern recognition and formal grammar techniques leading to knowledge representation as graphs. The second group of methods is solving language modelling problem by collecting large corpora of texts with sole aim to train a large scale machine learning models. The transformer-based models [45] have significant impact in the natural language processing systems enabling improvements over preceding encoder/decoder recurrent networks used for language to language translation. The transformer is performing block correlation analysis over window of input tokens without reliance on any recurrent connections. This improves model training performance for next token prediction task and also significantly reduce inference time.

The three levels in the construction of LLMs for practical uses [46] are:

- Pre-training on the large corpus or text data from diverse sources;
- Fine-tuning of the model to a target application domain with application specific data;
- Prompt adaptation providing context basis.

Technological advances of single instruction multiple data (SIMD) processors mainly employed in graphical processing units (GPU), such as increased video memory, reduced execution time, richer instruction set, allow optimization of models with hundred of billion of parameters. Still, a full pre-training of an LLM is slow and costly. The bigger the corpus and bigger the number of parameters more iterations are required. Therefore, a majority of LLM applications operate on the level of

 doi:10.20944/preprints202604.1316.v1

prompt adaptation where various techniques are possible - experiments with prompt structure or with system expression, optimization of a fixed prompt prefix. A notable technique is retrieval augmented generation where given input query, a similarity search is performed in an external dataset to obtain actual information, to be further fed into a LLM processor for summarization in respect to input query. Since LLM operate as a model of conditional probability distribution over input sequences, the aim in prompt engineering is to "fine-tune" the conditional distribution.

As a middle ground between full pre-training and prompt adaptation, a parametric fine-tuning of LLM over small number of parameters is possible. This techniques stem from the LLM property that when they are applied to specific knowledge domain, their intrinsic dimension is drastically reduced [46,47]. Therefore, the tuning in low dimensional projection of full parameter space can be as efficient as training the model over full parameter space. The fine tuning setup is the same as full scale training setup, after conditioning the model with trainable adapter blocks while fixing the parameters of the original model.

The ability of LLM models to generate fast and correct predictions over very large and diverse data stimulates LLM applications outside the NLP studies. The place of LLM in control theory is reviewed recently by [48] with corpus of 260 references. There, a direct LLM embodiments in a control systems, amongst others, are seen at higher stratification layers. For example to dynamically suggest parameters for a feedback loop controller like PID or LQR, to generate sequence of action trajectories or to explore design space for a system architecture. However, a direct substitution of a classical feedback controller with an LLM for real-time decision making is not advised.

In the present study we aim to fine-tune a small LLM to operate as an actual feedback loop controller for a metabolic process like glucose regulation in T1D. Such execution is possible, because of reasonable decision time requirement for AP systems varying between 1 to 5 minutes, which is enough for a compact LLM with several billion of parameters running on a middle grade GPU. Moreover, multiple requests to the model can be batched in a centralized fashion for processing on a cloud GPU, similar to one medic for multiple patients scenario.

The key problem in using LLM directly in control feedback loop is in the unmatched signals domains. While physical systems are converting quantities, LLMs are transforming high-dimensional representations of input token stream. The representation of physical quantities in the LLM input space can be naively approached by direct substitution of a decimal string of digits in the LLM prompt. However, LLM are reduced in performance when doing pure mathematical evaluations, especially, smaller models below 10 billion of parameters. The LLM treat such decimal string representation as a general character sequence and performing correlation arithmetic over it instead of arithmetic with quantities.

In this study, we propose that token meaning in the LLM embedding space is dominated by vector direction and a given physical quantity can be represented as a modulation of the amplitude of a selected carrier token. Therefore, a perturbation of the model vocabulary embedding vectors is introduced to smoothly vary the degree of membership of physical quantities to a collection of fuzzy sets mapped to this vocabulary. The benefit from such representation is that a single modulated carrier token is used to represent a numerical quantity instead of tokens corresponding to a string of decimals. Furthermore, to improve training convergence, a linear combination of embedding vectors is assigned to a physical quantity by using fuzzy set membership for modulation. Contributions in the preset work can be summarized as:

1. Using a fine-tuned LLM model to execute real-time decisions for insulin regulation in an artificial pancreas system.
2. Introduction of a novel technique to seamlessly represent quantified data into LLM embedding space as amplitude modulated carrier tokens.

 doi:10.20944/preprints202604.1316.v1

3. Closed-loop fine-tuning of fuzzy LLM controller with multiple randomized metabolic models of T1D virtual patients.
4. Validation of fine-tuned fuzzy LLM controller in UVa/Padova simulator for 10 virtual patients from adult population.

The organisation of the paper is as follows. Section 2 review briefly the Hovorka T1D model, fuzzy control theory applied for blood glucose regulation and the architecture of TinyLlama LLM used for experiments. The section 3 presents fuzzy embedding approach for LLM, gives the equations of the closed loop system and training objective. Then in section 4 brief presentation of fine tuning setup with code snippets is given. The section 5 summarize the results from fine-tuning and UVa/Padova simulation.

2. Preliminaries

2.1. Hovorka Model

There are several widely used metabolic models in the field of AP systems. In this article we use Hovorka [49] model as most commonly adapted in T1D studies for insulin control predictions with a short-term acting Lispro replacement therapy. The Hovorka model describe glucose-insulin interaction with a two-compartment pharmacokinetic model incorporating gut absorption dynamics obtained from the intake of carbohydrates, subcutaneous insulin absorption dynamics, insulin interaction with the plasma glucose, and rate of endogenous glucose production. The two compartments in the model are the subcutaneous fluid and the blood plasma.

The state-space equations of the Hovorka model include glucose metabolism model

$$\begin{aligned} \dot{Q}_1(t) &= EGP_0(1.0 - x_3(t)) + U_G - F_R - \left(x_1(t) + \frac{F^c_{01}}{V_G G(t)}\right) Q_1(t) + k_{12} * Q_2(t) \\ \dot{Q}_2(t) &= x_1(t) Q_1(t) - (k_{12} + x_2(t)) Q_2(t) \end{aligned}, \tag{1}$$

where Q_1 and Q_2 measured in *mmol* are glucose amounts in the accessible and non-accessible compartments. Accessible compartment is subcutaneous fluid where measurements can be taken with portable glucose sensor, while non-accessible compartment is blood plasma where blood sample is required in order to perform a measurement. The kinetic rate k_{12} characterizes the transfer between both compartments. The constant EGP_0 represents endogenous glucose production, which is present even without carbohydrate ingestion and extrapolated to zero insulin. Therefore $EGP = EGP_0(1.0 - x_3(t))$ is insulin dependent glucose production where one of insulin actions $x_3(t)$ is to dampen the glucose release from tissues. The measurable glucose concentration G in *mmol/liter* from the sensor is calculated by

$$G(t) = \frac{Q_1(t)}{V_G}, \tag{2}$$

where V_G is glucose distribution volume in accessible compartment. Glucose readings are either represented in *mmol/l* or in *mg/dl*, after applying a conversion factor of 18 mg/dl for every mmol/l.

The U_G term represents ingested carbohydrates which are absorbed in the gut with some versions of the model detailing the gut absorption model further if that is required.

$$U_G(t) = \sum_{m=1}^{N_m} \frac{D_{G,m}\, A_G\, \tau_m\, e^{-\tau_m / t_{max,G}}}{t^2_{max,G}}, \tag{3}$$

where $D_{G,m}$ is the ingested carbohydrate amount with meal $m \in 1 \ldots N_m$ in *mmol*, A_G is carbohydrate absorption ratio (bioavailablility), $t_{max,G}$ is time to peak of carbohydrate concentration in tissues and τ_m is relative meal time defined as

$$\tau_m = t - t_m, \quad t > t_m. \tag{4}$$

 doi:10.20944/preprints202604.1316.v1

The presented glucose metabolism model includes also renal excretion component for extreme hyperglycaemia as

$$F_R(t) = 0.003\ (G(t) - 9)\ V_G \tag{5}$$

which is non-zero for $G \geq 9$ mmol/l. Also we have total non-insulin dependent glucose flux (for example in CNS) corrected for ambient glucose concentration as

$$F_{01}^c(t) = \begin{cases} F_{01}, & G \geq 4.5 \\ F_{01}\ G(t)/4.5 \end{cases} \tag{6}$$

Insulin absorption subsystem is modelled again as two compartment kinetic system

$$\begin{aligned} \dot{S}_1(t) &= u(t) - \frac{S_1(t)}{t_{max,I}} \\ \dot{S}_2(t) &= \frac{S_1(t)}{t_{max,I}} - \frac{S_2(t)}{t_{max,I}} \end{aligned} \tag{7}$$

where S_1 and S_2 are a two-compartment subcutaneous volumes absorbing short acting insulin in mU, and $u(t)$ represents administration of insulin in mU/min, $t_{max,I}$ in min is the time-to-maximum insulin absorption. The plasma insulin concentration $I(t)$ in mU/L is obtained as

$$\dot{I}(t) = \frac{S_2(t)}{t_{max,I} V_I} - k_e I(t), \tag{8}$$

where V_I is insulin distribution volume and k_e is insulin elimination rate.

The insulin action subsystem is modelled as

$$\dot{x}_i(t) = k_{a,i}\ x_i(t) + k_{b,i}\ I(t), \quad i = 1,2,3 \tag{9}$$

where x_1, x_2 and x_3 represent the effects of insulin on glucose distribution/transport, glucose disposal and endogenous glucose production as evident from the above equations, $k_{a,i}$ are deactivation rate constants, and $k_{b,i}$ are activation rate constants

$$\begin{aligned} k_{b,1} &= S_I^T\ k_{a,1} \\ k_{b,2} &= S_I^D\ k_{a,2} \\ k_{b,3} &= S_I^E\ k_{a,3} \end{aligned} \tag{10}$$

where S_I^T is insulin sensitivity of glucose transport from interstitial to plasma, S_I^D is insulin sensitivity of glucose elimination from plasma into tissues and S_I^E is insulin sensitivity of endogenous glucose production.

Original paper of Hovorka [49] divides the model parameters into tunable and constant, because [50] proves some of the parameters to be likely unidentifiable from data. On the other hand, tunable model parameters aim to retain ability to represent the wide range of glucose variations observed with type 1 diabetes. Tunable model parameters are S_I^T, S_I^D, S_I^E, EGP_0, F_{01} and $t_{max,I}$. The numerical values of the model parameters are summarized in Table 1.

 doi:10.20944/preprints202604.1316.v1

Table 1. Model parameters.

Parameter	Symbol	Unit	Value
Glucose distrib. volume	V_G	L	0.16 BW
Insulin distrib. volume	V_I	L	0.12 BW
Non-insulin glucose flux	F_{01}	mmol/min	0.0097 BW
Transfer rate from Q_2 to Q_1	k_{12}	1/min	0.0066
Deactivation rate	$k_{a,1}$	1/min	0.006
Deactivation rate	$k_{a,2}$	1/min	0.06
Deactivation rate	$k_{a,3}$	1/min	0.03
Insulin sens. of glucose transport	S_I^T	L/min/mU	51.2×10^{-4}
Insulin sens. of glucose distribution	S_I^D	L/min/mU	8.2×10^{-4}
Insulin sens. of EGP	S_I^E	L/min/mU	520×10^{-4}
EGP at 0 insulin	EGP_0	mmol/min	0.0161 BW
Carbohydrate bioavailability	A_G	-	0.8
Time to max carbohydrate	$t_{max,G}$	min	40
Time to max insulin	$t_{max,I}$	min	55
Insulin elimination from plasma	k_e	1/min	0.138

2.2. Fuzzy Logic Control Basics

Because this paper is offering a new way of implementing a fuzzy inference controller by embedding linguistic variables in the LLM we look briefly over principles behind a Mamdani-type fuzzy controller. The Mamdani-type controller is known as universal fuzzy controller as proved by [51,52]. The control action is obtained as state feedback

$$u(t) = g(x(t)) = \sum_{l=1}^{m} g_l(x(t))\mu_l(x(t)) \tag{11}$$

where the non-linear function $g(\bullet)$ over n dimensional state space $\mathbb{R}^n$ is defined though application of m fuzzy inference rules

$$R^l: \text{ IF } x_1 \text{ is } F_1^l \text{ AND } \ldots \; x_n \text{ is } F_n^l \text{ THEN } u(t) = g_l(x(t)), \quad l = 1 \ldots m \tag{12}$$

$S_l = \prod_{i=1}^{n} F_i^l$ defines the lth fuzzy set, g_l is the lth local control law, $\mu_l(\bullet)$ is normalized membership function for the inferred fuzzy set S_l. For example

$$\mu_l(x) = \min_i \{\mu_i^l(x_1) | i = 1 \ldots n\} \tag{13}$$

where $\mu_i^l(\bullet)$ is the membership functions describing the fuzzy set F_i^l. In such inference the control signal $u(t)$ is obtained as fuzzy variable with different membership values to m output sets as (2.2) shows.

Equivalently for a finite dimensional system, the state of the system can be reconstructed by looking at n past values of output and input so we can have

$$x(t) \cong (G(t), G(t - T_s), \ldots, G(t - n\, T_s), u(t), u(t - T_s), \ldots, u(t - n\, T_s))^T \tag{14}$$

 doi:10.20944/preprints202604.1316.v1

For examination of the insulin delivery system we define 3 fuzzy sets for the glucose level corresponding to hypoglycemic (90-105 *mg/dl*), target (90-180 *mg/dl*) and hyperglycemic (150-300 *mg/dl*) regions (Figure 1) with

$$\begin{aligned} G_{hypo}: \ \mu_{hypo}(G) &= \max\Big(\min\Big(-\tfrac{18G-105}{15}, 1\Big), 0\Big) \\ G_{target}: \ \mu_{target}(G) &= \max\Big(\min\Big(\min\Big(\tfrac{18G-90}{15}, \tfrac{180-18G}{75}\Big), 1\Big), 0\Big) \\ G_{hyper}: \ \mu_{hyper}(G) &= \max\Big(\min\Big(\tfrac{18G-150}{150}, 1\Big), 0\Big) \end{aligned} \tag{15}$$

The insulin dose fuzzy sets describe the possible minute infusion per kg of body weight as zero,low and high dose (Figure 2) with

$$\begin{aligned} U_z: \ \mu_z(u) &= \max\big(\min\big(-\tfrac{u}{0.01}, 1\big), 0\big) \\ U_l: \ \mu_l(u) &= \max\Big(\min\Big(\min\Big(\tfrac{u}{0.01}, \tfrac{0.1-u}{0.99}\Big), 1\Big), 0\Big) \\ U_h: \ \mu_h(u) &= \max\Big(\min\Big(\tfrac{u-0.1}{1.4}, 1\Big), 0\Big) \end{aligned} \tag{16}$$

Therefore a simplified version of insulin controller would be defined as

$$\begin{aligned} R^1: &\text{ IF } G(t) \text{ is } G_{hypo} \text{ THEN } u(t) \text{ is } U_z \\ R^2: &\text{ IF } G(t) \text{ is } G_{target} \text{ THEN } u(t) \text{ is } U_l \\ R^3: &\text{ IF } G(t) \text{ is } G_{hyper} \text{ THEN } u(t) \text{ is } U_h \end{aligned} \tag{17}$$

Then the control signal fuzzy set $U_{out} = \sum_i U_i$ defined with

$$U_{out}: \ \mu_{out}(u) = \mu_z(u)\mu_{hypo}(G(t)) + \mu_l(u)\mu_{target}(G(t)) + \mu_h(u)\mu_{hyper}(G(t)) \tag{18}$$

A centroid defuzzification procedure is applied to obtain the crisp value of control signal

$$u(t) = BW \frac{\int_0^{u_{max}} u \ \mu_{out}(u) du}{\int_0^{u_{max}} \mu_{out}(u) du} \tag{19}$$

or after discretization with step h such that $u_i \in \{0, h, 2h, 3h, \ldots, u_{max}\}$

$$u(t) = BW \frac{\sum_i u_i \ \mu_{out}(u_i)}{\sum_i \mu_{out}(u_i)} \tag{20}$$

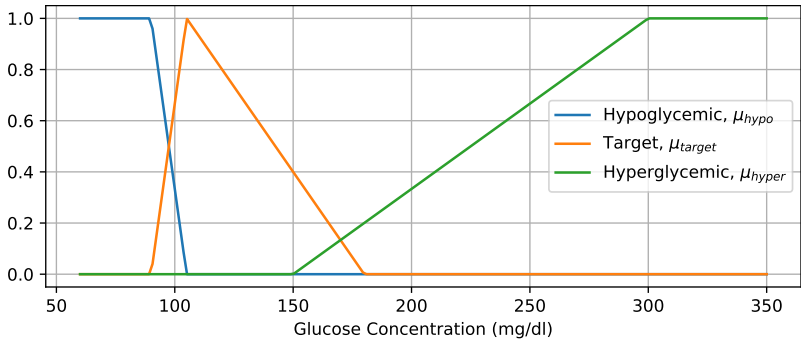

Figure 1. Membership functions for glucose concentration fuzzy sets.

 doi:10.20944/preprints202604.1316.v1

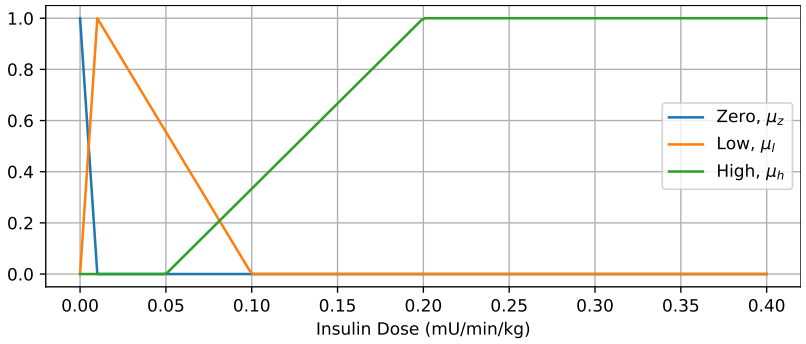

Figure 2. Membership functions for insulin dosage fuzzy sets.

2.3. Large Language Models

A paper from [45] defines the transformers architecture as optimization over preceding encoder/decoder recurrent networks used for language to language translation. The transformer is performing block correlation analysis over window of input tokens without reliance on any recurrent connections, which greatly improves model training performance for next token prediction task. Here we focus on few architecture decisions with reference to Llama group of LLM which are described as decoder only models.

The input to the language processing is a token sequence $w_{tok} = (w_1, w_2, \ldots, w_T)$. Depending on language model architecture the tokens can be whole words, workpieces or even sentences formed in natural language. Following tokenization is mapping of each token w_i into an embedding vector e_i by the embedding layer of the LLM without positional information. For TinyLlama model the embedding dimension is $d = 2048$ and token dictionary size is $N_{tok} = 32000$. Resultant token sequence after embedding is

$$e = (e_1, e_2, \ldots, e_T) \in \mathbb{R}^{d \times T}. \tag{21}$$

Such high dimensional representation is a key to distinguishing meaning of various tokens and to allow freedom in internal transformations by the model. A general observation is that tokens with more distinct or opposite meanings map to embedding vectors with larger cosine distance while the tokens with similar meaning cluster together in the $\mathbb{R}^d$.

At the core of LLM performance is self-attention mechanism which works as analogy to database query over index key. After producing query, key and value mappings of each input sequence token

$$\begin{aligned} q_m &= f_q(e_m, m) \\ k_n &= f_k(e_n, n) \\ v_n &= f_v(e_n, n) \end{aligned} \quad , \tag{22}$$

the output of the l-th hidden model layer in the transformer LLM is a function over its input sequence

$$h_l(e) = (h_{l,1}, h_{l,2}, \ldots, h_{l,T}) \tag{23}$$

where the components $h_{l,m}$ is computed as a weighted sum of the values v_n

$$h_{l,m}(e) = \sum_{n=1}^{T} a_{m,n}\, v_n, \tag{24}$$

and the weight $a_{m,n}$ of each value is reflecting the matching degree between query q_m and key k_n pairs, which is expressed as a scalar product $\langle q_m, k_n \rangle$

$$a_{m,n} = f_{softmax}(\langle q_m, k_1 \rangle, \ldots, \langle q_m, k_T \rangle) = \frac{e^{\langle q_m, k_n \rangle}}{\sum_{i=1}^{T} e^{\langle q_m, k_i \rangle}}. \tag{25}$$

 doi:10.20944/preprints202604.1316.v1

The scalar product between query and key sequences is how the information transfer between tokens at different positions happens in LLM. As can be seen in (22) the maps f_m, f_k and f_v require to incorporate position information of the m-th and n-th tokens in the sequence such that attention scores are increasing for the key values k_n, which are situated closer to a m-th query position q_m. Therefore the scalar product in attention scores calculation must encode explicitly the position information

$$\langle q_m, k_n \rangle = g(q_m, k_n, m-n) \tag{26}$$

A distinguishable characteristic of Llama models [53] is application of rotation matrix transformation $R^d_{\Theta,m}$ over the embedding vectors to reflect positional information [54] as

$$\tilde{e}_m = R^d_{\Theta,m} e_m, \tag{27}$$

where the d-dimensional vector space is decomposed into direct sum of $d/2$ two dimensional spaces and two dimensional rotation is applied in each of the subspaces

$$R^d_{\Theta,m} = diag\left(\begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 \\ \sin m\theta_1 & \cos m\theta_1 \end{pmatrix}, \ldots, \begin{pmatrix} \cos m\theta_{d/2} & -\sin m\theta_{d/2} \\ \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{pmatrix} \right), \tag{28}$$

where $\Theta = \{\theta_i = 10000^{-2(i-1)/d}, i = 1 \ldots d/2\}$. Therefore the input embedding vector components are rotated to angles proportional to their position in the sequence and to their dimension. The transformation functions of a hidden layer become

$$f_q(e_m, m) = R^d_{\Theta,m} W_q, \quad f_k(e_n, n) = R^d_{\Theta,n} W_k, \quad f_v(e_n, n) = R^d_{\Theta,n} W_v \tag{29}$$

and the scalar product function is

$$\langle q_m, k_n \rangle = \frac{(R^d_{\Theta,m} W_q e_m)^T (R^d_{\Theta,n} W_k e_n)}{\sqrt{d}} = \frac{e_m^T W_q^T \, R^d_{\Theta,m-n} W_k e_n}{\sqrt{d}} \tag{30}$$

with guaranteed relative position sensitivity. In practical implementation of Llama models where multiple hidden layers $h_{l,m}$ are stacked together, the rotational encoding is applied before the first hidden layer, and is followed by a normalization layer [55], which is related to training performance. Another feature of LLM is multi-head attention, where the input embedding space is decomposed into few lower dimensional components and attention weights are applied to each component separately, followed by concatenation of the results.

In our application we look into causal language modelling framework, where every processed token from the input window is allowed to attend only to the preceding ones through application of causal mask over attention weights. The mask is upper triangular matrix with entries

$$u_{i,j} = \begin{cases} 0, & i \geq j \\ -\infty, & i < j \end{cases} \tag{31}$$

The final layer of the transformer is a projection layer $L_{tok} : \mathbb{R}^d \to \mathbb{R}^{N_{tok}}$ mapping dimensions of embedding space into token scores. After the composition of L_{tok} with the transformation in embedding space performed by each hidden layer $h_l : \mathbb{R}^d \to \mathbb{R}^d$ over positionally encoded input sequence $\tilde{e}$ we have

$$y = (L_{tok} \circ h_{n_l} \circ h_{n_l - 1} \circ \ldots \circ h_1)(\tilde{e}), \tag{32}$$

where

$$y = (y_1, \ldots, y_T) \in \mathbb{R}^{N_{tok} \times T} \tag{33}$$

 doi:10.20944/preprints202604.1316.v1

is an output sequence of scores over model vocabulary. Each output vector $y_i \in \mathbb{R}^{N_{tok}}$ from the output sequence y of the LLM is containing class scores $y_{i,j}$ over model vocabulary of N_{tok} tokens, which allows classification for example by

$$y_{i,tok} = \text{argmax}_j(y_{i,j}) \in [1, N_{tok}], \tag{34}$$

which returns the index of the token where output scores are maximized.

The LLM are trained over large dataset of plain text, programming code and conversational data by looking to maximize probability $p(w_{T+1}|w_{1:T})$ for correct prediction of next token in a sequence based on the window of previous tokens, which corresponds to maximizing the scores for $y_{T,w_{T+1}}$. In practice this is performed by calculating cross entropy loss function

$$l(y_i, w_{i+1}) = -\log \frac{e^{y_{i,w_{i+1}}}}{\sum_{j=1}^{N_{tok}} e^{y_{i,j}}} \tag{35}$$

between output scores y_i and target token index w_{i+1} obtained from one step ahead shifted version of the input sequence w_{tok}.

In generation mode, the LLM decoder operates in an autoregressive manner. After initialized with a starting sequence w_{tok}, at each generation step the predicted token $\hat{w}_{T+1} = y_{T,tok}$ is appended to the input sequence. The process is repeated either for predefined maximal number or steps or till a special end of string token is produced. Therefore, for the following analysis we primary look into last embedding from the sequence $y_T = y_{-1} \in \mathbb{R}^d$ which can be expressed as

$$y_{-1} = F_{LLM}(e), \quad F_{LLM} : \mathbb{R}^{d \times T} \to \mathbb{R}^d \tag{36}$$

3. Proposed LLM-Fuzzy Framework

The key problem in using LLM directly in control feedback loop are unmatched signals domains. While control plant is operating as a functional mapping between signal spaces

$$G = \mathcal{F}_{Hov}(u), \quad u : \mathbb{R} \to \mathbb{R}, \quad G : \mathbb{R} \to \mathbb{R}, \tag{37}$$

the LLM is operating as a functional mapping over a high-dimensional space like $\mathbb{R}^{d \times T}$ of word embedding sequences with length T as (36). We are looking for natural connection between physical signals and high dimensional LLM-specific embeddings.

A trivial approach would be to convert numerical signal level to a string of model recognized tokens, which for $G(t)$ would look like

$$w_G(t) = (w_{d_1}, w_{d_2}, w_{d_3}, \ldots, w_{d_p}, w_{d_{-1}}, w_{d_{-2}}, \ldots, w_{d_{-q}}), \tag{38}$$

where

$$G(t) \approx \overline{d_1 d_2 \ldots d_p . d_{-1} d_{-2} \ldots d_{-q}} \tag{39}$$

is the decimal representations of signal level $G(t)$ using p digits for the integer part and q digits for the fractional part, and w_{d_i} is the LLM token corresponding to a digit d_i. The token sequence $w_G(t)$ after converted to a sequence of embedding vectors $e_{G,i}(t)$, can be fed into the LLM model prompt template. However, a known limitations exists in LLM to perform precise mathematical calculation. The processing of a tokenized decimal representation $w_G(t)$ through the model will treat the represented quantity $G(t)$ no different than other general character sequence and will calculate attention scores $a_{m,n}$ to perform a weighted average of value sequence v_n. This is not a conventional algebraic processing of $G(t)$ but instead a correlation analysis on string of digits. So if we want the model to calculate the control action

$$u(t) = \overline{d_{u,1} \ldots d_{u,p} . d_{u,-1} \ldots d_{u,-q}} \tag{40}$$

 doi:10.20944/preprints202604.1316.v1

will be generated digit by digit in probabilistic manner by sampling a distribution

$$p(d_{u,i}|d_{u,i-1}\ldots,d_1,\ldots,d_p,d_{-1},\ldots,d_{-q}). \tag{41}$$

Alternatively, in this study, we propose that token meaning is primary encoded as embedding vector direction obtained as

$$\hat{e}_i = e_i/\|e_i\|, \tag{42}$$

where $\|e_i\|$ is the 2-norm of the embedding vector. Therefore, embedding vectors $k_1\hat{e}_i$ and $k_2\hat{e}_i$, which point in the same direction, but have different lengths, still identify same token from the input vocabulary in maximum likelihood sense. Or formally, for every embedding vector $e_i \in \mathbb{R}^d$, there exists a positive $\epsilon(e_i) > 0$ such that for all $\delta < \epsilon(e_i)$

$$\min_j \|(1-\delta)e_i - e_j\|, \quad j \in 0\ldots N_{tok} \tag{43}$$

is obtained for $i = j$, i.e. no other token embedding vector is closer to the rescaled vector $(1-\epsilon)e_i$. Therefore, in such framework, the instantaneous amplitude of an observable signal $G(t)$ can be encoded through proportionally scaling a selected token from vocabulary e_i by

$$e_G(t) = \left(1 - \frac{G_{max} - G(t)}{G_{max} - G_{min}}\epsilon(e_i)\right)e_i. \tag{44}$$

As can be seen, this representation requires a single *carrier token* e_i to represent the instantaneous level of a signal, compared to a decimal-based token representation (38). Obviously, such encoding will create a perturbation $\epsilon(e_i)$ in the pre-learned embeddings, so a fine-tuning of the modified model will be required.

To generalize further, let $e_{i_1}, e_{i_2}, \ldots, e_{i_n} \in \mathbb{R}^d$ are selected embedding vectors. Then, there exists corresponding perturbation bounds $\epsilon(e_{i_k}) > 0$ such that for any $\delta_k < \epsilon(e_{i_k})$ a linear combination between vectors with $1-\delta_k$ when compared to the embedding vectors from the learned vocabulary with

$$\min_j \left\| \sum_{k=1}^{n} (1-\delta_k)e_{i_k} - e_j \right\|, \quad j \in 0\ldots N_{tok} \tag{45}$$

obtains a minimum for $j \in \{i_1, i_2, \ldots, n\}$. Therefore, the meaning of the linear combination of embedding vectors is still closer to any of the vectors in the linear combination, compared to the rest of learned embeddings.

A question remains, how to select a *carrier token* $w_{ct,G}$ corresponding for a given measurable quantity. A natural choice is to have the name (e.g. *blood glucose*) or physical unit(e.g. *mg/dL*) of the quantity as a token carrier. The benefit from such a choice is in the conditioning of the pre-learned by the model probability distribution function $p(w_{T+1}|w_{1:T}, w_{ct,G})$ on an application specific knowledge from the physical domain where the quantity belongs. The final outcome will be increased scores for appropriate output tokens corresponding to the same physical domain. If $w_{ct,u}$ is the *carrier token* of the LLM generated output signal $u(t)$ (e.g. *insulin dose*) then we assume

$$p(w_{ct,u}|w_{1:T}, w_{ct,G}) > p(w_{ct,u}|w_{1:T}), \tag{46}$$

meaning that output scores of $w_{ct,u}$ will increase if we use a contextual input carrier token. Of course, such correlation is not guaranteed, but can be tracked by initial experiments with the model when selecting appropriate input and output carrier tokens. Higher scores of the output token will accelerate consequent fine tuning of the model by requiring less training steeps in direction of the token.

 doi:10.20944/preprints202604.1316.v1

3.1. Fuzzy Embeddings

The fuzzy logic is a well founded control framework enabling encapsulation of expert knowledge expressed in relative linguistic terms amounting to non-linear controllers. In the proposed Fuzzy-LLM framework the linguistic terms describing fuzzy logic sets allow fine grained selection of multiple meaningful carrier tokens, and also giving a natural rules for scaling the corresponding embedding vectors through calculated membership values, instead to absolute physical values.

The first step is to encode fuzzy linguistic terms $\{l_1, l_2, \ldots, l_n\}$ for the input and output system variables into carrier token sequences recognizable from LLM as $w_{tok,l_i} = (w_1, w_2, \ldots, w_{T_{l_i}})$. The selected linguistic terms and their carrier encodings for the described fuzzy sets in 2.2 are given in Table 2. Note that some of the terms are encoded with a single token, while others with 2,4 or 5 tokens, which depends on model vocabulary. Modification or extension on the model vocabulary is also possible and requires a dedicated fine-tuning process.

Table 2. Fuzzy Carrier Tokenization with Padding.

Fuzzy Set	Term	Token IDs	Tokens	Embedding
G_{hypo}	hypoglycemia	10163, 468, 368, 19335, 423	hyp-og-ly-cem-ia	$e_{hypo} \in \mathbb{R}^{2048\times5}$
G_{target}	in range	297, 3464, 2, 2, 2	in -range	$e_{target} \in \mathbb{R}^{2048\times5}$
G_{hyper}	hyperglycemia	11266, 16808, 19335, 423, 2	hyper-gly-cem-ia	$e_{hyper} \in \mathbb{R}^{2048\times5}$
U_z	zero	5225	zero	$e_{zero} \in \mathbb{R}^{2048}$
U_l	low	4482	low	$e_{low} \in \mathbb{R}^{2048}$
U_h	high	1880	high	$e_{high} \in \mathbb{R}^{2048}$

'2' is padding token id in TinyLLama tokenizer.

The next step in preparation of model input is calculation of the embedding vectors for each token $e_{l_i} = (e_1, e_2, \ldots, e_{T_{l_i}}) \in \mathbb{R}^{d\times T_{l_i}}$. However in the proposed approach the sequences of embedding vectors corresponding to each input fuzzy set are scaled with corresponding membership functions to produce

$$e_{glu}(G) = \mu_{hypo}(G)\, e_{hypo} + \mu_{target}(G)\, e_{target} + \mu_{hyper}(G)\, e_{hyper} = E_{c,glu}\, \vec{\mu}_g(G), \tag{47}$$

where $E_{c,glu} = (e_{hypo}, e_{target}, e_{hyper}) \in \mathbb{R}^{d\times3}$ is a matrix of carrier tokens used to capture glucose level and $\vec{\mu}_g = (\mu_{hypo}, \mu_{target}, \mu_{hyper})^T$. The embedding sequence $e_{glu}(G) \in \mathbb{R}^{d\times5}$ is fuzzified representation of the numerical glucose concentration value in model embedding space as weighted sum of embedding vectors corresponding to input terms, where the weights are obtained from fuzzy set membership functions.

Similarly the output insulin dose can also be represented as weighted sum in input embedding space

$$e_{ins}(u) = \mu_z(u)\, e_{zero} + \mu_l(u)\, e_{low} + \mu_h(u)\, e_{high} = E_{c,ins}\, \vec{\mu}_i(u). \tag{48}$$

where $E_{c,ins} = (e_{zero}, e_{low}, e_{high}) \in \mathbb{R}^{d\times3}$ is a matrix of carrier tokens used to capture glucose level and $\vec{\mu}_i = (\mu_z, \mu_l, \mu_h)^T$. However because the embeddings of the insulin dose terms are one dimensional we have $e_{ins}(u) \in \mathbb{R}^d$.

3.2. LLM for Fuzzy Inference

Let start by defining input sequence w_{tok} to the TinyLLama model following conversational template using <|system|>, <|user|> and <|assistant|> special token as given in Listing 1. The <|system|> section is used to set general instructions the model, the <|user|> section defines user query, <|assistant|> section is framing the expected predictions and </s> is special token denoting end-of-string token.

 doi:10.20944/preprints202604.1316.v1

Listing 1. Tiny Llama prompt for fuzzy decision making.

<|system|> You are an automatic insulin delivery advisor for type 1 diabetes.</s> <|user|> Historical glucose: <G10><G9><G8><G7><G6><G5><G4><G3><G2><G1>. Historical insulin dosages: <U10><U9><U8><U7><U6><U5><U4><U3><U2><U1></s> <|assistant|> Next insulin dose: <U>

In this prompt we have placeholders <Gi> and <Ui> for the past 10 measured glucose concentrations and past applied insulin dosages and for the final dose <U>, which will be predicted by the model. These placeholders are not actually filled in textual form because proposed fuzzy embedding scheme is not offering reverse textual mapping but works directly in embedding space. Therefore we map the sections from the prompt 1 into embedding space where the placeholders for <Gi> and <Ui> are filled by concatenation with fuzzy embedding of past glucose and insulin dose values.

$$e_{in}(\vec{G},\vec{u}) = (e_{pre,1}, e_{glu,10}(\vec{G}), e_{pre,2}, e_{ins,10}(\vec{u}), e_{post}) \in \mathbb{R}^{d\times T}, \tag{49}$$

where $e_{pre,1}$ are prompt embeddings from beginning of the prompt up to placeholder for <G10>, $\vec{G}$ and $\vec{u}$ are vectors of past 10 glucose measurements and insulin dosages. Then the sequence

$$e_{glu,10}(\vec{G}) = (\ e_{glu}(G(t-9T_s)), e_{glu}(G(t-8T_s)), \ldots, e_{glu}(G(t))\ ), \tag{50}$$

where T_S is sampling period of AP controller. The $e_{pre,2}$ are prompt embeddings after placeholder for <G1> up to placeholder for <U10>, then

$$e_{ins,10}(\vec{u}) = (\ e_{ins}(u(t-10T_s)), e_{ins}(u(t-9T_s)), \ldots, e_{ins}(u(t-T_s))\ ), \tag{51}$$

and e_{post} are prompt embeddings after placeholder for <U1> up to placeholder for <U>. The overall prompt can be expressed as composition of fixed and variable part as

$$e_{in} = P_{glu}\, e_{glu,10} + P_{ins}\, e_{ins,10} + e_{fix}, \tag{52}$$

where $P_{glu}, P_{ins} : \mathbb{R}^{d\times 10} \to \mathbb{R}^{d\times T}$ are linear projection operators with appropriate tensor representations.

The output of the model after processing the input embedding sequence $e_{in}(\vec{G},\vec{u})$

$$y(\vec{G},\vec{u}) = (L_{tok} \circ h_{n_l} \circ h_{n_l-1} \circ \ldots \circ h_1)(\tilde{e}_{in}(\vec{G},\vec{u})). \tag{53}$$

And the embedding vector corresponding to the prediction for token <U> is

$$y_{-1}(\vec{G},\vec{u}) = F_{LLM}(e(\vec{G},\vec{u})) \in \mathbb{R}^d, \tag{54}$$

which is the last column from the sequence matrix $y(\vec{G},\vec{u})$. The membership values of this token to output fuzzy sets is obtained after applying projection to carrier token space

$$\bar{\mathbf{y}}_{-1} = I_u\, \mathbf{y}_{-1} \tag{55}$$

where

$$I_u = \begin{pmatrix} \mathbf{0}_{1,N_z-1} & 1 & \mathbf{0}_{1,N_{tok}-N_z-1} \\ \mathbf{0}_{1,N_l-1} & 1 & \mathbf{0}_{1,N_{tok}-N_l-1} \\ \mathbf{0}_{1,N_h-1} & 1 & \mathbf{0}_{1,N_{tok}-N_h-1} \end{pmatrix} \in \mathbb{R}^{3\times N_{tok}}, \tag{56}$$

 doi:10.20944/preprints202604.1316.v1

where $N_z = 5225$, $N_l = 4482$ and $N_h = 1880$ are the token indices for the output fuzzy terms according to Table 2. After softmax operation $f_{softmax}$ output scores are converted to normalized values

$$\mathbf{p}_u = f_{softmax}(\bar{\mathbf{y}}) = \frac{1}{e^{-y_{-1,N_z}} + e^{-y_{-1,N_l}} + e^{-y_{-1,N_h}}} (e^{-y_{-1,N_z}}, e^{-y_{-1,N_l}}, e^{-y_{-1,N_h}})^T = (p_z, p_l, p_h)^T, \quad (57)$$

which can be also interpreted as conditional probability distribution $p(|e_z, e_l, e_h)$.The resultant output fuzzy set membership function becomes

$$U_{out}: \ \mu_u(w) = \mu_z(w)p_z + \mu_l(w)p_l + \mu_h(w)p_h = \mathbf{p}_u.\vec{\mu}_i(w), \quad (58)$$

which allows the computation of new insulin dose $u(t)$ with (19) and (20).

As can be seen the proposed representation scheme for the numerical values of the signals as linear combinations of embedding vectors matches the natural internal representation of the linguistic information in the language model. However, the functional relationship between degree of amplification of input fuzzy embeddings into output class scores is not established and needs to be learned in the model by proper training loop.

3.3. Closed-Loop System with LLM

The components of the closed loop system with the Fuzzy-LLM controller are summarized in Figure 3. As presented the Hovorka metabolic model for type 1 diabetes is calculating a glucose concentration signal $G(t)$ as a dynamic function on insulin dose $u(t)$ and ingested meal. Let the state of the Hovorka model be given in a state vector

$$x = (Q_1, Q_2, x_1, x_2, x_3, S_1, S_2, I)^T, \quad (59)$$

then the dynamics of metabolic model is represented in state space form as

$$\begin{aligned} \dot{x}(t) &= H(t, x(t), u) \\ G(t) &= Q_1(t)/V_G = C_Q\, x(t). \end{aligned} \quad (60)$$

where $C_Q = (1/V_G, \mathbf{0}_{1,7})$.

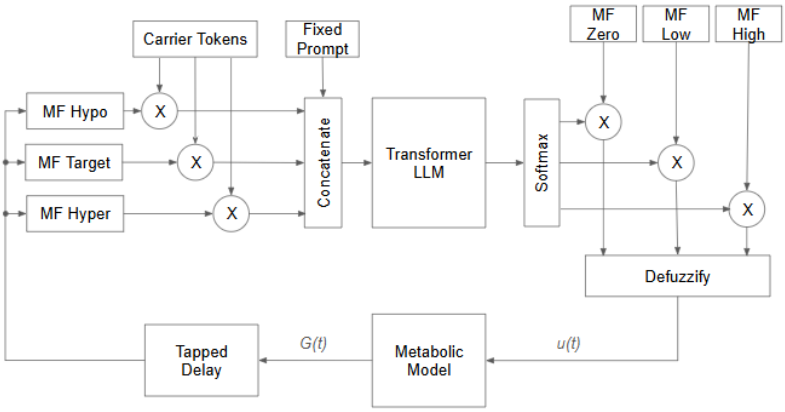

Figure 3. Closed-loop system diagram.

The full controller expression is obtained by composition of fuzzy embedding operation $e_{glu}(\bullet)$, tapped delay operator, concatenation with fixed prompt, application of llm function $F_{LLM}(\bullet)$, token selection projection I_u, softmax operation $f_{softmax}(\bullet)$, output membership function generation and defuzzifying. This can be expressed in compact form as

$$u(t) = \frac{\int_0^{u_{max}} w\ \mathbf{p}_u.\vec{\mu}_i(w)dw}{\int_0^{u_{max}} \mathbf{p}_u.\vec{\mu}_i(w)dw}, \quad (61)$$

 doi:10.20944/preprints202604.1316.v1

$$\mathbf{p}_u = f_{softmax}(I_u\, F_{LLM}(P_{glu}\tau_{10}(E_{c,glu}\vec{\mu}_g(G(t))) + P_{ins}\tau_{10}(E_{c,ins}\vec{\mu}_i(u(t))) + e_{fix}, \pi)) \tag{62}$$

where $\tau_k(\bullet)$ is tapped delay operator over the embedding vector space

$$\tau_k(e(t)) = (e(t-(k-1)T_S), e(t-(k-1)T_S), \ldots, e(t-T_S)), \tag{63}$$

and $\pi = \{\pi_i | i = 1 \ldots N_{par}\}$ is the vector of tunable parameters LLM model LoRA adapter.

3.4. Training Objective

It is known that optimal BG levels are around $G_{ref} = 105$ mg/dL. Therefore, we define a quadratic cost of population worst case deviation from the target for the current time period t as

$$J(\pi) = \max_{s_i \in S} \frac{1}{2}(G(t + kT_S, \pi(t), s_i) - G_{ref})^2, \tag{64}$$

where $s_i \in S$ is a subject from an examined T1D population characterized in general with specific initial state conditions, specific meal times and meal amounts, specific parameter settings for the glucose metabolic model, and $k = const$ is a fixed extrapolation horizon. If we aim to minimize $J(\pi)$ using a gradient descent algorithm we calculate

$$\pi(t) = \pi(t - T_S) - \alpha(t)\frac{\partial J}{\partial \pi}\bigg|_{\pi(t-T_S)}, \tag{65}$$

where the gradient of $J(\pi)$ with respect to each of tunable parameters π_i is

$$\frac{\partial J}{\partial \pi_i} = (G_k(\pi_i, s^*(t)) - G_{ref})\frac{\partial G_k}{\partial \pi_i}, \quad G_k = G(t + kT_S), \tag{66}$$

where $s^*(t) \in S$ is the subject from the population, where the maximum quadratic deviation is obtained at the time instant t. Therefore, the subject with worst deviation from the target is driving the fine-tuning of the LLM during the current time instant.

For a fixed $u(t) = u_0$ the solution of the state equation for k steps forward in time with T_S sample period can be approximated with Euler formula

$$x_k = x_{k-1} + T_S\, H(t_k, x_{k-1}, u_k), \tag{67}$$

where $x_k = x(t + kT_S)$, $x_{k-1} = x(t + (k-1)T_S)$, $t_k = t + kT_S$, $u_k = u(kT_S)$ and $t_{k-1} = t + (k-1)T_S$. The derivative of extrapolated G_k with respect to parameter π_i becomes

$$\frac{\partial G_k}{\partial \pi_i} = C_Q \frac{\partial x_k}{\partial u_0}\frac{\partial u_0}{\partial \pi_i}, \tag{68}$$

exposing sensitivity of extrapolated glucose to current control action and sensitivity of current control action to LLM parameters.

Without loss of generality, taking $t = 0$ for current time instant, $t > 0$ for future (extrapolated) time instants, and assuming $u_k = u_0$ for $k > 0$, then derivative of extrapolated state for k steps ahead with respect to current input signal u_0 is

$$\frac{\partial x_k}{\partial u} = \left(1 + T_S\frac{\partial H}{\partial x}\bigg|_{p_0}\right)\frac{\partial x_{k-1}}{\partial u}\bigg|_{p_0} + T_S\frac{\partial H}{\partial u}\bigg|_{p_0}, \tag{69}$$

where $p_0 = (t_k, x_{k-1}, u_0)$. This recursive relation expands to

$$\frac{\partial x_k}{\partial u}\bigg|_{p_0} = \left(1 + T_S\frac{\partial H}{\partial x}\bigg|_{p_0}\right)^k \frac{\partial x_0}{\partial u}\bigg|_{p_0} + \sum_{i=0}^{k-1}\left(1 + T_S\frac{\partial H}{\partial x}\bigg|_{p_0}\right)^i T_S\frac{\partial H}{\partial u}\bigg|_{p_0}, \tag{70}$$

 doi:10.20944/preprints202604.1316.v1

but because $\partial x_0/\partial u_0 = 0$

$$\left.\frac{\partial x_k}{\partial u}\right|_{p_0} = \sum_{i=0}^{k-1}\left(1 + T_S \left.\frac{\partial H}{\partial x}\right|_{p_0}\right)^i T_S \left.\frac{\partial H}{\partial u}\right|_{p_0}. \tag{71}$$

On the other hand the control derivative term in (68)

$$\frac{\partial u}{\partial \pi_i} = \frac{\int_0^{u_{max}} w \, \frac{\partial \mathbf{p}_u}{\partial \pi_i}.\vec{\mu}_i(w)dw \int_0^{u_{max}} \mathbf{p}_u.\vec{\mu}_i(w)dw - \int_0^{u_{max}} w \, \mathbf{p}_u.\vec{\mu}_i(w)dw \int_0^{u_{max}} \frac{\partial \mathbf{p}_u}{\partial \pi_i}.\vec{\mu}_i(w)dw}{\left(\int_0^{u_{max}} \mathbf{p}_u.\vec{\mu}_i(w)dw\right)^2} \tag{72}$$

where

$$\frac{\partial \mathbf{p}_u}{\partial \pi_i} = f'_{softmax}\left(I_u \left.\frac{\partial F_{LLM}}{\partial \pi_i}\right|_{e_{in}(t),\pi_i(t)}\right), \tag{73}$$

where $f'_{softmax}(\bullet)$ is the derivative of the softmax function and

$$e_{in}(t) = P_{glu}\tau_{10}(E_{c,glu}\vec{\mu}_g(G(t))) + P_{ins}\tau_{10}(E_{c,ins}\vec{\mu}_i(u(t))) + e_{fix} \tag{74}$$

is the fixed prompt embedding being function on the historic glucose and insulin values.

Therefore, the parameters of the LLM π_i are tuned only with respect to current control signal $u(t)$ to optimize its impact on the extrapolated glucose concentration for $k > 0$ steps ahead of current time t. For k we can take longer period between 1 to 5 hours, which is compatible with residual insulin action to boost the sensitivity of the G_k to $u(t)$. Otherwise, if k is small this sensitivity may vanish and compromise training convergence.

Important note on the cost $J(\pi)$ is that it is atypical for fine tuning LLM where usually token classification is aimed, hence, cross-entropy loss is employed. Contrary, in our application we look for closed loop performance with respect to inferred from LLM signal levels. We've also experimented with hybrid loss accounting for suggested fuzzy rule classification scheme through addition of cross entropy term, however the results were not as good for closed loop performance in this case.

4. Fine-Tuning Implementation

In this section we give the specifics around training loop implementation. To setup the training a couple of components must interact. First, Hovorka metabolic model have to be implemented in vectorized form in GPU to allow parallel simulation of multiple virtual patients. Then input prompt for the LLM needs to be constructed according to proposed fuzzy embedding schema. Appropriate adapter needs to be initialized for LLM fine tuning. The software components from the training phase a then reused during model inference. For the inference a server module is initialized to allow remote calculation of insulin dosage for multiple patients, which will be used for the simulation with the UVa/Padova. Note: some of the squeeze and repeat operations are omitted in the provided code snippets for clarity.

4.1. Metabolic Model in GPU

In Hovorka model a virtual subjects are parametrized using body weight parameter BW, which is initialized as torch vector. The initial state is also initialized and repeated for number of virtual subjects, which identifies the batch size. A predefined vector of three meal portions are scheduled for 8, 12 and 19 o'clock on daily basis (Listing 2).

 doi:10.20944/preprints202604.1316.v1

Listing 2. Virtual patients parametrization.

```
BW = torch.tensor([70., 75. ...])
state = torch.tensor(init_state).repeat(batch_size,1)
# ingestion time, min/ Digested CHO mmol
meals = [(8*60,250), (12*60,250), (19*60,250)]
```

The model derivatives are calculated according to the presented equations in section 2.1. Since the model parameters are one dimensional torch tensors with length `batch_size`, the vectorized model states become a two dimensional torch tensor `state` with dimension `batch_size`×`Nstate`. Internally, each state is assigned to a variable used in calculating derivatives along batch dimensions. Then the derivatives are stacked in a tensor along the state dimension (Listing 3).

In Listing 4 we show the Euler integration of Hovorka model derivatives. The current `state` is updated with the latest calculated control signal. On the other hand, state extrapolation is carried out for a constant input signal. Note that tensor `state` is detached at each iteration from torch autograd graph.

Listing 3. Model derivative calculation.

```
def ap_model(t,state,uin):
    VG = 0.16*BW
    ...
    Q1 = state[:,0]
    Q2 = state[:,1]
    ...
    G = Q1/VG # mmol/L
    ...
    Q1dot = EGP0*(1.0-x3) + UG - FR - (x1+F01c/(VG*G))*Q1 + k12*Q2
    ...
    dstate = torch.hstack([
        Q1dot.unsqueeze(1),
        Q2dot.unsqueeze(1),...])
    return (dstate,G)
```

Listing 4. Hovorka model integration.

```
(dstate,G) = ap_model(tt,state,u_new)
state = state + dstate*Ts

state_extrap = state
for k in range(0,Textrap):
    (dstate,Gk) = ap_model(tt,state_extrap,u_new)
    state_extrap = state_extrap + dstate*Ts
```

4.2. LoRA Configuration

For efficient training we employ the so called low rank adaptation framework where the pre-trained linear weights $W \in \mathbb{R}^{d\times d}$ of the LLM layers are not modified. Only a low dimensional correction $\Delta W = BA$ is applied where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times d}$ and $r \ll d$. This reduces the number of trainable parameters more than 100 times, which have effect on required memory and model evaluation time. The initialization of LoRA adapter is given in Listing 5. There we set $r = 16$ and

 doi:10.20944/preprints202604.1316.v1

scaling 32 for ΔW, introduce 10% random dropout during training and apply the adapter to query and value matrices of the self-attention layers. Using these settings we reduce trainable parameters from 1,102,301,184 to 2,252,800.

Listing 5. LLM Initialization with LoRA Adapter.

```
llm_name = "TinyLlama/TinyLlama-1.1B-Chat-v1.0"
llm = AutoModelForCausalLM.from_pretrained(llm_name,device_map="auto")
tokenizer = AutoTokenizer.from_pretrained(llm_name)
lora_config = LoraConfig(r=16,lora_alpha=32,...)
llm = get_peft_model(llm, lora_config)
```

4.3. Fuzzy Embeddings

Listing 6 shows how we generate input embedding vector for given glucose measurement. After definition of the fuzzy variables in `input_terms` array, the corresponding indices of these variables in the LLM vocabulary are recovered with the tokenizer function. As in our case, multiple tokens correspond to a fuzzy variable, hence, a padding is added to align the lengths of representations. Then using the `embed_token` layer of the LLM, the corresponding embedding vectors of the selected vocabulary tokens are obtained in `input_terms_vec0`. Finally, the fuzzy membersip functions are applied to actual glucose measurements contained in yt tensor, multiplied with embedding vectors and aggregated in a single embedding.

Listing 6. LLM embedding of glucose concentration.

```
input_terms = ['hypoglycemia','in range','hyperglycemia']
input_terms_tok = tokenizer(input_terms,padding=True,...)...
input_terms_vec0 = llm.model.embed_tokens(input_terms_tok)
...
input_terms_vec = hypo_glucose(yt)*input_terms_vec0[0,:,:]
    +target_glucose(yt)*input_terms_vec0[1,:,:]
    +hyper_glucose(yt)*input_terms_vec0[2,:,:]
```

4.4. Inference and Defuzzification

The input prompt `combined_embeds` to LLM is concatenation of parts where some are fixed pre-generated embedding like `vect_msg1` and other contain the fuzzy representation of signals in embedding space. The last column from model output is extracted, which components represent the next token prediction scores. The pytorch `gather` function extracts only the scores matching output fuzzy variable token indices in LLM vocabulary. Using softmax the extracted scores are normalized, after which, they are multiplied with discretized output membership function values in `mf_vals` through pytorch `einsum` function. The deffuzification is performed by weithed integration with pytorch `tapz` function.

 doi:10.20944/preprints202604.1316.v1

Listing 7. Calculating new control action.

```
combined_embeds = torch.cat([vect_msg1,input_terms_vec,...],...
outputs = llm(inputs_embeds=combined_embeds,use_cache=False)
output_logits = outputs.logits[:,-1]

zero_dose_p = output_logits.gather(1,output_terms_tok[0,0])
low_dose_p = output_logits.gather(1,output_terms_tok[1,0])
high_dose_p = output_logits.gather(1,output_terms_tok[2,0])

mf_block_norm = softmax(torch.cat([zero_dose_p,low_dose_p,high_dose_p],...
agg_mu = torch.einsum('br,rx->bx', mf_block_norm, mf_vals)

u_new = BW*(torch.trapz(agg_mu * x_vals.unsqueeze(0), x_vals, dim=1)
/ (torch.trapz(agg_mu, x_vals, dim=1) + 1e-8))
```

4.5. Training Loop

A custom training loop is implemented in pytorch (Listing 8) using a variation of stochastic gradient optimization [56], known as `AdamW` torch optimizer. The learning rate is set to 2×10^{-5}. The loss function is computed at every training step as quadratic deviation of extrapolated glucose concentration from the target glucose level of 105 mg/dL for a given insulin dosage. During training multiple virtual patients are propageted through the closed loop system in parallel. Therefore, the maximal deviation of quadratic loss over the population is taken. Then the gradients of the parameters are obtained for the subject with maximal deviation as well. The tensors `yt` and `ut` contain the historical values of glucose and injected insulin. They are updated at each training step by shifting left previous values and concatenating the current values, i.e. implementing tapped delay line. Historical value tensors are detached from automatic gradient computation graph in the beginning of every iteration.

Listing 8. Training Loop.

```
for t in range(0,num_training_steps):
    ...
    optimizer.zero_grad()
    total_loss = ((Gk - 105/18)*(Gk - 105/18)).max()
    total_loss.backward()
    optimizer.step()
    yt = torch.hstack([yt[:,input_toks_per_term:],G])
    ut = torch.hstack([ut[:,output_toks_per_term:],u_per_kg])
```

4.6. Deployment

Running a simulation with even small LLM as TinyLLama require significant computation resources due to massive number of parameters. Therefore a feasible setup is having the actual Fuzzy-LLM controller installed on a computation server with GPU capability. For our experiments we setup the model on NVIDIA RTX A6000 GPU with 48GB of video RAM, even though. For this specific model a lower grade GPU would be suitable too. The architecture from Figure 4 can handle multiple virtual patients simulations by independently keeping the controller state - previous values of glucose and insulin dosages for the tapped delay line, specific patient parameters like identification and body weight. The setup allows parallel evaluation of multiple virtual patients through batching them together. A HTTP REST API endpoint is the way to interface with the centralized controller. In UVa/Padova simulation a stub controller is implemented as an Interpreted MATLAB Function, which

 doi:10.20944/preprints202604.1316.v1

only purpose is to collect current input signals and send to the server in blocking model, till waiting for the response. The response contains the suggested insulin dosage.

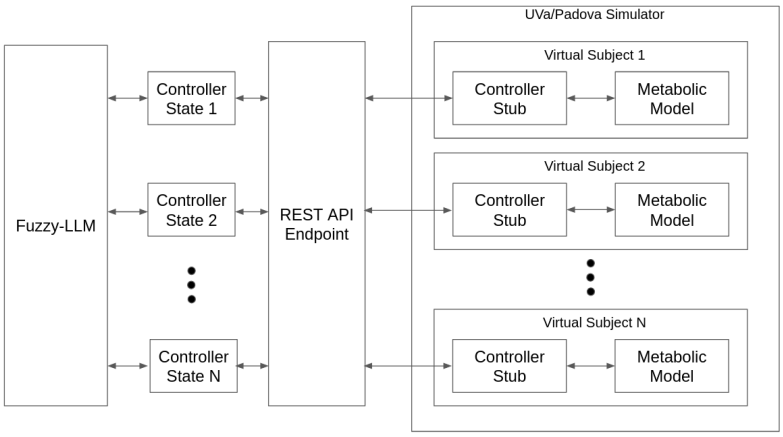

Figure 4. Centralized server deployement of Fuzzy-LLM controller.

5. Results

5.1. Fine-Tuning Performance

The Figure 5 shows blood glucose trajectories for multiple virtual subjects over 48 hours during training, with the curves tightly clustered around the target region near 100 mg/dL for much of the simulation. In the initial 15-20 hours from the training we see postprandial hyperglycemic peaks after meal events above 180 mg/dL, as well as, downward dips toward hypoglycemic range of 70 mg/dL. The transient spikes after meals become less disruptive with training, even though they are not fully eliminated. During later training phases the blood glucose look smoother around the 90-110 mg/dL region, with fewer deep drops and less persistent overshoot after peaks. Therefore, the fine-tuning is improving both robustness and timing of insulin dosage. The insulin is being delivered earlier or more appropriately relative to glucose rises, so the closed loop compensates faster without overcorrecting.

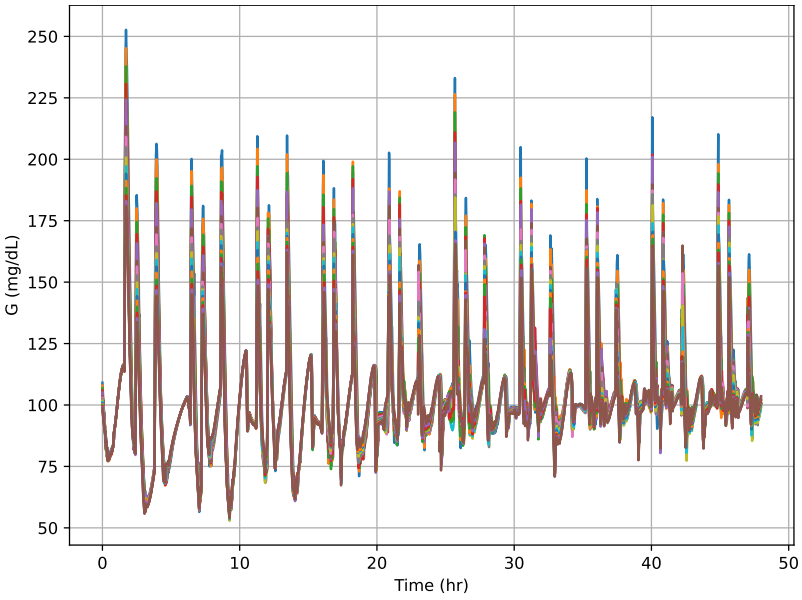

Figure 5. Blood Glucose for multiple virtual subjects during training.

Despite subject-to-subject variability, the trajectories remain bounded and eventually exhibit a recurrent pattern around the target range, which suggests the controller is learning a reasonable closed-loop policy. As training progresses, the glucose trajectories of the virtual subjects become

 doi:10.20944/preprints202604.1316.v1

more tightly organized around the target region, with the population mean settling near the desired range and the spread across subjects gradually shrinking. Early in training, the curves show larger oscillations and taller post-meal excursions, including some pronounced hyperglycemic peaks. Later in training, the responses remain dynamic around meals, but the baseline is better regulated and the inter-subject variability is visibly reduced.

The insulin dosage plot (Figure 6 shows a clear progression with training toward more structured and repeatable control actions. During early training phases, doses vary more smoothly and stay relatively modest, but later the controller produces sharper, better-timed micro-boluses that align with glucose disturbances. This suggests the model is learning when to hold back insulin during stable periods and when to intervene decisively after meal-related rises or persistent hyperglycemia. The learned policy remains individualized across virtual subjects while still converging to a common dosing pattern. The peaks become more pronounced in the later segments, but they are not random spikes; they appear at consistent times and with similar magnitudes across the batch, which indicates the controller is capturing the recurring meal structure in the simulation.

The training loss curve (Figure 7) shows a decreasing trend over time, which indicates that the controller is progressively learning a better mapping from the fuzzified glucose history to the insulin action. The large spikes at the beginning are consistent with the model still exploring poor dosing decisions and encountering large glucose deviations, so the objective is temporarily high. As training continues, the baseline loss drops close to zero for long intervals, suggesting that the learned policy is increasingly able to keep the predicted glucose near the target level. The remaining isolated spikes are also informative: they likely correspond to harder scenarios, such as meal-induced disturbances or subjects with stronger variability, where the controller must react more aggressively. Importantly, these spikes become less frequent and the low-loss segments become longer, which implies improved stability and better generalization across virtual subjects.

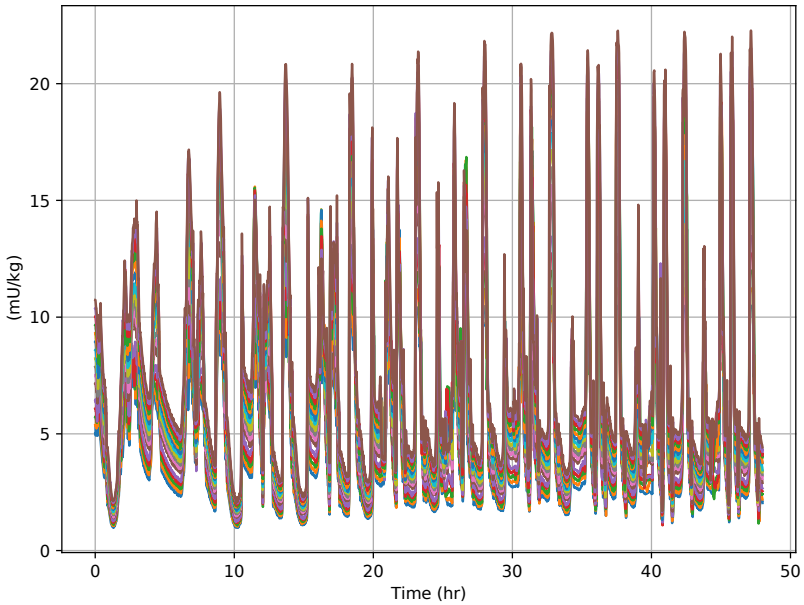

Figure 6. Insulin dosage for multiple virtual subjects during training.

 doi:10.20944/preprints202604.1316.v1

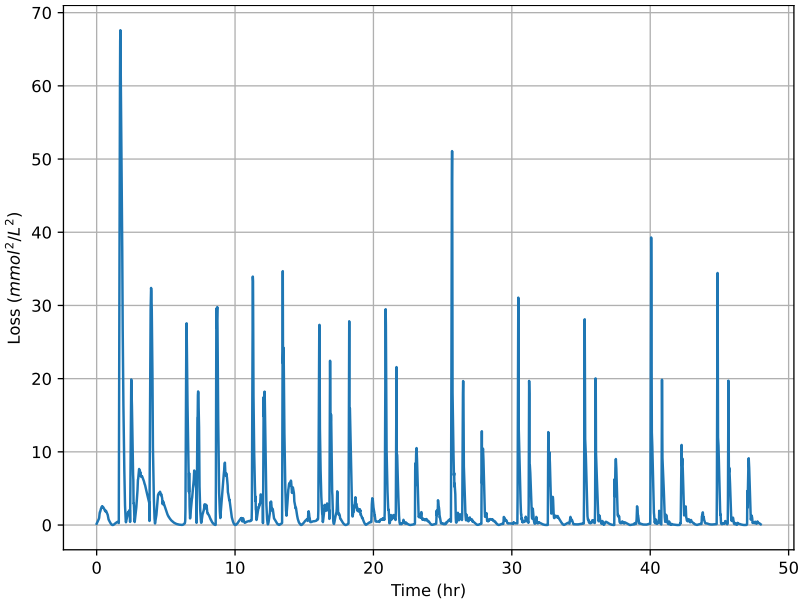

Figure 7. Loss function during training.

The Figure 8 should be interpreted as a monitoring signal rather than an optimization objective. The cross-entropy here measures how well the current input-side glucose membership terms, such as hypo, target, and hyper, align with the target insulin classes zero, low, and high. Because it is not directly minimized, the curve does not need to decrease monotonically, and the visible fluctuations are expected. The overall level stays in a fairly narrow band for much of training, which suggests that the linguistic mapping between glucose patterns and insulin categories remains reasonably stable. The spikes indicate moments where the current fuzzy representation is less consistent with the target insulin label, likely due to harder meal-driven transitions or borderline glucose states. In other words, these peaks show mismatch or ambiguity in the class correspondence, indicating a trivial dosing strategy (high insulin when glucose is high) wouldn't be sufficient for quadratic performance. Much of the trace remains centered around a moderate entropy level. That means the fuzzy tokenization still carries useful information for classification, even as the controller is being trained by the separate closed-loop loss.

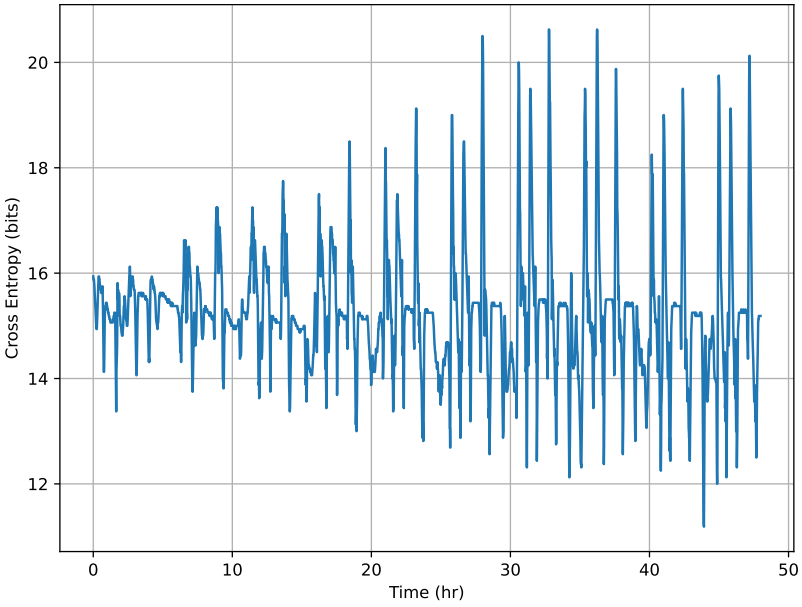

Figure 8. Cross entropy during training. (Cross entropy is reported only as an auxiliary diagnostic of the consistency between glucose membership patterns and insulin class labels; it is not used as the training loss.)

 doi:10.20944/preprints202604.1316.v1

The Figure 9 shows how the insulin-dose membership levels evolve during training for the three output classes: zero dose, low dose, and high dose. The high-dose membership remains dominant for long intervals, especially after the first several hours, which indicates that the controller often interprets the current glucose context as requiring an active corrective response. At the same time, the zero-dose curve is strongest mainly in the early part of the horizon and then tends to remain near zero except for brief impulses, suggesting that the policy becomes less conservative once the training loop learns the recurrent hyperglycemic patterns.

The low-dose membership acts as an intermediate channel and stays active more continuously than the zero-dose class, but with smaller amplitude than the high-dose class. This is a useful sign because it means the model is not collapsing into a purely binary strategy; instead, it preserves graded control behavior where modest corrections are still available when the glucose state is near the transition region. The short upward excursions in the low- and zero-dose curves likely correspond to ambiguous or boundary cases where the fuzzy representation allows multiple insulin actions to remain partially plausible.

Another clear trend is that the high-dose membership becomes more stable and repeatedly saturates at a high level later in training. That pattern suggests the learned controller has become more confident in associating many of the observed glucose histories with stronger insulin action, which is consistent with meal-driven hyperglycemic excursions in the simulation. The frequent sharp drops and recoveries across all three curves reflect switching behavior in a fuzzy controller, where the output is not a single crisp class but a soft membership distribution over insulin options.

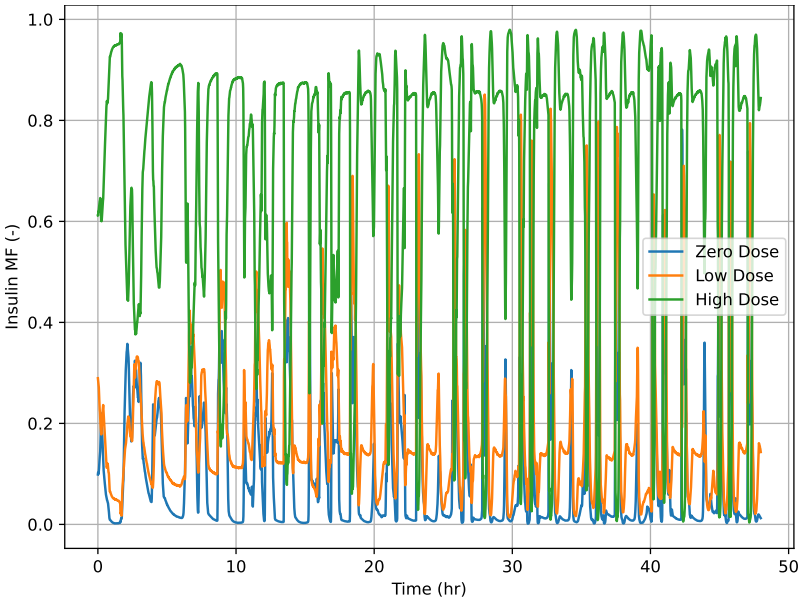

Figure 9. Insulin dose membership function levels during training.

5.2. Simulation with UVa/Padova

The controller was also evaluated in the environment of the UVa/Padova simulator, which is a state-of-the-art tool for proving the feasibility of closed-loop controllers for type 1 diabetes. The controller is applied to the provided simulator group of 10 adult subjects, plus 1 subject representing an average adult population. The selected scenario is a 31 h period with three main meals. The timings of the meals were set to 7:00, 14:00 and 21:00 with the amount of 40 g of carbohydrates and meal duration of 15 min. Detailed results are presented for all subjects in Table 3.

 doi:10.20944/preprints202604.1316.v1

Table 3. Result summary from the UVa/Padova simulation with the μ-controller for 10 subjects from the adult population.

ID	BG	$T_{<50}$	$T_{<90}$	T_{ref}	$T_{>180}$	$T_{>300}$	LBGI	HBGI	BGRI	RoC	A+B	E+F
1	143	0	0	97	3	0	0	3	3	0.5	63	0
2	131	0	0	100	0	0	0	1	1	0.3	90	0
3	142	0	0	100	0	0	0	2	2	0.5	62	0
4	140	0	0	99	1	0	0	2	2	0.5	64	0
5	158	0	0	73	26	0	0	5	5	0.7	32	0
6	133	0	0	95	5	0	0	2	2	0.5	79	0
7	168	0	0	60	40	0	0	7	7	0.9	27	0
8	112	0	3	96	0	0	1	1	2	0.7	80	0
9	135	0	0	98	2	0	0	2	2	0.5	78	0
10	125	0	0	100	0	0	0	1	1	0.4	86	0
AVG	138	0	0	97	3	0	0	2	2	0.6	75	0

Columns: ID—subject identification, BG—blood glucose concentration in mg/dL, $T_{<50}$—time below 50 mg/dL (%), $T_{<90}$—time below 90 mg/dL (%), T_{ref}—time in range (%), $T_{>180}$—time above 180 mg/dL (%), $T_{>300}$—time above 300 mg/dL (%), LBGI—low blood glucose index, HBGI—high blood glucose index, BGRI—blood glucose risk index, RoC—standard deviation of BG rate of change, A + B—% of time in A and B zones from CVGA, E + F—% of time in A and B zones from CVGA.

Several well-recognized metrics in the AP field are presented in Table 3. The UVa/Padova results show that the Fuzzy-LLM controller trained on the Hovorka model achieves generally good glycemic regulation across the 10 adult virtual subjects. The final blood glucose for averaged subject is 138 mg/dL, with 97% time in range and essentially no time below 50 mg/dL or above 300 mg/dL, which indicates a strong safety profile and effective prevention of severe hypo- and hyperglycemia. A key strength of the table is the consistency across subjects: most cases stay close to the target zone, with only moderate excursions above 180 mg/dL in a few harder scenarios, such as subjects 5 and 7. Note subject 7 pose a significant challenges for many AP controllers since it is an adult with 46kg body weight. Even there, the controller still avoids dangerous extremes, and the risk indices remain low to moderate, with average LBGI and HBGI both around 2. This suggests that the controller is conservative enough to remain safe, while still responsive enough to keep the majority of the trajectory within the clinically desirable range.

The CVGA-related metrics also support this interpretation as well CVGA plot in Figure 12. The average RoC is only 0.6, which implies relatively smooth glucose evolution rather than unstable oscillatory behavior, and the A+B metric is high at 75%, showing that most trajectories lie in the safer CVGA zones. The E+F value is 0% for all subjects, confirming that the controller successfully avoids the most dangerous regions of the clinical risk map.

The population graph trace of the BG variation for the selected scenario is presented in Figure 10, where the mean BG for the population is plotted along with minimal, maximal, and standard deviation bounds. All curves are in the acceptable range. Figure 11 presents the hourly calculated glucose risk indices where HBGI is positive number and LBGI is a negative number, along with their standard deviations taken for the examined population.

The glucose density function (Figure 13) is sharply concentrated in the clinically relevant region between the two green thresholds, with the dominant mass sitting around roughly 110-190 mg/dL. This indicates that most simulated glucose values remain near the target and upper target range, rather than spreading broadly across extreme hypo- or hyperglycemic regions. The tall, narrow peak near about 115 mg/dL suggests a strong clustering around the preferred operating point of the controller. The annotated percentages show that only a very small fraction of samples lie below the lower bound, while the majority, about 91%, lies in the central band, and a smaller but non-negligible tail, about 8%, extends into the hyperglycemic side. That shape is consistent with a controller that is generally effective but still allows occasional postprandial overshoots. In other words, the learned policy maintains most

 doi:10.20944/preprints202604.1316.v1

trajectories inside or close to the target zone, but does not completely eliminate excursions after meals or during more difficult subject-specific dynamics.

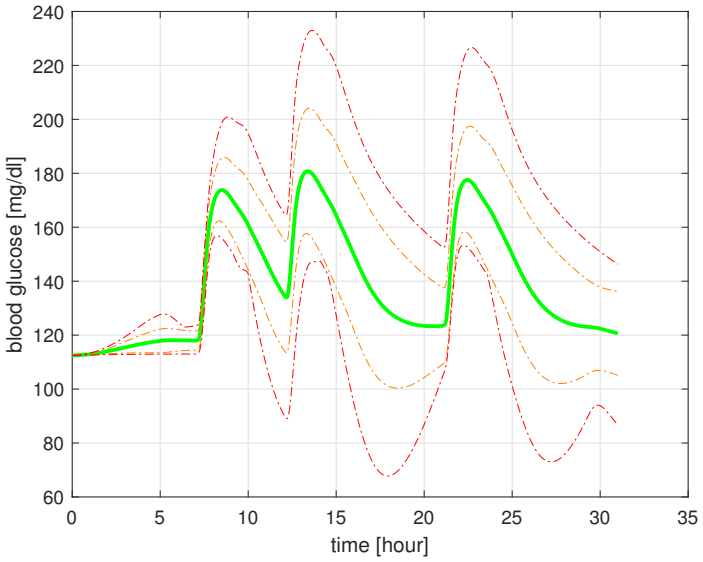

Figure 10. Glucose trace for the 10-adult population from the UVA/Padova simulator. The green line represents the average glucose, the orange line represents the ± 1 standard deviation interval, and the red line is the minimal and maximal values from the envelope.

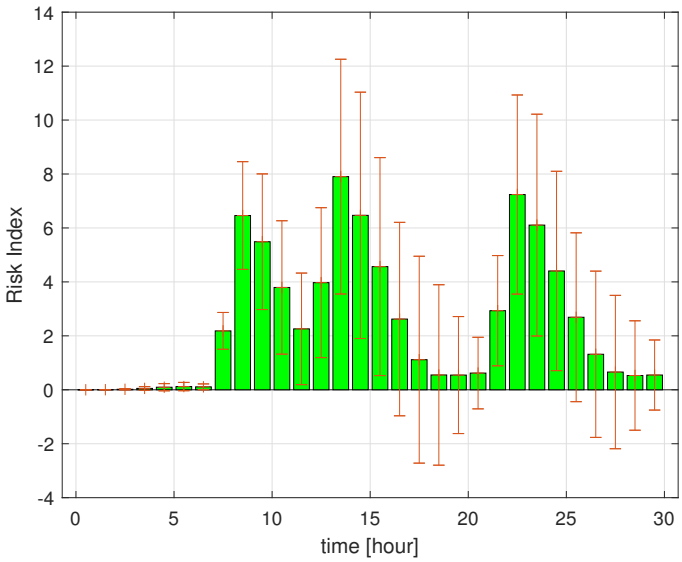

Figure 11. Glucose risk index calculated for each hour with ± 1 standard deviation confidence.

 doi:10.20944/preprints202604.1316.v1

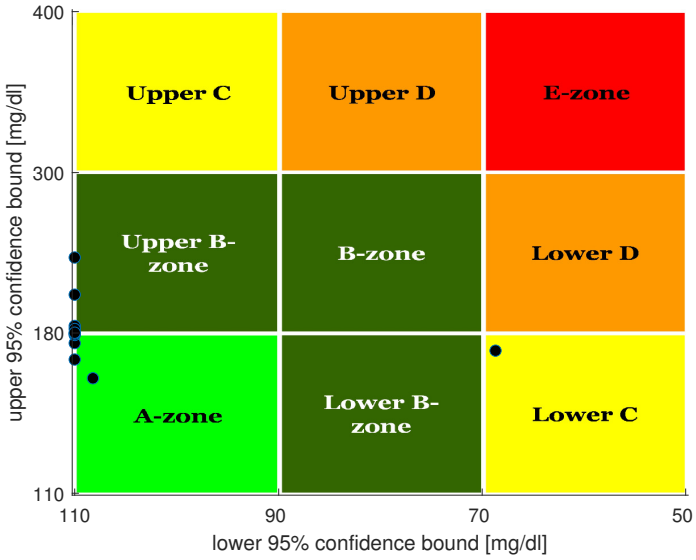

Figure 12. CVGA analysis.

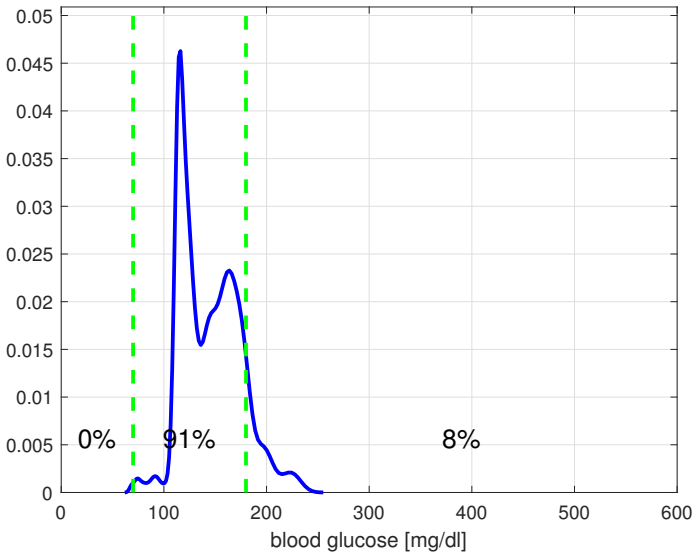

Figure 13. Glucose density function.

6. Conclusions

The results show that the proposed Fuzzy-LLM controller can regulate glucose effectively in both the internal Hovorka-based training environment and the external UVa/Padova benchmark. In the training simulations, glucose trajectories remain largely within the clinically acceptable zone, while the UVa/Padova validation confirms that this behavior transfers to a standardized simulator with multiple virtual adult subjects. This is important because it indicates that the controller is not simply overfit to one model instance, but can generalize across a broader population of patient dynamics.

A major strength of the approach is that it combines the interpretability of fuzzy logic with the sequence-handling capacity of a fine-tuned language model. The fuzzy membership representation provides a natural bridge between physiological variables and the LLM token space, while the closed-loop training objective aligns the learned policy with actual glycemic regulation rather than isolated classification accuracy. The figures and tables support this claim: glucose remains mostly in range, severe hypo- and hyperglycemia are avoided, and the risk indices stay low across the evaluated subjects.

 doi:10.20944/preprints202604.1316.v1

At the same time, the results also show that the task is not trivial. Some subjects still experience moderate postprandial excursions, which is expected in a setting with meal disturbances and inter-subject variability. This suggests that the controller is robust but still conservative, especially when it must trade off between aggressive correction and safety. The cross-entropy diagnostic and insulin membership evolution also indicate that the fuzzy representation remains meaningful throughout training, even though the main optimization target is the closed-loop glucose error.

From a control perspective, the UVa/Padova outcomes are especially encouraging because they are obtained under a benchmark environment commonly used to assess artificial pancreas algorithms. The fact that the controller avoids dangerous regions in the control-variability grid and keeps the time in range high supports the viability of the proposed Fuzzy-LLM framework as a real-time decision-support strategy. This can be seen as evidence that language-model fine-tuning can be repurposed beyond text generation and opens possibility for structured biomedical control when the input and output signals are carefully embedded.

In conclusion, the study suggests a promising direction for closed-loop glucose regulation by combining Hovorka-based simulation, fuzzy membership encoding, and LLM fine-tuning. Future work should focus on longer simulation horizons, broader meal variability, additional safety constraints, and eventually evaluation on more extensive datasets.

Funding: The present study is carried out within the project Infrastructure for Fine-tuning Pre-trained Large Language Models, Grant Agreement No. ПВУ – 55 from 12.12.2024 /BG-RRP-2.017-0030-C01/.

Data Availability Statement: Due to privacy or ethical restrictions no data is available.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. American Diabetes Association. *Standards of Care in Diabetes to Guide Prevention, Diagnosis, and Treatment for People Living with Diabetes*; American Diabetes Association: Arlington, VA, USA, 2023.
2. National Institute for Health and Care Excellence. *Type 1 Diabetes in Adults: Diagnosis and Management*; National Institute for Health and Care Excellence: London, UK, 2017.
3. Nakrani, M.N.; Wineland, R.H.; Anjum, F. Physiology, Glucose Metabolism. In *StatPearls [Internet]*; StatPearls Publishing: Treasure Island, FL, USA, 2023. Available online: https://www.ncbi.nlm.nih.gov/books/NBK560599/ (Accessed on 1 June 2023).
4. Melo, K.F.S.; Bahia, L.R.; Pasinato, B.; Porfirio, G.J.M.; Martimbianco, A.L.; Riera, R.; Calliari, L.E.P.; Minicucci, W.J.; Turatti, L.A.A.; Pedrosa, H.C.; et al. Short-acting insulin analogues versus regular human insulin on postprandial glucose and hypoglycemia in type 1 diabetes mellitus: A systematic review and meta-analysis. *Diabetol Metab. Syndr.* **2019**, *11*.
5. Man, C.D.; Micheletto, F.; Lv, D.; Breton, M.; Kovatchev, B.; Cobelli, C. The UVA/PADOVA Type 1 Diabetes Simulator: New Features. *J Diabetes Sci Technol.* **2014** https://doi.org/10.1177/1932296813514502 *8(1)*, 26–34.
6. Fushimi, E.; De Battista, H.; Garelli, F. A Dual-Hormone Multicontroller for Artificial Pancreas Systems. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 4743–4750. https://doi.org/10.1109/JBHI.2022.3182581.
7. Peng, Z.; Xie, X.; Tan, Q.; Kang, H.; Cui, J.; Zhang, X.; Li, W.; Feng, G. Blood glucose sensors and recent advances: A review. *J. Innov. Opt. Health Sci.* **2022**, *15*. https://doi.org/10.1142/S1793545822300038
8. Huyett, L.M.; Dassau, E.; Zisser, H.C.; Doyle F.J. Glucose Sensor Dynamics and the Artificial Pancreas. *IEEE Control. Syst. Mag.* **2018**, *38*, 30–46.
9. Berget, C.; Messer, L.H.; Forlenza, G.P. A Clinical Overview of Insulin Pump Therapy for the Management of Diabetes: Past, Present, and Future of Intensive Therapy. *Diabetes Spectr.* **2019**, *32*, 194–204.
10. Lewis, D.; Leibrand, S. Real-World Use of Open Source Artificial Pancreas Systems. *J. Diabetes Sci. Technol.* **2016**. https://doi.org/10.1177/1932296816665635
11. Knebel, T.; Neumiller, J.J. Medtronic MiniMed 670G Hybrid Closed-Loop System. *Clin. Diabetes* **2019**, *37*, 94–95.

 doi:10.20944/preprints202604.1316.v1

12. Kublin, O.; Stępień, M. The Nightscout system—Description of the system and its evaluation in scientific publications. *Pediatr. Endocrinol. Diabetes Metab.* **2020**, *26*, 140–143.
13. Gomez, E.J.; Pérez, M.E.H.; Vering, T.; Cros, M.R.; Bott, O.; García-Sáez, G.; Pretschner, P.; Brugués, E.; Schnell, O.; Patte, C.; et al. The INCA System: A Further Step Towards a Telemedical Artificial Pancreas. *IEEE Trans. Inf. Technol. Biomed.* **2008**, *12*, 470–479. https://doi.org/10.1109/TITB.2007.902162.
14. Pfeiffer, E.F. Artificial pancreas: State of the Art. *Int. J. Artif. Organs* **1988**, *11*, 13–26.
15. Bondia, J.; Romero-Vivó, S.; Ricarte, B.; Díez J.L. Insulin Estimation and Prediction. *IEEE Control. Syst. Mag.* **2018**, *38*, 47–66.
16. Seron, M.M.; Braslavsky, J.H.; Goodwin, G.C. *Fundamental Limitations in Filtering and Control*; Springer: London, UK, 1997; ISBN 9781447112440.
17. Ramkissoon, C.M.; Aufderheide, B.; Bequette, B.W.; Vehi, J. A Review of Safety and Hazards Associated With the Artificial Pancreas. *IEEE Rev. Biomed. Eng.* **2017**, *10*, 44–62. https://doi.org/10.1109/RBME.2017.2749038.
18. Borri, A.; Cacace, F.; Gaetano, A.; Germani, A.; Manes, C.; Palumbo, P.; Panunzi, S.; Pepe, P. Observers for Nonlinear Time-Delay Systems with Application to the Artificial Pancreas *IEEE Control. Syst. Mag.* **2017**, *37*, 33–49.
19. Sanz, R.; Garcia, P.; Diez, J.-L.; Bondia, J. Artificial Pancreas System With Unannounced Meals Based on a Disturbance Observer and Feedforward Compensation. *IEEE Trans. Control. Syst. Technol.* **2021**, *29*, 454–460.
20. Turksoy, K.; Samadi, S.; Feng, J.; Littlejohn, E.; Quinn, L.; Cinar, A. Meal Detection in Patients With Type 1 Diabetes: A New Module for the Multivariable Adaptive Artificial Pancreas Control System. *IEEE J. Biomed. Health Inform.* **2016**, *20*, 47–54. https://doi.org/10.1109/JBHI.2015.2446413.
21. Paoletti, N.; Liu, K.S.; Chen, H.; Smolka, S.A.; Lin, S. Data-Driven Robust Control for a Closed-Loop Artificial Pancreas. *IEEE/Acm Trans. Comput. Biol. Bioinform.* **2020**, *17*, 1981–1993. https://doi.org/10.1109/TCBB.2019.2912609.
22. Lee, S.; Kim, J.; Park, S.W.; Jin S.-M.; Park, S.-M. Toward a Fully Automated Artificial Pancreas System Using a Bioinspired Reinforcement Learning Design: In Silico Validation. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 536–546. https://doi.org/10.1109/JBHI.2020.3002022.
23. Chakrabarty, A.; Zavitsanou, S.; Doyle, F.J.; Dassau, E. Event-Triggered Model Predictive Control for Embedded Artificial Pancreas Systems. *IEEE Trans. Biomed. Eng.* **2018**, *65*, 575–586. https://doi.org/10.1109/TBME.2017.2707344.
24. Chakrabarty, A.; Healey, E.; Shi, D.; Zavitsanou, S.; Doyle, F.J.; Dassau, E. Embedded Model Predictive Control for a Wearable Artificial Pancreas. *IEEE Trans. Control. Syst. Technol.* **2020**, *28*, 2600–2607.
25. Batmani, Y.; Khodakaramzadeh, S.; Moradi, P. Automatic Artificial Pancreas Systems Using an Intelligent Multiple-Model PID Strategy. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 1708–1717. https://doi.org/10.1109/JBHI.2021.3116376.
26. Meneghetti, L.; Terzi, M.; Favero, S.; Susto, G.A.; Cobelli, C. Data-Driven Anomaly Recognition for Unsupervised Model-Free Fault Detection in Artificial Pancreas. *IEEE Trans. Control. Syst. Technol.* **2020**, *28*, 33–47.
27. Ruiz-Velázquez, E.; García-Rodríguez, J.; Quiroz, G.; Femat, R. Robust μ-synthesis: Towards a unified glucose control in adults, adolescents and children with T1DM. *J. Frankl. Inst.* **2020**, *357*, 9633–9653. https://doi.org/10.1016/j.jfranklin.2020.07.030.
28. Cassany, L.; Gucik-Derigny, D.; Cieslak, J.; Henry, D.; Franco, R.; Ferreira de Loza, A.; Ríos, H.; Olcomendy, L.; Pirog, A.; Bornat, Y.; Renaud, S.; Catargi, B. A Robust H-∞ Control Approach for Blood Glucose Regulation in Type-1 Diabetes. *IFAC-PapersOnLine* **2021**, *54*, 460–465. https://doi.org/10.1016/j.ifacol.2021.10.299.
29. Cassany, L.; Gucik-Derigny, D.; Cieslak, J.; Henry, D.; Franco, R.; De Loza, A.F.; Rios, H.; Olçomendy, L.; Pirog, A.; Bornat, Y.; et al. A Robust Control solution for Glycaemia Regulation of Type-1 Diabetes Mellitus. In *2021 European Control Conference (ECC)*; IEEE: Delft, Netherlands, 2021; pp. 327–332 https://doi.org/10.23919/ECC54610.2021.9654888.
30. Mandal, S.; Sutradhar, A. Robust controller for artificial pancreas for patients with type-1 diabetes. *Res. Biomed. Eng.* **2023**, *39*, 437–450. https://doi.org/10.1007/s42600-023-00285-9.
31. Zadeh, L.A. Fuzzy Logic. *Computer* **1988** *21(4)* 83–93 https://doi.org/10.1109/2.53.
32. Lee, C.C. Fuzzy Logic in Control Systems: Fuzzy Logic Controller. I. *IEEE Transactions on Systems, Man, and Cybernetics* **1990** *20(2)* 404–18. https://doi.org/10.1109/21.52551.
33. Mamdani, E.H.; Assilian, S. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies* **1975** *7(1)* 1–13. https://doi.org/10.1016/S0020-7373(75)80002-2.

 doi:10.20944/preprints202604.1316.v1

34. Grant, P. A New Approach to Diabetic Control: Fuzzy Logic and Insulin Pump Technology. *Medical Engineering & Physics* **2007** *29(7)*. https://doi.org/10.1016/j.medengphy.2006.08.014.
35. Mauseth, R.; Wang, Y.; Dassau E.; et.al. Proposed Clinical Application for Tuning Fuzzy Logic Controller of Artificial Pancreas Utilizing a Personalization Factor. *Journal of Diabetes Science and Technology* **2010** *4(4)*. https://doi.org/10.1177/193229681000400422.
36. Yan, S.-R.; Alattas, K.A.; Bakouri, M.; Alanazi, A.K.; Mohammadzadeh, A.; Mobayen, S.; Zhilenkov, A.; Guo, W. Generalized Type-2 Fuzzy Control for Type-I Diabetes: Analytical Robust System. *Mathematics* **2022**, *10(690)*. https://doi.org/10.3390/math10050690
37. Liu, M.; Zhang, H.; Xu, Z.; Ding. K. The fusion of fuzzy theories and natural language processing: A state-of-the-art survey. *Appl. Soft Comput.* **2024** *162*. https://doi.org/10.1016/j.asoc.2024.111818
38. Zhang,H.; Shang,J.Natural Language Processing and Applications. *Tsinghua University Press* **2025** 9789819797387.
39. Adel, N.; Crockett, K.; Carvalho, J.P.; Cross, V. Fuzzy Influence in Fuzzy Semantic Similarity Measures, *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* **2021** *IEEE* 1–7.
40. Adel, N.; Crockett, K.; Livesey, D.; Carvalho, J.P. An interval type-2 fuzzy ontological similarity measure, *IEEE Access* **2022** *10* 81506–81521.
41. Hoang, C.; Sachan, D.; Mathur, P.; et. al. Improving retrieval augmented neural machine translation by controlling source and fuzzy-match interactions. *In Findings of the Association for Computational Linguistics: EACL 2023* **2023** 289-295.
42. Yan, R.; Yu, Y.; Qiu, D. Emotion-enhanced classification based on fuzzy reasoning. International Journal of Machine Learning and Cybernetics, 13(3), 839-850.
43. Li, Q.; Li, L.; Li, Q.; Zhong, J. A comprehensive exploration on spider with fuzzy decision text-to-SQL model. *IEEE Transactions on Industrial Informatics* **2019** *16(4)* 2542-2550. https://doi.org/10.1109/TII.2019.2952929
44. Novák, V. Fuzzy logic in natural language processing. *2017 IEEE international conference on fuzzy systems (FUZZ-IEEE)* **2017** *1-6. https://doi.org/10.1109/FUZZ-IEEE.2017.8015405*
45. *Vaswani, A.; Shazeer, N.; Parmar, N. et al. Attention Is All You Need.* arXiv **2023** *https://arxiv.org/abs/1706.03762*
46. *Hu, E.J.; Shen, Y.; Wallis, P.; et al. LoRA: Low-Rank Adaptation of Large Language Models.* arXiv **2021** *https://arxiv.org/abs/2106.09685*
47. *Aghajanyan, A.; Zettlemoyer, L.; Gupta, S. Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning.* arXiv **2020** *https://arxiv.org/abs/2012.13255*
48. *Nosrati, K.; Tepljakov, A.; Belikov, J.; Petlenkov, E. When control meets large language models: From words to dynamics.* arXiv **2026** *https://arxiv.org/pdf/2602.03433v1*
49. *Hovorka, R.; Canonico, V.; Chassin, L.J.; Haueter, U.; Massi-Benedetti, M.; Federici, M.O.; Wilinska M.E. Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes.* Physiol. Meas. **2004**, 25, *905.*
50. *Carson, E.R.; Cobelli, C.; Finkelstein, L. The mathematical modeling of metabolic and endocrine systems : model formulation, identification, and validation.* New York: Wiley **1983** *https://api.semanticscholar.org/CorpusID:83410068*
51. *Cao, S.G.; Rees, N.W.; Feng, G. Analysis and design of fuzzy control systems using dynamic fuzzy global models.* Fuzzy Sets and Systems **1995** 75(1), *47–62. https://doi.org/10.1016/0165-0114(94)00323-Y*
52. *Cao, S.G.; Rees, N.W.; Feng, G. Mamdani-type fuzzy controllers are universal fuzzy controllers.* Fuzzy Sets and Systems **2001** 123(3), *359–367. https://doi.org/10.1016/S0165-0114(01)00015-X*
53. *Touvron, H.; Lavril, T.; Izacard G. et al. LLaMA: Open and Efficient Foundation Language Models.* arXiv **2023** *https://arxiv.org/abs/2302.13971*
54. *Su, J.; Lu, Y.; Pan, S. et al. RoFormer: Enhanced Transformer with Rotary Position Embedding.* arXiv **2023** *https://arxiv.org/abs/2014.09864*
55. *Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer Normalization.* arXiV **2016** *https://arxiv.org/abs/1607.06450*
56. *Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization.* arXiv **2017** *https://arxiv.org/abs/1412.6980*