Personalized open world plan generation for safety-critical human centered autonomous systems: A case study on Artificial Pancreas

Ayan Banerjee and Sandeep K.S. Gupta
IMPACT Lab, SCAI,
Arizona State University, Tempe, Az
`abanerj3@asu.edu, sandeep.gupta@asu.edu`

Abstract

Design-time safety guarantees for human-centered autonomous systems (HCAS) often break down in open-world deployment due to uncertain human interaction. In practice, HCAS must follow a user-personalized safety plan, with the human providing external inputs to handle out-of-distribution events. Open-world safety planning for HCAS demands modeling dynamical systems, exploring novel actions, and rapid replanning when plans are invalidated or dynamics shift. No single state-of-the-art planner meets all these needs. We introduce an LLM-based architecture that automatically generates personalized safety plans. By itself, the LLM fares poorly at producing safe usage plans, but coupling it with a safety verifier—which evaluates plan safety over the planning horizon and feeds back quality scores—enables the discovery of safe plans. Moreover, fine-tuning the LLM on personalized models inferred from open-world data further enhances plan quality. We validate our approach by generating safe usage plans for artificial pancreas systems in automated insulin delivery for Type 1 Diabetes patients. Code: `https://github.com/ImpactLabASU/LLMOpen`

1 Introduction

Human centered autonomous systems (HCAS) are often safety critical (Sadigh et al., 2016), where actions taken by a reactive module (RM) in response to percepts from the environment can cause harm to the human. The human in a HCAS can assume two roles (Fig. 1): a) human in the loop (HIL), where the human is observing the environment and can provide additional inputs to the environment or RM to change their states, and b) human in the plant (HIP), where the human acts as a passive physical entity with time variant dynamics that gets directly affected by the actions of the RM with the intent to reach a goal state. As such assurance of safety under human inputs is an essential property to establish trust and widespread acceptance of HCAS such as Artificial Pancreas (AP) systems for glucose management in Type 1 Diabetes (T1D) or autonomous cars. Given that the human is inherently coupled with the operation of HCAS, any safety assurance process will need to accurately model RM-human interaction (Sadigh et al., 2016).

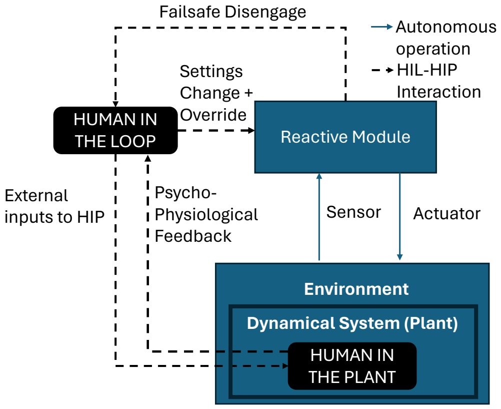

Figure 1: HIL-HIP autonomous systems.

The complexity of human interactions with the HCAS makes it impossible to accurately enlist all possible scenarios (Banerjee and Gupta, 2014). As such, in the design phase, safety critical systems utilize certain limiting assumptions on the distribution of HIL actions and HIP properties; a human who satisfies the assumed distribution is referred to as the “average human". Then a set of *safety plans*, HIL action sequences of the “average user", is developed under which the HCAS is tested to be safe. Some of the major problems in assuring that a tested safe HCAS with a set of safety plans still operates safely in deployment are:

P1: Out of distribution HIL - The set of safety plans may not optimize performance for a human user who may not fit the assumptions of the average human user. As a result, the user in its HIL capacity may operate the HCAS under out of distribution (OOD) usage plans that are not tested to be safe and hence can potentially cause safety violations in deployment.

P2: Long term time varying HIL-HIP contexts - The HCAS is often intended for long term usage and in the course, user contexts such as HIP dynamics or HIL action space can fundamentally change. Such changes may lead to human interactions with

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 22409–22422
November 4-9, 2025 ©2025 Association for Computational Linguistics

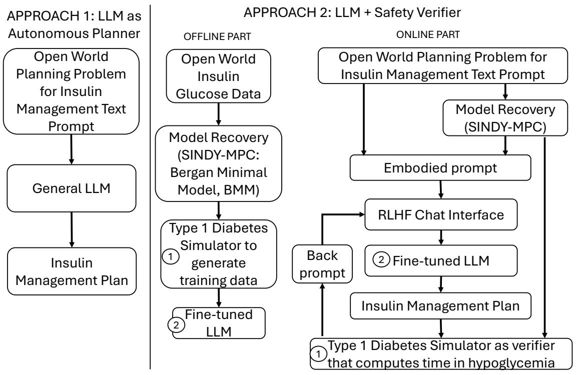
APPROACH 1: LLM as Autonomous Planner
Open World Planning Problem for Insulin Management Text Prompt
General LLM
Insulin Management Plan
APPROACH 2: LLM + Safety Verifier
OFFLINE PART
Open World Insulin Glucose Data
Model Recovery (SINDY-MPC: Bergan Minimal Model, BMM)
① Type 1 Diabetes Simulator to generate training data
② Fine-tuned LLM
ONLINE PART
Open World Planning Problem for Insulin Management Text Prompt
Model Recovery (SINDY-MPC)
Embodied prompt
RLHF Chat Interface
Back prompt
② Fine-tuned LLM
Insulin Management Plan
① Type 1 Diabetes Simulator as verifier that computes time in hypoglycemia

Figure 2: Solution to the personalized usage plan generation in open world for insulin management. Approach 1 uses LLMs as autonomous planners, while approach 2 uses LLMs as plan suggester with iterative plan update using feedback from a safety verifier.

HCAS that may not adhere to the set of safety plans or the safety plans themselves may be invalidated due to the changed HIP dynamics.

P3: Short term unexpected plan invalidation - Human contexts are characterize by unexpected events that may affect HIP or may require novel HIL actions for safe and effective performance, resulting in deviation from the safety plan.

In state-of-art commercialized HCAS, the problems $P1$ through $P3$ are tackled by exploring novel usage plans in the open world through consultation with experts, detecting safety violations through safety monitors, and preventing hazards through fail safe modes that may entail shutting off HCAS autonomy and requiring emergency HIL action. In this paper, an *open world* (Talamadupula et al., 2010) is characterized by occurrences of one or more instances of the problems $P1$ through $P3$. In the standard modus operandi of HCAS, the exploration and execution of novel plans in the open world is a *burden on the user*. As such increased *plan management burden*: a) is often a deterrent to usage or adherence to safety plans, or b) may restrict the HIL actions which may in turn affect HIP dynamics through the psycho-physiological feedback pathways.

In this paper, we develop a framework to automatically generate usage plans in the open world for a HCAS that are: a) personalized for a given human user, in that it optimizes certain efficacy criteria for the human user, b) safe for human use even if they are originally not included in the set of safe usage plans, and c) reduces plan management burden by decreasing the number of inputs required from the human user.

1.1 Solution Overview

We leverage large language models (LLMs) to solve the problem of usage plan generation in open world with two approaches (Fig. 2):

a) Approach 1: Using LLM as an autonomous planner. In this approach, the HIL user interacts with the chat interface of the LLM to provide an open world scenario in the form of an embodied prompt (a textual prompt interleaved with sensor data from the HCAS operation) as shown in Fig. 2. The LLM has two components: i) re-inforcement learning with human feedback (RLHF), that contextualizes an open world scenario embodied prompt based on human feedback through back prompting and converts it into a query for the LLM, and ii) a transformer based language model, which takes a query and searches for an usage plan. We assume that the training set of the transformer architecture may include domain specific usage plan for large set of human users. The usage plan from the transformer is provided back to the human user, which then modifies the plan through back prompting (Valmeekam et al., 2023) to finally reach a potentially safe plan. Back prompting is a method where the human user gives a detailed feedback on the plan quality of the LLM which is then used to update the LLM response.

We show that using LLMs as autonomous planner has several drawbacks including the following:

C1: Physically infeasible plans generated by LLMs (Evaluated in Section 6).

C2: Unsafe plan: Even if LLMs generate a feasible plan, there is no guarantee that the LLMs may generate a plan that is safe (Section 6).

C3: Agnostic of personalized HIP dynamical context: LLMs may generate plans that are not aware of the personalized human user contexts. Hence, even if the plans may be safe for the average user, it may not be safe for the out of distribution HIL (Evaluated in Section 6). To overcome such drawbacks, we explore Approach 2.

b) Approach 2: Using LLM as plan suggester with safety verification. This approach uses plan safety feedback from a forward safety simulation module to iteratively modify an initial plan into a personalized safety plan (Fig. 2).

Tackle drawback C1: We contextualize the RLHF module of the LLM with a physics driven model of the HIP. This contextualization process is done by generating domain specific prompts regarding the physics model.

Tackle drawback C3: We first recover the physics guided model of the HIP from the open world scenario data (Section 5.2.1). The LLM is then fine tuned using embodied instruction prompts that encode the relationship between open world scenario data and physics model parameters. The fine tuned LLM is capable of correlating model parameters with open world scenario data and incorporates such causal relations in its plan search mechanism.

Tackle drawback C2: We use the recovered model as a forward safety simulator, instantiated with the usage plan derived by the LLM. Safety is quantified using metrics such as robustness of signal temporal logic formula. This safety evaluation is then passed to the RLHF module to modulate the plan quality score. If a plan is unsafe, then a heavy penalty is imposed in the plan quality score.

We show the efficacy of Approach 2 in yielding safe plans for various instances of problems $P1$ through $P3$ in the domain of AP for automated insulin delivery in T1D patients. Approach 2 was tested on 102 open world scenarios and yielded safe plans in 93 scenarios within two iterations of the forward safety simulation feedback. If it cannot generate a safe plan, the Approach 2 defaults to fail safe HCAS shutdown. We also show that user burden in terms of number of HIL inputs to the RM, can be reduced through further back prompting the LLM, once a safe usage plan is achieved.

2 Related Works

Why can't we use classical planners? Classical planners such as STRIPS (Fikes and Nilsson, 1971), GraphPlan (Blum and Furst, 1997), or Fast Downward (Helmert, 2006) are designed to operate with discrete states, with limited uncertainty. The usage plan generation problem is in continuous state space with open world. Modification of classical planners to continuous states and open world scenarios may lead to state explosion. Moreover, addressing problems $P2$ and $P3$ will require re-planning, which with classical planners maybe very time consuming. Moreover, classical planners cannot optimize a plan by reducing the number of steps and hence cannot reduce user burden.

Why can't POMDPs be used? Partially observable Markov decision process (POMDP) (Cassandra et al., 1999), although designed for discrete planning domain, can be extended to continuous domain through sampling (Pineau et al., 2003). POMDPs are good at addressing short term uncertainties ($P3$). However, POMDPs suffer from curse of dimensionality especially in the open world which makes it very time consuming to learn and replan with personalized out-of-distribution contexts ($P1$ and $P2$). Searched plan may be optimized by putting negative reward on plan length.

Why can't we use traditional reinforcement learning for usage plan generation in open world? Reinforcement learning (RL) approaches are best suited to solve the planning problems in open world (Brockman et al., 2016) with their plan exploration capabilities. However, in the open world if uncertain changes occur, then the underlying Markov decision process (MDP) needs to be relearned and the plan search process has to be executed again. While relearning MDP can be optimized through iterative training (Palacios and Geffner, 2016), plan search is often performed using computationally complex methods such as genetic algorithms (Rodriguez-Aguilar et al., 2010), non-linear optimization (Li and Allison, 2017), or constraint satisfaction (Albrecht and Ramamoorthy, 2015) methods. This relearning process is slow and during this process there is no guarantee on the safety of the HCAS.

Paper	**NA**	**MA**	**PI**	**DS**	**QRP**
(Zhuo et al., 2013)	No	Yes	No	No	No
(Ding et al., 2022)	No	Yes	Yes	No	No
(Chen et al., 2024)	No	Yes	Yes	No	No
(Cardellini et al., 2023)	No	No	No	Yes	No
(Huang et al., 2024)	No	Yes	Yes	No	No
(Gestrin et al., 2024)	Yes	Yes	Yes	No	No
(Tantakoun et al., 2024)	No	Yes	Yes	No	No
(Li et al., 2024)	No	Yes	Yes	No	Yes
(Wang et al., 2025)	No	Yes	Yes	No	No
(Goel et al., 2024)	No	Yes	Yes	No	Yes
(Wang et al., 2023)	Yes	Yes	Yes	No	No
This work	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

Table 1: Comparison of open–world planning approaches on capabilities, NA-Novel Action, MA, Model Adaptation, PI-Plan invalidation, DS-dynamical systems, QRP-quick replanning upon change in dynamics.

What potential advantages do we have with LLMs? LLMs can potentially solve the slow replanning problem in RL. A pre-trained LLM represents large databases into very sparse embedding space. We hypothesize that the pre-trained LLM has been trained on textual content related to domain specific usage plans. This is the case in the example domain of AP considered and also highly likely for many other domains. Hence, given a prompt describing an open world scenario, it can map the prompt to the embedding space and search for similar embeddings. This search is fast since it is a sub-linear time search over n embeddings using similarity metrics (Huang et al., 2020). Reverse embedding of the output of this search may result in a valid plan which can be quickly evaluated by the RLHF quality score.

What are the previous attempts at using LLMs as planners? Huang et al. (Huang et al., 2022) show that if LLMs are appropriately prompted, they can effectively decompose high-level tasks into mid-level plans without any further training. Valmeekam et al. (Valmeekam et al., 2023) show that LLMs as autonomous planners have dismal performance (3% success rate). Iterative correction by humans through back prompting enhanced their ability to solve benchmark tasks. Sharan et al. (Sharan et al., 2023) present a hybrid planning strategy to improve closed-loop planning in autonomous driving where the LLM plan was evaluated using collision risk metrics to enhance the overall system's adaptability and performance. Banerjee et al (Banerjee et al., 2024) showed that fine tuning with a calibrated continuous dynamical model can enable LLMs to evaluate plans in the continuous domain for closed world problems whereas this work focuses on open world plan generation. Our approach is motivated by Valmeekam et al. (Valmeekam et al., 2023), where instead of human correction, we utilize a forward safety simulation engine coupled with a model recovery tool such as SINDY (Kaiser et al., 2018) or EMILY (Banerjee and Gupta, 2024b,a; Xu et al., 2025) as plan evaluator.

Existing work on open world planning: Table 1 lists several previous work focussed on open world planning. As shown, there exists no single planner that can tackle all components of open world planning in real deployment viz. novel action search, adaptation to model change, planning with continuous dynamics, and quick re-planning upon change in continuous dynamics.

3 Formal Problem Definition

Let the state of the HCAS be expressed by the state variable X which follows the dynamics in Eqn 1.

$$\dot{X} = f_\omega(X, \pi(X, S_p) + U_{ex}), \tag{1}$$

where $f_\omega(.)$ is any Lipshitz continuous (Nesterov, 1982) function parameterized by coefficient set ω, and $\pi(.,.)$ describes the action of a RM that computes an input to the environment based on the environment state expressed by $X_p \subset X$ and RM configuration set (such as set point) S_p (Fig. 1). For convenience in expressing the planning problem and with restricted focus on the case study of AP, we will assume that the environment state is accurately expressed by the HCAS state and hence $X_p = X$. In a HIL-HIP architecture, the input to the environment is given by: $u = \pi(X, S_p) + u_{ex}$, where $u_{ex} \in U_{ex}$ is an external input from the HIL, and S_p can be manually changed by the HIL. A *usage plan*, pl, consists of a temporal sequence of b external inputs ($u_{ex}(q_i)$) at times $0 \leq q_i \leq T_H$ and/or a system configuration changes ($s_p(p_i) \in S_p$) at times $0 \leq p_i \leq T_H$, T_H is the planning horizon. We denote the set of all possible plans as $\mathcal{P}\infty$.

Safety is defined as a logical predicate $Safe(X, f_\omega, pl)$ on the state of the HIP X, for a given dynamics f_ω, and a given usage plan pl. A plan pl is a *safety plan* if the predicate $Safe(X, f_\omega, pl)$ is satisfied at all times $t \in [0, T_H]$. The user burden for executing a plan $pl \in \mathcal{P}\infty$ is simply the length $|pl|$.

Safety tested HCAS: We assume that the HCAS (Fig. 1) is safety tested, which implies the existence of an "average user" denoted by coefficient distribution $\mathcal{D}(\omega)$, external input distribution $\mathcal{D}(U_{ex})$, and configuration distribution $\mathcal{D}(S_p)$, and a set $\mathcal{P} \subset \mathcal{P}\infty$ of safety plans with respect to the predicate $Safe(X, f_\omega, pl) : \omega \in \mathcal{D}(\omega)$ and $pl \in \mathcal{P}$.

Open World: It is characterized by open world events $(\omega', u'_{ex}, s'_p, X)$, such that either, a) $\omega' \notin \mathcal{D}(\omega)$, or b) $u'_{ex} \notin \mathcal{D}(U_{ex})$, or c) $s'_p \notin \mathcal{D}(S_p)$.

Personalized plan generation in open world: Given an open world event $(\omega', u'_{ex}, s'_p, X)$, find a safety plan $pl \in \mathcal{P}\infty$ with minimum length $|pl|$, such that $Safe(X, f_{\omega'}, pl)$ is satisfied in $[0, T_H]$.

4 Usage plan in open world case study

AP is an exemplary safety critical HIL-HIP HCAS with the open world problem. APs automatically infuse insulin, known as *micro bolus*, to control blood glucose levels around a *set point* S_p, while preventing hypoglycemia when blood glucose level falls below 70 mg/dl. All AP systems that are approved for human use by Food and Drug Administration (FDA) require the human user to provide external insulin in addition to the AP controller input to manage glucose variability due to meal intake, also called *meal bolus* (u_{meal}). This meal bolus is proportional to the carbohydrate content of the meal C, with *carb insulin ratio (CIR)* as the proportionality constant. While administering meal bolus any residual insulin in the body due to past insulin infusion, characterized by insulin on board (IoB) is subtracted (Eqn. 2).

$$u_{meal} = C/CIR - iob, \tag{2}$$

In addition, external insulin, *correction bolus* (u_{corr}), unrelated to meal can also be administered if the CGM reading is greater than the set point. u_{corr} is proportional to the difference between the current glucose value $G(t)$ and the set point, with *insulin sensitivity factor (ISF)* as the proportionality constant shown in Eqn. 3.

$$u_{corr} = (G(t) - S_p)/ISF - iob. \tag{3}$$

The residual insulin or IOB depends on the insulin pharmacokinetics, (Eqn. 4), obtained from Bergman Minimal Model (BMM) (Bergman, 2021), and is difficult for a human to guess.

$$\frac{dy}{dt} = z,\ \frac{dz}{dt} = -2k_1 z - k_1^2 y + k_1^2 u_{ex},\ \frac{diob}{dt} = -niob + p_1(y + I_b), \tag{4}$$

where $X = y, z, iob$, k_1 is the diffusion coefficient for insulin, and n and p_1 are patient specific metrics. Here, we assume that y and z are internal state variables of the BMM and are not measurable.

A safe usage plan for the AP HCAS is a sequence of set point changes, carbohydrate, meal bolus, and correction bolus intake actions such that the safety criteria of percentage time below 70 mg/dl in 24 hrs is less than 4% is satisfied. In STL the safety condition can be written as $G\ glucose > 70$, where G

is the globally true STL operator. An example safe usage plan is as follows: "*set point is 110 mg/dL at 6 am; set ISF to 50 at 6 am; set CIR to 15 at 6 am; breakfast with 20 g of carbohydrate at 8:30 am with 1 U of meal bolus; if CGM > 180 mg/dL, take correction bolus from Eqn. 3; lunch with 40 g of carbohydrate at 1 pm with meal bolus in Eqn. 2; if CGM > 210 mg/dL, take correction bolus from Eqn. 3; dinner with 30 g of carbohydrate at 6 pm with meal bolus in Eqn. 2, set set point at 90 mg/dL at 10 pm*" This is deemed safe by a sample patient from the virtual patient registry available in the FDA approved Type 1 Diabetes simulator developed at UVA PADOVA (Man et al., 2014). This is an example of the average user that follows the constrained distribution manifested by the virtual patient registry. The outcome is measured using four metrics: a) percentage time in range (TIR), $70mg/dl \leq CGM \leq 180mg/dl$, and b) time below range (TBR), when $CGM < 70mg/dl$.

4.1 Example of open world problems

P1: Out of distribution HIL with novel actions- Exercise induces changes in the glucose insulin dynamics of the HIP that are still un-characterized in clinical literature (Bally et al., 2020). As such none of the FDA approved AP can operate safely if the human user performs exercise, which makes it an OOD HIL action. Each AP system have their own open world usage plan for managing exercise, where the user is required to change the set point and take a small snack with 7g carbohydrate, 30 mins prior to exercise (Bally et al., 2020). Post exercise, the actions in the plan changes based on the type of exercise. If it is an aerobic exercise then the user needs to carefully monitor for hypoglycemia and take 7 g snack upon occurrence (Bally et al., 2020). Else if it is interval training, then the user needs to monitor for hyper-glycemia and then administer correction bolus.

P2: Long term time variance - Pregnancy induces fundamental long term change in the HIP dynamics, which none of the AP systems are designed to control (Bally et al., 2020). Safe and effective usage plan requires significant manual external bolus (meal or correction) decisions that increases plan management workload (Parent et al., 2023).

P3: Short term unexpected plan invalidation - Meal schedule is very important in maintaining safe usage plan. However, the human user may deviate from the meal plan on certain occasions. Such deviations again require careful manual intervention resulting in user burden.

5 Detailed Solution

We provide details of the two approaches.

5.1 Approach 1: LLM as autonomous planner

We use three LLMs, GPT o4 mini (OpenAI et al., 2023), Gemini 2.5 Flash (Team et al., 2023), and Llama 2 (Touvron et al., 2023) as autonomous planners in Approach 1. Llama 2 is chosen for autonomous planner since it establishes baseline for Approach 2 since given the computational resources available Llama 2 could be fine tuned.

5.2 Approach 2: LLM with safety feedback

It has two new compnents: a) physics model recovery module, and b) safety simulator.

5.2.1 Physics model recovery module

This module extracts the coefficients of Eqn. 4, which is given by the BMM. For this purpose we utilize sparse identification of non-linear dynamics strategy (SINDY-MPC) (Kaiser et al., 2018). Given the temporal traces of the state variables, SINDY-MPC gives the model ceofficient set ω for the HIP.

5.2.2 Safety simulator

Safety of the LLM-generated plan is evaluated using forward simulation. For the AP system, we instantiated virtual patients in the UVA PADOVA T1D simulator (Man et al., 2014) with BMM model coefficients obtained from the SINDY-MPC based model recovery. The model was simulated for planning horizon T_H to determine whether the LLM generated plan is safe (Details in supplement).

5.2.3 Safety guarantees

We have explored Control Lyapunov–Barrier Function (CLBF) theory (Romdlony and Jayawardhana, 2016) that guarantee safety under uncertainty.

Safety is a predicate $Safe(X, f_w, pl)$, state: X, dynamics: f_w, plan: pl. Existence of forward invariant (FI) set F is a theoretical safety guarantee, since if $X(0) \in F$ satisfies $Safe(X, f_w, pl)$ then $\forall$ subsequent $X(t)$ under f_w, pl, $X(t) \in F$ and satisfy $Safe(X, f_w, pl)$. A set F is FI, if $\exists$ CLBF $V(X, pl)$ where: a) $V(X, pl) > 0, \forall X \in F$, b) $V(Sp) = 0$, Sp is the set point, and c) $\forall X \in F, \exists \lambda > 0$ such that $L_{f_w}(V(X, pl)) + \lambda V(X, pl) < 0$, where L_{f_w} is the Lie derivative of $V(X, pl)$ with respect to f_w. Neural CLBF architecture, CLBFNN (Dawson et al., 2022), can search for a CLBF. To verify safety of a plan pl, we use the following steps:

Step 1: Calibrate the T1D simulator with real world data to obtain the individualized parameters.

Step 2: Instantiate a CLBFNN with dynamics f_w from calibrated T1D simulator, plan pl, and $X(0)$. Simulator is used to derive glucose trajectory and compute Lie derivatives of the CLBFNN penultimate layer, to obtain loss function, $||L_{f_w}(V(X, pl)) + \lambda V(X, pl) - \epsilon||^2$, $\epsilon > 0$ is a relaxation variable.

Step 3: Run the CLBFNN until loss function reaches 0 for $\epsilon = 0.01$.

Step 4: If the CLBFNN reaches zero loss in $< N(300)$ epochs then $\exists$ a CLBF, and plan pl is guaranteed safe.

We introduce a new metric v^g_{safe} to compute the number of guaranteed safe plans.

5.2.4 Physics contextualization of RLHF

The physics model contextualization is performed through instruction based tuning of the RLHF using the following question response prompt in each LLM repeated with different parameter set.

> Question: The insulin glucose dynamics of an individual with Type 1 Diabetes is given by the Bergman Minimal Model. Please use the Bergman Minimal Model with parameter set $k_1 = 0.09$, $n = 0.142$, and $p_1 = 0.02$ to compute the insulin on board in the next 30 mins when a bolus insulin of 2 U is taken now.
> Response: It is 1.2 U.

c) Instruction-Based Fine-tuning of LMs: Here, we fine-tune the LLM with embodied instruction prompts. These prompts explain the relationship between the dynamical coefficients of the HCAS ω and the functions $f_\omega(.)$.We used the ALPACA prompt response format (Chen et al., 2023) to generate training prompts, with three parts:

> "### Instruction: Find out the diffusion parameter from the Bergman Minimal Model with the following time series. The 40 values corresponding to 400 seconds of IOB values
> ### Input: 1.0 0.99948 0.99747 0.99411 0.98975 0.98473 0.97931 0.97371 0.96808 0.96254 0.95717 0.95205 0.94719 0.94264 0.93839 0.93446 0.93084 0.92752 0.92448 0.92171 0.9192 0.91693 0.91488 0.91303 0.91137 0.90988 0.90855 0.90735 0.90629 0.90534 0.90449 0.90374 0.90307 0.90248 0.90195 0.90148 0.90107 0.90071 0.90038 0.9001
> ### Response: 0.015"

Fine tuning is performed using 20,000 such sample prompts generated from the T1D simulator.

5.2.5 Deployment of Approach 2

In approach 2 due to resource limitations we fine tuned Phi 2 and Llama 2 models. To maintain consistency in the RLHF combinations we used the BARD RLHF in both the LLM configurations. During deployment, the HIL is expected to provide two types of inputs to facilitate the generation of the safe usage plan.

Inputs: a) **A Natural Language Prompt** that is provided by the HIL which describes a HCAS usage plan discovery task through a chat interface, BARD in this case (GoogleAI, 2023).

b) **A Trace** of physical dynamics of the HCAS denoted by $\tau = \{X(t) \forall t \in [t_0 - t_h, t_0]\}$, where t_0 is the current time and the t_h is the past horizon.

Steps for the Generation of a Safe Plan:

Step 1: The trace τ is used to recover the personalized dynamics coefficients for the real user ω^P using the SINDY-MPC based model recovery method (Kaiser et al., 2018).

Step 2: The coefficient ω^P is then used in an embedded prompt to solve the inverse inference problem for the physical dynamics, where the fine-tuned Llama-2 model is instructed to derive a trace $X(t) : \forall t \in [t_0, t_0 + t_f]$, where t_f is the future horizon for the given ω^P and the current state $X(t_0)$.

Step 3: This trace is used by a chat RL interface BARD to map to the appropriate plan.

Step 4: The plan is then evaluated for safety through forward simulation of the plant dynamics. Time below range (TBR) computed by counting the number of glucose values less than 70 mg/dl in 24 hrs plan horizon as a percentage. A plan is deemed safe if $TBR < 4\%$. In addition to TBR another plan evaluation is the number of human inputs required by the plan. These two metrics are used to provide feedback about plan safety to the RLHF through back prompting (done manually)

Step 5: If the plan is safe, then it is executed and the cycle continues. If unsafe, then the LLM is prompted to generate a new plan and Step 4 is re-executed. **Output:** A safe usage plan.

6 Evaluation

The open world scenario is introduced to each LLM using the following prompt sequence.

T1, Contextualization Prompt:

> I am a 30 year old woman with Type 1 Diabetes. I am using an automated insulin delivery system. Please learn insulin delivery algorithm from the following prompts.
> Q1: I am eating 30g carbs. Carb ratio is 5. Insulin on board is 3 U. How much bolus should I take?
> Answer: You should take 3 U bolus
> ...
> Q6: I am eating 7g carbs to avoid hypoglycemia. Carb ratio is 5. Insulin on board is 1 U. How much bolus should I take?
> Answer: You should take 0 U bolus
> **Full prompt in supplement**

T2, Prompts to test temporal dynamics understanding: These are test prompts where, a question about the dynamical properties of the environment such as insulin glucose interaction, is asked to the LLM and the answer is manually verified. Following prompt is an example.

> My diffusion coefficient is 0.3 $U/kg/hr$. What should be my IoB value in 2 hrs if my current glucose is 110 mg/dL and I just tool 2 U of insulin, I dont have any active insulin or active carbs, and I do not eat for the next 2 hrs.

The answer to this prompt should be 1.2 U according to the BMM used in UVA PADOVA simulator (Man et al., 2014).

T3, Prompts to test feasible action: These are a set of prompts that test whether the LLM is providing physically feasible actions. As an example:

> My carbohydrate to insulin ration is 15. What should be the meal bolus dose if I eat 30 grams of carbohydrate?

The answer to this prompt should be that the user should take 2 U of insulin according to Eqn. 2.

T4, Closed world planning prompt:

> I am going on a trip and I will be in downtown Chicago. My average blood glucose during daytime is 110 mg/dL. I like to eat three meals a day. What should I have for breakfast, lunch and Dinner so that my average CGM does not vary more than 10%?

T5, Open world planning prompt: We utilize three types of open world planning prompt:

Prompt with P1 to test novel action (NA):

> I want to do interval training for 30 mins in the next hour. I dont have a healthcare provider to help me now. My current CGM reading is 85 mg/dL. My Insulin sensitivity factor setting is 50, and my carbohydrate to insulin ratio is 0.36. I dont want hypoglycemia after exercise. How should I configure the set point of my device? Should I eat a snack to avoid hypoglycemia? Should I take any insulin with the snack.

Prompt with P2 to test model adaptation (MA):

> I am in sixth week of pregnancy. What should be my meal plan throughout the day and exercise plan to maintain $>$ 70% time in range?

Prompt with P3 to test plan invalidation (PI):

> It is 6 pm now. I followed your meal plan. But I feel like having a quarter piece of a 0.5 pound tiramisu cake. My current glucose is 121 mg/dL and I ate afternoon snacks at 3 pm. How much should I eat so that my glucose does not go above 180 mg/dL? Also should I take insulin with this cake? If so how much?

6.1 Dataset Description

Real world dataset for personalization: We used two real world datasets: a) 10 Individuals with T1D using Control IQ model predictive control automated insulin delivery system for 2 weeks obtained from JAEB center (JAEB center, 2023), and b) 24 T1D women with pregnancy using Control IQ AP for 30 weeks in LOIS-P study (O'Malley et al., 2021). The data are used to show model adaptation (MA) and quick replanning (QPR) capacities of our approach (Table 1).

Simulation data: We used a virtual patient with BMM parameters shown in Table 3 as simulation settings. We generated 218 meal instances of sizes ranging from 7 g to 50 g for various carb ratio settings ranging from 10 to 25. We set up the virtual patients with prior insulin usage starting from 30 mins before a meal to 3 hrs before a meal. We integrated an MPC controller similar to Control IQ that generates the insulin outputs $u = \pi(X, s)$ in addition to the prior bolus and also the meal bolus. The data is used to validate the accuracy of the SINDY-MPC model recovery technique and also to show novel action search (NA), plan invalidation (PI) tackling capacities and understanding of dynamical systems (DS) properties (Table 1).

6.2 Evaluation Experiments

We evaluate each approach for performance on:

Model Adaptation (MA) is evaluated in two parts.

a) Model recovery accuracy - We measure the accuracy of the recovered model using SINDY-MPC method in replying real world data from normal T1D and pregnant T1D individuals using real world datasets. **Metrics:** Accuracy is measured using the root mean square error (RMSE) between replayed glucose and groung truth data. In addition, we also observe the variance in the underlying model coefficients across the two sub-populations.

b) Responding to contextualization prompts ($E_{context}$)*:* We used 100 T1 prompts with 10 prompts each from from 5 non-pregnant patients, and 5 pregnant patients. **Metrics:** For each approach we evaluate the percentage of bogus responses v_b, as evaluated by a human user, a measure of hallucination, and the %-age of responses which are within 10% error of the ground truth v_g.

Protocol for hallucination evaluation: A set protocol was used to flag hallucination which are objective, verifiable, and reproducible:

i) LLM suggests 2 consecutive meals in $< 15 mins$

ii) LLM suggests a meal > 300 g of carbohydrate

iii) consecutive > 20 U (medically approved maximum single dose) insulin bolus suggested in $< 15 mins$.

iv) LLM suggested a setpoint of < 20 mg/dl (lowest medically allowed).

v) Wrong correction bolus Insulin (Eqn. 3) computed by LLM.

c) understanding temporal dynamics (E_{time})**:** We generated 12 T2 and 12 T3 prompts as described in Section 6 and evaluate each approach using the **metrics** v_b and v_g on the response.

d) generating safe usage plans for prompts in the closed world (E_{close})**:** We generated 20 T4 plans and evaluated each approach using the v_b metric and percentage of times the LLM generates safe plans v_{safe} and the length of the plan v_n as a quantifier of user burden.

e) generating safe usage plans for prompts in the open world (E_{open})**:** We generated 102 open world T5 prompts with equal distribution across the three problems $P1$ (evaluate NA), P2 (evaluate MA), and $P3$ (evaluate PI). We used v_{safe} and v_n as **metrics** for evaluation.

Table 2: Performance of LLM as autonomous planner in the evaluations. NAN means not applicable

LLM	Evaluation	Hallucination v_b	Accuracy v_g	Safety v_{safe}	User burden v_n
GPT o4 mini	$E_{context}$	32%	68%	NAN	NAN
GPT o4 mini	E_{time}	41.6%	25%	NAN	NAN
GPT o4 mini	E_{close}	0%	NAN	20%	$13(\pm 2)$
GPT o4 mini	E_{open}	0%	NAN	14.7%	$9(\pm 4)$
Gemini 2.5 Flash	$E_{context}$	22%	46%	NAN	NAN
Gemini 2.5 Flash	E_{time}	33.3%	25%	NAN	NAN
Gemini 2.5 Flash	E_{close}	5%	NAN	25%	$11(\pm 3)$
Gemini 2.5 Flash	E_{open}	0%	NAN	30.4%	$8(\pm 2)$
Llama 2	$E_{context}$	14%	86%	NAN	NAN
Llama 2	E_{time}	16.6%	25%	NAN	NAN
Llama 2	E_{close}	0%	NAN	50%	$12(\pm 4)$
Llama 2	E_{open}	0%	NAN	40.2%	$9(\pm 4)$

Table 3: BMM coefficients derived using SINDY-MPC for AP in real world datasets.

Data Type	$k_1(10^{-2})$	$n(10^{-2})$ 1/min	$p_1(10^{-2})$ 1/min	RMSE
Normal	9.8(± 0.3)	14.06(± 0.9)	2.8(± 0.3)	11.1(± 1.3)
Pregnancy	7.18 (± 0.4)	15.9 (± 1.2)	2.5 (± 0.2)	12.8(± 3.4)
Difference	2.6 ($p < 0.01$)	1.8 ($p = 0.04$)	0.3 ($p = 0.2$)	1.7 (p=0.7)

6.3 Approach 1: autonomous planners

Table 2 shows that LLMs are poor autonomous planners usage plan generation. Although LLAMA 2 and Gemini 2.5 Flash models had overall less hallucinations, still their accuracy in extracting temporal properties is poor. Moreover, they fail to generate safety plans more than 50% of the time in either closed or open world scenarios. The plan lengths are variable across the LLMs but required a high number (best average 8) of responses from the human over 24 hrs.

6.4 Approach 2: LLM + safety verifier

Model Adaptation evaluation:

Accuracy of Model coefficient estimation using SINDY-MPC The SINDY-MPC model could extract model coefficients from the real world datasets with good replay RMSE as shown in Table 3. There was no statistical difference (ttest p value) between the RMSE of pregnant and normal T1D individuals indicating that the model recovery process recovered diverse models which performed equally well on both the sub-populations. This is further corroborated by the significant differences between the model coefficients of the two cohorts (Table 3).

Contextualization performance: Despite differences in cohort characteristics Table 4 shows that fine tuned LLAMA 2 model was 92% accurate in answering contextualization prompts. The above two results show that Approach 2 exhibits model adaptation capability.

Understanding temporal dynamics: Table 4 shows that finetuned LLAMA 2 integrated with BARD RLHF is 83.3% accurate as compared to 16.6% if not finetuned on T2 prompts. This is significant since the finetuned LLAMA 2 can accurately answer prompts related to temporal characteristics of dynamics showing understanding of dynamical properties and beats the untuned latest GPT o4 mini or Gemini 2.5 Flash (Table 2).

Table 4: Performance of LLM + safety verifier in the evaluations. NA means not applicable

LLM	Evaluation	Hallucination v_b	Accuracy v_g	Safety v_{safe}	User burden v_n
BARD RLHF + LLAMA 2	$E_{context}$	0%	92%	NA	NA
–	E_{time}	0%	83.3%	NA	NA
–	E_{close}	0%	NA	100%	$9(\pm 3)$
–	E_{open}	0%	NA	91.2%	$6(\pm 2)$
BARD RLHF + Phi 2	$E_{context}$	13%	65%	NA	NA
–	E_{time}	0%	58.3%	NA	NA
–	E_{close}	0%	NA	90%	$10(\pm 4)$
–	E_{open}	0%	NA	69.6%	$6(\pm 5)$

PI and QPR performance: Table 4 shows that the BARD+LLama2 LLM architecture achieve 91.2% safe plans without any hallucination as identified by the human evaluator for plan invalidation and quick replanning tasks. The average delay in plan regeneration was 1 min 2s (± 45 s) on the LLAMA 2 model indicating fast enough response for the insulin management example. Moreover, we see that the burden on the user is also significantly reduced as compared to Approach 1 as demonstrated by the decreased plan length.

Novel action search performance: Table 5 shows the comparison of Control IQ MPC + HIL and MPC + LLM in the closed world setting with the Approach 2 on LLAMA 2, MPC+LLM+verifier+open. We see that the Approach 2 has the least TBR showing safe operation. Further, the correction bolus and set point suggestions are significantly different from closed world setting indicating novel actions suggested by the Approach 2 which also leads to safe operation.

Table 5: Comparison of usage plan generation methods with novel action exploration.

Method	**TIR**	**TBR**	**Correction bolus (U)**	**Set point (mg/dl)**
MPC + HIL	78.2%(±21.1)	4.4%(±4.2)	5.4(±4.2)	105(±12)
MPC + LLM + closed	81.2%(±18)	8.8%(±1.1)	5.6(±3.1)	100(±10)
MPC + LLM + verifier + open	**81%(±12)**	**3%(±2)**	**11.6 (±2)**	**75 (± 13)**

6.5 Ablation analysis

We demonstrate the relative importance of each step of Approach 2 by taking the BARD+Llama 2 model and re-evaluating for the closed and open world planning task by removing the components one by one. We generate the following LLM configurations for the ablation study:

Approach 2 - safety verifier, where we remove the safety verifier and iterate twice incorporating plan length as the only plan quality metric.

Table 6: Performance of LLM + safety verifier in the evaluations. NA means not applicable

Approach 2 -	Evaluation	Safety v_{safe}	User burden v_n
safety verifier	E_{close}	75%	9(±4)
safety verifier	E_{open}	61.8%	6(±2)
plan length	E_{close}	100%	12(±0)
plan length	E_{open}	87.2%	8(±4)
fine tuning	E_{close}	100%	9(±7)
fine tuning	E_{open}	39.2%	7(±3)
model contextualization	E_{close}	85%	14(±7)
model contextualization	E_{open}	81.4%	10(±6)

Table 7: Safety guarantees of LLM autonomous planner.

LLM	**Evaluation**	$\mathbf{v^g_{safe}}$
GPT o4-mini	E_{close}	19%
GPT o4-mini	E_{open}	14.4%
Gemini 2.5 Flash	E_{close}	23%
Gemini 2.5 Flash	E_{open}	30.0%
Llama 2	E_{close}	47%
Llama 2	E_{open}	39.1%

Approach 2 - plan length, where we remove the plan length quality metric from the back prompts. **Approach 2 - fine tuning**, where we remove personalized model based fine tuning. **Approach 2 - physics model contextualization**, where the RLHF is not aware of the physics model.

Table 6 shows that removal of safety verifier has significant effect on the safety of the generated plan. Removal of plan length feedback not only has impact on user burden but also results in more safety violation for open world scenarios. This may be due to increased insulin delivery actions in the updated plans. Removal of fine tuning surprisingly has no effect on the safety of the close world plans but has significant effect on the safety of open world plans. This is intuitive since nearly 30% of the open world scenarios involved change in HIP dynamics which cause the close world safe plans to be unsafe. Model contextualization has reduced plan safety and increased user burden.

6.6 Results for safety guarantees

Baselines in Table 7 show modest safety generalization: GPT o4-mini (19/14.4), Gemini 2.5 Flash (23/30.0), and Llama 2 (47/39.1) for v^g_{safe} under E_{close}/E_{open}. Table 8 shows that RLHF-based hybrids substantially raise performance: BARD RLHF + Llama 2 (93.5/89.1) and BARD RLHF + Phi 2 (84/65.8), narrowing the $E_{close} \to E_{open}$ gap. Table 9 ablations ("Approach 2 minus") indicate contribution hierarchy: removing the *safety verifier* hurts most (72/60.6), removing *model contextualization* also degrades (83/80.1), removing *plan length* has minor effect (99/86.7), and removing *fine tuning* preserves $E_{close} = 100$ but collapses $E_{open} = 30.9$. Overall, verifier and contextualization drive the largest gains over baselines, fine tuning is crucial for E_{open} (harder, more OOD-like), and E_{open} remains the more challenging regime.

Table 8: Safety guarantees for LLM with safety verifier.

LLM	**Evaluation**	$\mathbf{v^g_{safe}}$
BARD RLHF + Llama 2	E_{close}	93.5%
BARD RLHF + Llama 2	E_{open}	89.1%
BARD RLHF + Phi 2	E_{close}	84%
BARD RLHF + Phi 2	E_{open}	65.8%

Table 9: Safety guarantees for ablation studies.

Approach 2 minus	**Evaluation**	$\mathbf{v^g_{safe}}$
safety verifier	E_{close}	72%
safety verifier	E_{open}	60.6%
plan length	E_{close}	99%
plan length	E_{open}	86.7%
fine tuning	E_{close}	100%
fine tuning	E_{open}	30.9%
model contextualization	E_{close}	83%
model contextualization	E_{open}	80.1%

6.7 Quantification of Backprompting effort

We measure average back-prompts b_{avg}, no-back-prompt rate p_b, and fail-safe trigger rate as shown in the table below for Approach 2's E_{close} and E_{open}.

Table 10: Back-prompting performance.

LLM	**Evaluation**	**Back prompting effort** $(\mathbf{b_{avg}})$	**% no back-prompts** $(\mathbf{p_b})$	**% fail-safe mode triggers**
BARD + Llama 2	E_{close}	0.2	84%	3%
BARD + Llama 2	E_{open}	0.81	67%	8.4%
BARD + Phi 2	E_{close}	0.36	76%	4.3%
BARD + Phi 2	E_{open}	0.94	59%	11.2%

7 Conclusions

This paper demonstrates using LLMs to plan the personalized operation of an HCAS. It highlights a key property of LLM planners: they can explore novel actions and reason about dynamical systems with rapid replanning. Our main observations are that LLMs can plan control tasks when two steps are carefully designed: (a) contextualization of the chat RL, and (b) fine-tuning the LLM's internal weights via embodied training, where textual instructions and interpretations intertwine with traces from the real-world system. Our method applies to any open-world planning over a dynamical system with Lipschitz-continuous dynamics and temporally concatenated inputs—e.g., artificial pancreas with human in the loop operation, or semi-autonomous cars that drive under autonomy and hand off control to the driver in critical scenarios or perception failures.

Future Work: Safety assured LLM based open world planners can be integrated into HIL control design (Banerjee et al., 2025a) and assistive technologies such as AIIM (Banerjee et al., 2025b) to seamlessly integrate open world planning capabilities in safety critical systems.

8 Acknowledgments

The work is partly funded by DARPA AMP-N6600120C4020, DARPA FIRE-P000050426, NSF FDT-Biotech 2436801, Mayo ASU seed grant (ARI-332627) and Helmsley Charitable Trust - 2-SRA-2017-503-M-B. We thank Aranyak Maity for the Llama implementation and Payal Kamboj for helping with literature survey.

9 Limitations

The proposed framework effectively automates the generation of safety plans for HCAS while optimizing user burden measures using the plan size. However, in addition to safety and plan size there are several other metrics that are important for the human user which is yet to be studied.

Quality of life for a safety plan: A safety plan should not restrict the human user from doing day to day activities. As seen in case of exercise for T1D, users are reluctant to exercise because of the burden of glucose management using the AP. In our paper, we did not characterize quality of life as a plan quality metric. Quantification of quality of life for a plan can have significant individual variance and needs to be studied in more detail.

Guarantees on LLM performance: In the demonstrated plan generation technique, we enforce the LLM to either produce a safe plan in two iteractions of plan quality measurement or relinquish control and default to fail safe modes. Hence as such if the system is not in fail safe mode then the plan is guaranteed safe. However, this strategy may often result in unnecessary fail safe mode trigger due to failure to find a suitable plan. Formal guarantees on LLM performance with unrestricted iterations is difficult to provide.

Quantifying back prompting overhead: Back prompting was necessary in all LLMs to obtain domain specific safety plans. Especially all LLMs seem to have amnesia of contexts and through back prompting required the human user to remind about context information previously provided in the conversation. We have not quantified the overhead of back prompting but is an important efficacy metric.

10 Potential Risks

Usage of LLMs in critical medical applications is not devoid of its risks. Unsafe decisions has safety risks. This is addressed to some extent in this paper. However, safety certification requires some form of safety gurantees. Such guarantees cannot be currently provided for LLMs. Without such guarantees it may not pass certification studies. Hence, LLM based techniques have to be used off label. This is a significant risk and has to be prevented until research on safety guarantees on LLMs mature.

References

Stefano V. Albrecht and Subramanian Ramamoorthy. 2015. Combining symbolic and numeric representations of uncertainty in constraint-based planning. *Journal of Artificial Intelligence Research*, 52:113–157.

Lia Bally, Patrick Kempf, Thomas Zueger, Christian Speck, Nicola Pasi, Carlos Ciller, Katrin Feller, Hannah Loher, Roland Rosset, Matthias Wilhelm, and 1 others. 2020. Exercise management in type 1 diabetes: A consensus statement. *The Lancet Diabetes & Endocrinology*, 8(10):811–825.

Ayan Banerjee and Sandeep Gupta. 2024a. Emily: Extracting sparse model from implicit dynamics. In *ECAI Workshop on Machine Learning Meets Differential Eqns: From Theory to Applications*, pages 1–11.

Ayan Banerjee and Sandeep KS Gupta. 2014. Analysis of smart mobile applications for healthcare under dynamic context changes. *IEEE Transactions on Mobile Computing*, 14(5):904–919.

Ayan Banerjee and Sandeep KS Gupta. 2024b. Recovering implicit physics model under real-world constraints. In *ECAI 2024*, pages 737–744. IOS Press.

Ayan Banerjee, Imane Lamrani, and Sandeep K. S. Gupta. 2025a. Synthesizing operationally safe controllers for human-in-the-loop human-in-the-plant hybrid close loop systems. In *Pattern Recognition*, pages 17–35, Cham. Springer Nature Switzerland.

Ayan Banerjee, Aranyak Maity, Payal Kamboj, and Sandeep KS Gupta. 2024. Cps-llm: large language model based safe usage plan generator for human-in-the-loop human-in-the-plant cyber-physical system. *arXiv preprint arXiv:2405.11458*.

Ayan Banerjee, Aranyak Maity, Imane Lamrani, and Sandeep K.S. Gupta. 2025b. Towards certified safe personalization in learning enabled human-in-the-loop human-in-the-plant systems. *J. Emerg. Technol. Comput. Syst.* Just Accepted.

Richard N Bergman. 2021. Origins and history of the minimal model of glucose regulation. *Frontiers in endocrinology*, 11:583016.

Avrim L. Blum and Merrick L. Furst. 1997. Graphplan: A planning graph-based automated planning system. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1125–1134.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI gym. *arXiv preprint arXiv:1606.01540*.

Matteo Cardellini, Marco Maratea, Francesco Percassi, Enrico Scala, and Mauro Vallati. 2023. Taming discretised pddl+ through multiple discretisations. In *Knowledge Engineering for Planning and Scheduling (KEPS) Workshop at ICAPS*.

Anthony R. Cassandra, Leslie P. Kaelbling, and James Kurien. 1999. Point-based value iteration: An anytime algorithm for pomdps. In *Proceedings of the Fifteenth International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 55–62.

Guanqi Chen, Lei Yang, Ruixing Jia, Zhe Hu, Yizhou Chen, Wei Zhang, Wenping Wang, and Jia Pan. 2024. Language-augmented symbolic planner for open-world task planning. *arXiv preprint arXiv:2407.09792*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and 1 others. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. 2022. Safe nonlinear control using robust neural lyapunov-barrier functions. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1724–1735. PMLR.

Yan Ding, Xiaohan Zhang, Saeid Amiri, Nieqing Cao, Hao Yang, Chad Esselink, and Shiqi Zhang. 2022. Robot task planning and situation handling in open worlds. *arXiv preprint arXiv:2210.01287*.

Richard E. Fikes and Nils J. Nilsson. 1971. Strips: A new approach to the application of theorem proving to problem solving. *Artif. Intell.*, 2(3-4):189–208.

Elliot Gestrin, Marco Kuhlmann, and Jendrik Seipp. 2024. Nl2plan: Robust llm-driven planning from minimal text descriptions. In *HAXP*.

Shivam Goel, Panagiotis Lymperopoulos, Ravenna Thielstrom, Evan Krause, Patrick Feeney, Pierrick Lorang, Sarah Schneider, Yichen Wei, Eric Kildebeck, Stephen Goss, Michael C. Hughes, Liping Liu, Jivko Sinapov, and Matthias Scheutz. 2024. A neurosymbolic cognitive architecture framework for handling novelties in open worlds. *Artificial Intelligence*, 331:104111.

GoogleAI. 2023. Bard: A large language model from google ai. https://ai.googleblog.com/2022/01/lamda-language-model-for-dialogue.html. Accessed December 23, 2023.

Malte Helmert. 2006. Fast downward: A fast, deterministic planner. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 368–387.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2553–2561.

Sukai Huang, Nir Lipovetzky, and Trevor Cohn. 2024. Planning in the dark: Llm-symbolic planning pipeline without experts. *arXiv preprint arXiv:2409.15915*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR.

JAEB center. 2023. JAEB center dataset. https://public.jaeb.org/datasets/diabetes.

Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. 2018. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A*, 474(2219):20180335.

Alicia Li, Nishanth Kumar, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. 2024. Learning to bridge the gap: Efficient novelty recovery with planning and reinforcement learning. *arXiv preprint arXiv:2409.19226*.

Ruoyu Li and James T. Allison. 2017. Nonlinear optimization for planning problems with complex constraints and objectives. *IEEE Transactions on Automation Science and Engineering*, 14(3):1419–1432.

Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. 2014. The uva/padova type 1 diabetes simulator: new features. *Journal of diabetes science and technology*, 8(1):26–34.

Yurii Nesterov. 1982. Lipschitzian optimization without the lipschitz constant. *Optimization Methods and Software*, 1(1):167–179.

Grenye O'Malley, Basak Ozaslan, Carol J Levy, Kristin Castorino, Donna Desjardins, Camilla Levister, Shelly McCrady-Spitzer, Mei Mei Church, Ravinder Jeet Kaur, Corey Reid, and 1 others. 2021. Longitudinal observation of insulin use and glucose sensor metrics in pregnant women with type 1 diabetes using continuous glucose monitors and insulin pumps: the lois-p study. *Diabetes technology & therapeutics*, 23(12):807–817.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, and 263 others. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Héctor Palacios and Hector Geffner. 2016. Learning action models for re-planning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1670–1676.

Cassandra Parent, Elodie Lespagnol, Serge Berthoin, Sémah Tagougui, Joris Heyman, Chantal Stuckens, Iva Gueorguieva, Costantino Balestra, Cajsa Tonoli, Bérengère Kozon, and 1 others. 2023. Barriers to physical activity in children and adults living with type 1 diabetes: A complex link with real-life glycemic excursions. *Canadian journal of diabetes*, 47(2):124–132.

Joelle Pineau, Geoff Gordon, and Sebastian Thrun. 2003. Point-based value iteration for continuous pomdps. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025–1030.

Juan A. Rodriguez-Aguilar, Ulle Endriss, Sarvapali D. Ramchurn, and Michael Luck. 2010. Efficient genetic algorithms for multi-agent plan coordination. *Journal of Artificial Intelligence Research*, 39:59–101.

Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. 2016. Stabilization with guaranteed safety using control lyapunov-barrier function. *Automatica*, 66(C):39–47.

Dorsa Sadigh, Shankar Sastry, Sanjit A Seshia, and Anca D Dragan. 2016. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and systems*, volume 2, pages 1–9. Ann Arbor, MI, USA.

SP Sharan, Francesco Pittaluga, Manmohan Chandraker, and 1 others. 2023. Llm-assist: Enhancing closed-loop planning with language-based reasoning. *arXiv preprint arXiv:2401.00125*.

Kartik Talamadupula, J Benton, Subbarao Kambhampati, Paul Schermerhorn, and Matthias Scheutz. 2010. Planning for human-robot teaming in open worlds. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 1(2):1–24.

Marcus Tantakoun, Christian Muise, and Xiaodan Zhu. 2024. Fixmyplan: Leveraging large language models to fix ill-defined models and incorrect plans. In *Bridging Planning and Reasoning in Natural Languages with Foundational Models Workshop*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Karthik Valmeekam, Sarath Sreedharan, Matthew Marquez, Alberto Olmo, and Subbarao Kambhampati. 2023. On the planning abilities of large language models (a critical investigation with a proposed benchmark). *arXiv preprint arXiv:2302.06706*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Zhu Wang, Ishika Singh, Robin Jia, and Jesse Thomason. 2025. Language models can infer action semantics for symbolic planners from environment feedback. In *NAACL*.

Bin Xu, Ayan Banerjee, and Sandeep KS Gupta. 2025. Model recovery at the edge under resource constraints for physical ai. In *ECAI 2025*, pages 1–8. IOS Press.

Hankz Hankui Zhuo, Tuan Nguyen, and Subbarao Kambhampati. 2013. Refining incomplete planning domain models through plan traces. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2451–2456.

A LLM Prompts

A.1 LLM Prompts for plan generation - GPT o4 mini

[sharp corners, colback=white, colframe=black, boxrule=1pt] **Learn from the following prompts**

Q1: I am eating 30g carbs. Carb ratio is 5. Insulin on board is 3 U. How much bolus should I take?

Answer: You should take 3 U bolus

Q2: I am eating 20g carbs. Carb ratio is 5. Insulin on board is 1 U. How much bolus should I take?

Answer: You should take 3 U bolus

Q3: I am eating 7g carbs to avoid hypoglycemia. Carb ratio is 5. Insulin on board is 2 U. How much bolus should I take?

Answer: You should take 0 U bolus

Q4: I am eating 60g carbs. Carb ratio is 5. Insulin on board is 4 U. How much bolus should I take?

Answer: You should take 8 U bolus

Q5: I am eating 25g carbs. Carb ratio is 5. Insulin on board is 3 U. How much bolus should I take?

Answer: You should take 2 U bolus

Q6: I am eating 7g carbs to avoid hypoglycemia. Carb ratio is 5. Insulin on board is 1 U. How much bolus should I take?

Answer: You should take 0 U bolus [sharp corners, colback=white, colframe=black, boxrule=1pt] **GPT o4 mini Response**

It seems like you've provided a set of prompts and corresponding answers related to insulin bolus calculations in the context of carbohydrate intake, carb ratio, and insulin on board. Each prompt presents a scenario with specific values, and the corresponding answer provides the recommended bolus insulin amount based on those values.

If you have any specific questions or if there's a particular aspect you'd like me to focus on or elaborate on regarding these prompts, please let me know, and I'll do my best to assist you.

[sharp corners, colback=white, colframe=black, boxrule=1pt] **Prompt** I am eating 45 g carbs. Carb ratio is 5. Insulin on board is 2 U. How much bolus should I take?

[sharp corners, colback=white, colframe=black, boxrule=1pt] **GPT o4 mini Response**

Based on the patterns observed in the provided prompts, the calculation for bolus insulin is determined by the formula: Bolus dose=(Carb intake/-Carb ratio)+Insulin on board Substituting in the values: Bolus dose=(45/5)+2=9+2=11 U Therefore, based on the provided information, you should take 11 units of bolus insulinfor 45g of carb intake with a carb ratio of 5 and 2 units of insulin on board.

A.2 LLM Prompts for fine-tuning Llama2 Model

[sharp corners, colback=white, colframe=black, boxrule=1pt] **### Instruction:** Find out the diffusion parameter from the Bergman Minimal Model with the following time series. The 40 values corresponding to 400 seconds of IOB values
### Input: 1.0 0.99948 0.99747 0.99411 0.98975 0.98473 0.97931 0.97371 0.96808 0.96254 0.95717 0.95205 0.94719 0.94264 0.93839 0.93446 0.93084 0.92752 0.92448 0.92171 0.9192 0.91693 0.91488 0.91303 0.91137 0.90988 0.90855 0.90735 0.90629 0.90534 0.90449 0.90374 0.90307 0.90248 0.90195 0.90148 0.90107 0.90071 0.90038 0.9001
### Response: 0.015

A.3 LLM Prompts for final query to fine-tuned Llama2 Model (LLAMA 2)

[sharp corners, colback=white, colframe=black, boxrule=1pt] **Prompt to LLAMA 2**

Below is an instruction that describes the task of finding the Insulin On Board of a type 1 diabetic patient paired with a diffusion parameter of the Bergman Minimal Model for an insulin intake. Write a corresponding output that is the Insulin On Board time series.

### Instruction: My diffusion parameter is 0.015081113389040546 and I took an insulin dosage. What is my Insulin On Board percentage time series?

### Response:

[sharp corners, colback=white, colframe=black, boxrule=1pt] **Response from LLAMA 2**

### Response: 1 0.9992 0.99622 0.9915 0.98568 0.98 0.97398 0.96744 0.9614 0.95591 0.95099 0.94646 0.94225 0.93842 0.93502 0.93198 0.92925 0.92684 0.92469 0.9227 0.92087 0.91922 0.91777 0.91647 0.91526 0.91416 0.91315 0.91222 0.91138 0.9106 0.90987 0.90919 0.90859 0.90803 0.9075 0.90699 0.9065 0.90599 0.90551 0.90505 0.9046 0.90415 0.90372 0.9033 0.90289 0.9025 0.9021 0.90172 0.90137 0.90105 0.90076 0.90049 0.90022 0.900 0.89978 0.89958

B T1D Simulator

The T1D simulator is implemented in Simulink following the architecture shown in Figure 3. The simulator can be reun by providing a meal input, meal time, bolus input, bolus time, initial glucose, and duration of simulation. In addition, the insulin sensitivity factor, carbohydrate ratio can also be specified.

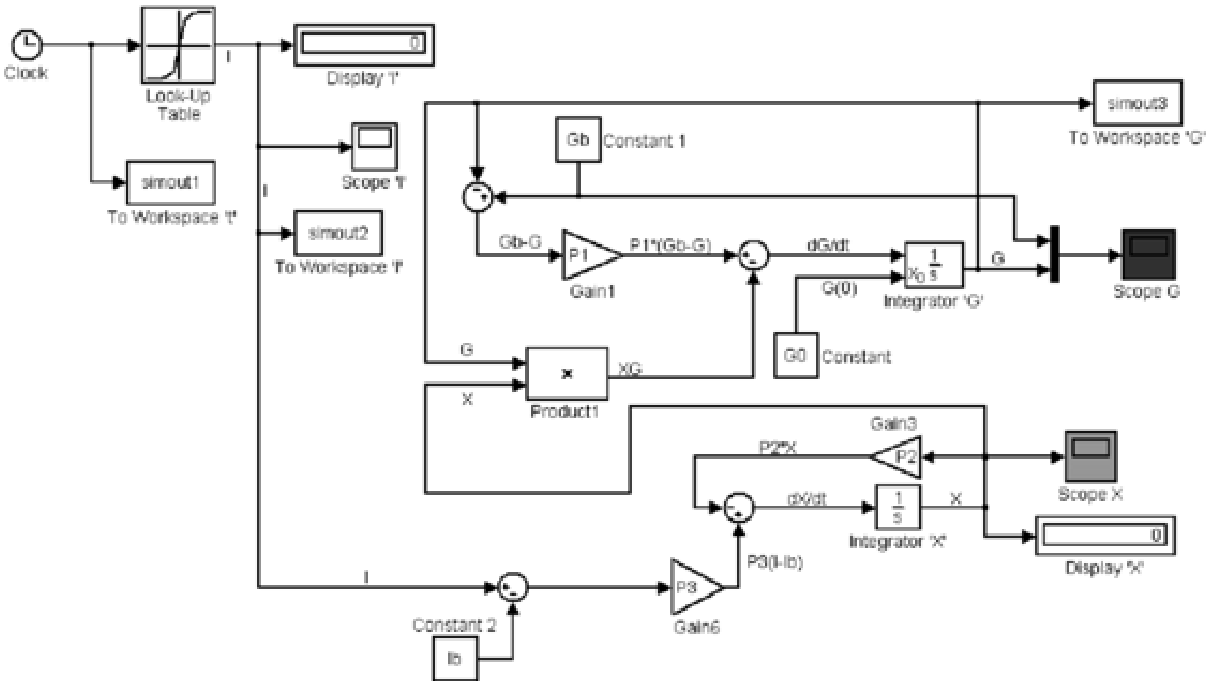

Figure 3: Simulink architecture of Type 1 Diabetes Simulator.