UNIVERSITAT DE BARCELONA

MiceLab
Modeling, Identification & Control Engineering

Final Degree Project

Biomedical Engineering Degree

A Foundation Model for Glucose Dynamics in Type 1 Diabetes: Pre-training, Evaluation, and Knowledge Transfer

Un Model Fundacional per a la Dinàmica de la Glucosa en la Diabetis Tipus 1: Preentrenament, Avaluació i Transferència de Coneixement

Barcelona, 8th of June 2026

Author: Marc Font Oliveras

Director: Josep Vehí Casellas

Tutor: Juan Ignacio Barrios

Abstract

Personalised management of Type 1 Diabetes (T1D) depends on understanding each patient's glucose dynamics, yet supervised labels for clinical models are scarce and costly. The growing availability of continuous glucose monitoring (CGM) wearables creates an opportunity: self-supervised pre-training on unlabelled CGM sequences could learn physiological representations that transfer to clinical tasks.

Two transformer foundation models were trained on 934 adult T1D patients from the METABONET and T1DEXI datasets. The first is a BERT-style bidirectional encoder trained with masked span reconstruction; the second a GPT-style causal decoder trained with next-token prediction. Both take CGM alongside ODE-derived plasma insulin and carbohydrate absorption as input, providing richer physiological context than glucose alone.

The encoder's reconstruction capability was first validated on gap imputation. Lightweight task heads were then fine-tuned on short-term glucose forecasting and nocturnal hypoglycaemia risk prediction, and compared against LSTM baselines trained from scratch. Fine-tuned foundation model variants matched or exceeded baseline performance across all tasks without exposure to any task labels during pre-training.

Patient embeddings from both architectures were analysed using linear probes and dimensionality reduction. The embeddings recovered insulin sensitivity factor and carbohydrate-to-insulin ratio more accurately than a regression model with access to the patient's full 24-hour insulin and absorption profiles, parameters the models were never trained to predict. Both embedding spaces stratified patients by glycaemic risk without supervision.

These results show that self-supervised pre-training on CGM sequences learns representations that transfer across clinical tasks and encode physiological information beyond what the pre-training objective explicitly demands.

Keywords: Type 1 Diabetes, Continuous Glucose Monitoring, Foundation Model, Transformer, Self-supervised Learning, Transfer Learning, Representation Learning

Resum

La gestió personalitzada de la diabetis tipus 1 (DT1) depèn de la comprensió de la dinàmica de la glucosa de cada pacient, però les etiquetes supervisades per a models clínics són escasses i costoses. La disponibilitat creixent de dispositius de monitoratge continu de glucosa (MCG) ofereix una oportunitat: el preentrenament autosupervisat en seqüències de MCG no etiquetades per aprendre representacions fisiològiques transferibles a tasques clíniques.

S'han entrenat dos models fonamentals transformer amb 934 pacients adults amb DT1 de les fonts METABONET i T1DEXI. El primer és un codificador bidireccional d'estil BERT entrenat a partir de la reconstrucció de segments emmascarats; el segon, un descodificador causal d'estil GPT entrenat mitjançant predicció del següent token. Ambdós utilitzen com a entrada el MCG juntament amb la insulina plasmàtica i l'absorció de carbohidrats, proporcionant un context fisiològic més ric que la glucosa sola.

La capacitat de reconstrucció del codificador s'ha validat primer en la imputació de buits. Després, s'han integrat models lleugers per a la predicció de la glucosa a curt termini i del risc d'hipoglucèmia nocturna, i s'han comparat amb models base LSTM entrenats des de zero. Les variants del model fonamental han igualat o superat el rendiment dels models base en totes les tasques sense haver estat exposats a etiquetes de tasca durant el preentrenament.

Els embeddings dels cohorts d'ambdues arquitectures s'han analitzat mitjançant models lineals i reducció de dimensionalitat. Els embeddings han recuperat el factor de sensibilitat a la insulina i la ràtio carbohidrat-insulina amb més precisió que un model de regressió amb accés al perfil complet de 24 hores d'insulina i absorció del pacient, paràmetres que els models mai han estat entrenats per predir. Per altra banda, ambdós espais d'embeddings estratifiquen els pacients per risc glicèmic de forma no supervisada.

Aquests resultats demostren que el preentrenament autosupervisat en seqüències de MCG aprèn representacions que es transfereixen entre tasques clíniques i codifiquen informació fisiològica més enllà de la corresponent a l'objectiu de preentrenament.

Paraules clau: Diabetis Tipus 1, Monitoratge Continu de Glucosa, Model Fonamental, Transformer, Aprenentatge Autosupervisat, Aprenentatge per Transferència, Aprenentatge de Representacions

Acknowledgements

This project would not have been possible without the support and guidance of the people around me.

I would like to express my sincere gratitude to Josep Vehí, for giving me the opportunity to develop this project at MiceLab and for the trust and freedom to explore a research direction that was new to the group. Special thanks also to Omer Mujahid, for his expertise and invaluable guidance throughout the work. I would also like to thank Juan Barrios for his enthusiasm and support.

Finally, to my family and partner. This would not have been possible without you.

List of Abbreviations

ADA	American Diabetes Association
AID	Automated Insulin Delivery
BERT	Bidirectional Encoder Representations from Transformers
CGM	Continuous Glucose Monitor
CR	Carbohydrate Ratio
EDA	Exploratory Data Analysis
GDL	Generative Deep Learning
GRI	Glycaemia Risk Index
HbA1c	Glycated Haemoglobin
HCL	Hybrid Closed Loop
ISF	Insulin Sensitivity Factor
LLM	Large Language Model
MCR^2	Maximal Coding Rate Reduction
MDI	Multiple Daily Injections
MTSM	Masked Time Series Modelling
NTP	Next-Token Prediction
NLP	Natural Language Processing
ODE	Ordinary Differential Equation
PCA	Principal Component Analysis
PI	Plasma Insulin
RA	Rate of Carbohydrate Absorption
SAP	Sensor-Augmented Pump
T1D	Type 1 Diabetes
T2D	Type 2 Diabetes
TIR	Time in Range
UMAP	Uniform Manifold Approximation and Projection
FT	Fine Tuning

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Motivation

Diabetes mellitus affects more than 537 million adults worldwide, a number projected to surpass 700 million by 2045 [1]. Among its forms, Type 1 Diabetes (T_1D) is distinct in both severity and management complexity: an autoimmune condition in which pancreatic beta cells are destroyed, leaving patients entirely dependent on exogenous insulin for life [2]. Its management requires continuous vigilance; a person living with T_1D has been estimated to make over 180 diabetes-related decisions every single day [3], balancing the requirements of preventing hyperglycaemia and avoiding hypoglycaemia across meals, exercise, stress, and sleep.

The appearance of Continuous Glucose Monitoring (*CGM*) has changed clinical practice substantially. Modern *CGM* sensors sample interstitial glucose every five minutes, generating longitudinal time series that record an individual's glycaemic patterns continuously. The result is a growing number of large multi-centre clinical datasets containing *CGM* records from thousands of T_1D patients, spanning months or years of monitoring.

These large clinical datasets have driven significant methodological advances, such as the application of generative deep learning (*GDL*) to develop highly realistic in silico T_1D simulators [4]. However, in predictive and analytical applications, glucose modelling still is very task-specific: algorithms are typically designed for a single clinical objective, trained from scratch on relatively small datasets, and rarely generalise across tasks. Rather than building isolated architectures for each clinical need, a more principled approach is to learn the underlying dynamics of glucose metabolism from large-scale data and then adapt this shared representation to clinical applications.

The concept of foundation models (large models pre-trained on broad data and then adapted to many downstream tasks) has transformed fields from natural language processing to computer vision [5]. The growing availability of large-scale *CGM* datasets makes this paradigm increasingly viable for glucose modelling: GluFormer [6], CGMformer [7] and CGM-LSM [8] have all shown that self-supervised pre-training on *CGM* data produces representations that transfer across patient cohorts and clinical endpoints, establishing proof of concept for the approach.

These models, however, share two key limitations. All three were developed on non-diabetic or Type 2 diabetic populations and do not incorporate the most important physiological signals relative to T_1D management, exogenous insulin delivery and carbohydrate absorption, leaving a clear gap for a T_1D-specific foundation model. The second limitation goes deeper: none of these works compares encoder and decoder architectures, which process the same data under fundamentally different assumptions

and may produce representations with different structure and clinical utility.

This thesis addresses these questions through a comparison of two complementary Transformer architectures pre-trained exclusively on adult T_1D data. A bidirectional BERT-style encoder pre-trained with masked span reconstruction, a task that corresponds directly to glucose gap imputation, and a causal GPT-style decoder trained with next-token prediction. Both are pre-trained on 934 patients from two clinical studies [9, 10], integrating *CGM* with physiological driver signals derived from pharmacokinetic modelling at 5-minute resolution. The central questions are what do these architectures, originally designed for natural language, learn from the dynamics of glucose metabolism, and whether that knowledge transfers to task-specific models. Transferability is evaluated on two clinical tasks: short-horizon glucose forecasting and overnight hypoglycaemia risk stratification. Their physiological content is further probed by recovering patient-level insulin sensitivity and carbohydrate ratio from the frozen embeddings, and each embedding space is characterised through unsupervised glycaemic phenotyping.

1.2 Objectives and Scope

The project has five main objectives:

1. **Build and characterise a large-scale T1D dataset.** Integrate and harmonise data from multiple clinical studies, conduct an Exploratory Data Analysis (EDA) to quantify *CGM* signal quality, missing data patterns, driver annotation rates, and patient heterogeneity across the datasets.

2. **Design and implement a preprocessing pipeline for population-scale T_1D data.** Harmonise raw *CGM* and dosing records from multiple clinical studies, derive continuous physiological driver signals via pharmacokinetic modelling, and handle incomplete annotations robustly, a common condition in real-world clinical datasets.

3. **Pre-train and compare two complementary Transformer architectures for T_1D glucose dynamics (Stage 1).** Design and train a bidirectional BERT-style encoder with masked span reconstruction and a causal GPT-style decoder with next-token prediction, both pre-trained exclusively on 934 adult T_1D patients integrating *CGM* with physiological driver signals. Systematically compare the two architectures to assess whether their different pre-training objectives produce representations with different structure and downstream utility.

4. **Evaluate the transferability of the pre-trained representations to downstream clinical tasks (Stage 2).** Attach lightweight task-specific heads to fine-tuned variants of both architectures and assess performance on short-horizon glucose forecasting and overnight hypoglycaemia risk stratification, comparing encoder and decoder representations against each other and against equivalent architectures trained from scratch.

5. **Characterise the structure and physiological content of the learned representations.** Test

whether the learned representations carry clinically meaningful information beyond glucose statistics, by recovering individual insulin sensitivity and carbohydrate ratio from the frozen embeddings of both architectures. Visualise and compare each embedding space to assess whether the two architectures capture individual metabolic variation differently.

This work covers a full development cycle from dataset harmonisation and preprocessing through the design and pre-training of two complementary Transformer architectures to downstream task evaluation and representation analysis. The implementation is developed in Python using TensorFlow 2.17, trained on an NVIDIA RTX 5070 GPU (12 GB VRAM). Code is available at https://github.com/marcfont14/tvae.

1.3 Institutions and Work Period

This research project has been conducted at Micelab, a biomedical control systems research group within the Institute of Informatics and Applications at the University of Girona (Parc Científic i Tecnològic, c/ Pic de Peguera 15, 17003 Girona) [11]. The work is co-supervised by Prof. Josep Vehí, director of Micelab; Dr. Omer Mujahid, senior researcher at Micelab; and Prof. Juan Barrios. The main working period for this project spans from February to June 2026.

Chapter 2

Background: General Concepts and State of the Art

2.1 Type 1 Diabetes and Glucose Homeostasis

2.1.1 Insulin, Carbohydrates, and Glycaemic Control

In a healthy human, blood glucose is kept within a narrow range through the coordinated action of insulin and glucagon. Insulin, secreted by pancreatic beta cells in response to rising glucose, promotes cellular glucose uptake and suppresses hepatic glucose output. Glucagon, secreted by alpha cells when glucose falls, stimulates the liver to release stored glycogen. In T_1D this feedback loop is lost. Without endogenous insulin, glucose from meals enters the bloodstream directly while the liver continues to produce glucose, as it no longer receives the inhibitory signal. Exogenous insulin must replace the entire axis. The mismatch between the administered insulin and the body's actual requirements is the cause of glycaemic variability in T_1D.

Three main metrics define glycaemic management in T_1D. Time in Range (*TIR*), the fraction of time spent between 70 and 180 mg dL^{-1}, is the primary measure [12]. On the other hand, sustained hyperglycaemia (glucose $>$180 mg dL^{-1}) is the main cause of microvascular complications, such as neuropathy and retinopathy. This risk is clinically monitored via glycated haemoglobin (HbA1c), which reflects the average glycaemia over the last three months [13]. Finally, hypoglycaemia (glucose $<$70 mg dL^{-1}) is the more immediate concern: risks include loss of consciousness, making it the primary safety limit for any dosing strategy. Managing all three metrics simultaneously is difficult given the high inter- and intra-patient variability in insulin response. Figure 1 illustrates these three zones on a representative 24-hour CGM trace.

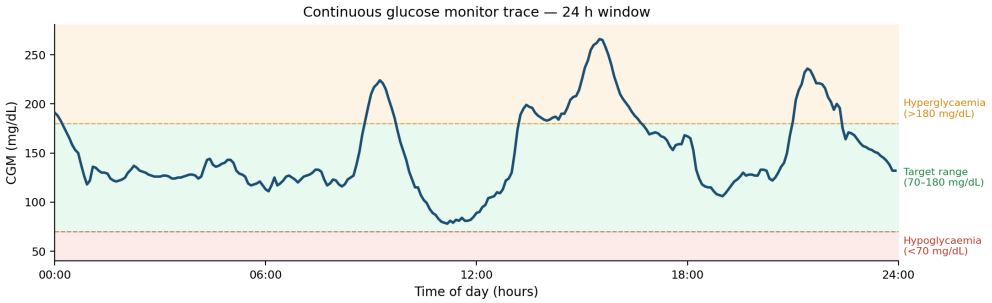

Figure 1. Glycaemic zones on a representative 24-hour CGM trace. The three bands correspond to the clinical targets defined by the international TIR consensus [12]: hypoglycaemia (glucose $<$70 mg dL^{-1}), time in range (70–180 mg dL^{-1}), and hyperglycaemia ($>$180 mg dL^{-1}).

A composite metric that captures both hypo- and hyperglycaemia simultaneously is the Glycaemia Risk Index (GRI), a weighted combination of both [14]. A higher GRI reflects worse overall glycaemic control.

In this thesis it will be used as a primary summary statistic for patient-level phenotyping.

Finally, two physiological parameters determine individual insulin dose requirements. The Insulin Sensitivity Factor (ISF), expressed in mg dL^{-1} per unit of insulin, quantifies how much one unit of rapid-acting insulin lowers blood glucose; a higher ISF indicates a patient that is more insulin-sensitive. The Carbohydrate Ratio (CR), expressed in grams of carbohydrate per unit of insulin, determines how many grams are covered by a single unit. Both parameters vary substantially between patients and cannot be measured directly from CGM alone, making their estimation from physiological time-series data a clinically valuable objective.

2.1.2 Automated Insulin Delivery and Therapy Modalities

Three insulin delivery modalities appear in the patient population studied in this thesis. In Multiple Daily Injections (MDI), the simplest approach, patients administer discrete manual bolus doses at mealtimes and a long-acting basal insulin once or twice daily. Sensor-Augmented Pump therapy (SAP) adds a continuous subcutaneous insulin pump, which delivers a programmable basal rate and boluses triggered by the patient, alongside a CGM sensor, but doesn't link the two. Automated Insulin Delivery (AID) closes that loop: a control algorithm reads the CGM signal and adjusts the pump's infusion rate continuously without patient input. Randomised trials have shown that AID increases TIR and reduces both hypoglycaemia exposure and HbA1c compared to open-loop therapy [15].

2.1.3 Continuous Glucose Monitoring

CGM devices measure glucose in the subcutaneous interstitial fluid via an electrochemical sensor inserted just below the skin [16]. Nowadays, they transmit a reading every five minutes. Because interstitial glucose equilibrates with blood glucose by diffusion, there is an inherent lag of approximately 5–15 minutes behind capillary values.

For data-driven modelling, the shift from finger-prick measurements (typically 4–8 per day) to 5-minute resolution is significant: a single 24-hour window yields 288 timesteps, capturing the full glucose profile. CGM signals still require preprocessing, as sensor dropout, calibration drift, and motion artefact introduce missing values; this is addressed in Chapters 4 and 5.

2.2 Glucose Dynamics and Circadian Rhythms

2.2.1 The Dawn Phenomenon

Glucose dynamics in T_1D are not uniform across the 24-hour cycle. Rising growth hormone and cortisol between 03:00 and 08:00 promote hepatic glucose release and reduce insulin sensitivity, producing a characteristic early-morning glucose rise known as the dawn phenomenon [17, 18]. In T_1D this cannot be compensated automatically, making nocturnal windows the most clinically sensitive part of the glucose record [19]. This circadian structure motivates the inclusion of time-of-day encoding as a

model input and the focus on overnight hypoglycaemia risk as a downstream task.

2.3 From Mechanistic to Data-Driven Glucose Models

The availability of large-scale CGM datasets shifted glucose modelling away from physiological equations and towards data-driven approaches. This section covers that progression: from classical compartmental ODE models, through supervised machine learning, to the Transformer-based architectures most relevant to this work.

2.3.1 Pharmacokinetic and Pharmacodynamic Models

Before machine learning, glucose dynamics in T_1D were studied using compartmental models based on ordinary differential equations (ODEs). The Bergman minimal model (1979) was the first to capture the glucose-insulin system quantitatively [20]: it describes glucose disappearance as a function of a remote insulin effect compartment. Simple as it is, it remains a standard reference in glucose physiology.

The Hovorka model (2004) extended this to the full subcutaneous delivery scenario relevant to T_1D management, modelling subcutaneous insulin absorption and gut carbohydrate absorption as two coupled ODE subsystems [21]. Its mathematical formulation is described in detail in Chapter 5.

These compartmental models have fundamental limitations. They are structurally rigid, require patient-specific parameter tuning, and do not improve as more data becomes available. This is where data-driven approaches have the advantage.

2.3.2 Classical Machine Learning Approaches

Early data-driven models took advantage of the growing availability of CGM data to predict glucose levels over short horizons (15-60 min). Initial attempts used ARIMA-based approaches, which treated blood sugar like a standard time series by fitting coefficients to recent history. Later, Support Vector Machines (SVMs) and gradient-boosted methods allowed for more complex risk classification. By using ”handcrafted” features (like combining meal logs, insulin history, and CGM trends) these models could learn to predict specific hypoglycemia or hyperglycemia events [22].

A good example of this is the work by Vehí et al., where different tools were combined for specific tasks: SVMs predicted post-meal drops, while neural networks handled overnight monitoring [23]. These classical approaches share a common limitation: they rely on manual feature engineering and process inputs as fixed-length vectors rather than sequences, giving the model no access to the temporal shape of the glucose curve. This motivated the move towards deep learning architectures that learn representations directly from the raw signal.

2.3.3 Deep Learning for Glucose Modelling

Recurrent neural networks and long short-term memory networks (LSTMs) were the first architectures to treat the CGM signal as a true sequence. By using a ”hidden state” to carry information from one timestep to the next, LSTMs can track long-term trends that classical models miss. This allows them to automatically weigh the impact of past glucose values, meals, and insulin doses without needing manual feature design [24]. Temporal Convolutional Networks (TCNs) emerged as a faster alternative, using ”dilated convolutions” to look far back into the patient's history without the high computational cost of recurrent layers [25]. However, both families still face a major limitation: they are built for one specific task at a time. A model trained to predict glucose 30 minutes ahead cannot be easily reused for something else, like hypoglycemia risk classification, without being completely retrained from scratch.

Generative deep learning (GDL) introduced a different capability. Instead of predicting a single scalar outcome, GDL models learn the joint distribution of glucose dynamics conditioned on insulin and carbohydrate inputs, making it possible to create plausible glucose trajectories for unseen scenarios. Mujahid et al. demonstrated this with a conditional GAN for glucose synthesis from insulin inputs [26], and later developed a full virtual patient simulator based on a sequence-to-sequence GAN with a 90-minute causal shift between driver inputs and glucose output, validated under both open-loop and closed-loop delivery scenarios [4].

Despite these advances, deep sequence models for glucose face a documented structural risk: they tend to default to autocorrelation, reproducing glucose patterns from recent history while ignoring the insulin and carbohydrate inputs that should inform the forecast [27], an effect referred to here as input neglect. Attention mechanisms give a direct way to address this, since self-attention weights each input signal by its relevance to the current prediction rather than compressing all history into a fixed hidden state. This observation motivates the move towards the Transformer-based foundation model paradigm discussed in Section 2.4.

2.3.4 Transformer-Based Glucose Models

The Transformer [28] introduced an architecture built entirely on attention. Vaswani et al. in their paper *”Attention is All You Need”* originally proposed an encoder-decoder architecture for machine translation, shown in Figure 2.

The core mechanism is **self-attention**. For each position in the sequence, three vectors are computed: a **query** (**q**), a **key** (**k**), and a **value** (**v**). The query represents what a specific position is ”looking for”, while the key represents what this position has to offer. The attention weight between two positions is the dot product of one's query with the other's key; positions whose keys best match the query get a higher weight. The final output for that position is a weighted sum of the value vectors across the full sequence. Because these weights are learned directly from the data, no relationships need to be predefined.

Multi-head attention runs this process h times in parallel, each with independently learned projections.

This allows the model to simultaneously track different patterns. For example, one head may focus on the glucose rise following a bolus while another tracks the slower overnight hormonal drift. Each encoder layer also includes a **position-wise feed-forward** sublayer, a small two-layer network applied independently at each timestep, with residual connections and layer normalisation around both.

The decoder, shown on the right of Figure 2, extends the architecture to handle sequence generation tasks. In this thesis, both components are used as independent pre-training models: the encoder produces a global contextualised representation via bidirectional attention, while the decoder produces a causally ordered sequential representation via next-token prediction, with each serving as a backbone for downstream task heads.

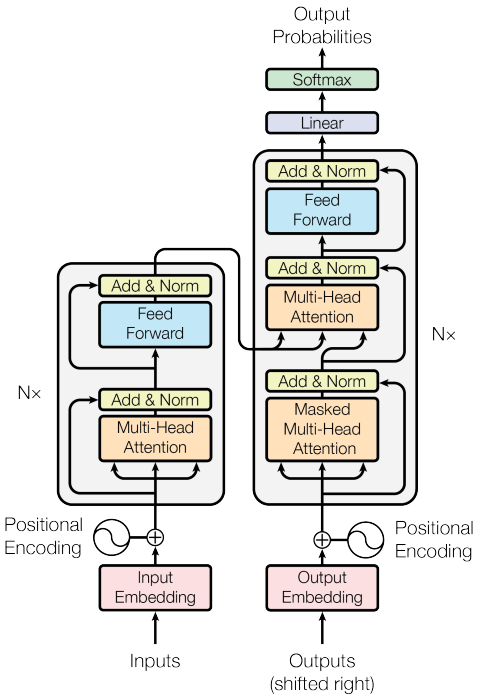

Figure 2. The Transformer architecture proposed by Vaswani et al. [28]. The encoder (left) stacks multi-head self-attention and position-wise feed-forward sublayers with residual connections and layer normalisation. The decoder (right) adds a cross-attention sublayer attending to the encoder output. This thesis adapts both components independently as separate pre-training models: the encoder for bidirectional masked reconstruction, and the decoder for causal next-token prediction.

What made the design unique was the combination of full context access and parallelism. Every position attends to every other in a single matrix operation, which eliminates the sequential bottleneck and the risk of gradients vanishing across long chains of timesteps. For a 288-step CGM window, for example, this means the encoder can process the glucose signal, insulin pharmacokinetics, carbohydrate absorption, and the time-of-day encoding in parallel, learning which combinations matter from data alone, without any of those relationships being specified by hand.

2.4 Foundation Models in Healthcare

2.4.1 Definition and Key Properties

The term *"Foundation Model"* was introduced by Bommasani et al. [5] to describe models pre-trained on broad, unlabelled data at scale and then adapted to many downstream tasks. The key idea is that pre-training happens once and adaptation happens many times: instead of a separate model for each clinical question, a shared encoder or decoder is trained and then reused or fine-tuned for specific applications [29].

CGM data fits this well. Large longitudinal datasets exist with no task-specific annotation, and the same 24-hour glucose window with its insulin and meal context can be the input for multiple clinically relevant tasks: glucose forecasting, hypoglycaemia risk estimation, missing data imputation, and insulin sensitivity profiling.

2.4.2 Existing Work on Biosignal Foundation Models

The foundation model paradigm began in natural language processing (NLP). BERT [30] proved that a bidirectional encoder could learn to reconstruct masked tokens from their surrounding context, producing representations that transferred across multiple tasks without needing labels. MOMENT [31] showed the same principle holds for time series: a single encoder pre-trained on a heterogeneous dataset transfers to forecasting, classification, and anomaly detection.

Three recent works have brought this paradigm to CGM data: GluFormer [6], a causal decoder pre-trained on non-diabetic adults; CGMformer [7], a bidirectional encoder pre-trained on T_2D and prediabetic Chinese cohorts; and CGM-LSM [8], a decoder-only model pre-trained on mixed T_1D/T_2D data. Their architectures, training populations, and downstream applications are compared in detail in Chapter 3.

2.4.3 Masked Reconstruction vs Autoregressive Pre-training

Two dominant self-supervised pre-training objectives have emerged for sequence models, corresponding to the two architectural families evaluated in this thesis.

Masked reconstruction trains a bidirectional encoder by randomly masking a subset of input tokens and asking the model to reconstruct them from their surrounding context. Because every position can attend to every other, the model must integrate global information across the full sequence, it cannot rely on local interpolation. Devlin et al. [30] introduced this objective for NLP, producing representations that transferred across diverse downstream tasks without further pre-training. Goswami et al. [31] and Dong et al. [32] showed the same principle applies to time series: encoders pre-trained with masked reconstruction on heterogeneous datasets transfer to forecasting, anomaly detection, and classification. CGMformer [7] adapted this objective specifically to CGM data.

Autoregressive next-token prediction trains a causal decoder to predict the next token from all pre-

ceding ones. Attention is restricted to past positions, so the model learns a sequential model of how the signal unfolds over time rather than a global summary. Radford et al. [33] demonstrated that this objective produces highly transferable representations at scale. In time series, TimesFM [34] and Chronos [35] showed that a decoder-only model pre-trained with next-token prediction can generalise across forecasting tasks from diverse domains. GluFormer [6] and CGM-LSM [8] applied the same objective to CGM data.

These two objectives encode fundamentally different inductive biases: masked reconstruction encourages a rich global context representation that transfers to tasks where the full input window is observed, while autoregressive prediction learns the sequential structure of how the signal unfolds and transfers naturally to generation and forecasting tasks.

2.5 Representation Learning and Low-Dimensional Structure

2.5.1 The Low-Dimensional Structure Hypothesis

A central premise of modern ML is that high-dimensional data, despite its apparent complexity, tends to concentrate near low-dimensional structures embedded in the ambient space [29, 36]. A 24-hour *CGM* window discretised at 5-minute resolution, combined with physiological driver signals, constitutes a high-dimensional object; yet the population of such windows does not fill that space uniformly. The physiological constraints of glucose-insulin dynamics, circadian patterns, and inter-patient metabolic variation confine real patient data to a manifold of much lower dimensionality. Representation learning is the task of finding a compact mapping onto this manifold, one that discards noise and redundancy while preserving the variation that matters.

The key test of such a mapping is whether it encodes information that was never explicitly provided. If a model has genuinely learned the geometry of the glucose manifold rather than memorising surface statistics, its frozen embeddings should recover patient-level physiological properties such as insulin sensitivity or carbohydrate ratio, even though no such labels were available during pre-training. This is the central hypothesis of the representation analysis in this thesis, and it motivates evaluating the embeddings of both architectures through physiological probing.

2.5.2 Principles of Good Representations: Maximal Coding Rate Reduction

Wright and Ma [36] formalise what it means for a representation to be good through the Maximal Coding Rate Reduction (MCR^2) framework, later extended by Buchanan et al. [29] to the setting of deep representation learning. The core idea comes from data compression. Given a set of learned embeddings, the "coding rate" R_ε measures how many bits are needed, on average, to record the location of each embedding up to a precision ε. If the embeddings are spread over a large volume, many bits are required and R_ε is high; if they are tightly clustered, few bits are needed and R_ε is low.

A good representation must satisfy two properties at once. First, embeddings from the *same* group of

data should cluster tightly, so the within-group coding rate is low. Second, embeddings from *different* groups should spread apart, occupying directions in the space that are as orthogonal as possible, so the overall coding rate is high. MCR^2 captures both requirements in a single objective, the coding rate reduction:

$$\Delta R_\varepsilon(\mathbf{Z}) \;=\; R_\varepsilon(\mathbf{Z}) \;-\; R_\varepsilon^c(\mathbf{Z}) \tag{2.1}$$

where the first term measures the spread of all embeddings treated jointly and the second term is the average coding rate within each group. Maximising ΔR_ε simultaneously pushes different groups apart and compresses each group into a compact subspace.

This framework provides a concrete prediction for the glucose representations learned in this thesis. If the encoder has genuinely captured the structure of T_1D glucose dynamics, patients with similar metabolic profiles should form tight clusters in the embedding space, while phenotypically distinct groups, such as patients at opposite ends of the glycaemic risk spectrum or on different therapy modalities, should occupy well-separated regions. This is precisely what the UMAP visualisation, GRI stratification, and within-versus-between group variance experiments of this thesis are designed to test. A complementary physiological probing analysis then asks whether the same frozen embeddings can recover patient-level insulin parameters, specifically ISF and CR, that were never observed during pre-training.

The connection to the Transformer architecture is direct: Buchanan et al. [29] show that multi-head self-attention can be interpreted as performing a gradient step on the coding rate objective at each layer. This thesis evaluates whether the architectural difference between bidirectional and causal attention produces representations with measurably different geometric structure and clinical utility.

2.6 The Driver-Blindness Problem

One of the main concerns of real-world T_1D data is that it is rarely fully annotated. In MDI therapy, for example, bolus doses and meals are recorded manually; so omissions are frequent, particularly for snacks, corrective boluses, and unplanned meals. While AID systems log micro-boluses automatically, carbohydrate intake is still the patient's responsibility, and often not recorded. Therefore, a large fraction of windows in any T_1D dataset will contain zero values in the bolus and carbohydrate columns.

Missing driver annotations create a systematic problem: a model that treats an absent bolus record as confirmation that nothing was injected will see glucose rising without an apparent cause and attribute the rise to some other, often incorrect, variable. This is the driver blindness problem.

Chapter 3

Market Analysis

3.1 Historical Evolution and Current Players

Glucose monitoring has evolved substantially over the past five decades, from portable blood glucose meters in the 1970s to CGM devices in the late 1990s that gradually reduced calibration requirements until the sensor became a passive five-minute wearable. In 2016 the first hybrid automated insulin delivery (AID) system was approved, closing the control loop between the sensor and the pump and establishing the current standard of care for T1D in most high-income countries [37].

The CGM hardware market is large and continues to grow rapidly. Recent industry analysis values the global CGM market at approximately USD 13.38 billion in 2025, with a projected compound annual growth rate (CAGR) of 15.1% through 2033, as detailed in Figure 3 [38]. Three major manufacturers dominate the sensor segment: Abbott (FreeStyle Libre), Dexcom (G6/G7), and Medtronic (Guardian).

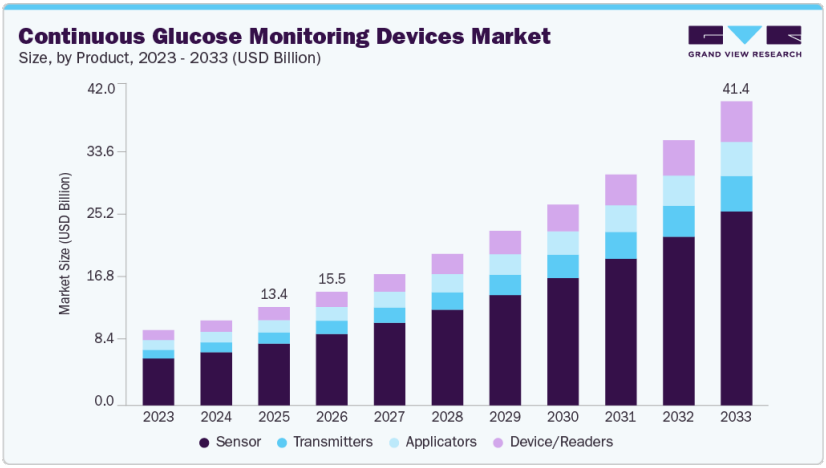

Figure 3. Continuous glucose monitoring market size forecast by product, 2023–2033 (USD Billion). The market demonstrates substantial growth, with a projected value of $41.4 billion by 2033, driven by the expanding sensor segment (dark purple). Data adapted from [38].

On the academic side, the group most directly relevant to this thesis is MiceLab at Universitat de Girona [11], which has worked in T1D glucose modelling spanning physiological simulation, hypoglycaemia prediction, and digital twin development [4, 23, 26].

3.2 Existing Solutions and Added Value

Three foundation models have brought large-scale pre-training to CGM data. Each targets a different population and task scope.

GluFormer [6], from the Weizmann Institute of Science, is an autoregressive Transformer decoder pre-trained with next-token prediction on more than ten million CGM measurements from 10,812 primarily non-diabetic adults. It generalises to 19 external cohorts and demonstrates utility for long-term risk stratification, including prediction of HbA1c and 12-year cardiovascular mortality from frozen representations.

CGMformer [7], from a consortium of Chinese academic hospitals, is a bidirectional encoder pre-trained with masked reconstruction on a combined dataset of roughly 60,000 participants spanning normoglycaemic, prediabetic, and T_2D populations. A downstream supervised module combines frozen encoder embeddings with static clinical features for T_2D and complication screening. In its unsupervised mode it identifies six metabolic subtypes. The encoder is trained exclusively on T_2D and prediabetic data, so its representations reflect insulin-resistant metabolism rather than the insulin-deficient dynamics of T_1D.

CGM-LSM [8], from Johns Hopkins University and Welldoc, is a decoder-only Transformer pre-trained with next-token prediction on a dataset of 15.96 million CGM readings from 592 patients (291 T1D, 301 T2D). It achieves strong forecasting on the OhioT1DM benchmark (RMSE 15.90 mg dL^{-1} at one hour), but is restricted to a single task and uses CGM as its only input. The authors report accuracy degrades during mealtimes and at glucose extremes [8].

These models establish that large-scale pre-training on CGM sequences can give meaningful representations and that multi-task transfer is achievable. None, however, is pre-trained exclusively on T_1D. Furthermore, while GluFormer incorporates discrete meal tokens and CGMformer combines its embeddings with static biomarkers in a separate supervised module, no existing model feeds continuous pharmacodynamic drivers directly into the pre-trained model. The main architectural differences are presented in Table 1.

Table 1. Comparison of existing CGM foundation models and the model proposed in this thesis.

Dimension	GluFormer	CGMformer	CGM-LSM	This work
Pre-training objective	Next-token prediction	Masked reconstruction	Next-token prediction	MTSM (encoder) + NTP (decoder)
Attention type	Causal (unidirectional)	Bidirectional	Causal (unidirectional)	Bidirectional + Causal
Training population	Non-diabetic adults	NGT / prediabetes / T2D	T1D + T2D (mixed)	**T1D adults (exclusive)**
Input modalities	CGM + diet tokens	CGM only (encoder)	CGM only	CGM + insulin and carbohydrate events
Physiological drivers	Dietary tokens (discrete)	None (clinical in CGMformer_C)	None	Hovorka-derived PI and RA (continuous)
Patient representation	Max-pool (1,024-d)	Mean-pool	Autoregressive generation	$\mathbf{h}_{cls}$ (encoder) / $\mathbf{h}_{last}$ (decoder)
Multi-task scope	General metabolic health	T2D / prediabetes	Glucose forecasting only	**T1D clinical management**
Data availability	Proprietary	Proprietary	Proprietary	**Public (METABONET + T1DEXI)**

Table 1 highlights three structural gaps shared by all existing models: none is pre-trained exclusively on T_1D, none incorporates continuous pharmacodynamic driver signals alongside CGM, and none systematically compares bidirectional and causal architectures on the same population and downstream tasks. The present work addresses all three.

3.3 Target Sectors and Future Prospects

Existing tools leave a clear gap in T1D-specific, multi-task modelling. The model proposed in this thesis addresses this across three sectors: clinical decision support, medical device development, and biomedical research.

Clinical decision support

Clinicians and diabetes care teams stand to benefit most directly. Hypoglycaemia risk estimation is the natural first application; the ADA standards of care increasingly encourage individualised, data-driven management [39], and a model that produces patient-specific risk and insulin sensitivity estimates from routine CGM data fits naturally within this workflow.

Medical device and digital health

Automated insulin delivery is the fastest-growing commercial sector, yet manufacturers still rely on classical control algorithms for the glucose forecasting component. A model pre-trained on the full range of T1D therapy modalities and meal patterns offers a richer foundation than task-specific approaches trained on small labelled datasets.

Clinical research and pharmaceutical development

CGM has become a standard endpoint in T1D clinical trials [12, 13], but current summary metrics capture little of the underlying signal. Studies of new insulin formulations or beta-cell therapies could benefit from a model trained on the full dynamical profile of glucose response, applied as a digital biomarker extraction layer.

Future prospects

T1D prevalence is rising globally [1] and CGM adoption continues to expand, which means more training data and a larger population that could benefit. The broader shift toward foundation models in medicine is creating an expectation that pre-trained representations will be reused [5]; in clinical time-series data this is only beginning, and as T1D-specific encoders become available, each new downstream application reduces to designing a task head and running a targeted validation study.

Chapter 4

Conception Engineering

This chapter defines the design of the experimental framework: training data, feature set, architecture, and evaluation strategy. For each component, alternatives are compared and the rationale for the final choice is made explicit. The mathematical specification of the selected designs is explained in Chapter 5.

4.1 The Training Dataset

We train on two complementary T1D datasets: METABONET and T1DEXI.

4.1.1 METABONET

METABONET [9] is a collection of T1D management studies combined and standardised to address the fragmentation of existing CGM datasets. Released in 2026, it consolidates 21 individual studies into a standardised format, covering 3,135 subjects and 1,228 patient-years of overlapping CGM and insulin data. The collection spans an age range of 1 to 82 years and includes all three main therapy modalities: automated insulin delivery (AID), sensor-augmented pump (SAP), and multiple daily injections (MDI).

Table 2. Studies included in the METABONET public release. Duration is the average per patient.

Study	Primary modality	Patients	Avg. duration per patient
AZT1D	AID / MDI	23	0.9 months
BrisT1D	AID / MDI	19	3.7 months
CTR3	AID	30	1.7 months
DCLP3	AID	112	5.2 months
DCLP5	AID	100	5.5 months
Flair	AID	113	4.1 months
HUPA-UCM	AID / MDI	22	1.0 months
IOBP2	AID	332	2 months
Loop	AID	845	7.4 months
PEDAP	AID	65	4.3 months
ReplaceBG	SAP / MDI	208	7.1 months
Shanghai T1DM	MDI	12	0.1 months
T1D-UOM	AID / MDI	14	0.1 months
Total		**1,895**	3.32 months

Note: Counts are per sub-study. 241 participants contributed to more than one study, so the total of 1,895 records corresponds to 1,291 unique patients.

4.1.2 T1DEXI

The Type 1 Diabetes and Exercise Initiative (T1DEXI) [10] is one of the largest observational studies conducted in adults with T1D. It enrolled 497 participants aged 18-69 years (mean 37 ± 14 years) across multiple sites and randomised them to six structured exercise sessions over four weeks, covering aerobic, interval, and resistance exercise modalities. Glucose was monitored continuously using the Dexcom G6. T1DEXI is included as a large T1D-exclusive reference dataset; its exercise annotations are not used as model inputs because no equivalent recording exists across the METABONET cohort, as discussed in Section 4.2.4.

4.2 Feature Selection

4.2.1 Primary Signal: CGM

The central target signal is glucose: most of the clinical tasks the model must support are predictions or summaries of past and future glucose behaviour. CGM is therefore the feature around which the representation is built.

4.2.2 Insulin and Carbohydrate Drivers

Insulin and carbohydrate records in the raw datasets are discrete events: a bolus entry at a single timestep, a meal at another. These impulses are an incomplete picture of what is actually happening physiologically. A bolus given 90 minutes ago is still driving down glucose; a meal ingested at $t = 0$ will continue to raise blood glucose for the next hour or more. An encoder that only sees the raw event flags cannot observe these ongoing effects.

To address this, both signals are converted to continuous pharmacokinetic curves using the Hovorka model [21], described in full in Section 5.1.5.

The continuous PI (Plasma Insulin Approximation) and RA (Rate of Exogenous Glucose Appearance) curves are the primary driver signals. Ablation experiments on imputation quantify their individual contributions: removing PI alone drops R^2 by 0.10 at a 4-hour gap, growing to 0.23 at 8 hours; removing RA costs 0.05 at 4 hours and 0.09 at 8 hours. PI matters more for two reasons: its pharmacological effect lasts several hours after a bolus, while carbohydrate absorption peaks and fades more quickly; and insulin doses are logged automatically by pumps, while meal records are entered manually and frequently omitted.

We also retain the raw event spikes as binary flags alongside the driver curves. The curves tell the model how much insulin is active or how fast glucose is rising at each moment. The flags add something the curves cannot: the exact moment the injection or meal was given, which helps the model anticipate an upcoming peak response. Averaged across all test windows the cost of removing the flags is small ($\Delta R^2 = -0.02$ at 4 hours), but the average is misleading. When an event falls inside the masked gap, removing the flags costs $\Delta R^2 = -0.04$ to -0.05; when no event falls in the gap, the cost is

negligible ($\Delta R^2 = -0.002$), as shown in Table 3. The average looks small simply because most windows contain no event in the masked region.

Table 3. Imputation R^2 with and without binary flags, stratified by whether a bolus or carbohydrate event occurs inside the masked gap.

	4h gap		6h gap	
Window stratum	**Full**	**No flags**	**Full**	**No flags**
Any event in gap	0.779	0.738 ($\Delta = -0.041$)	0.707	0.659 ($\Delta = -0.048$)
No event in gap	0.780	0.778 ($\Delta = -0.002$)	0.685	0.682 ($\Delta = -0.003$)

4.2.3 Temporal Features and Positional Encoding

Glucose dynamics follow a 24-hour pattern. Insulin sensitivity varies across the day [18], meals tend to cluster at predictable hours, and the dawn phenomenon (a glucose rise in the early morning) is a known feature of T1D [17]. The model needs to know what time of day it is processing.

The raw hour of day (0–23) is a poor input because of the midnight discontinuity: 23:55 and 00:05 are adjacent in time but 23 units apart as numbers. Cyclic encoding resolves this by converting the hour to a sine and cosine pair, so midnight is continuous and the model sees the full 24-hour periodicity without any artificial break:

$$\text{hour\_sin} = \sin\left(\frac{2\pi \cdot \text{hour}}{24}\right), \qquad \text{hour\_cos} = \cos\left(\frac{2\pi \cdot \text{hour}}{24}\right). \tag{4.1}$$

The global contribution to imputation is modest ($\Delta R^2 \approx -0.008$ without time features), but the effect concentrates where it matters. Dawn windows (04:00–08:00) and nocturnal windows (20:00–04:00) show a R^2 drop of roughly 0.016 and 0.015 at a 6-hour gap, about eight times the cost for afternoon windows ($\Delta R^2 \approx -0.002$). During these periods, the time of day carries information about glucose trajectory that the driver signals alone cannot provide. A full analysis is provided in Appendix A.

4.2.4 Features Considered and Excluded

Two additional feature groups were considered during design and excluded after evaluation.

Therapy modality. AID, SAP, and MDI are different in how insulin is delivered. This information is available as patient metadata and could in principle help the model interpret the shape of the PI curve. Ablation shows a small average imputation drop ($\Delta R^2 = -0.04$ at 4 hours). More decisively, a linear probe was tested on the frozen encoder representations and achieved only 0.33 accuracy (so, random guessing). If the encoder has not learned to separate modalities from the PI and RA patterns it already observes, an explicit one-hot label is unlikely to add anything. The feature was excluded.

Exercise. T1DEXI records exercise events, and physical activity has meaningful glucose effects. The

problem is that the METABONET cohort doesn't include this exercise information, which accounts for most patients. Including it as a feature would require imputing missing records as zero. We excluded it.

A detailed feature importance analysis (gradient saliency, zero-out ablation across all gap lengths, and principal component correlations with per-patient physiological summaries) is provided in Appendix A.

4.3 Feature Normalisation

Three continuous input channels require normalisation: CGM, plasma insulin PI (column 1), and carbohydrate absorption rate RA.

CGM

The choice for CGM is between per-patient z-scoring and global z-scoring. Per-patient normalisation collapses all patients to a common zero mean, making a hyperglycaemic patient's windows statistically indistinguishable from those of a well-controlled patient. Global z-scoring (computed once across the 934-patient pre-training cohort) preserves absolute glucose level as a learnable feature. Windows from a patient with chronically high glucose retain distinctly higher z-score values, allowing the model to condition its predictions on the patient's overall glycaemic state. Global parameters: $\mu = 144.40$ mg dL^{-1}, $\sigma = 57.11$ mg dL^{-1}.

PI and RA

To evaluate which normalisation strategy is better, we check whether the resulting representations preserve physiological information. ISF and CR (defined in Chapter 2) are patient-specific parameters that the model never sees during pre-training. If a simple linear probe on frozen representations can recover them, the model has captured genuine patient physiology. The evaluation methodology is described in Chapter 5.

The normalisation choice for the driver signals required careful consideration. The initial hypothesis was that therapy modality heterogeneity would make per-patient scaling necessary: AID systems deliver continuous micro-doses with low PI amplitude, while MDI patients receive discrete large boluses that produce much higher PI peaks. The worry was that a population-level normalisation would mistake this difference in dose magnitude for a difference in patient physiology, when in fact it reflects the delivery method.

Individual z-scoring of PI and RA was therefore the first approach, used in the initial pre-training runs. It eliminates inter-patient dose magnitude differences, keeping each patient's driver signals on a consistent internal scale regardless of therapy type.

This hypothesis was wrong. Switching to global z-scoring (population-level mean and standard deviation applied uniformly to all patients) produced a decisive improvement in physiological recovery

(Table 4). The reason is that the inter-patient amplitude differences encode insulin sensitivity. A patient who requires large PI doses to achieve the same glucose response has low insulin sensitivity: that is precisely the ISF signal. Per-patient normalisation erases exactly this information. The dose magnitude is not noise to be normalised away: it is exactly the patient-level signal the model needs.

Table 4. Effect of normalisation strategy for PI and RA on downstream physiological recovery. Scores are R^2 from Ridge regression probes on frozen encoder ($\mathbf{h}_{\text{cls}}$, for CR) and decoder ($\mathbf{h}_{\text{last}}$, for ISF) representations.

Normalisation	ISF R^2 (decoder)	CR R^2 (encoder)
Per-patient z-score	0.354	0.012
Global z-score	**0.499**	**0.406**

Global parameters: PI: $\mu = -3.06$, $\sigma = 4.22$; RA: $\mu = 0.63$, $\sigma = 2.48$.

The complete 7-feature input vector is given in Table 5.

Table 5. Input feature tensor at each timestep. CGM, PI, and RA are globally z-scored using population-level statistics. All other features are bounded and require no normalisation.

Index	Feature	Source	Encoding
0	CGM	CGM device (5-min)	global z-score
1	PI	Hovorka 3-compartment ODE	global z-score
2	RA	Hovorka 2-compartment ODE	global z-score
3	hour_sin	Timestamp	$\sin(2\pi \cdot \text{hr}/24)$
4	hour_cos	Timestamp	$\cos(2\pi \cdot \text{hr}/24)$
5	bolus_logged	Dosing record	Binary
6	carbs_logged	Meal record	Binary

4.4 Two-Stage Architecture Design

The design follows the foundation model paradigm introduced in Chapter 2: a backbone pre-trained on unlabelled physiological data in Stage 1, then reused with lightweight task-specific heads in Stage 2.

4.4.1 Stage 1: Pre-training

Stage 1 trains two Transformer models on the 934-patient pre-training cohort, using only the unlabelled physiological signal: a bidirectional encoder pre-trained with masked reconstruction, and a causal decoder pre-trained with next-token prediction. The architectural differences between these two variants and the rationale for training both are discussed in Section 4.5. Both models produce two types of output from each 24-hour window. The first is a full sequence: 288 vectors, one per timestep, used by tasks that act at each point in time. The second is a single summary vector that compresses the entire

window: $\mathbf{h}_{cls}$ for the encoder and $\mathbf{h}_{last}$ for the decoder. Classification and profiling tasks use this single vector. Their construction is described in detail in Appendix B.

4.4.2 Stage 2: Downstream Task Heads

Stage 2 attaches a lightweight head to each pre-trained model for each clinical task. The heads are evaluated in two modes.

In the first mode (*frozen backbone*), the pre-trained model is kept exactly as it was after Stage 1; its weights do not change, and the head must work with whatever representation the backbone already produces. This tests what the model learned during pre-training alone.

In the second mode (*fine-tuned backbone*), the pre-trained model is allowed to adapt its weights alongside the task head. The result is compared against an LSTM baseline trained entirely from scratch, without any pre-training. This tests whether pre-training knowledge is transferable to tasks it was not explicitly trained on.

Four downstream applications are included in this thesis, selected for their clinical relevance to T1D management.

Gap imputation (frozen). CGM sensors produce gaps from warm-ups, compression artefacts during sleep, and calibration failures. Standard linear interpolation fills these gaps mechanically, ignoring the physiological context. The frozen encoder reconstructs missing segments using the driver activity and surrounding glucose context within the window.

Short-horizon glucose forecasting (fine-tuned vs from-scratch). Given the current 24-hour window, the head predicts glucose over the following 2 hours.

Overnight hypoglycaemia risk (fine-tuned vs from-scratch). Nocturnal hypoglycaemia is a significant safety concern in T1D [19], and the risk is difficult to assess at bedtime from the current glucose level alone. The model receives a window ending between 21:00 and 23:59 and estimates the probability that glucose will drop below 70 mg dL^{-1} within the next 8 hours.

ISF and CR profiling (frozen). ISF and CR govern correction doses and meal boluses but are currently estimated by clinical experience and patient trial and error. Linear probes on the frozen representations test whether the pre-trained model has encoded enough patient physiology to estimate parameters it was never trained to predict.

In addition to these four tasks, a *representation analysis* characterises what each model has learned. Patient-level embeddings are extracted from the frozen models and examined for clinically meaningful structure: do patients with similar glycaemic profiles cluster together? Can clinical targets such as GRI, ISF, and CR be recovered from the embeddings? This is not a prediction task but an interpretability study. Full methodology and results are reported in Chapters 5 and 6.

Two further applications came up during design but did not make the final scope.

Personalised digital twin. A generative application that samples counterfactual glucose trajectories under hypothetical insulin regimes, using per-patient fine-tuning. Integrating a generative decoder with the pre-trained backbone is a task too complex to complete within the thesis timeline. We treat it as a natural next step after Stage 2.

Next-day TIR prediction and anomaly detection. Next-day TIR is already available retrospectively and does not directly inform management decisions. Artefact detection addresses a data quality problem rather than a clinical question. Neither fits the scope of this thesis.

4.5 Model Selection

The Transformer architecture [28] is the basis of Stage 1. Its defining property is self-attention: every position in the input sequence can directly access every other position, regardless of how far apart they are in time. In a 24-hour glucose window, a reading can attend directly to a bolus or meal event from hours earlier, all in a single operation. Recurrent networks (LSTMs) can only reach distant positions by passing information through every intermediate step, making long-range relationships harder to learn. This direct access is particularly relevant for glucose dynamics, where a glucose value at any moment depends on insulin and meal events that may have occurred hours earlier.

Within the Transformer family, three structural variants exist, and the choice determines how the model processes the input and what pre-training objective it can use.

4.5.1 Transformer Architecture Variants

Encoder-only (bidirectional). An encoder reads the full window at once: every position can attend to every other position, both forward and backward in time. This means each output vector is informed by the entire 24-hour context. The design follows BERT [30] and has been adopted for time series by MOMENT [31] and CGMformer [7].

Decoder-only (causal). A decoder reads a 24-hour glucose window strictly left to right: each position can only attend to positions that come before it. The output at the last position has seen the entire window, but every earlier position has only a partial view. This is the design used by GluFormer [6] and GPT [33]. Its natural strength is sequential prediction, where the model generates one value at a time conditioned on everything it has seen so far.

Full encoder-decoder. The original Transformer [28] pairs a bidirectional encoder with a causal decoder, connected by cross-attention. This is designed for sequence-to-sequence tasks such as translation. For this thesis, full configuration is unnecessary: the goal is to learn a representation of glucose dynamics, not to translate between two different sequences. A full encoder-decoder would add parameters that are discarded after pre-training.

Both the encoder-only and decoder-only variants are retained. The encoder provides full bidirectional context, which is well suited to tasks like imputation where both past and future readings inform the

reconstruction. The decoder provides a causally ordered summary that may be better suited to forecasting and generation. Training both on the same data and evaluating them on the same downstream tasks is the only way to understand which captures T1D physiology better.

4.5.2 Pre-training Objectives

Each architecture variant requires a different pre-training strategy, determined by how the model processes the input.

Encoder: masked time series modelling

Because the encoder sees the full window at once, it cannot be trained to predict the next value (it already sees it). Instead, the training task is reconstruction: a contiguous span of CGM readings is masked, and the model must predict the missing values from the surrounding context [30, 31]. This is the time series analogue of BERT's masked language modelling, referred to here as Masked Time Series Modelling (MTSM).

The mask spans 5 to 8 hours of the 24-hour window. This length is chosen deliberately: a short gap (one or two hours) can be filled by simple interpolation of the CGM values on either side, without needing to understand the underlying physiology. A gap of 5 hours or more cannot be bridged this way. The glucose curve over that period depends on how much insulin was active, when meals occurred, and how the patient's metabolism responded. To fill a long gap accurately, the encoder must extract this information from the driver signals. This forces the model to learn the relationships between insulin, carbohydrates, and glucose, rather than taking a shortcut through local interpolation.

Two alternative encoder pre-training strategies were considered and rejected. *Contrastive learning* [40] trains the model by comparing augmented versions of the same window, but defining valid augmentations for physiological time series is difficult: any modification to CGM or driver signals risks altering their clinical meaning. *Latent-space prediction* [41], where the model predicts the representation of masked regions rather than the actual values, was tested during development but collapsed to near-constant outputs regardless of the input.

Decoder: next-token prediction

Because the decoder reads left to right, its natural training task is next-token prediction (NTP): at each timestep, the model predicts the next glucose reading from all preceding observations and the driver sequence. The glucose signal is discretised into bins and the model outputs a probability distribution over them; the bin resolution is chosen to stay well below the measurement noise of clinical CGM devices, so no clinically meaningful distinction is lost. The exact discretisation is specified in Chapter 5.

A variant using delta targets (predicting the change in glucose rather than the absolute value) was

tested but showed no improvement on downstream tasks; absolute-target prediction was retained.

The full mathematical specification of both architectures and their training objectives is given in Chapter 5.

4.6 Evaluation Strategy

The four downstream tasks described in Section 4.4.2 are organised into two evaluation tracks, each answering a different question.

Track 1 (frozen backbone): does pre-training on unlabelled T1D data cause the model to learn glucose dynamics? The three frozen-backbone tasks (gap imputation, ISF/CR profiling, representation analysis) answer this by measuring what the pre-training has encoded without any task-specific adaptation.

Track 2 (fine-tuned backbone): does pre-training knowledge transfer to clinical prediction tasks? The two fine-tuned tasks (forecasting, hypoglycaemia risk) answer this by comparing the pre-trained backbone against an LSTM trained from scratch on the same task-labelled data.

The encoder–decoder comparison runs through both tracks: which paradigm encodes more physiological structure in its frozen representations (Track 1), and which provides a better starting point for clinical prediction (Track 2).

4.7 Summary of Design Decisions

Table 6 collects the key design decisions made in this chapter, the alternatives considered, and the rationale for each choice.

Table 6. Key design decisions made during the conception of the experimental framework.

Decision	Options	Choice	Rationale
Pre-training paradigm	Encoder only, Decoder only, Full encoder-decoder	Encoder-only and Decoder-only	No prior establishes which context direction transfers best on T1D signals; both are retained and compared downstream.
CGM normalisation	Per-patient or Global z-score	Global z-score	Preserves absolute glucose level as a feature, keeping clinically meaningful differences.
PI/RA normalisation	Per-patient or Global z-score	Global z-score	Absolute dose magnitude gives valuable physiological information.
Encoder pre-training objective	Contrastive learning, latent-space prediction or MTSM	MTSM	Contrastive requires valid augmentations, latent-space prediction collapsed.
Decoder pre-training objective	Delta targets with auxiliary losses, NTP	Absolute-target NTP (CE)	Delta variant showed no improvement.
Therapy modality	Include one-hot (AID/S-AP/MDI); exclude	Excluded	Ablation shows marginal benefit ($\Delta R^2 \leq 0.04$ at 4h); representation probe returns random-guess accuracy.
Exercise data	Include or exclude from T1DEXI	Excluded	Absent from the majority of METABONET patients; imputing zero would confound absence of data with absence of activity.

Chapter 5

Detail Engineering

With the design concept explained in Chapter 4, this chapter details the implementation: data preprocessing, model architecture, training configuration, and evaluation.

5.1 Data Preprocessing Pipeline

We begin with preprocessing: merging and cleaning the two source datasets, extracting continuous physiological features from discrete event records, normalising the signals, and packaging everything into the windowed format the models train on.

5.1.1 Dataset Acquisition and Harmonisation

The two source datasets described in Section 4.1 share the same clinical focus but use different data formats. Our first step is to bring them into a common structure.

METABONET was downloaded from its public repository in its already standardised form. T1DEXI was obtained through MiceLab and harmonised to METABONET's format using the `metabonet_processor` pipeline [42]. After merging, the combined dataset contains approximately 126 million rows at 5' frequency.

5.1.2 Quality Filtering

We process the two datasets separately before merging. Firstly, we keep only patients with at least ten logged bolus and carbohydrate events each for METABONET, as some of the studies included didn't contain these variables, leaving 914 unique METABONET patients. T1DEXI adds a further 497 adults. The merged cohort therefore contains 1,411 patients.

We then apply three exclusion criteria to the merged cohort. We exclude patients under 18, since paediatric T1D physiology cannot be directly extrapolated to adults [43–45]. We then exclude patients whose *CGM* signal has a standard deviation below $15\ mg\,dL^{-1}$, to avoid having flat curves due to sensor errors. We also discard patients with more than 50% null *CGM* values, considering such records would only negatively affect the training. The full inclusion pipeline is shown in Figure 4, with 1,037 adult patients in the final cohort.

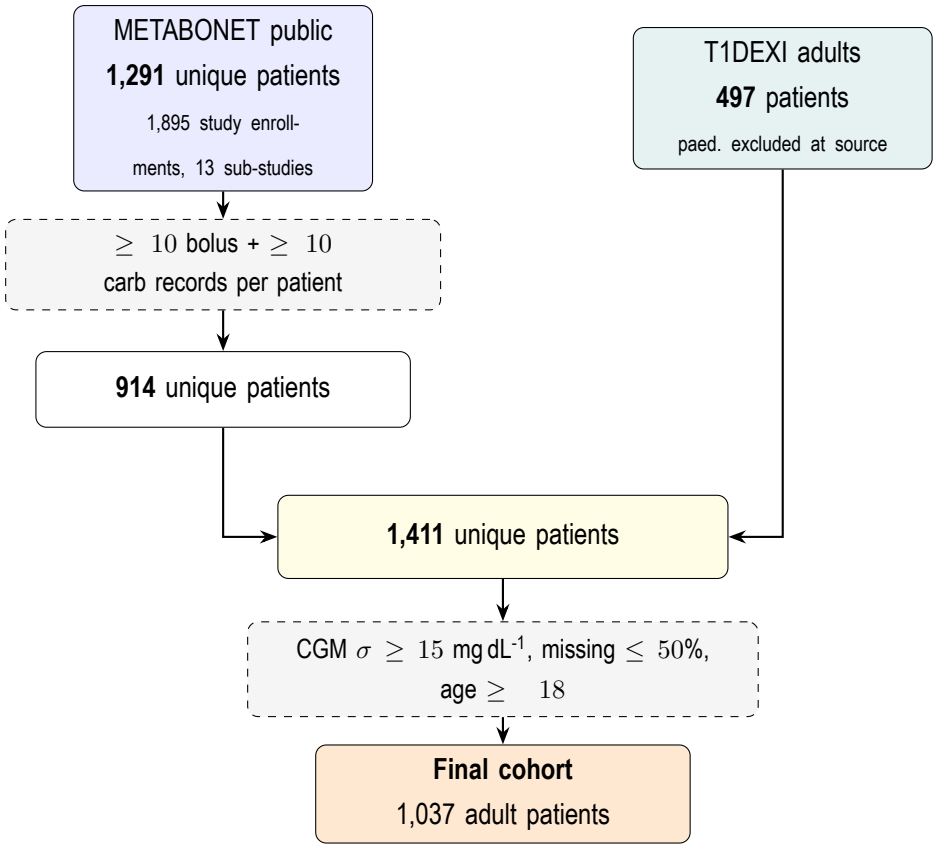

Figure 4. Patient inclusion pipeline.

5.1.3 Data Normalisation

As justified in Section 4.3, three channels are normalised by subtracting the population mean and dividing by the population standard deviation:

$$\hat{x} = \frac{x - \mu}{\sigma}. \tag{5.1}$$

The parameters are estimated from all 1,037 patients: *CGM* ($\mu = 144.40\, mg\, dL^{-1}, \sigma = 57.11\, mg\, dL^{-1}$), PI ($\mu = -3.06$, $\sigma = 4.22$), and RA ($\mu = 0.63$, $\sigma = 2.48$).

5.1.4 Missing Data Handling

CGM gaps and missing driver records require different strategies. Short gaps in the CGM signal of up to one hour are filled by linear interpolation; longer gaps are left as null and potentially discarded at the windowing stage. For insulin and carbohydrate records, however, for null values we don't know if no event happened or if it was simply not logged. Therefore, we cannot impute at all. An approximation of the scale of this annotation gap across the cohort is quantified in Section 5.2.2.

5.1.5 Pharmacokinetic Transformations: Plasma Insulin Approximation and Rate of Exogenous Glucose Appearance

As justified in Section 4.2.2, from insulin and carbohydrate discrete records we derive two continuous features per timestep from the Hovorka (2004) pharmacokinetic model [21]: plasma insulin approximation (PI), the active insulin concentration in plasma, and rate of carbohydrate absorption (RA), the glucose flux entering the bloodstream from the gut.

Plasma Insulin Approximation. PI is computed via a three-compartment subcutaneous absorption system. Combined bolus and basal insulin enter a first depot compartment S_1, drain into S_2, and populate the plasma concentration I (mU L^{-1}):

$$\frac{dS_1}{dt} = u(t) - \frac{S_1}{t_{\max I}}, \tag{5.2}$$

$$\frac{dS_2}{dt} = \frac{S_1}{t_{\max I}} - \frac{S_2}{t_{\max I}}, \tag{5.3}$$

$$\frac{dI}{dt} = \frac{S_2}{t_{\max I} \cdot V_I} - k_e\, I, \tag{5.4}$$

where $u(t)$ is the total insulin input (mU min^{-1}), $t_{\max I} = 55$ min is the time of peak subcutaneous absorption, $V_I = 0.12$ L kg^{-1} is the insulin volume of distribution, and $k_e = 0.138$ min^{-1} is the plasma elimination rate constant [21].

Rate of Exogenous Glucose Appearance. RA follows a two-compartment gut model. Ingested carbohydrates $G(t)$ (g min^{-1}) pass through compartments D_1 and D_2 before entering the bloodstream as the absorption flux RA:

$$\frac{dD_1}{dt} = \frac{A_G\, G(t)}{\tau_D} - \frac{D_1}{\tau_D}, \tag{5.5}$$

$$\frac{dD_2}{dt} = \frac{D_1}{\tau_D} - \frac{D_2}{\tau_D}, \tag{5.6}$$

$$\mathrm{RA}(t) = \frac{D_2}{\tau_D}, \tag{5.7}$$

where $\tau_D = 40$ min is the gut transit time constant and $A_G = 0.8$ is the carbohydrate bioavailability fraction [21].

5.1.6 Temporal Feature Engineering

As motivated in Section 4.2.3, glucose dynamics follow a 24-hour pattern that the model must be able to see. The time of day is encoded as a circular feature pair at each timestep:

$$\texttt{hour\_sin} = \sin\left(\frac{2\pi \cdot h}{24}\right), \qquad \texttt{hour\_cos} = \cos\left(\frac{2\pi \cdot h}{24}\right), \tag{5.8}$$

where h is the fractional hour in $[0, 24)$. This unit-circle representation keeps 23:55 adjacent to 00:05 in feature space, which a plain numeric hour would not.

These features tell the model what time of day each step is, but not where it falls within the window. A window starting at 06:00 contains a step at position 0 (06:00) and a step at position 287 (around 05:55 the next morning); both share nearly the same hour values, but they are very far apart. Sinusoidal positional encodings [28], added to the projected input inside the transformer backbone (Section 5.3.1), give this information to the model: they ensure the model knows that position 0 is the start and position 287 is the end, regardless of the clock time at each position.

5.1.7 Sliding Window Serialisation

After having all the features prepared and normalised, we cut each patient's time series into 24-hour windows of 288 steps, advancing by 6 hours (72 steps) between consecutive windows.

Windows with more than 20% null *CGM* readings are discarded. Across all 1,037 patients this gives 999,224 usable 24-hour windows to train and evaluate the models.

5.1.8 Patient-Level Data Split

After having the data prepared, we proceed with the splitting. We split by patients, not windows, into training (831), validation (103), and test (103) sets. A window-level split would let the model see some windows from a test patient during training, inflating the performance. Splitting at the patient level guarantees that all windows from a given patient fall in exactly one pool.

5.2 Exploratory Data Analysis

Before training, we characterise the final cohort across two dimensions: demographic composition and annotation coverage. An extended EDA can be found in Appendix C.

5.2.1 Cohort Demographics

Figure 5 shows the cohort across four dimensions: age, therapy modality, dataset source, and sex.

Age peaks between 25 and 40. Therapy modality is AID-dominant. This reflects study composition, as several of the trials included we're only for AID users. The two source datasets are quite balanced: 549 patients (53%) from METABONET vs 488 (47%) from T1DEXI. Sex was recorded for 992 patients, of whom 650 (63%) are female and 342 (33%) male.

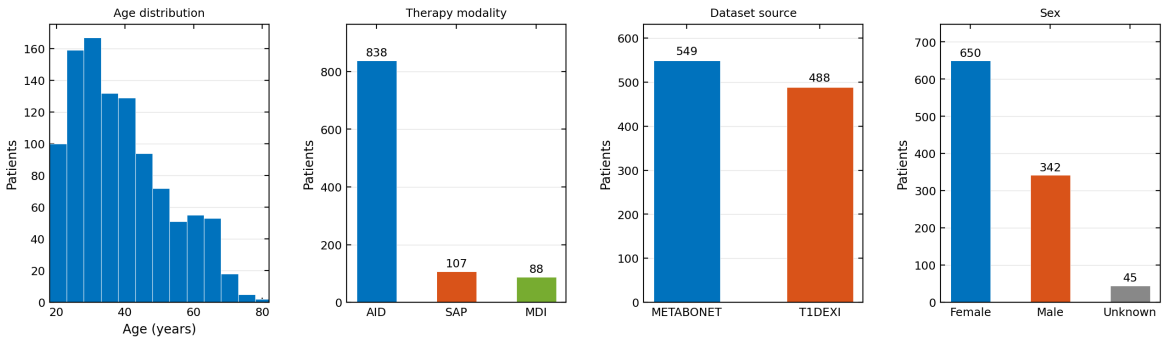

Figure 5. Cohort demographics of the 1,037-patient training cohort, across age distribution, therapy modality (AID / SAP / MDI), dataset source (METABONET / T1DEXI), and sex.

5.2.2 The Driver-Blindness Problem: Quantitative Analysis

Annotation quality directly affects how much of each patient's data can be used for training: a window with no logged events (the patient forgot to log them) gives a weaker training signal than one with confirmed insulin and carbohydrate records. Figure 6 shows the carbohydrate annotation rate for each patient, defined as the fraction of days with at least one logged carbohydrate event, for both the initial 1,788-patient raw cohort and the final 1,037-patient cohort.

We can see that filtering has improved annotation coverage: More than 350 patients with no carbs logged were removed. The spread remains wide, however; in the final cohort the median is 75%.

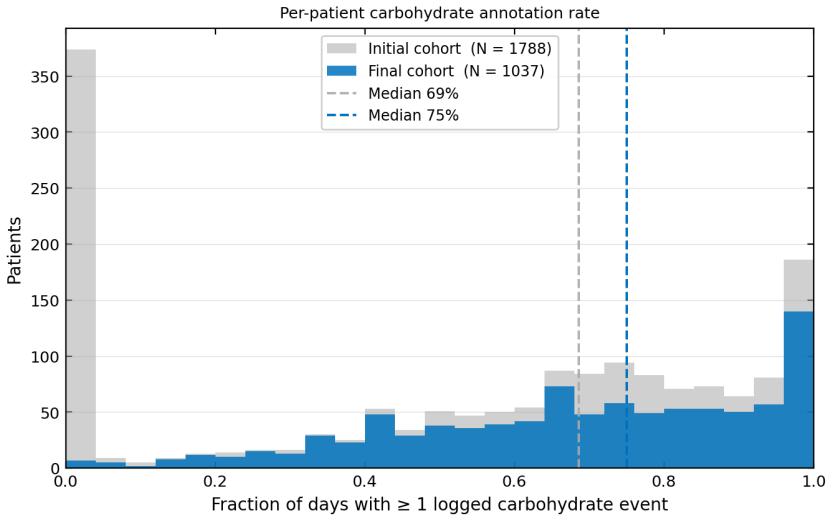

Figure 6. Per-patient carbohydrate annotation rate (fraction of calendar days with at least one logged carbohydrate event) for the initial 1,788-patient raw cohort and the final 1,037-patient retained cohort. Dashed lines mark the respective medians (69% and 75%).

5.3 Model Architecture

Both encoder-only and decoder-only models share a common transformer backbone and differ only in their attention pattern and pre-training objective. We will describe the shared backbone first, then the

encoder and decoder separately, and finally the task heads we attach at Stage 2. Figure 7 shows an overview of the full pipeline.

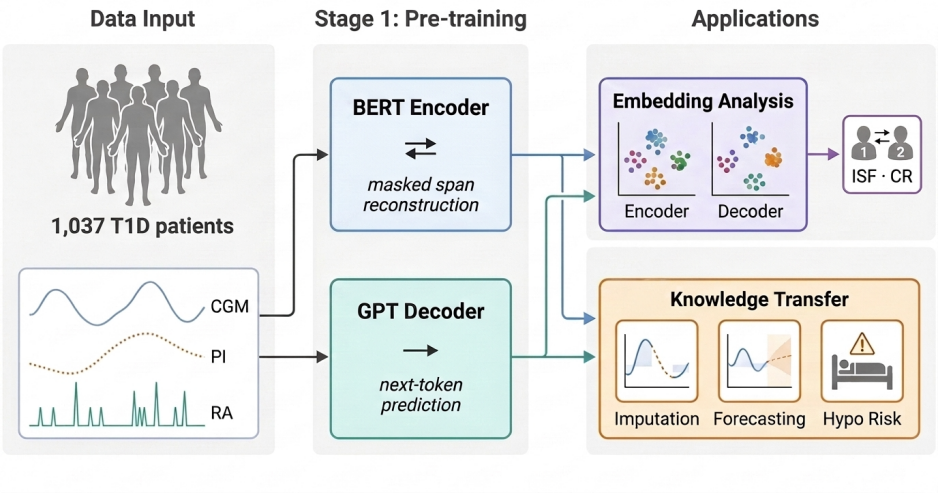

Figure 7. Overview of the two-stage foundation model pipeline. Stage 1 pre-trains the encoder (BERT-style, masked span reconstruction) and decoder (GPT-style, next-token prediction) on *CGM* data from 1,037 patients. The learned representations feed two evaluation branches: an embedding analysis (UMAP visualisation and Ridge probes) and a knowledge transfer branch with three downstream clinical tasks. Patient-level profiling (ISF · CR) is derived from the patient embeddings produced by the embedding analysis step.

5.3.1 Common Transformer Backbone

Each input window has shape $(288,\ 7)$: 288 five-minute steps and 7 features. We first project these 7 features into a 128-dimensional embedding using a Dense layer applied identically at every timestep. Sinusoidal positional encodings [28] are then added so the model knows where in the day each step falls.

The embedded sequence passes through five transformer blocks. Each block has a four-head self-attention layer and a position-wise feed-forward sublayer with inner dimension $d_{\text{ff}} = 256$, with pre-layer normalisation and residual connections throughout. Dropout of 0.2 is applied to both attention weights and feed-forward activations during training. The full backbone has 663,936 parameters. Table 7 summarises each component and Figure 8 shows the full architecture.

Table 7. Backbone architecture shared by the encoder and decoder. CLS token applies to the encoder only.

Component	Output shape	Params
Input projection (Dense 7→128)	(288, 128)	1,408
Sinusoidal PE	(288, 128)	0 (fixed)
CLS token (encoder)	(289, 128)	128
Transformer block $\times 5$	(288/289, 128)	132,480 $\times$ 5
Total		663,936

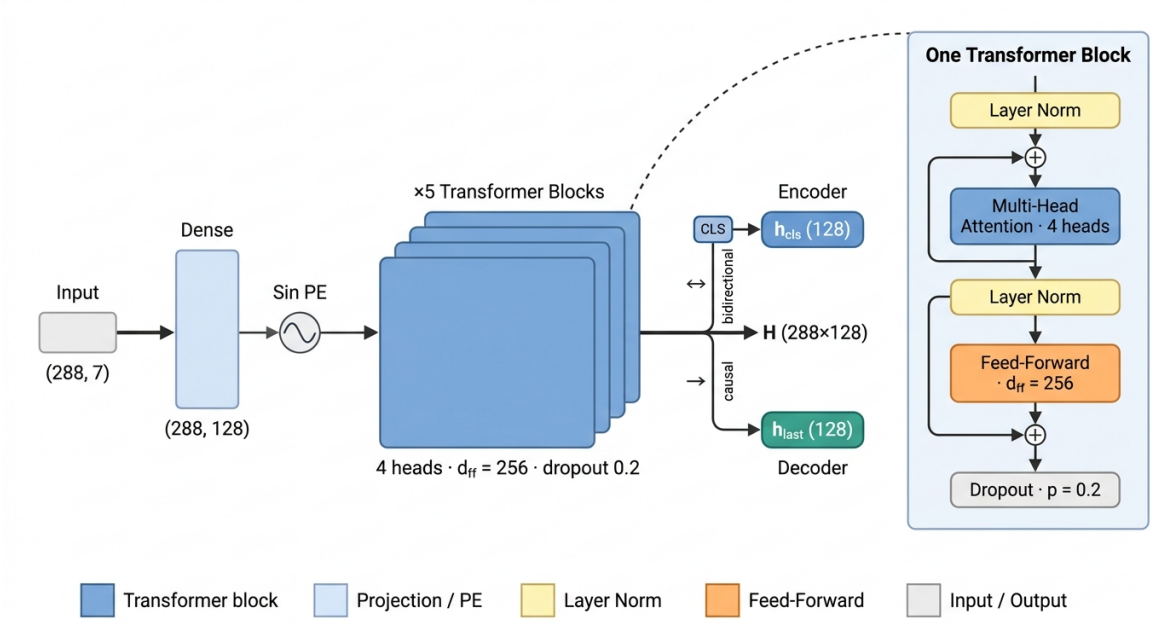

Figure 8. Shared transformer backbone. Each input window is projected to $d_{\text{model}} = 128$ and passed through five transformer blocks (4 attention heads, $d_{\text{ff}} = 256$, dropout 0.2). The encoder prepends a CLS token and produces both the sequence representation **H** and the global summary $\mathbf{h}_{\text{cls}}$; the decoder uses causal masking and takes $\mathbf{h}_{\text{last}}$ as its global summary.

5.3.2 Stage 1a: MTSM Encoder

The encoder uses bidirectional self-attention as explained in Section 4.5.2. A learnable CLS token is prepended to the input, extending the sequence to 289 steps. After the five transformer blocks, the CLS output becomes the global window summary $\mathbf{h}_{\text{cls}} \in \mathbb{R}^{128}$; the remaining 288 positions form the sequence representation $\mathbf{H} \in \mathbb{R}^{288\times128}$.

The pre-training task is masked span reconstruction [30, 31] as described in Section 4.5.2. A contiguous span of 60 to 96 steps (five to eight hours) is chosen at random, the *CGM* values at those positions are replaced with a learned mask-token embedding, and the encoder must reconstruct the original glucose from the context given by other features.

The loss is cross-entropy over $K = 200$ equal-width bins covering $[40,\ 400]\ mg\,dL^{-1}$. Rather than predicting a single value, the model outputs a probability distribution over bins at each masked position, which handles uncertainty across the full glucose range [6].

5.3.3 Stage 1b: NTP Decoder

The decoder is causal [33]: we apply a triangular attention mask so that position i can only attend to positions $\leq i$, never to the future. The final hidden state at position 287 is the global causal summary $\mathbf{h}_{\text{last}} \in \mathbb{R}^{128}$, because it has to predict the next value considering the full window.

The pre-training objective is next-token prediction on *CGM*. At each position i the model predicts the bin index of the next step $i+1$, using the same $K = 200$ discretisation. The loss is the mean cross-

entropy over all 287 prediction positions in the window.

5.3.4 Patient Embeddings

Both foundation models produce a summary for each 24-hour window: the encoder outputs $\mathbf{h}_{\text{cls}} \in \mathbb{R}^{128}$ from the CLS token, and the decoder has no CLS token so we mean-pool its 288 hidden states to obtain a window summary $\bar{\mathbf{h}} \in \mathbb{R}^{128}$.

A patient is monitored over many days, so the model produces a separate summary for each of their 24-hour windows rather than a single patient-level vector. To represent the whole patient as a single vector, we run the frozen model once per window, collect the 128-dimensional summary from each, and average across all of them. Formally, for a patient with W valid windows:

$$\mathbf{z} = \frac{1}{W}\sum_{w=1}^{W} \mathbf{h}_{\text{cls}}^{(w)} \quad \text{(encoder)}, \qquad \mathbf{z} = \frac{1}{W}\sum_{w=1}^{W} \bar{\mathbf{h}}^{(w)} \quad \text{(decoder)}. \tag{5.9}$$

Averaging is the standard approach for building patient or document-level representations from transformer sequence models: it pools information across all available recordings without introducing additional trainable parameters or task-specific bias [6, 7, 30]. No gradient flows through $\mathbf{z}$; it is a direct read-out of what the pre-trained model has learned. This vector is the input to the physiological profiling probe (App 4) and to the embedding analysis in Section 5.6.

5.3.5 Stage 2 Task Heads

With the pre-trained representations defined, we attach a small task-specific head for each downstream application. Track 1 keeps the foundation model frozen and trains only the head; Track 2 fine-tunes the full model end-to-end from the Stage 1 checkpoint.

App 1: Gap Imputation. Imputation requires no additional training at all (the encoder training itself is imputation). At test time we remove a contiguous span from the window, replace it with the mask token, and run the encoder. The pre-training classification head converts the hidden states at masked positions into bin logits, and we take the argmax bin centre as the predicted *CGM* value. This is the most direct test of what the encoder learned during pre-training.

App 2: Short-Horizon Forecasting. In this downstream application we predict the next two hours beyond the end of a window. The sequence representation $\mathbf{H} \in \mathbb{R}^{288\times128}$ is first compressed into a single context vector $\mathbf{h} \in \mathbb{R}^{128}$ by an attention-pooling layer [46]. Rather than averaging all 288 hidden states equally, a learnable query vector scores each one, the scores are normalised through a softmax to sum to one, and the output is their weighted sum. This lets the model focus on the most informative timesteps when summarising the window. We inject $\mathbf{h}$ as the initial hidden state of a single-layer LSTM with 128 units, which unrolls autoregressively for 24 steps. At each step a learnable horizon embedding

of dimension 32 serves as the LSTM input; the hidden state passes through Dense(64) and Dense(1), and the result is added as a residual correction to the last observed *CGM* value. Figure 9 contrasts the two approaches.

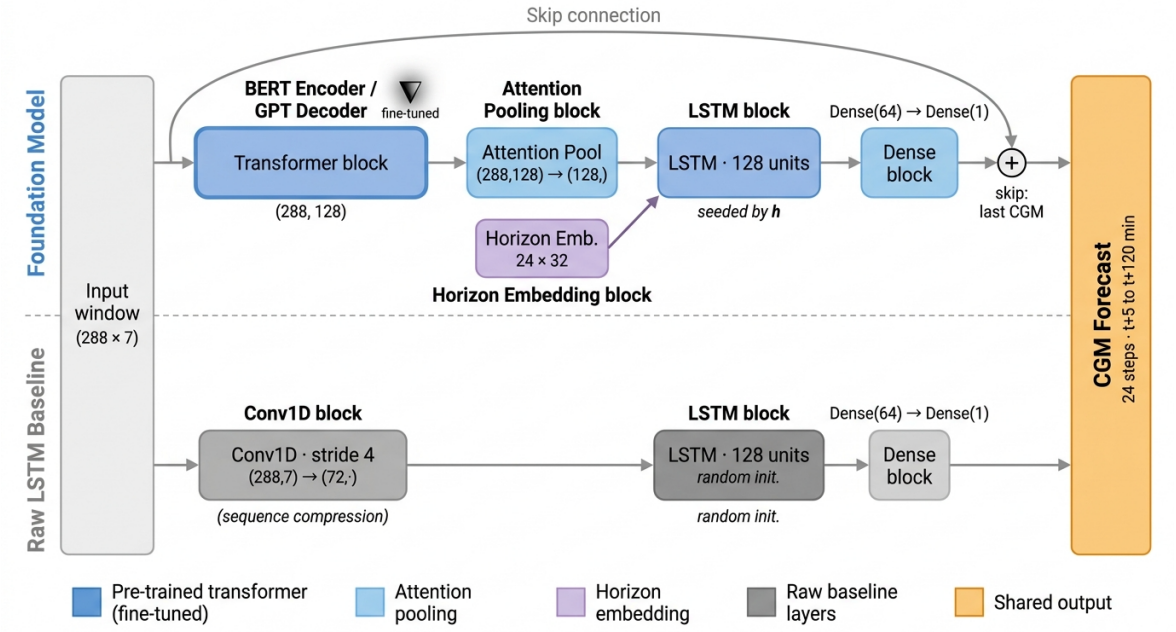

Figure 9. Comparison of the two forecasting architectures. The foundation model path (top) compresses the pre-trained sequence representation **H** via attention pooling into a single context vector **h**, which seeds the LSTM decoder; the full model is fine-tuned end-to-end from the Stage 1 checkpoint. The raw LSTM baseline (bottom) processes the input directly after a Conv1D compression step and is trained from random initialisation. Both paths share the same LSTM head structure and prediction target.

App 3: Nocturnal Hypoglycaemia Risk. We filter the dataset to bedtime windows: those whose final step falls between 21:00 and 23:59. Each window gets a binary label: positive if any *CGM* reading in the following eight hours falls below $70\ mg\,dL^{-1}$, negative otherwise. The encoder global summary $\mathbf{h}_{cls}$ passes through Dense(64, ReLU) followed by Dense(1, sigmoid) to produce a risk probability. The loss is binary cross-entropy.

App 4: Patient-Level Physiological Profiling. While Apps 1–3 test knowledge transfer to new tasks, App 4 asks how much clinical information is already encoded in the frozen representations. We attempt to recover each patient's insulin sensitivity factor (ISF) and carbohydrate ratio (CR) from the patient embedding **z** defined in Section 5.3.4. The patient vector **z** is the input to a Ridge regression probe: a simple linear model with L2 regularisation, fit on top of the frozen embeddings without modifying them. Using a linear model is a deliberate choice. If a straight line through the 128-dimensional space can predict ISF or CR, it means that information is already directly encoded in the representation and does not require a complex decoder to extract. The regularisation strength is chosen by five-fold

cross-validation on training patients to avoid overfitting on the 128-dimensional input. No gradient flows back to the foundation model at any point.

5.4 Training Configuration

5.4.1 Stage 1 Pre-training

We pre-train both models on the 934 non-test patients using TensorFlow 2.17.0 on a single NVIDIA RTX 5070 GPU (12 GB VRAM, CUDA 12.9). Both use the Adam optimiser with an initial learning rate of 10^{-3}, batch size 32, and dropout 0.2. The encoder is trained with early stopping on validation loss (patience 10) and converges at epoch 14 (val_loss $= 7.93$). The decoder runs for a fixed 70 epochs; next-token prediction is a harder objective and validation loss continues to decrease slowly throughout.

5.4.2 Stage 2 Task Heads

Fine-tuned Stage 2 variants initialise from the Stage 1 checkpoint and train end-to-end with Adam at 10^{-4}, batch 32, and early stopping (patience 15). We use the lower learning rate to avoid disrupting the pre-trained representations with large task-specific gradient steps at the start of fine-tuning.

The raw LSTM baseline trains from random initialisation with the same learning rate and patience. All variants use Huber loss for forecasting and binary cross-entropy for hypo risk.

5.5 Evaluation Metrics

We use different metrics for each task depending on what the output is and what matters clinically.

5.5.1 Gap Imputation

We measure RMSE (mg dL^{-1}), MAE (mg dL^{-1}), and R^2, computed only over the masked timesteps. All three are reported for 4-hour, 6-hour, and 8-hour gaps so we can see how quickly performance degrades. Baselines are: linear interpolation between the bounding valid values, and a raw LSTM trained on the same objective.

5.5.2 Short-Horizon Forecasting

We measure RMSE (mg dL^{-1}) at $t{+}5$ min, $t{+}30$ min, and $t{+}120$ min. A naive persistence baseline is included to frame the task difficulty.

5.5.3 Nocturnal Hypoglycaemia Risk

We measure two metrics: AUROC and AUPRC. AUROC is the primary metric and captures discrimination across all decision thresholds. We add AUPRC because the bedtime subset has a positive-class prevalence of approximately 27%, and AUPRC is more sensitive to performance at the high-precision

operating points that matter most in a clinical alerting system where false alarm fatigue is a real concern.

5.5.4 Patient-Level Physiological Profiling

We evaluate each probe by R^2 on the 103 held-out test patients, reported separately for ISF and CR. We compare three representations: encoder $\mathbf{h}_{\text{cls}}$, decoder mean-pooled $\mathbf{H}$ and a composite baseline that concatenates the mean 24-hour *CGM*, PI, and RA profiles and applies Ridge + PCA(32). The latest is basically a linear model with access to the full average daily driver curves. Beating it means the transformer captures inter-variable dynamics that are not recoverable from the raw profiles alone.

5.5.5 Confidence Intervals and Significance Tests

Point estimates alone do not tell us how much a reported difference might shrink or grow if we drew a different random sample of test patients or test windows. To quantify this uncertainty, we attach 95% confidence intervals (CIs) to the main metrics for the three downstream tasks.

The method is *bootstrap resampling*. Concretely, after collecting all window predictions on the 103 test patients, we draw $B = 1{,}000$ bootstrap samples: each sample is formed by resampling the evaluation windows with replacement (i.e. the same window can appear more than once, and some windows may not appear at all), keeping the same total count as the original. We compute the metric of interest (RMSE, AUROC, or AUPRC) on each bootstrap sample, giving us a distribution of 1,000 values. The 95% CI is then simply the interval from the 2.5th to the 97.5th percentile of that distribution.

For selected comparisons (decoder fine-tuned versus raw LSTM on AUROC, and the two fine-tuned variants versus raw LSTM on RMSE at $t{+}30$ min) we also report a two-sided paired bootstrap p-value. This answers the question: "could the observed gap between two models be explained by chance alone?" The procedure is: (1) compute the observed difference in metric between the two models on the full test set; (2) for each of the 1,000 bootstrap samples, compute the same difference; (3) shift the bootstrap distribution to be centred at zero, as required under the null hypothesis of no true difference; (4) count the fraction of shifted bootstrap values that are at least as extreme as the observed difference. That fraction is the p-value.

5.6 Embedding Analysis

Alongside the downstream tasks we also examine the decoder and encoder's representation space as a whole, asking whether pre-training on glucose dynamics produces a patient embedding that captures clinically meaningful structure without any supervision signal.

5.6.1 Extracting Patient-Level Embeddings

Patient embeddings are extracted as defined in Section 5.3.4, yielding a $1{,}037 \times 128$ matrix for each model.

5.6.2 Unsupervised Glycaemic Phenotyping

We apply UMAP [47] to project the 128-dimensional embeddings down to three dimensions for visualisation. Patients are coloured by their Glucose Risk Index (GRI) [14], a composite metric that weights hypoglycaemic and hyperglycaemic exposure by clinical severity, and are grouped into quintiles. If the layout separates GRI quintiles without any supervision during pre-training, it suggests the representations encodes the glycaemic control of these patients, not just the patterns needed for short-horizon prediction.

5.6.3 Linear Probes for Representation Quality

We use the same Ridge regression probe setup as App 4 (Section 5.3.5) to assess representation quality on ISF and CR. The three representations and baselines compared are as defined in Section 5.5.4: encoder $\mathbf{h}_{\text{cls}}$, decoder mean-pooled $\mathbf{H}$, and the composite Ridge + PCA(32) baseline.

Chapter 6

Results

In this chapter we present the experimental outcomes for each stage of the system, including pre-training convergence, downstream task performance, and embedding analysis. Numbers are reported as obtained; conclusions are drawn in Chapter 7.

6.1 Stage 1 Pre-training

Figure 10 shows the training loss for both models.

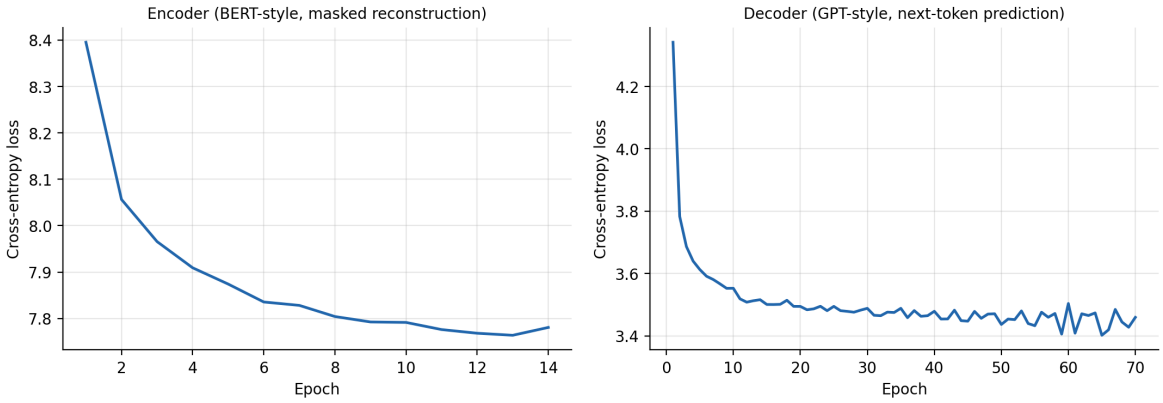

Figure 10. Training loss curves for the encoder (left, 14 epochs) and decoder (right, 70 epochs). Both models optimise cross-entropy loss over K=200 CGM bins.

One epoch is a full pass over the pre-training corpus of roughly 606,000 daily windows from the 831 pre-training patients: 4,734 gradient updates at batch size 128. On a single RTX 5070 (12 GB), one epoch took approximately 15 minutes.

Encoder. The encoder trained for 14 epochs before early stopping triggered, about 3.5 hours of GPU time and 66,000 gradient updates. Table 8 reports reconstruction quality on held-out masked spans, evaluated on the 103 test patients.

Table 8. Encoder masked-span reconstruction on the test set.

Metric	Value
MAE (z-score)	0.374
RMSE (z-score)	0.555
R^2	0.626
MAE (mg/dL)	21.3
RMSE (mg/dL)	31.7

Decoder. The decoder trained for the full 70 epochs (about 17 hours) without triggering early stopping. Next-token prediction is a denser objective than masked-span reconstruction, incurring a loss at every one of the 288 positions rather than only on the masked spans, so optimisation benefits from the longer schedule. We retain the epoch-70 checkpoint for all downstream experiments.

6.2 Gap Imputation

We evaluate gap imputation on the 103 test patients. Synthetic gaps of 4, 5, 6, and 8 hours are evaluated. We compare against piecewise linear interpolation and a trained raw LSTM baseline.

Table 9. Gap imputation by gap duration. FM: frozen encoder, no imputation-specific training. Linear: piecewise linear interpolation. Raw: trained LSTM. RMSE in mg/dL; values show point estimate [95% CI].

	FM (encoder)		Linear		Raw LSTM	
Gap	RMSE [95% CI]	R^2	RMSE [95% CI]	R^2	RMSE [95% CI]	R^2
4h	25.9 [24.6, 27.2]	0.782	33.5 [32.0, 35.1]	0.635	52.0 [49.4, 54.6]	0.120
5h	28.6 [26.8, 30.7]	0.737	37.5 [35.8, 39.1]	0.550	52.4 [50.0, 54.7]	0.120
6h	30.5 [28.9, 32.3]	0.704	41.5 [39.8, 43.3]	0.451	52.5 [50.3, 55.1]	0.121
8h	34.5 [32.7, 36.4]	0.620	46.8 [45.0, 48.5]	0.301	52.5 [50.3, 54.7]	0.122

RMSE is in mg/dL. FM R^2 falls from 0.782 at 4h to 0.620 at 8h as the gap lengthens. The raw LSTM baseline stays near $R^2 \approx 0.12$ across all gaps, as it cannot use the visible context after the gap. It suggests that the LSTM has collapsed.

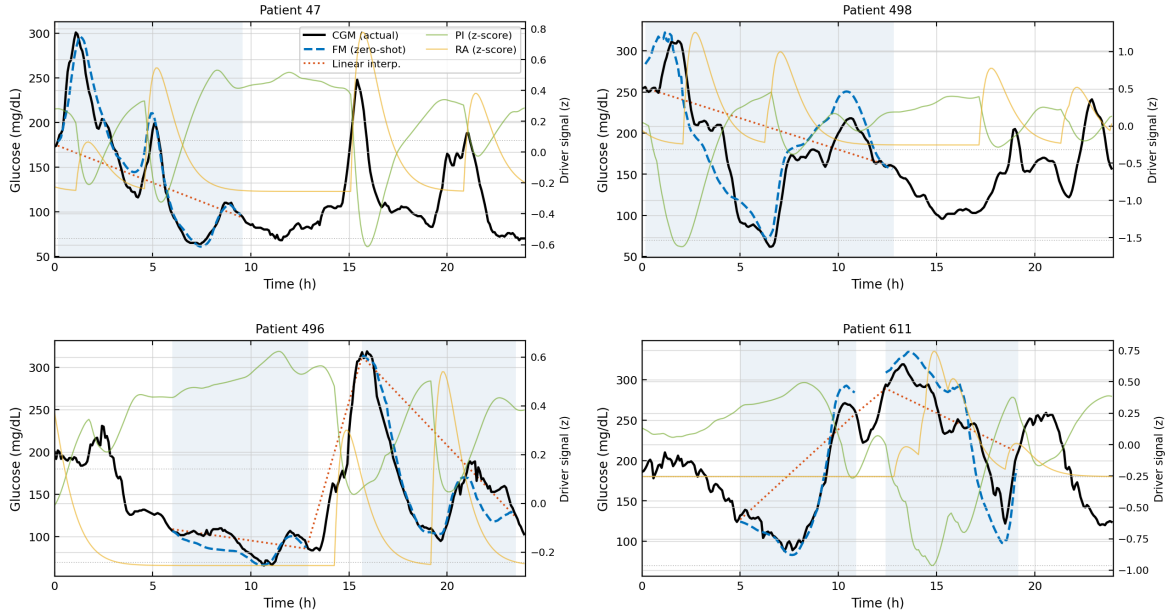

Figure 11. Gap imputation on four held-out test patients. Blue shading marks the masked region; black: actual CGM; blue dashed: FM reconstruction; red dotted: linear interpolation. Green and yellow curves (right axis): PI and RA driver signals.

6.3 Short-Horizon Glucose Forecasting

We evaluate four variants on the 103 test patients: naive persistence, raw LSTM, encoder fine-tuned, and decoder fine-tuned. Table 10 reports RMSE and R^2 at three horizons.

Table 10. Forecasting performance. RMSE in mg/dL; values show point estimate [95% CI]. Paired bootstrap p-values at $t{+}30$: raw vs encoder FT $p < 0.001$, raw vs decoder FT $p < 0.001$.

Model	RMSE $t{+}5$ [95% CI]	R^2 $t{+}5$	RMSE $t{+}30$ [95% CI]	RMSE $t{+}120$ [95% CI]
Naive (persistence)	8.80 [8.70, 8.90]	0.974	25.78 [25.58, 25.97]	55.33 [55.03, 55.62]
Raw LSTM	7.94 [7.83, 8.05]	0.979	21.76 [21.55, 21.95]	49.06 [48.80, 49.36]
Encoder fine-tuned	8.10 [8.00, 8.21]	0.978	22.54 [22.34, 22.76]	50.09 [49.81, 50.36]
Decoder fine-tuned	9.21 [9.12, 9.31]	0.972	24.23 [24.02, 24.43]	45.65 [45.39, 45.93]

Both fine-tuned foundation model variants are competitive with the raw LSTM across all horizons.

6.4 Nocturnal Hypoglycaemia Risk

We evaluate binary hypoglycaemia risk on the 103 test patients. The positive class is any CGM < 70 mg/dL in the following 8 hours. Prevalence is 23.8%. Table 11 reports AUROC and AUPRC for each variant.

Table 11. Nocturnal hypoglycaemia risk classification on bedtime test windows. Prevalence = 23.8%. Values show point estimate [95% CI]. $^{\dagger}p = 0.004$ vs raw LSTM (paired bootstrap); $^{\ddagger}p = 0.31$ vs raw LSTM.

Model	AUROC [95% CI]	AUPRC [95% CI]
Decoder fine-tuned†	0.737 [0.726, 0.748]	0.550 [0.532, 0.569]
Raw LSTM	0.727 [0.715, 0.739]	0.534 [0.514, 0.553]
Encoder fine-tuned‡	0.723 [0.712, 0.735]	0.529 [0.510, 0.548]

The decoder fine-tuned variant leads on both metrics, with the advantage over the raw LSTM being statistically significant ($p = 0.004$). The encoder fine-tuned variant trails the raw LSTM by a small margin that does not reach significance ($p = 0.31$).

6.5 Patient-Level Physiological Profiling

We apply the Ridge regression probe described in Section 5.3.5 to recover ISF, CR, and HbA1c from each patient embedding. Table 12 reports the cross-validated R^2 for each representation.

Table 12. Patient-level Ridge probe R^2 (mean $\pm$ SD from 5-fold CV).

Target	Enc h_{cls}	Dec H	CGM+PI+RA
ISF (mg/dL/U)	0.372 ± 0.092	0.499 ± 0.051	0.151 ± 0.075
CR (g/U)	0.406 ± 0.067	0.271 ± 0.134	0.258 ± 0.108

Encoder and decoder exceed the CGM+PI+RA curve baseline on both targets. Figure 12 shows predicted vs actual ISF and CR.

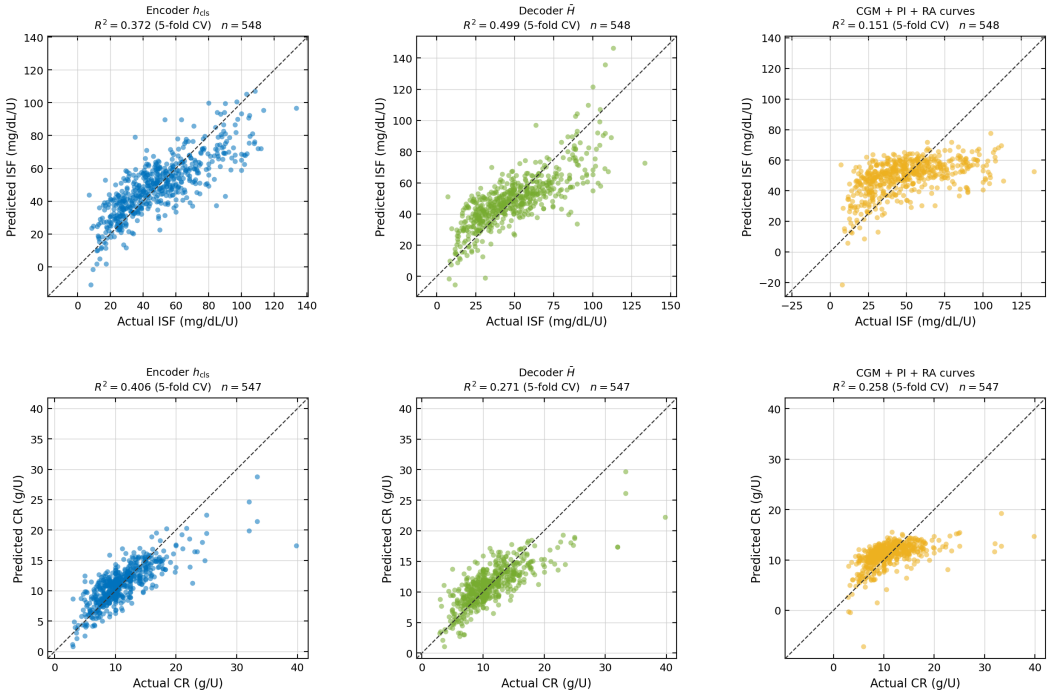

Figure 12. Predicted vs actual ISF (top row) and CR (bottom row). Columns: encoder h_{cls}, decoder mean-pooled H, and CGM+PI+RA curve baseline.

6.6 Embedding Analysis

We extract one embedding per patient, yielding a 1,037 $\times$ 128 matrix for each model. Table 13 summarises the geometry of both embedding spaces (discussed in Section 2.5) and linear probe accuracy for the Glucose Risk Index (GRI).

Table 13. Embedding space summary. PCA dims: components needed to explain 90% of variance. GRI probe: Ridge R^2 from patient embeddings to GRI. MCR^2 metrics computed under GRI quartile grouping.

	Encoder h_{cls}	Decoder H
PCA dims (90% var)	8	3
GRI linear probe R^2	0.949	0.997
MCR^2 ΔR (GRI quartile)	14.351	13.510
Subspace incoherence (GRI quartile)	0.719	0.746

The decoder embedding space is lower-dimensional (3 components for 90% variance vs 8 for the encoder). Both models satisfy the MCR^2 criterion under GRI quartile grouping: coding rate reduction is high for both, and subspace incoherence above 0.71 confirms that the four GRI groups occupy largely orthogonal directions in each embedding space.

Figure 13 visualises the 3D UMAP projections of both embedding spaces, coloured by GRI quintile. Both spaces separate patients by glycaemic risk without supervision.

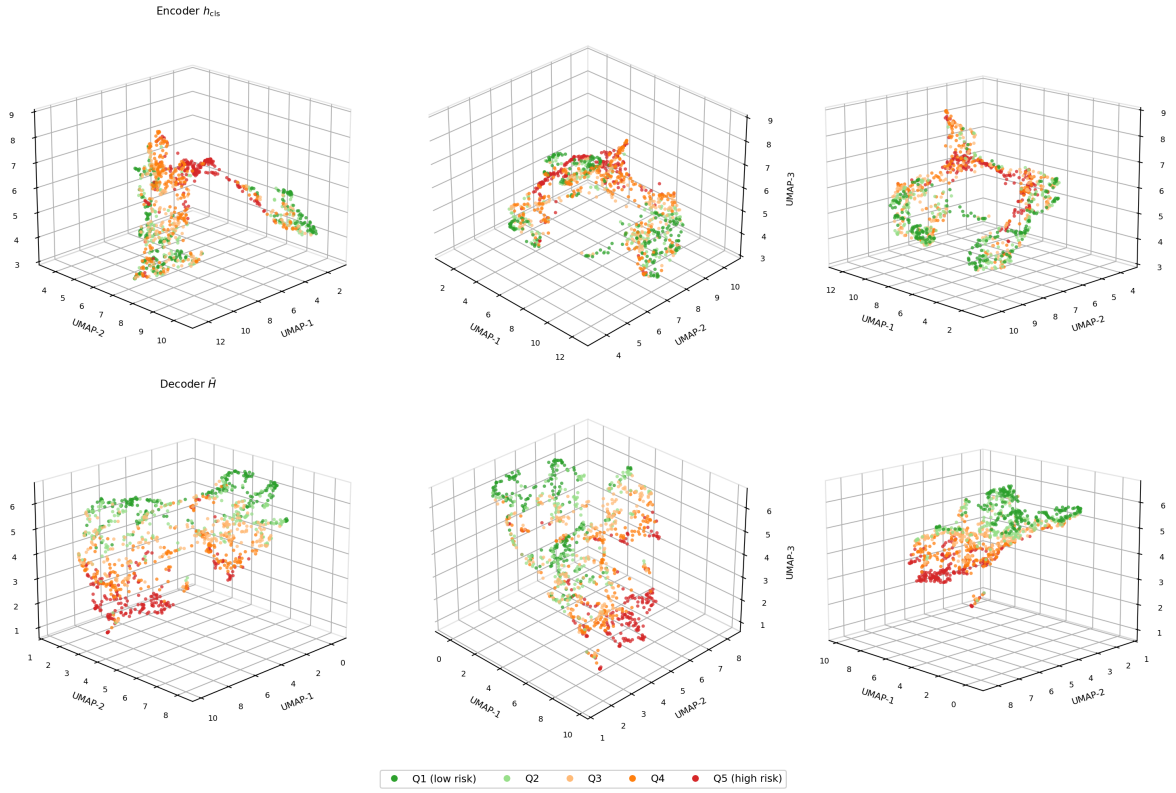

Figure 13. 3D UMAP projections of patient embeddings coloured by GRI quintile (Q1 best glycaemic control, Q5 worst; n_neighbours=15, min_dist=0.1). Top row: encoder h_{cls}. Bottom row: decoder mean-pooled H. Each row shows three viewing angles of the same embedding. Both spaces separate glycaemic risk without supervision.

Chapter 7

Discussion

This chapter discusses what the experimental results mean, where they align with the thesis objectives, and where they fall short as well as the limitations, biases and future work.

7.1 What the Pre-training Task Actually Teaches

The models were given two self-supervised objectives. The encoder was asked to reconstruct masked spans of glucose from bidirectional context. The decoder was asked to predict the next CGM value step by step. Neither objective mentions insulin sensitivity, carb ratio, or nocturnal hypoglycaemia. These labels were never given.

Yet the representations built by these objectives encode all three. The reason lies in the structure of the task itself. To correctly reconstruct a masked 4-hour glucose span after a meal, the model must integrate the magnitude of the carb absorption signal, the timing of the insulin response, and the patient's historical glucose trajectories in the same window.

Self-attention can directly model these long-range dependencies; a standard LSTM must compress them into its hidden state, losing temporal resolution along the way. Transformers were not designed for physiological time series, but the causal structure of glucose dynamics fits their inductive bias well.

7.2 Downstream Task Performance

Gap imputation. Gap imputation is the cleanest result. The encoder receives a 24-hour window with up to 8 hours of glucose zeroed out and reconstructs the missing segment in a single forward pass. At a 4-hour gap, the FM achieves R^2=0.782, a more than solid performance considering glucose variability in this time ranges (Table 9).

Short-horizon forecasting. Forecasting is more complex. At 30 minutes, the trained-from-scratch LSTM achieves RMSE=21.3 mg/dL; the encoder fine-tuned reaches 22.2 mg/dL and the decoder fine-tuned 24.4 mg/dL. (Table 10). The absolute margins are narrow, and neither FM variant outperforms the LSTM trained from scratch on this task. What the results do show is that a model pre-trained without ever seeing a forecasting label reaches the same performance as a task-specific baseline, which is the knowledge transfer claim. Neither model, however, reaches clinical-grade forecasting performance.

CGM-LSM [8] reports RMSE=9.02 mg/dL at 30 minutes on its full mixed cohort, but its T1D-specific results (RMSE=29.81 mg/dL at 2 hours) are closer to ours. These numbers are not directly comparable: different datasets, different patient populations, different preprocessing. We mention them only to place

our results in the broader landscape.

Nocturnal hypoglycaemia risk. The hypo risk results are where the FM variants show the clearest advantage over baseline. The decoder FT leads (AUROC=0.737, AUPRC=0.550), followed by the trained from scratch LSTM (0.727, 0.534) and encoder FT (0.723, 0.529) (Table 11).

The decoder's win is the most surprising result in this section. The pre-training task, NTP of glucose, shares no obvious relationship with binary hypoglycaemia classification 8 hours ahead. The decoder was never shown a hypoglycaemia label during training. Yet the summary it builds from 24 hours of glucose dynamics encodes enough about active insulin, absorption dynamics, and circadian phase to support a better classification head than an LSTM trained from scratch directly on the task.

Knowledge transfer. Across all three tasks, fine-tuned FM variants have reached performance comparable to or better than LSTMs trained end-to-end on task-specific labels. In imputation, the FM wins. In forecasting, the encoder FT is within 5% of the LSTM at 30 minutes. In hypo risk, the decoder FT leads. This is the core knowledge transfer result: a model pre-trained on self-supervised glucose reconstruction learns representations that transfer across tasks it was never shown during training. The performance is not yet clinical-grade. We frame these as proof of concept.

7.3 The Physiological Profiling Surprise

The patient-level profiling results are the ones we did not anticipate.

The key baseline is the CGM+PI+RA curve baseline: a Ridge regression with access to each patient's 24-hour CGM, PI, and RA profiles compressed to 32 PCA components. This amounts to 864 input features summarising the average daily pattern of all three signals, a substantial amount of information. It achieves ISF R^2=0.151 and CR R^2=0.258, proving that regression on these parameters is not an easy task, (Table 12).

The transformer embeddings outperform this baseline. The decoder mean-pooled H achieves ISF R^2=0.499, more than three times higher than the curve baseline. The encoder h_{cls} achieves CR R^2=0.406, 57% higher.

The model was never told this was what it needed to learn. It was told to predict the next glucose value, or to reconstruct a masked span. The ISF and CR signal emerged implicitly from the physics of the problem: to do the self-supervised task well, the model must encode the patient's insulin response, and this encoding turns out to be the signal needed for physiological profiling.

This result changed how we thought about what the model was doing. It is not a time series compressor. It is building a patient-specific physiological profile from the dynamics of glucose.

7.4 Representation Structure

The encoder and decoder embed patients into structurally different spaces. The encoder h_{cls} requires 8 PCA components to explain 90% of variance; the decoder mean-pooled H requires only 3 (Table 13). The encoder distributes information across more independent directions; the decoder collapses to fewer.

This structural difference explains the downstream probe results. The decoder's low-dimensional effective space is dominated by mean glucose, which is why its GRI probe achieves R^2=0.997 (GRI is strongly correlated with mean glucose) but its CR probe reaches only R^2=0.271. The encoder's richer manifold supports both ISF R^2=0.372 and CR R^2=0.406, encoding more independent aspects of the patient's physiology in separate representational directions.

The UMAP projections confirm this pattern visually (Figure 13). Both spaces stratify patients by GRI quintile without supervision, but for different reasons. The decoder space is approximately a line ordered by mean glucose. The encoder space is a more complex manifold that separates patients along multiple axes simultaneously, consistent with the higher intrinsic dimensionality.

The bidirectional masking objective likely contributes to this richer geometry. To reconstruct masked glucose from full temporal context, the encoder must encode both the absolute glycaemic level and the dynamic pattern of each window. The decoder's next-token objective has weaker pressure to encode dynamics beyond the immediate history, since the current value is the dominant predictor of the next one. The result is a qualitative difference in what each representation space captures, and in what clinical information is recoverable from it.

These geometric differences are confirmed quantitatively by the MCR^2 metrics in Table 13, which leads the criterion for structured representations introduced in Chapter 2. Under GRI quartile grouping, coding rate reduction reaches $\Delta R = 14.351$ for the encoder and $\Delta R = 13.510$ for the decoder: within each GRI quartile, embeddings are substantially more compact than the full patient population, and subspace incoherence above 0.71 confirms that the four GRI groups occupy largely orthogonal directions in each space. Both models satisfy the criterion; the encoder's slightly higher ΔR is consistent with its richer eight-dimensional geometry, which distributes more representational capacity across directions that separate patients by glycaemic control.

7.5 Limitations

Data quality. The most fundamental challenge in this work, and in the field more broadly, is data quality. CGM data from real-world studies contains sensor calibration drift and gaps caused by sensor failures. More importantly for this work, insulin doses and meal records are mostly entered manually, making them prone to omissions. Plasma insulin and carb absorption are not measured directly but computed from these logged records: any logging error propagates into the PI and RA signals that are the model's secondary inputs. All available preprocessing steps were applied (step-change filtering,

quality thresholds, artefact removal), but residual noise remains and its effect on representation quality is difficult to quantify.

Clinical readiness. The best AUROC for nocturnal hypoglycaemia risk is 0.737. This is meaningful for a proof-of-concept study but falls well below the threshold required for a clinical safety application. Gap imputation results (R^2=0.782 at 4h) are stronger in relative terms, but CGM gap filling is already handled in clinical practice by device-level interpolation. No results in this thesis have been validated prospectively or in a deployment setting.

Explainability and interpretability. The representations are effective but opaque. We show that clinical parameters such as ISF and CR are recoverable from the embeddings, yet not how the model encodes them: which attention patterns or representational directions drive a given prediction is unknown. Attention weights offer a partial view but are not full explanations of model behaviour. For a clinical decision-support tool this is a real barrier, since clinicians need to understand why a prediction is made before acting on it.

Pre-training objective choices. The encoder and decoder use MTSM and NTP, the direct analogues of BERT and GPT. These are established objectives with well-understood properties, but they are not the only options. Contrastive objectives, predictive coding architectures, or objectives designed around known physiological constraints (for example, the half-life of insulin) might learn qualitatively different representations. We have compared BERT-style and GPT-style objectives within this framework; comparing against other pre-training paradigms is left for future work.

No fair comparison with prior CGM foundation models. CGMformer [7] was trained on a large multi-centre dataset and evaluated primarily on classification tasks (T2D screening, diabetic complications). GluFormer [6] used 10,812 predominantly non-diabetic adults and focused on long-term outcome stratification. Neither model was evaluated on T1D patients for short-horizon physiological tasks, and neither attempts ISF or CR recovery. The comparison class in this thesis, short-term physiological task performance on T1D, is not addressed by prior work, and we have been careful to interpret literature metrics.

Population scope. The 1,037 patients are all adults with T1D drawn from U.S. and European research cohorts (METABONET, T1DEXI). Generalisability to paediatric populations, T2D, or patients recorded outside study-grade CGM protocols is unknown.

7.6 Dataset Biases and Representativeness

CGM technology is not equally accessible. In the United States, roughly 49.5% of White T1D patients use a CGM, compared to 17.7% of Black patients [**t1dexchange_equity**]. Income matters too: CGM use is substantially lower among lower-income households, and private insurance is strongly associated with uptake [**cgm_equity_review**]. Any large CGM dataset therefore reflects who can afford and access the technology, not the full T1D population.

METABONET, one of our two data sources, acknowledges this directly: White patients are overrepresented because the contributing studies were run primarily in North America and Europe [9]. Our combined 1,037-patient cohort does not include a demographic breakdown, so we cannot say by how much.

This matters for the model. A recent study found that an LSTM glucose predictor made larger errors for Black patients than for White patients when the training data was predominantly White [**algorithmic_bias_cgm**].

Any clinical use of this kind of model requires evaluation across diverse populations, not just the research cohorts it was trained on.

7.7 Clinical and Research Implications

The most direct clinical application suggested by this work is insulin dose personalisation. ISF and CR, the two parameters that govern how a patient's insulin dose is calculated, are today estimated through clinical experience and patient trial and error. The results here show that a model trained only on glucose dynamics can recover these parameters better than giving a regression model the patient's full profile. If this holds in larger and more diverse cohorts, it points toward a way to initialise insulin doses from wearable data.

More broadly, the approach itself has value beyond any single task. The pre-trained model is fixed; adding a new downstream task requires only a small output layer trained on that task's labels. This means the work of learning glucose dynamics can be done once, on a large unlabelled dataset, and reused for many clinical questions.

The comparison between the encoder and decoder also has a practical lesson: not all pre-training objectives are equal. The encoder's bidirectional objective leads to a richer, more distributed representation that captures dose-response dynamics. The decoder collapses toward mean glucose.

These are proof-of-concept results, not a finished product. The performance gains are real but modest. The physiological profiling results and the architectures comparison are the strongest results. What the work demonstrates, taken as a whole, is that learning from unlabelled CGM data is a viable path, and that the representations built along the way carry more clinical information than the pre-training task explicitly demands.

Chapter 8

Execution Cronogram

This chapter presents the project planning, including the work breakdown structure, task precedence analysis, and Gantt and PERT diagrams for the full project timeline.

8.1 Work Breakdown Structure

The project is structured into five main phases: planning, data engineering, model pre-training, downstream evaluation, and thesis writing. A detailed dictionary describing each work package (including acceptance criteria, deliverables, resources, and estimated costs) is in Appendix D.

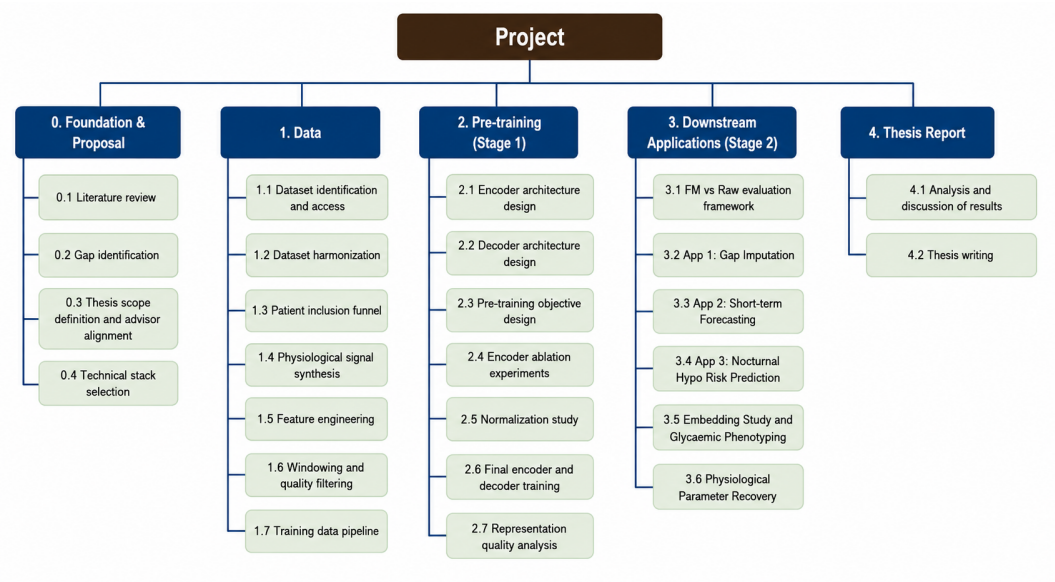

Figure 14. Work breakdown structure (WBS) for the project.

8.2 Milestone Plan

Table 14. Project milestones.

Milestone	Description	Completion
M1	Project scope and technical stack defined	End of week 2
M2	Dataset and data pipeline ready	End of week 5
M3	Foundation models trained	End of week 13
M4	All downstream tasks evaluated	End of week 17
M5	Thesis submitted	End of week 21

8.3 PERT and Gantt Diagrams

Table 15 lists all tasks with their durations and dependencies. Figures 15 and 16 show the resulting PERT network and Gantt chart.

Table 15. Precedence analysis of the tasks defined in the WBS. Durations are expressed in working days.

ID	WBS	Activity	Duration (days)	Predecessors
A	0.1	Literature review	7	—
B	0.2	Gap identification	3	A
C	0.3	Scope definition & advisor alignment	2	B
D	0.4	Technical stack selection	3	B
E	1.1	Dataset identification and access	3	C, D
F	1.2	Dataset harmonisation	3	E
G	1.3	Patient inclusion funnel	3	E
H	1.4	Physiological signal synthesis	3	F
I	1.5	Feature engineering	2	G
J	1.6	Windowing and quality filtering	3	H, I
K	1.7	Training data pipeline	2	J
L	2.1	Encoder architecture design	5	K
M	2.2	Decoder architecture design	3	K
N	2.3	Pre-training objective design	2	K
O	2.4	Encoder ablation experiments	12	L, N
P	2.5	Normalisation study	3	O
Q	2.6	Final encoder and decoder training	3	P, M
R	2.7	Representation quality analysis	3	Q
S	3.1	FM vs. raw evaluation framework	2	Q
T	3.2	App 1: Gap imputation	2	S
U	3.3	App 2: Short-term forecasting	5	S
V	3.4	App 3: Nocturnal hypo risk prediction	8	S
W	3.5	Embedding study & glycaemic phenotyping	3	R
X	3.6	Physiological parameter recovery	3	W
Y	4.1	Analysis and discussion of results	5	T, U, V, X
Z	4.2	Thesis writing	20	Y

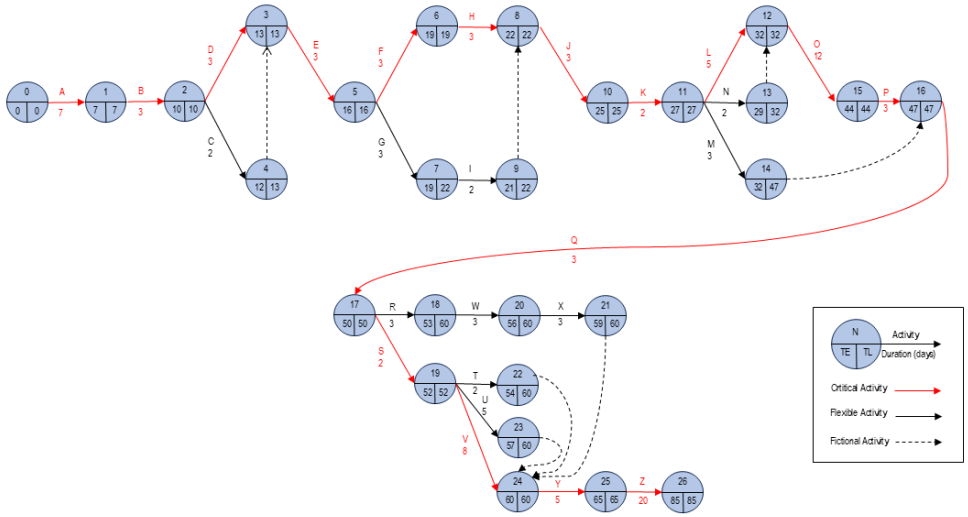

Figure 15. PERT network diagram. Critical path is highlighted.

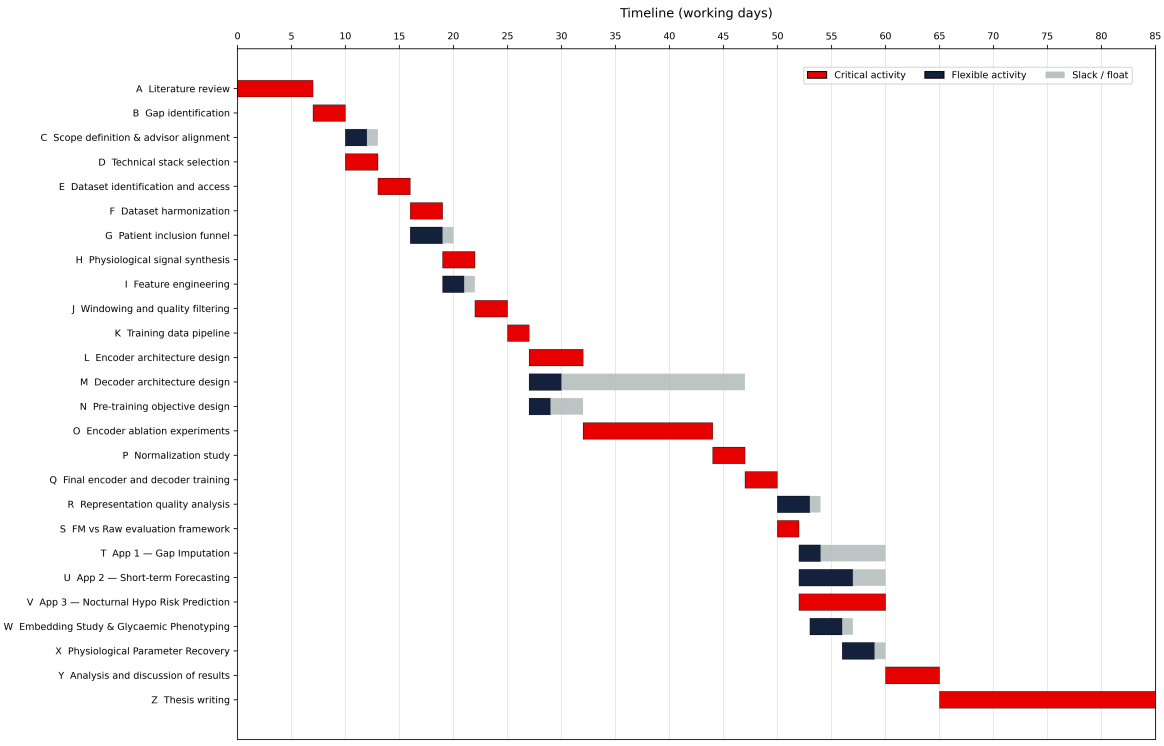

Figure 16. Gantt chart for the full project timeline (February – June 2026).

Chapter 9

Technical Feasibility

9.1 Infrastructure and Resources

The project was developed on a lab workstation provided by Micelab research group. The primary compute device is an NVIDIA GeForce RTX 5070 with 12 GB of VRAM, supported by an Intel Core i7-14700K processor (28 logical cores) and 16 GB of system RAM. Dataset storage and streaming are handled by a Samsung 990 Pro 1 TB NVMe SSD (PCIe 4.0), whose sequential read throughput mitigates the I/O bottleneck introduced by batched disk access. The operating system is Ubuntu 24.04 running under the Windows Subsystem for Linux 2 (WSL2, kernel 6.6).

The training environment is held in the NVIDIA NGC Docker image `nvcr.io/nvidia/tensorflow:24.02-tf2-py3`, which provides Python 3.12.3, TensorFlow 2.17.0, and CUDA 12.9 (driver 591.86). The use of this specific image was a hard requirement rather than a convenience: the RTX 5070 belongs to NVIDIA's Blackwell generation (CUDA Compute Capability 12.0), which the standard TensorFlow distribution does not support at the time of the project development.

Even though Deep Learning models training requires significant computational power, GPU was not the main limitation, system RAM was. With approximately one million training windows, the full dataset cannot reside in RAM simultaneously, so data was passed in batches from disk during both pre-training and downstream fine-tuning. The full pipeline (encoder pre-training on ~ 1 million windows followed by fine-tuning of multiple lightweight heads) therefore runs on a single GPU, confirming that the project is technically feasible within the resources available to a university research group.

9.2 SWOT Analysis

Table 16 summarises the project's position across the four SWOT dimensions.

Strengths	Weaknesses
• The only foundation model built exclusively on T1D data, filling a gap left by prior work trained on non-diabetic or mixed cohorts. • Novel research question with no direct prior work: first systematic BERT vs GPT comparison on T1D CGM data. • Physiological parameters recovered from unlabelled CGM without ever being explicitly supervised for them. • Knowledge transfer across three clinically distinct tasks without task-specific pre-training.	• Cohort size modest by foundation model standards. • Plasma insulin and carb absorption are ODE-derived, not directly measured; logging errors in insulin and meal records propagate into both signals. • No prospective or external validation; all results rest on a held-out split of the same study cohorts. • Cohort limited to adults with predominantly White demographics, introducing unquantified representativeness risk.
Opportunities	**Threats**
• Rapid CGM adoption generates growing volumes of unlabelled data, directly strengthening the pre-training regime over time. • The physiological profiling results point toward data-driven personalisation of closed-loop insulin delivery, a high-value clinical application. • Pre-trained backbone reusable for future clinical questions with few labels, amortising the training cost across many downstream tasks. • Regulatory frameworks for software as a medical device are maturing, providing a clearer path to clinical translation.	• Industry actors hold proprietary datasets orders of magnitude larger; better-resourced models may outperform at scale. • GDPR and dataset access restrictions limit cross-institutional data aggregation in an academic setting. • Algorithmic bias: a predominantly White training cohort may yield unequal performance across patient populations. • Rapid publication pace means results may be superseded before clinical translation.

Table 16. SWOT analysis of the thesis project.

Chapter 10

Economical Feasibility

This chapter presents an estimate of the costs associated with the execution of the project, covering hardware acquisition and personnel time.

10.1 Project Budget

The primary hardware cost is the workstation assembled for this project. Table 17 lists each component with its technical reference and purchase price.

Component	Technical Reference	Price (€)
CPU	Intel Core i7-14700K 3.4 GHz	385.00
Motherboard	MSI MAG Z790 TOMAHAWK WIFI	220.00
GPU	NVIDIA GeForce RTX 5070 12 GB	640.00
RAM	32 GB DDR5 (2×16 GB)	315.00
SSD	Samsung 990 PRO 1 TB NVMe PCIe 4.0	210.00
Power Supply	Corsair RM750e 750 W 80 Plus Gold Modular	115.00
Chassis	MSI MAG Series Silent	85.00
CPU Cooling	Nox Hummer H-224 Air Cooler	35.00
Monitor	Asus ProArt PA278CV 27 ” IPS	343.79
Keyboard/Mouse	Logitech Desktop MK120	18.93
Assembly	PC Componentes assembly service	37.18
Fees/Licences	Canon Digital (LPI) + Windows 11 Pro	25.00
Total		**2,430.90**

Table 17. Workstation hardware components and purchase prices.

Equipment depreciation

Equipment costs are not charged in full to the project; only the fraction corresponding to the project duration is attributed. A standard useful life of four years (48 months) is assumed for computer hardware, giving a monthly depreciation of:

$$D_{\text{month}} = \frac{2{,}430.90}{48} \approx 50.64\,€/\text{month}.$$

The project runs for 5 months (February–June 2026, as per Chapter 8), giving an attributable depreciation of:

$$D_{\text{project}} = 50.64 \times 5 = 253.20\,€.$$

Personnel

The only personnel cost is the researcher's time, valued at 8 €/hour (junior biomedical engineer rate). The full breakdown by project phase follows from the WBS Dictionary in Appendix D.

Phase	Description	Hours	Cost (€)
0	Planning	120	960.00
1	Data Engineering	152	1,216.00
2	Model Pre-training	248	1,984.00
3	Downstream Evaluation	184	1,472.00
4	Thesis Writing	200	1,600.00
Total		**904**	**7,232.00**

Table 18. Personnel cost by project phase at 8 €/hour.

Total project cost

Table 19 consolidates all cost items.

Item	Cost (€)
Hardware depreciation (5 months)	253.20
Personnel (904 h at 8 €/h)	7,232.00
Total	**7,485.20**

Table 19. Total estimated project cost.

Chapter 11

Regulatory Affairs

The models produced in this thesis are research artefacts; no clinical deployment is intended. This chapter identifies the regulatory frameworks that would apply to any future clinical translation and summarises how the data were handled under applicable data protection rules.

11.1 Software as a Medical Device

Under Regulation (EU) 2017/745 (MDR) [48], software intended for diagnosis, prognosis, or patient monitoring qualifies as Software as a Medical Device (SaMD). The models in this thesis make no diagnostic or therapeutic claim and are not intended for clinical use, so the regulation does not currently apply. Integration into a patient-facing decision support tool would likely qualify as Class IIa or Class IIb SaMD under Annex VIII, Rule 11, requiring a conformity assessment and CE marking.

11.2 The EU AI Act

Regulation (EU) 2024/1689 (the AI Act) [**eu_aiact2024**] establishes a risk-based framework for AI systems in the EU. Systems intended for clinical diagnosis, prognosis, or patient monitoring are classified as high-risk under Annex III, point 5(a), and are subject to risk management documentation, transparency obligations, data governance standards, and a conformity assessment before deployment. The models in this thesis are not placed on the market in the sense of the Act; any future clinical deployment would trigger these requirements.

11.3 Data Protection and Ethics

The datasets used (METABONET [9] and T1DEXI) contain health data, which is special-category data under Article 9 of the GDPR [49]. Both were collected under institutional review board approved protocols and distributed under defined data use agreements; all handling in this thesis was conducted within those agreements. No patient identifiers appear in this thesis, no new data was collected, and individual patients are not identifiable from any reported result.

Chapter 12

Conclusions

This thesis set out to build a T1D-specific glucose foundation model, compare two complementary transformer architectures, evaluate whether the representations transfer to clinical tasks, and probe what they actually encode. The results addressed all five objectives.

12.1 Objectives Revisited

Objective 1 and 2: Dataset and preprocessing. A cohort of 1,037 adult T1D patients was assembled from METABONET and T1DEXI, processed through a pipeline that derives continuous plasma insulin and carb absorption signals from dosing records using the Hovorka pharmacokinetic model, and split into 934 pre-training patients and 103 held-out test patients. The whole pipeline is documented in Chapters 4 and 5.

Objective 3: Pre-training. Both architectures were pre-trained on the 934-patient cohort. The main lesson from this stage is that the pre-training objective is not a neutral initialisation choice: it shapes what each model learns. The bidirectional encoder, forced to reconstruct glucose from full context, builds representations rich in event-level dynamics. The causal decoder, predicting one step at a time, builds a representation more tightly coupled to the recent trajectory of glucose. Each strategy is appropriate for different downstream needs, and the choice between them should be driven by what clinical information needs to be recoverable.

Objective 4: Downstream task evaluation. Across three clinical tasks, fine-tuned foundation model variants reach performance comparable to or better than LSTMs trained from scratch with full task labels. The model was never shown forecasting targets or hypoglycaemia labels during pre-training, yet it competes with task-specific baselines on all three. Knowledge transfer is demonstrated.

Objective 5: Representation structure and physiological content. The most surprising finding of the thesis. Frozen embeddings from both architectures encode insulin sensitivity and carb ratio better than a regression given the full mean 24-hour insulin and absorption profile of each patient, and neither model was given these labels at any point. Patient embeddings also cluster by glycaemic risk without supervision. The two architectures organise this information differently: the encoder spreads it across more independent dimensions, while the decoder's representation is more tightly anchored to glycaemic level. The difference reflects the objectives they were trained on.

12.2 Future Work

The limitations of Chapter 7 already mark the immediate next steps: alternative pre-training objectives, prospective validation in larger and more diverse cohorts, and interpretability of the learned representations.

One further direction was scoped out of this thesis but stands out as the most ambitious: the personalised digital twin. The representations here are discriminative; they read physiology off a window. A twin must instead generate it, simulating how a patient's glucose would respond to insulin and carbohydrate inputs that were never observed. Coupling the patient profile recovered by this model with a conditional generative model is, in my view, the most promising path from representation learning toward an actionable clinical tool.

12.3 Key Takeaways

Two things stand out from this work. The first is that the pre-training task is not just a technical trick to initialise weights: it genuinely shapes what the model learns. The encoder and decoder were given different tasks, and they came back with different representations, with different clinical content and different downstream strengths.

The second is the ISF and CR result. A linear probe on a 128-dimensional embedding, trained on glucose reconstruction, outperforms a regression given the full 24-hour insulin profile. That the model encoded insulin sensitivity and carb ratio without ever being told they were relevant is the result we would have found hardest to believe before running the experiments. It suggests that the transformer is not memorising glucose trajectories but learning the patient-specific physiology.

None of this is ready for clinical use; the classifiers need better performance before anyone should trust them near a patient. What this thesis offers is a proof of concept: that learning glucose dynamics from unlabelled CGM data is a viable path, that the representations carry more clinical information than the pre-training task explicitly demands, and that the BERT versus GPT question, which nobody had systematically asked for T1D data, has an interpretable answer.

I believe that is a solid starting point.

References

[1] Pouya Saeedi, Inga Petersohn, Paraskevi Salpea, et al. “Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045”. In: *Diabetes Research and Clinical Practice* 157 (2019), p. 107843.

[2] Mark A Atkinson, George S Eisenbarth, and Aaron W Michels. “Type 1 diabetes”. In: *The Lancet* 383.9911 (2014), pp. 69–82.

[3] Cornelis J. Tack, Gerardus J. Lancee, Barend Heeren, et al. “Glucose Control, Disease Burden, and Educational Gaps in People With Type 1 Diabetes: Exploratory Study of an Integrated Mobile Diabetes App”. In: *JMIR Diabetes* 3 (2018), e17.

[4] Omer Mujahid et al. “Generative deep learning for the development of a type 1 diabetes simulator”. In: *Communications Medicine* 4.1 (2024), p. 51.

[5] Rishi Bommasani et al. “On the opportunities and risks of foundation models”. In: *arXiv preprint arXiv:2108.07258* (2021).

[6] Guy Lutsker, Gal Sapir, Smadar Shilo, et al. “A foundation model for continuous glucose monitoring data”. In: *Nature* 650 (2026), pp. 978–986.

[7] Yurun Lu, Dan Liu, Zhongming Liang, et al. “A pretrained transformer model for decoding individual glucose dynamics from continuous glucose monitoring data”. In: *National Science Review* 12.5 (2025), nwaf039.

[8] Jian Luo, Abhinav Kumbara, Mehdi Shomali, et al. “A large sensor foundation model pretrained on continuous glucose monitor data for diabetes management”. In: *npj Health Systems* 2 (2025), p. 35.

[9] Miriam K Wolff, Peter Calhoun, Eleonora Maria Aiello, Yao Qin, and Sam F Royston. *MetaboNet: The largest publicly available consolidated dataset for type 1 diabetes management*. 2026.

[10] Michael C. Riddell, Zoey Li, Robin L. Gal, et al. “Examining the Acute Glycemic Effects of Different Types of Structured Exercise Sessions in Type 1 Diabetes in a Real-World Setting: The Type 1 Diabetes and Exercise Initiative (T1DEXI)”. In: *Diabetes Care* 46.4 (2023), pp. 704–713.

[11] MICELab. *Modeling, Identification and Control Engineering Laboratory*. https://micelab.udg.edu/. Universitat de Girona, 2024.

[12] Tadej Battelino, Thomas Danne, Richard M Bergenstal, et al. “Clinical Targets for Continuous Glucose Monitoring Data Interpretation: Recommendations From the International Consensus on Time in Range”. In: *Diabetes Care* 42.8 (2019), pp. 1593–1603.

[13] The Diabetes Control and Complications Trial Research Group. “The Effect of Intensive Treatment of Diabetes on the Development and Progression of Long-Term Complications in Insulin-Dependent Diabetes Mellitus”. In: *New England Journal of Medicine* 329.14 (1993), pp. 977–986.

[14] David C. Klonoff, Jing Wang, David Rodbard, et al. “A Glycemia Risk Index (GRI) of Hypoglycemia and Hyperglycemia for Continuous Glucose Monitoring Validated by Clinician Ratings”. In: *J Diabetes Sci Technol* 17.5 (2023), pp. 1226–1242.

[15] Qin Yang, Baoqi Zeng, Jiayi Hao, Qingqing Yang, and Feng Sun. “Real-world glycaemic outcomes of automated insulin delivery in type 1 diabetes: A meta-analysis”. In: *Diabetes, Obesity and Metabolism* 26 (2024), pp. 3753–3763.

[16] Thomas Danne, Revital Nimri, Tadej Battelino, et al. “International Consensus on Use of Continuous Glucose Monitoring”. In: *Diabetes Care* 40.12 (2017), pp. 1631–1640.

[17] Michael I Schmidt, Anastasios Hadji-Georgopoulos, Marc Rendell, Simeon Margolis, and Arlan Kowarski. “The dawn phenomenon, an early morning glucose rise: implications for diabetic intraday blood glucose variation”. In: *Diabetes Care* 4.6 (1981), pp. 579–585.

[18] Eleonora Poggiogalle, Humaira Jamshed, and Courtney M Peterson. “Circadian regulation of glucose, lipid, and energy metabolism in humans”. In: *Metabolism* 84 (2018), pp. 11–27.

[19] Philip E Cryer. “Hypoglycemia during therapy of diabetes”. In: *New England Journal of Medicine* 369.4 (2013), pp. 362–372.

[20] Richard N Bergman, Y Ziya Ider, Charles R Bowden, and Claudio Cobelli. “Quantitative estimation of insulin sensitivity”. In: *American Journal of Physiology – Endocrinology and Metabolism* 236.6 (1979), E667–E677.

[21] Roman Hovorka, Valentina Canonico, Ludovic J Chassin, et al. “Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes”. In: *Physiological Measurement* 25.4 (2004), pp. 905–920.

[22] Silvia Oviedo, Josep Vehí, Remei Calm, and Joaquim Armengol. “A review of personalized blood glucose prediction strategies for T1DM patients”. In: *International Journal for Numerical Methods in Biomedical Engineering* 33.6 (2017), e2833.

[23] Josep Vehí, Iván Contreras, Silvia Oviedo, Lyvia Biagi, and Arthur Bertachi. “Prediction and prevention of hypoglycaemic events in type-1 diabetic patients using machine learning”. In: *Health Informatics Journal* 26.1 (2020), pp. 703–718.

[24] Sadegh Mirshekarian, Hui Shen, Razvan Bunescu, and Cindy Marling. “LSTMs and Neural Attention Models for Blood Glucose Prediction: Comparative Experiments on Real and Synthetic Data”. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2019, pp. 706–712.

[25] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling”. In: *arXiv preprint arXiv:1803.01271* (2018).

[26] Omer Mujahid, Iván Contreras, Aleix Beneyto, Ignacio Conget, Marga Giménez, and Josep Vehí. “Conditional Synthesis of Blood Glucose Profiles for T1D Patients Using Deep Generative Models”. In: *Mathematics* 10.20 (2022), p. 3741.

[27] Heman Shakeri. “The Driver-Blindness Phenomenon: Why Deep Sequence Models Default to Autocorrelation in Blood Glucose Forecasting”. In: *arXiv preprint arXiv:2511.20601* (2025).

[28] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[29] Sam Buchanan, Druv Pai, Peng Wang, and Yi Ma. *Principles and Practice of Deep Representation Learning*. https://ma-lab-berkeley.github.io/deep-representation-learning-book/. Online, Aug. 2025.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 2019, pp. 4171–4186.

[31] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. “MOMENT: A Family of Open Time-series Foundation Models”. In: *Proceedings of the 41st International Conference on Machine Learning (ICML)*. 2024.

[32] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. “SimMTM: A Simple Pre-Training Framework for Masked Time-Series Modeling”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 36. 2023.

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. “Language Models are Unsupervised Multitask Learners”. In: *OpenAI Blog* 1.8 (2019), p. 9.

[34] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. “A Decoder-Only Foundation Model for Time-Series Forecasting”. In: *Proceedings of the 41st International Conference on Machine Learning (ICML)*. 2024.

[35] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, et al. “Chronos: Learning the Language of Time Series”. In: *Proceedings of the 41st International Conference on Machine Learning (ICML)*. 2024.

[36] John Wright and Yi Ma. *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press, 2022.

[37] Charlotte K Boughton and Roman Hovorka. “New closed-loop insulin systems”. In: *Diabetologia* 64.5 (2021), pp. 1007–1015.

[38] Grand View Research. *Continuous Glucose Monitoring Devices Market Size, Share & Trends Analysis Report By Product (Standalone, Integrated), By Connectivity, By Indication, By Distribution Channel, By Region, And Segment Forecasts, 2025 - 2033*. Market Analysis Report GVR-4613458. Accessed: 2026-04-13. Grand View Research, Inc., 2026.

[39] American Diabetes Association Professional Practice Committee. "Standards of Care in Diabetes—2024". In: *Diabetes Care* 47.Supplement_1 (2024), S1–S321.

[40] Rayan Krishnan, Pranav Rajpurkar, and Eric J. Topol. "Self-supervised learning in medicine and healthcare". In: *Nature Biomedical Engineering* 6 (2022), pp. 1346–1352.

[41] Mahmoud Assran, Quentin Duval, Ishan Misra, et al. "Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 15619–15629.

[42] replicahealth. *metabonet_processor: Parsing pipeline for the MetaboNet dataset*. 2024.

[43] Jane L Chiang, David M Maahs, Katharine C Garvey, et al. "Type 1 diabetes in children and adolescents: a position statement by the American Diabetes Association". In: *Diabetes Care* 41.9 (2018), pp. 2026–2044.

[44] Jia Zhu, Lisa K Volkening, and Lori M Laffel. "Distinct patterns of daily glucose variability by pubertal status in youth with type 1 diabetes". In: *Diabetes Care* 43.1 (2020), pp. 22–29.

[45] Shin-Hye Kim and Mi-Jung Park. "Effects of growth hormone on glucose metabolism and insulin resistance in human". In: *Annals of Pediatric Endocrinology & Metabolism* 22.3 (2017), pp. 145–152.

[46] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, et al. "A Structured Self-Attentive Sentence Embedding". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2017.

[47] Leland McInnes, John Healy, and James Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[48] European Parliament and Council. *Regulation (EU) 2017/745 on medical devices*. Tech. rep. Official Journal of the European Union, 2017.

[49] European Parliament and Council. *Regulation (EU) 2016/679 — General Data Protection Regulation*. Tech. rep. Official Journal of the European Union, 2016.

Appendix A

Feature Importance Analysis

This appendix documents the experiments used to justify the 7-feature input set described in Section 4.2. The central question is: how much does each input feature contribute to the model's ability to reconstruct missing glucose readings?

Three complementary methods are applied, each answering a different aspect of this question. *Zero-out ablation* measures how much performance drops when a feature is removed. *Gradient saliency* measures how sensitive the model's output is to small changes in each feature. *Principal component analysis* examines whether the features are reflected in the structure of the learned patient-level representations.

All analyses use the imputation task (gap filling without any task-specific fine-tuning) as the main task, evaluated on the 103-patient held-out test set. The baseline model includes all 7 active features (CGM, PI, RA, hour_sin, hour_cos, bolus_logged, carbs_logged).

A.1 Zero-out Ablation

Method

The most direct way to measure how much the model depends on a feature is to remove it and observe what happens to performance. In a zero-out ablation, one feature group is replaced by zeros at all timesteps while all other features remain unchanged. The model is not retrained; the change is applied at inference time only. The metric is $\Delta R^2 = R^2_{\text{ablated}} - R^2_{\text{full}}$, evaluated over 2,000 test windows per gap length. A large negative ΔR^2 means the model relies heavily on that feature; a value near zero means removing it has no measurable effect.

Results

Figure 17 shows the R^2 drop for four feature groups across four gap lengths (4h, 5h, 6h, 8h). Table 20 gives the corresponding numerical values.

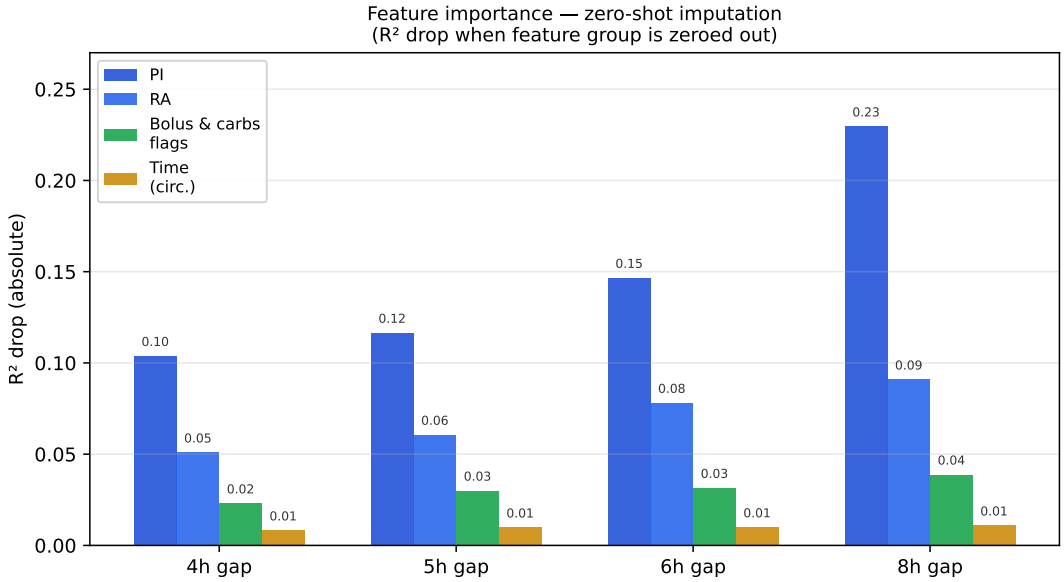

Figure 17. R² drop on zero-shot imputation when each feature group is zeroed. A larger bar means the model depends more on that feature. PI dominates at all gap lengths; the effect grows substantially with gap duration.

Table 20. ΔR^2 from zero-out ablation by feature group and gap length. Values are negative (a drop from the full-model baseline).

Feature group	4h	5h	6h	8h
PI (col 1)	-0.104	-0.117	-0.147	-0.230
RA (col 2)	-0.051	-0.061	-0.078	-0.091
Bolus & carbs flags (cols 5–6)	-0.023	-0.030	-0.031	-0.039
Time encoding (cols 3–4)	-0.008	-0.010	-0.010	-0.011

PI is the dominant feature at every gap length. Its cost grows with gap duration (from -0.10 at 4h to -0.23 at 8h), which reflects the long pharmacological action of plasma insulin: after a bolus, PI remains elevated for several hours, so even an 8-hour gap is still influenced by insulin activity that preceded it. RA shows a similar trend but roughly half the magnitude, consistent with the shorter timescale of gut carbohydrate absorption (a meal is typically absorbed within 2–3 hours). Binary flags and time encoding contribute modestly in aggregate but show concentrated effects under stratification, as described below and in Sections 4.2.2 and 4.2.3.

Time-stratified ablation for circadian features

The small global cost of removing the cyclic time encoding ($\Delta R^2 \approx -0.008$ at 4h) can be misleading: it averages across all times of day, diluting periods where time information matters most. Figure 18 breaks down the R^2 drop by the time of day at the centre of the gap, for 4h and 6h gaps.

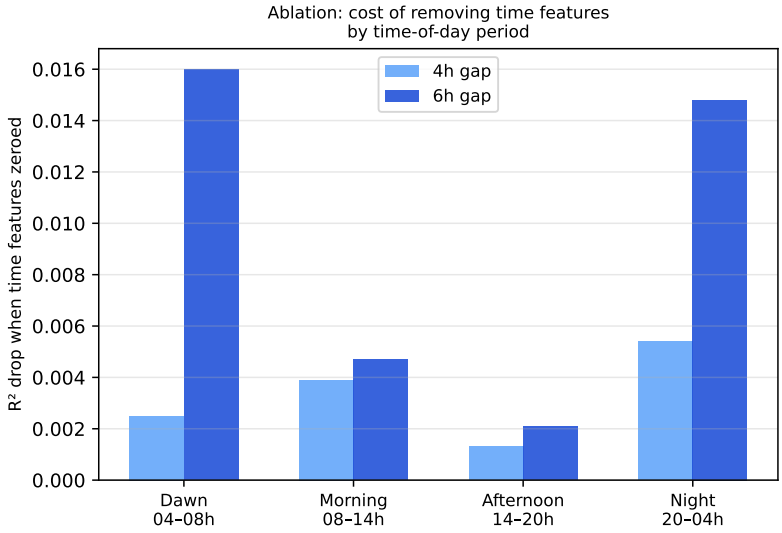

Figure 18. R^2 drop when cyclic time features are zeroed, stratified by time-of-day period. Dawn (04:00–08:00) and Night (20:00–04:00) show roughly 8× larger drops at the 6h gap than Afternoon (14:00–20:00), consistent with the dawn phenomenon and nocturnal metabolic quiescence.

Dawn and Night windows show a 6h-gap R^2 drop of 0.016 and 0.015 respectively, compared to 0.002 for Afternoon. These are the periods where glucose trajectory is most predictably shaped by time of day independent of driver signals: the dawn phenomenon produces a consistent early-morning glucose rise driven by counter-regulatory hormones rather than insulin or meals, and overnight quiescence means the driver signals carry little event information.

A.2 Gradient Saliency

Method

Ablation measures what happens when a feature is completely removed. Gradient saliency asks a complementary question: how sensitive is the model's output to small changes in each input channel? Rather than zeroing a feature entirely, gradients measure the rate at which the output would change if a feature were perturbed by a tiny amount. This provides a finer picture of which features the model is actively using at each timestep.

Formally, for a window with mask **m** and input tensor **W**, the saliency score for input channel j is:

$$s_j = \left\langle \left| \frac{\partial \sum_t m_t \hat{y}_t}{\partial W_{:,j}} \right| \right\rangle_{\text{windows}} \tag{A.1}$$

where the average is over all masked timesteps and all test windows. Gradients are computed using `tf.GradientTape` on the imputation head without any parameter update.

Results

Figure 19 shows the mean $|\partial \text{output} / \partial \text{input}|$ per channel for all four gap lengths.

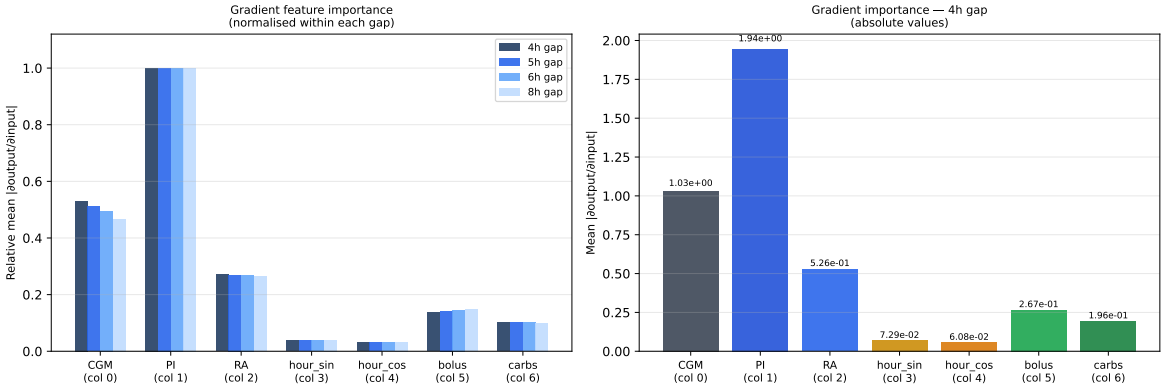

Figure 19. Mean absolute gradient of imputed CGM with respect to each input channel, averaged over masked timesteps and test windows. Left: relative importance (normalised to PI=1.0 within each gap). Right: absolute values for the 4h gap. PI shows the highest gradient by a wide margin; time features (hour_sin, hour_cos) show the lowest.

The ranking is consistent across all gap lengths: PI $>$ CGM $>$ RA $>$ bolus $\approx$ carbs $>$ hour_sin $\approx$ hour_cos. This ordering agrees with the ablation results. Relative to PI (normalised to 1.0), CGM scores 0.53, RA 0.27, the binary flags 0.10–0.14, and the cyclic time features 0.03–0.04.

One result worth explaining: CGM itself has a substantial gradient score (0.53 relative to PI), even though CGM is the variable being predicted. This is not circular. In windows where the gap is surrounded by visible glucose readings, the model uses those nearby values as context when reconstructing the missing segment. The gradient captures this local reliance on surrounding CGM.

A.3 Encoder Representation Structure

Method

The previous two analyses ask how much each feature contributes to imputation accuracy. This final analysis asks a different question: has the encoder organised its internal representation around the input features? If a feature is important to the model, it should leave a trace in the structure of the learned embeddings.

To test this, the encoder's window-level summaries ($\mathbf{h}_{\text{cls}}$, the 128-dimensional CLS vector described in Appendix B.2) are averaged across all windows for each patient, producing one 128-dimensional profile per patient. Principal Component Analysis (PCA) is then applied to these 1,037 patient profiles to find the main axes of variation. For each of the top principal components (PCs), Spearman correlation ρ is computed against six per-patient summaries of the input features: mean CGM, mean PI, mean RA, night/dawn fraction, bolus rate, and carb rate. If a PC correlates strongly with a feature summary, it means the encoder has learned to vary its representation along that feature dimension.

Results

Eight PCs explain 90% of the variance in the patient-level embeddings (40.7%, 19.2%, 12.4%, 9.1%, 3.8%, 2.5%, 1.7%, 1.7%). Figure 20 shows the Spearman ρ between each PC and each per-patient feature summary; greyed cells are not significant ($p > 0.05$).

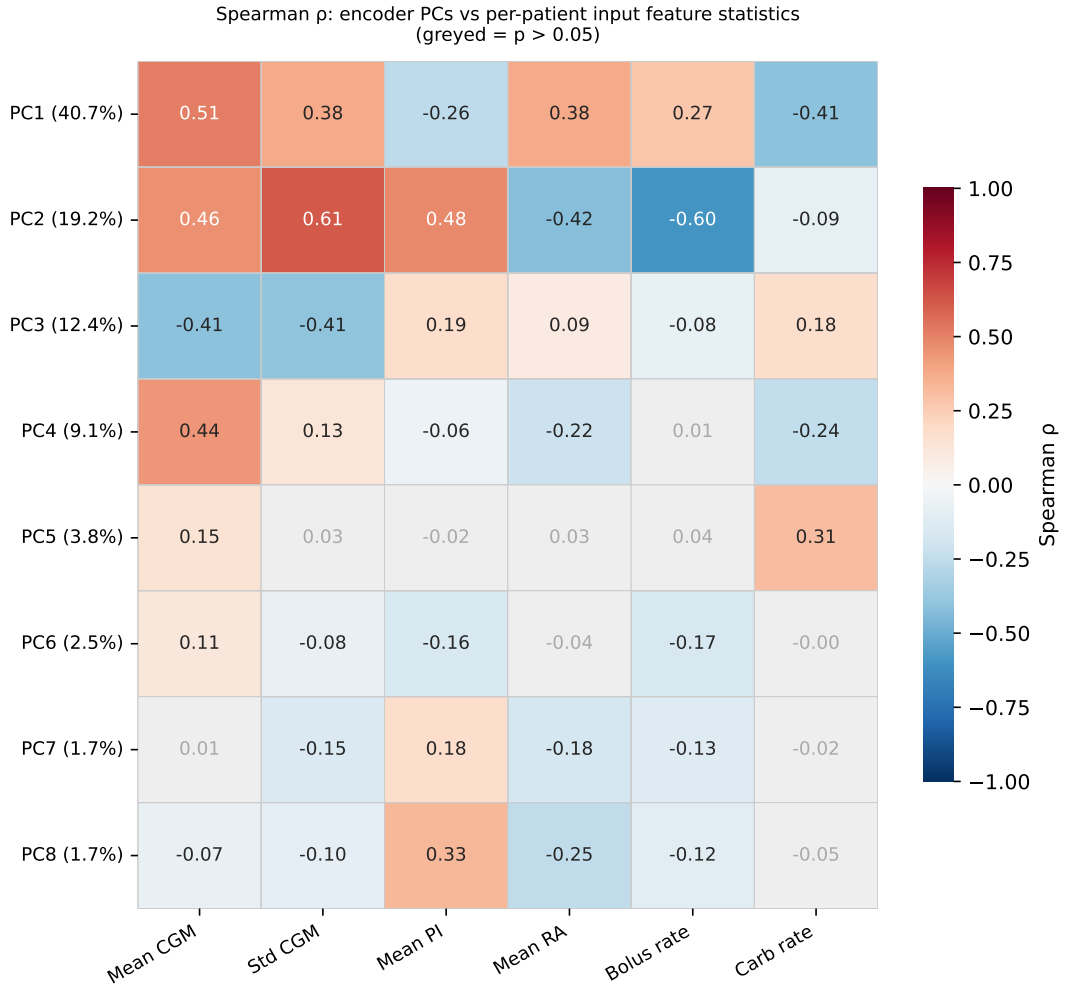

Figure 20. Spearman ρ between the top PCs of the 1,037 patient-level encoder embeddings and per-patient summaries of the 7 input features. Greyed cells: $p > 0.05$. The first two PCs align cleanly with glucose level and glycaemic variability axes, with strong contributions from the continuous physiological signals (CGM, PI, RA).

The first two PCs capture 60% of the variance and align with interpretable clinical axes:

- **PC1** (40.7%): correlates with mean CGM ($\rho = 0.51$), mean RA ($\rho = 0.38$), and carb rate ($\rho = -0.41$). This is a *glucose level* axis: patients with higher average glucose and more carbohydrate absorption score high; patients with lower glucose and less meal-driven variability score low.
- **PC2** (19.2%): correlates with mean PI ($\rho = 0.48$) and bolus rate ($\rho = -0.60$). This is an *insulin delivery pattern* axis. A high score means high continuous insulin action but few discrete

bolus events, the signature of AID therapy; a low score means infrequent but large boluses, typical of MDI.

Appendix B

Transformer Representation Objects

The Transformer models in this thesis produce three distinct outputs, each suited to a different type of downstream task. This appendix defines what each output contains and how it is constructed from the architecture introduced in Section 2.3.4.

B.1 The Sequence Representation H

Some tasks need to make a prediction at every timestep: imputation must reconstruct each missing glucose reading, and forecasting must generate each future value. These tasks require a representation that preserves the full temporal resolution of the input.

The sequence representation $\mathbf{H}$ is exactly this. A Transformer processes the input through L stacked layers. At each layer, every position computes attention over other positions: it projects its own state into query, key, and value vectors, computes a softmax-normalised compatibility score against every other position's key, and aggregates the corresponding values weighted by those scores. This operation is repeated across multiple attention heads in parallel, each head learning to attend to a different aspect of the sequence (Section 2.3.4). After all L layers, the model produces one output vector per timestep, each of dimension d_{model}. Stacking these as rows gives:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{T-1} \end{bmatrix} \in \mathbb{R}^{T \times d_{\text{model}}}. \tag{B.1}$$

The key point is that each row $\mathbf{h}_t$ is not simply the input at timestep t passed through a projection. At each of the L layers, $\mathbf{h}_t$ is updated by attending to other positions, and the result is added back to the previous value through a residual connection and stabilised by layer normalisation. By the time it exits the final layer, $\mathbf{h}_t$ has been refined through L rounds of this attention-residual-normalisation cycle, accumulating context from progressively more distant positions.

In this thesis, $\mathbf{H} \in \mathbb{R}^{288 \times 128}$: 288 timesteps (one full day at 5-minute resolution) and 128 dimensions per timestep. $\mathbf{H}$ is used directly for imputation and forecasting. Tasks that require a single summary of the entire window (such as risk classification or patient profiling) use $\mathbf{h}_{\text{cls}}$ or $\mathbf{h}_{\text{last}}$, described below.

B.2 The CLS Token and $\mathbf{h}_{\text{cls}}$

Some tasks need a single vector that summarises the entire 24-hour window: for example, estimating overnight hypoglycaemia risk or profiling a patient's insulin sensitivity. The simplest approach would be to average all 288 rows of **H**, but averaging treats every timestep equally, whether it is clinically informative or not.

The CLS token, introduced by Devlin et al. [30], is a more flexible alternative. A special learnable vector is added to the beginning of the input sequence before the first Transformer layer, extending the sequence from T to $T + 1$ positions. This extra position has no corresponding input timestep; its only role is to collect information. Because the bidirectional encoder applies no causal mask, the CLS position attends to all T input timesteps at every layer: its query vector computes attention scores against every key in the sequence, and the resulting softmax-weighted sum of values flows into the CLS output. After L layers, this output $\mathbf{h}_{\text{cls}} \in \mathbb{R}^{d_{\text{model}}}$ contains a learned summary of the full window.

The advantage over simple averaging is that the attention weights are learned, not uniform. Through the multi-head attention mechanism, the CLS token can allocate different heads to different aspects of the window: one head might attend to post-meal glucose excursions, another to the period around a bolus, another to the overnight baseline. This selective weighting emerges from the pre-training objective, not from hand-designed rules.

In the original BERT model, the CLS token is explicitly supervised during pre-training through a next-sentence prediction task. In this thesis, no such supervision is applied: $\mathbf{h}_{\text{cls}}$ emerges as a natural byproduct of the shared Transformer layers.

B.3 The Causal Last Position and $\mathbf{h}_{\text{last}}$

The CLS token relies on bidirectional attention: the summary position can look at the entire window at once. The causal decoder does not have this option. The triangular attention mask sets all entries above the diagonal to $-\infty$ before the softmax, so position t can only compute attention scores against positions $0, 1, \ldots, t$. This means that most positions in the sequence have only a partial view of the window. The exception is the very last position, $t = T - 1$: since it comes after every other timestep, the mask leaves its entire row unmasked, allowing it to attend to all preceding positions. After L layers of causal self-attention:

$$\mathbf{h}_{\text{last}} = \mathbf{h}_{T-1} \in \mathbb{R}^{d_{\text{model}}} \tag{B.2}$$

This vector is the decoder's equivalent of $\mathbf{h}_{\text{cls}}$: a single summary of the entire window, used by the same downstream tasks (risk classification, patient profiling).

Appendix C

Extended Exploratory Data Analysis

This appendix complements the EDA in Section 5.2 with two additional characterisations of the final 1,037-patient cohort: glycaemic control and training data volume.

C.1 Glycaemic Profile

Figure 21 shows two per-patient statistics computed across all retained windows. Mean CGM is centred near 144 mg dL^{-1}, close to the population normalisation anchor, with a right tail of chronically hyperglycaemic patients. Time-in-range (TIR, 70–180 mg dL^{-1}) has a median of 77%, above the 70% clinical guideline target. The spread is wide in both panels, covering patients from near-normal control to persistent hyperglycaemia.

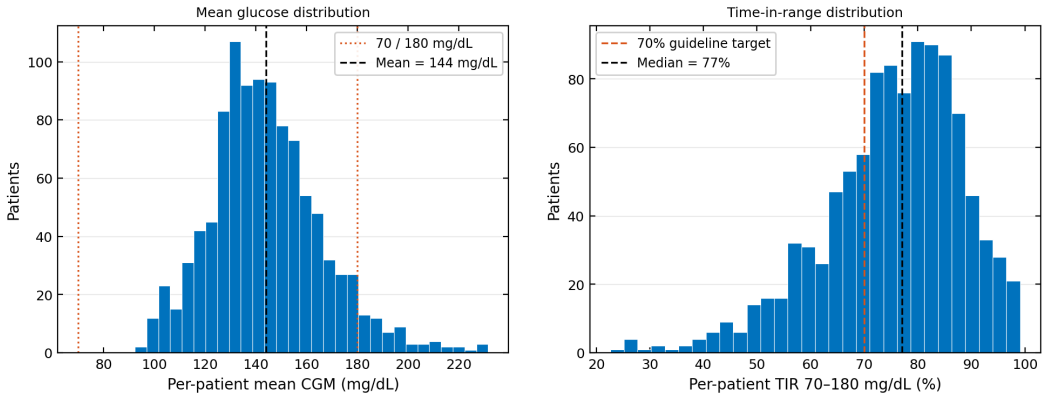

Figure 21. Per-patient glycaemic profile of the 1,037-patient cohort. Left: distribution of mean CGM, dashed line at cohort mean (144 mg dL^{-1}). Right: time-in-range (TIR 70–180 mg dL^{-1}), dashed line at the 70% clinical guideline target.

C.2 Cohort Data Quality

Figure 22 characterises the training data volume and glycaemic variability of the final cohort. The median patient contributes 678 training windows (approximately 169 days of monitoring at a 6-hour stride), though the distribution is right-skewed: a subset of long-term participants provides several thousand windows each. CGM standard deviation peaks near 50 mg dL^{-1}.

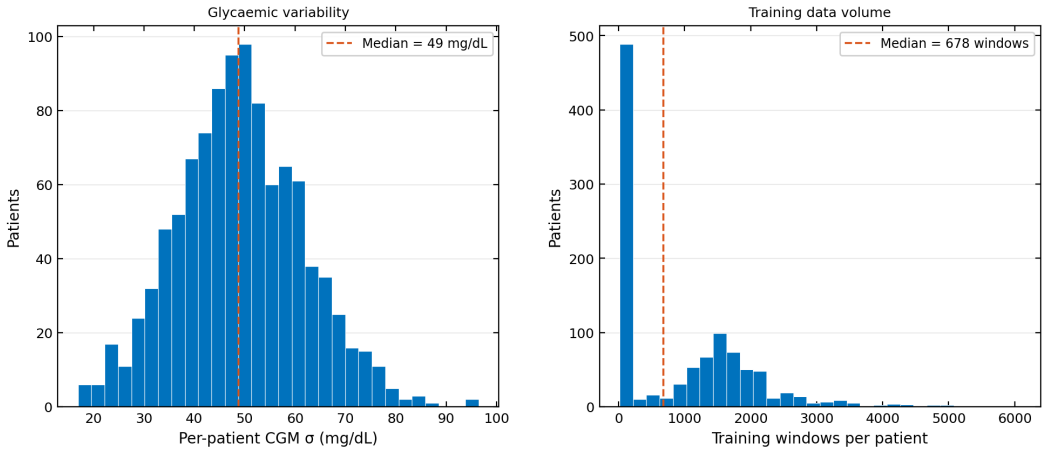

Figure 22. Training data characterisation of the 1,037-patient cohort. Left: per-patient CGM standard deviation. Right: number of 24-hour training windows per patient (288-step windows, 6-hour stride).

Appendix D

WBS Dictionary

This appendix provides a detailed description of each work package in the WBS (Section 8.1). Each card lists the name, WBS code, description, acceptance criterion, deliverables, resources, estimated cost, limit date, and responsible person. Rate: 8 €/hour. Responsible and approver throughout: Marc Font.

Literature Review	Preliminary project
Code in WBS: 0.1	
Description	
Survey of existing CGM foundation models (GluFormer, CGMformer), transformer architectures for time series, and T1D clinical task literature.	
Acceptance criterion	
Annotated bibliography of $\geq$20 papers; research gaps identified with supporting evidence.	
Approved by: Omer Mujahid	
Deliverables	
Literature review notes; annotated bibliography.	
Resources	
Researcher time; scientific database access.	
Estimated cost	448 €
Estimated cost (h)	56 h
Limit date	Feb 10, 2026
Responsible	Marc Font

Gap Identification	Preliminary project
Code in WBS: 0.2	
Description	
Identify specific research gaps: no BERT vs GPT comparison on T1D CGM data; no ISF/CR recovery from embeddings; no short-horizon physiological task evaluation.	
Acceptance criterion	
Gap list with supporting evidence from literature; each gap linked to at least one missing reference.	
Approved by: Omer Mujahid	
Deliverables	
Gap analysis document.	
Resources	
Researcher time.	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Feb 13, 2026
Responsible	Marc Font

Scope Definition	Preliminary project
Code in WBS: 0.3	
Description	
Define thesis objectives, pre-training architectures, downstream tasks, and evaluation strategy. Align scope and methodology with supervisor.	
Acceptance criterion	
Thesis proposal with objectives confirmed by supervisor; evaluation plan approved.	
Approved by: Omer Mujahid	
Deliverables	
Approved thesis proposal document.	
Resources	
Researcher time; supervisor meetings.	
Estimated cost	128 €
Estimated cost (h)	16 h
Limit date	Feb 17, 2026
Responsible	Marc Font

Technical Stack Selection	Preliminary project
Code in WBS: 0.4	
Description	
Select and configure the development environment: TensorFlow 2.17, RTX 5070 GPU (CUDA 12.9), Docker container, and data management approach.	
Acceptance criterion	
Functional GPU environment with verified package versions and adequate training throughput.	
Approved by: Omer Mujahid	
Deliverables	
Configured development environment; technical stack document.	
Resources	
Researcher time; computing hardware.	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Feb 18, 2026
Responsible	Marc Font

Dataset Identification & Access	Preliminary project
Code in WBS: 1.1	
Description	
Identify and access METABONET and T1DEXI datasets of adult T1D patients. Verify data availability, format, and temporal coverage.	
Acceptance criterion	
Both datasets successfully downloaded and parsed; raw file inventory confirmed.	
Approved by: Omer Mujahid	
Deliverables	
Raw dataset files; data access documentation.	
Resources	
Researcher time; institutional data access agreements.	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Feb 21, 2026
Responsible	Marc Font

Dataset Harmonisation	Preliminary project
Code in WBS: 1.2	
Description	
Harmonise METABONET and T1DEXI into a unified schema: CGM timestamps, insulin dose records, meal records, and patient metadata with consistent naming.	
Acceptance criterion	
Unified patient records with consistent timestamp format and variable naming across both sources; no systematic misalignments.	
Approved by: Omer Mujahid	
Deliverables	
Harmonised patient data files; preprocessing code.	
Resources	
Researcher time; Python.	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Feb 26, 2026
Responsible	Marc Font

Patient Inclusion Funnel	Preliminary project
Code in WBS: 1.3	
Description	
Apply inclusion criteria (age $\geq$18, data quality thresholds, minimum CGM recording duration) and document patient counts at each filtering step.	
Acceptance criterion	
Final cohort of $\geq$1,000 adult T1D patients; inclusion funnel fully documented.	
Approved by: Omer Mujahid	
Deliverables	
Inclusion funnel report; final patient list (1,037 patients).	
Resources	
Researcher time; Python scripts.	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Feb 26, 2026
Responsible	Marc Font

Physiological Signal Synthesis	Preliminary project
Code in WBS: 1.4	
Description	
Implement the Hovorka ODE model to derive plasma insulin (PI) and carbohydrate absorption rate (RA) at 5-minute resolution from logged insulin doses and meal records.	
Acceptance criterion	
PI and RA signals generated for all patients; pharmacokinetic profiles pass sanity checks.	
Approved by: Omer Mujahid	
Deliverables	
Per-patient PI and RA arrays; ODE implementation code.	
Resources	
Researcher time; Python (SciPy ODE solver).	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Mar 3, 2026
Responsible	Marc Font

Feature Engineering	Preliminary project
Code in WBS: 1.5	
Description	
Construct the 7-feature input tensor: CGM, PI, RA (z-scored), hour sin/cos (circadian encoding), bolus logged and carbs logged (binary event flags).	
Acceptance criterion	
Per-patient feature tensors generated; each feature confirmed to contribute positively to reconstruction performance.	
Approved by: Omer Mujahid	
Deliverables	
Per-patient feature matrices; feature engineering code.	
Resources	
Researcher time; Python.	
Estimated cost	128 €
Estimated cost (h)	16 h
Limit date	Feb 28, 2026
Responsible	Marc Font

Windowing & Quality Filtering	Preliminary project
Code in WBS: 1.6	
Description	
Segment continuous data into 24h windows (288 steps at 5-min resolution, stride 72 steps). Apply step-change filter to remove sensor artefact windows.	
Acceptance criterion	
Final windowed dataset with documented acceptance rate ($\geq$90%); per-patient `.npz` files generated.	
Approved by: Omer Mujahid	
Deliverables	
Per-patient windowed `.npz` files; filtering statistics report.	
Resources	
Researcher time; Python; disk storage.	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Mar 6, 2026
Responsible	Marc Font

Training Data Pipeline	Preliminary project
Code in WBS: 1.7	
Description	
Build `tf.data` pipeline for efficient batched GPU loading. Implement 831/103/103 patient split (train/val/test) with no data leakage.	
Acceptance criterion	
Pipeline loads all splits without data leakage; throughput adequate for pre-training; split files saved.	
Approved by: Omer Mujahid	
Deliverables	
`tf.data` pipeline code; patient split files.	
Resources	
Researcher time; Python/TensorFlow.	
Estimated cost	128 €
Estimated cost (h)	16 h
Limit date	Mar 10, 2026
Responsible	Marc Font

Encoder Architecture Design	Preliminary project
Code in WBS: 2.1	
Description	
Design BERT-style bidirectional Transformer encoder: 5 layers, 4 attention heads, $d_{\text{model}} = 128$, $d_{\text{ff}} = 256$, CLS token, sinusoidal PE, driver-weighted masking loss ($\sim$641K parameters).	
Acceptance criterion	
Model runs forward pass on sample batch without errors; architecture matches all design specifications.	
Approved by: Omer Mujahid	
Deliverables	
Encoder model code; architecture documentation.	
Resources	
Researcher time; TensorFlow; GPU.	
Estimated cost	320 €
Estimated cost (h)	40 h
Limit date	Mar 17, 2026
Responsible	Marc Font

Decoder Architecture Design	Preliminary project
Code in WBS: 2.2	
Description	
Design GPT-style causal Transformer decoder with identical hyperparameters to the encoder but with a triangular attention mask and a next-token prediction head.	
Acceptance criterion	
Causal mask verified at inference time; forward pass produces valid next-step predictions.	
Approved by: Omer Mujahid	
Deliverables	
Decoder model code; architecture documentation.	
Resources	
Researcher time; TensorFlow; GPU.	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Mar 13, 2026
Responsible	Marc Font

Pre-training Objective Design	Preliminary project
Code in WBS: 2.3	
Description	
Define masked span reconstruction (encoder) and next-token prediction (decoder) objectives. Implement cross-entropy loss over 200-bin CGM discretisation in [40, 400] mg/dL.	
Acceptance criterion	
Both objectives produce decreasing training loss in first 5 epochs; bin range validated against full CGM distribution.	
Approved by: Omer Mujahid	
Deliverables	
Loss function implementations; objective documentation.	
Resources	
Researcher time; TensorFlow.	
Estimated cost	128 €
Estimated cost (h)	16 h
Limit date	Mar 12, 2026
Responsible	Marc Font

Encoder Ablation Experiments	Preliminary project
Code in WBS: 2.4	
Description	
Systematic ablation of encoder design choices: masking strategy, driver weighting, positional encoding, CLS position, and number of layers. Compare by validation loss and representation quality.	
Acceptance criterion	
Ablation results table complete; final encoder configuration selected with documented rationale.	
Approved by: Omer Mujahid	
Deliverables	
Ablation experiment logs; selected configuration report.	
Resources	
Researcher time; TensorFlow; GPU (RTX 5070).	
Estimated cost	768 €
Estimated cost (h)	96 h
Limit date	Apr 2, 2026
Responsible	Marc Font

Normalisation Study	Preliminary project
Code in WBS: 2.5	
Description	
Compare per-patient vs global z-scoring of PI and RA signals. Evaluate the effect on downstream ISF and CR linear probe R^2 to select the optimal normalisation strategy.	
Acceptance criterion	
Comparison documented; global norm adopted if ISF/CR probe R^2 improves by $\geq$10% relative to per-patient norm.	
Approved by: Omer Mujahid	
Deliverables	
Normalisation study report; global scaler files.	
Resources	
Researcher time; Python; GPU.	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Apr 7, 2026
Responsible	Marc Font

Final Model Training	Preliminary project
Code in WBS: 2.6	
Description	
Train final encoder and decoder on the 831-patient training set using the validated configuration and global PI/RA normalisation. Apply early stopping on validation loss.	
Acceptance criterion	
Both models achieve stable convergence; final weights saved and verified on validation set.	
Approved by: Omer Mujahid	
Deliverables	
Trained encoder and decoder weight files.	
Resources	
Researcher time; TensorFlow; GPU; disk storage.	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Apr 10, 2026
Responsible	Marc Font

Representation Quality Analysis	Preliminary project
Code in WBS: 2.7	
Description	
Analyse trained representations: PCA intrinsic dimensionality, feature coverage, and Spearman correlations between principal components and clinical variables. Produce UMAP visualisations.	
Acceptance criterion	
PCA and UMAP results complete; at least 2 interpretable clinical axes identified in both encoder and decoder spaces.	
Approved by: Omer Mujahid	
Deliverables	
Analysis report; UMAP and PCA figures.	
Resources	
Researcher time; Python (scikit-learn, UMAP-learn).	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Apr 15, 2026
Responsible	Marc Font

FM vs Raw Evaluation Framework	Preliminary project
Code in WBS: 3.1	
Description	
Build evaluation framework comparing foundation model variants (frozen, encoder FT, decoder FT) against raw LSTM baselines. Standardise metrics and held-out test protocols.	
Acceptance criterion	
Framework runs all model variants on test set without data leakage; metric computation validated on held-out patients.	
Approved by: Omer Mujahid	
Deliverables	
Evaluation framework code; metric definitions document.	
Resources	
Researcher time; Python/TensorFlow.	
Estimated cost	128 €
Estimated cost (h)	16 h
Limit date	Apr 14, 2026
Responsible	Marc Font

App 1: Gap Imputation	Preliminary project
Code in WBS: 3.2	
Description	
Zero-shot gap imputation evaluation using the frozen encoder. Test on 4h, 6h, and 8h gaps on the 103-patient held-out test set. Compare against linear interpolation baseline.	
Acceptance criterion	
Results tables and figures complete; FM $R^2 \geq 0.60$ at the 4h gap.	
Approved by: Omer Mujahid	
Deliverables	
Imputation results tables and figures; evaluation code.	
Resources	
Researcher time; Python/TensorFlow; GPU.	
Estimated cost	128 €
Estimated cost (h)	16 h
Limit date	Apr 16, 2026
Responsible	Marc Font

App 2: Short-term Forecasting	Preliminary project
Code in WBS: 3.3	
Description	
Evaluate glucose forecasting at t+5 to t+120 min for FM frozen/FT variants and raw LSTM. Implement LSTM decoder seeded from transformer embeddings.	
Acceptance criterion	
Results tables and figures complete; at least one FM variant within 10% RMSE of the raw LSTM at 30 min.	
Approved by: Omer Mujahid	
Deliverables	
Forecasting results tables and figures; model and evaluation code.	
Resources	
Researcher time; TensorFlow; GPU.	
Estimated cost	320 €
Estimated cost (h)	40 h
Limit date	Apr 21, 2026
Responsible	Marc Font

App 3: Nocturnal Hypo Risk	Preliminary project
Code in WBS: 3.4	
Description	
Evaluate 8h nocturnal hypoglycaemia risk prediction using bedtime-filtered windows (21:00–23:59). Compare FM variants and raw LSTM on AUROC and AUPRC.	
Acceptance criterion	
Results tables and figures complete; best FM variant achieves AUROC ≥ 0.70.	
Approved by: Omer Mujahid	
Deliverables	
Hypo risk results tables and figures; evaluation code.	
Resources	
Researcher time; TensorFlow; GPU.	
Estimated cost	512 €
Estimated cost (h)	64 h
Limit date	Apr 24, 2026
Responsible	Marc Font

Embedding Study & Glycaemic Phenotyping	Preliminary project
Code in WBS: 3.5	
Description	
Generate patient-level embeddings for all 1,037 patients. Analyse structure via PCA and UMAP; identify glycaemic phenotype clusters without supervision.	
Acceptance criterion	
UMAP visualisations show clinically interpretable GRI stratification; PCA intrinsic dimensionality documented for both architectures.	
Approved by: Omer Mujahid	
Deliverables	
Embedding study report; UMAP figures; patient-level embedding files.	
Resources	
Researcher time; Python; GPU.	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Apr 20, 2026
Responsible	Marc Font

Physiological Parameter Recovery	Preliminary project
Code in WBS: 3.6	
Description	
Evaluate whether patient embeddings encode ISF, CR, and HbA1c via Ridge regression probes in 5-fold CV. Compare against CGM statistics and CGM+PI+RA curve baselines.	
Acceptance criterion	
Probe R^2 results for all three targets complete; decoder ISF $R^2 \geq 0.35$.	
Approved by: Omer Mujahid	
Deliverables	
Patient-level analysis report; probe R^2 tables; figures.	
Resources	
Researcher time; Python (scikit-learn).	
Estimated cost	192 €
Estimated cost (h)	24 h
Limit date	Apr 23, 2026
Responsible	Marc Font

Analysis and Discussion of Results	Preliminary project
Code in WBS: 4.1	
Description	
Synthesise all experimental results into an interpreted analysis of downstream task performance, physiological profiling findings, and representation structure differences between encoder and decoder.	
Acceptance criterion	
Discussion chapter draft reviewed and approved by supervisor; all results referenced and consistently interpreted.	
Approved by: Omer Mujahid	
Deliverables	
Discussion chapter draft.	
Resources	
Researcher time; supervisor feedback.	
Estimated cost	320 €
Estimated cost (h)	40 h
Limit date	May 1, 2026
Responsible	Marc Font

Thesis Writing	Preliminary project
Code in WBS: 4.2	
Description	
Write all thesis chapters in LaTeX (Introduction through Appendices), compile the full document, address supervisor feedback, and submit the final version to the university.	
Acceptance criterion	
Complete thesis submitted to university; all chapters reviewed and approved by supervisor.	
Approved by: Omer Mujahid	
Deliverables	
Final thesis PDF; submitted document.	
Resources	
Researcher time; LaTeX; supervisor feedback.	
Estimated cost	1280 €
Estimated cost (h)	160 h
Limit date	May 29, 2026
Responsible	Marc Font