A large sensor foundation model pre-trained on continuous glucose monitor data for diabetes management

npj Health Systems
Published: Sep. 2025

Saman Khamesian

02/04/2026

Introduction

- Continuous Glucose Monitoring (CGM) has become a core tool for diabetes management and real-time decision making.

- One particularly promising approach to meeting patient needs is to combine CGM with AI to predict near-future glucose values.

- Most prior studies rely on:
  - simulated CGM data
  - small patient cohorts

- As a result, prediction accuracy remains limited, especially for longer horizons (e.g. 2 h)

Introduction

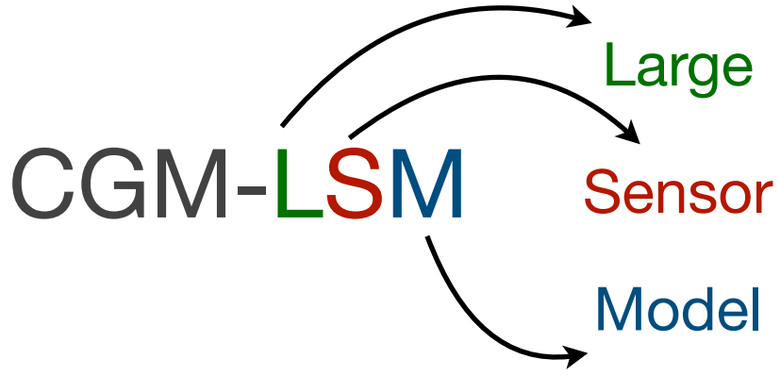
- Poor performance is driven not only by limited data, but also by:
  - training models from random initialization
  - lack of learned glucose generation knowledge

- How to extract and leverage latent patterns in large-scale CGM data remains an open question.

- To address this gap, they presented a new approach to glucose prediction for diabetes management that harnesses the pre-training technique exemplified in large language models (LLMs).

Methodology

Core Idea



- The key conceptual move is: *Treat CGM like language*.

Language Models	CGM-LSM
Token = word	Token = glucose value
Sentence = word sequence	Sequence = CGM time series
Next-token prediction	Next-glucose prediction
GPT-style decoder	Transformer decoder

- CGM-LSM is just a **GPT-style decoder** trained with **next-token cross-entropy**
- The only difference is that **tokens are glucose values** instead of words.
- They argue that glucose has latent structure, just like language, and that pre-training can uncover it.

Methodology

Tokenization

- They do NOT regress glucose as a real number. Instead:

- Each glucose value (0–400 mg/dL) is a categorical token

  - Vocabulary size ≈ 400

- For instance, a glucose level of 153 becomes the token “153”.

- In the pre-training task of next glucose value prediction, the model was expected to learn the semantic meanings of the embeddings for these glucose value tokens.

Methodology

Problem Definition for Pre-Training

- Teaches the model how glucose evolves over time, without any labels, by learning to predict the next glucose value from previous ones.

- They represent one CGM instance as a sequence:

$$s_1, s_2, s_3, \ldots, s_n$$

- Where:
  - s_i = glucose value (token) at time step i
  - Each token corresponds to a **5-minute CGM reading**
  - $n = 312$ (24h past + 2h future)

Methodology

Problem Definition for Pre-Training

- Mathematically, this can be described as maximizing the likelihood of a glucose value s_i given the preceding glucose values $s_1, s_2, s_3, \ldots, s_{i-1}$.

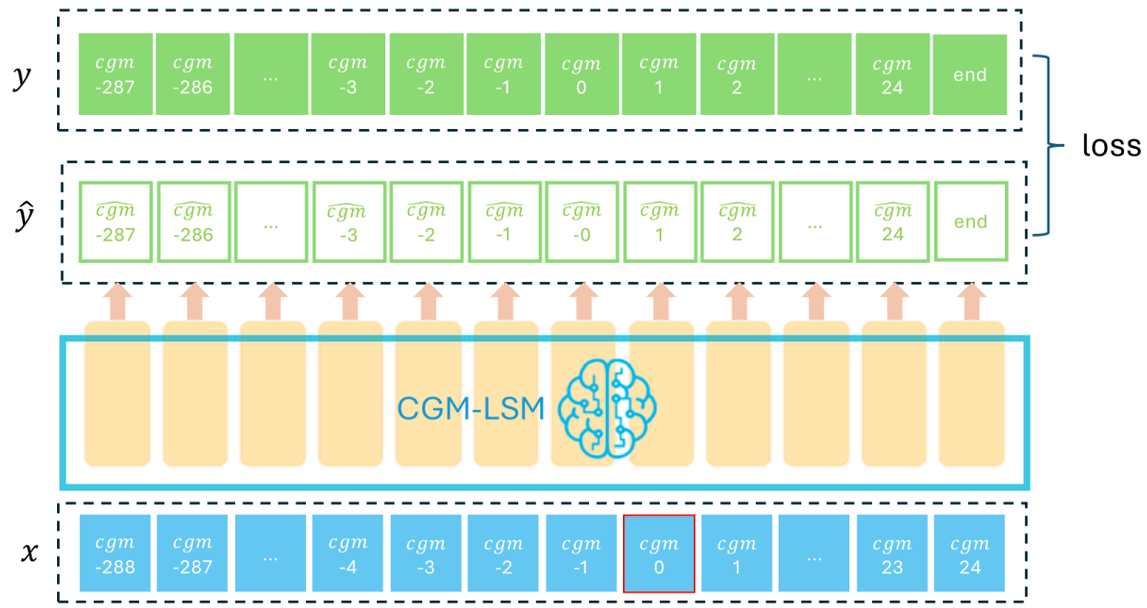
$$L(\theta) = \sum_{i=1}^{n} log\ p_\theta(s_i \mid s_1, \ldots s_{i-1})$$

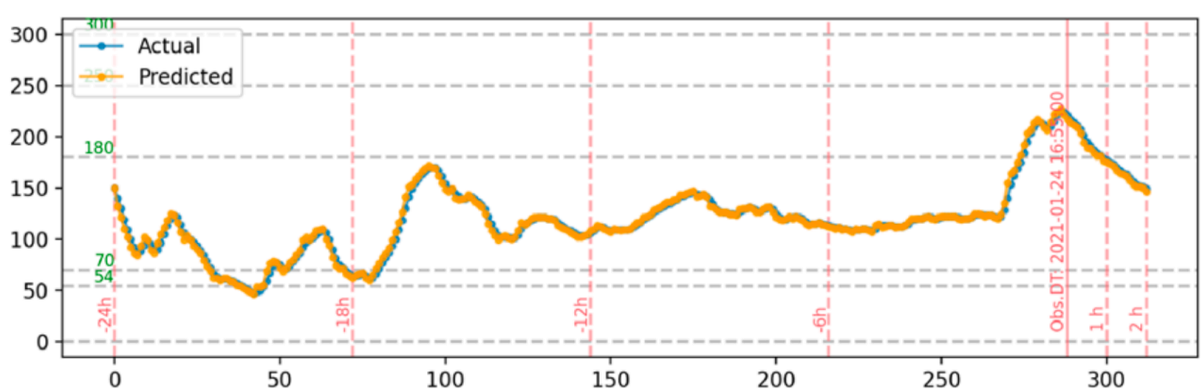
- Where:
  - θ : Model Parameters (Transformer Weights)
  - n : Length of the glucose sequence
  - s_i : Glucose token at time i
  - $p_\theta(\ .\ )$: Model's predicted probability

Methodology

Pre-Training Process Overview

- Autoregressive pre-training of CGM-LSM, where the model predicts each next glucose token from past tokens and is optimized using cross-entropy loss over the full sequence.





Methodology

Problem Definition for Prediction

- After the CGM-LSM was trained, it could be used to generate new glucose sequences (similar to that of the GPT models):

$$p(s_{i+1} \mid s_1, \ldots, s_i \,; \theta) = \text{softmax}(h_i W)$$

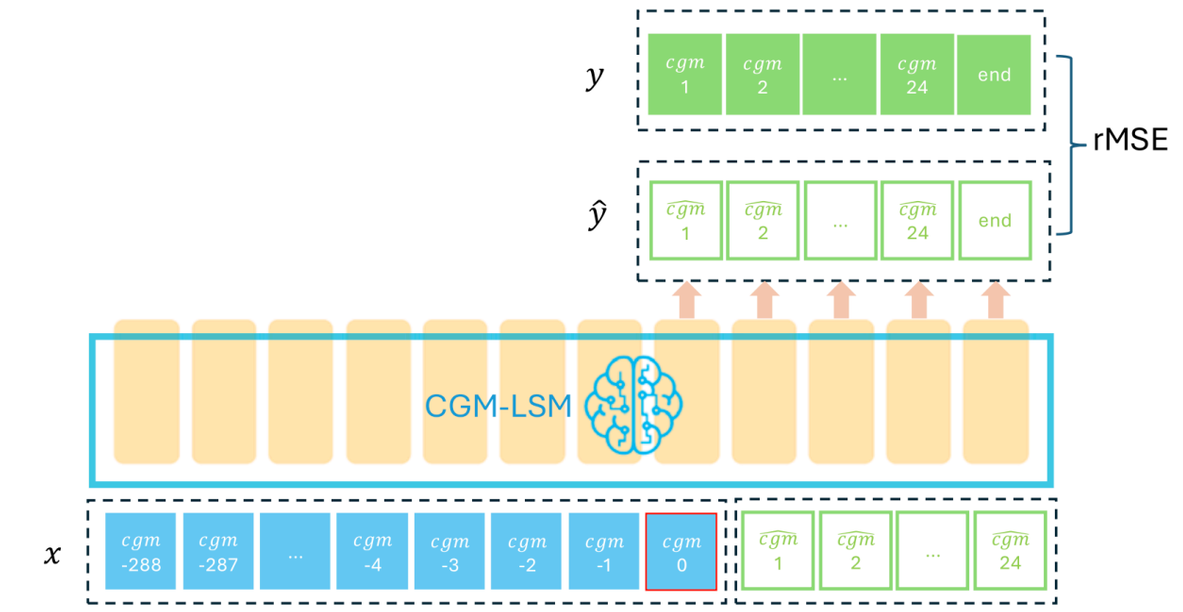
- Where:
  - θ : Model parameters
  - h_i : Hidden states derived from transformer blocks
  - W : Output projection matrix
- They use greedy decoding:

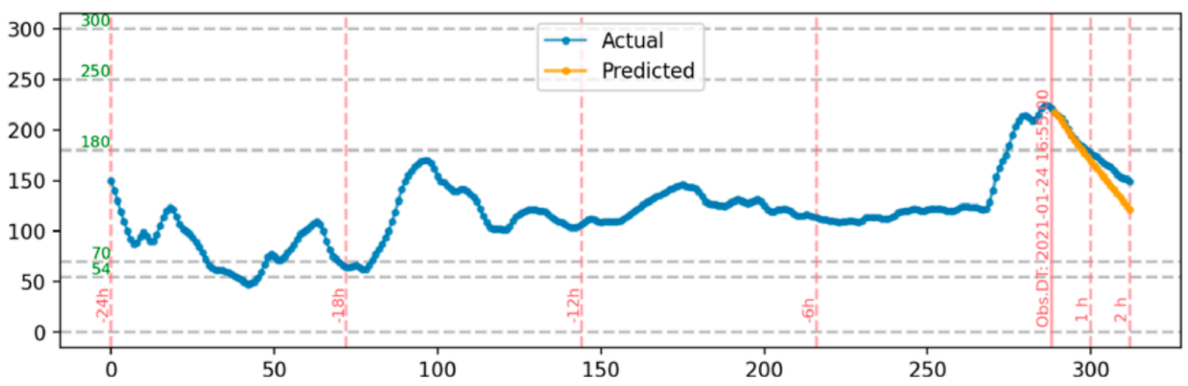
$$s_{i+1} = \text{argmax}\; p(s_{i+1} \mid s_1, \ldots s_{i-1})$$

Methodology

Prediction Process Overview

- During prediction, CGM-LSM auto-regressively generates future glucose values from past CGM context, with performance evaluated using rMSE.





Methodology

Structure

- CGM-LSM uses a decoder-only transformer architecture, similar to GPT-style language models

- The model relies entirely on self-attention to process glucose sequences.

- In a decoder-only transformer, a stack of decoder blocks generates one glucose value token at a time in an autoregressive manner.

- Each decoder block in the transformer comprises two main components:
  - a multi-head self-attention mechanism
  - and a position-wise fully connected feed-forward network.

Model Evaluation

Pre-Training Dataset

- The model is pre-trained on a large, real-world dataset collected by Welldoc.
- The dataset contains CGM sampled every 5 minutes.
- After preprocessing and filtering, the final dataset includes:
  - ~15.96 million valid CGM instances
  - 592 patients in total
  - Both T1D and T2D patients
  - Diverse age groups and genders
- Each CGM instance represents a 26-hour window:
  - 24 hours of past glucose values (input)
  - 2 hours of future glucose values (prediction target)
- Only CGM data are used (No insulin, carbs, or lifestyle features)

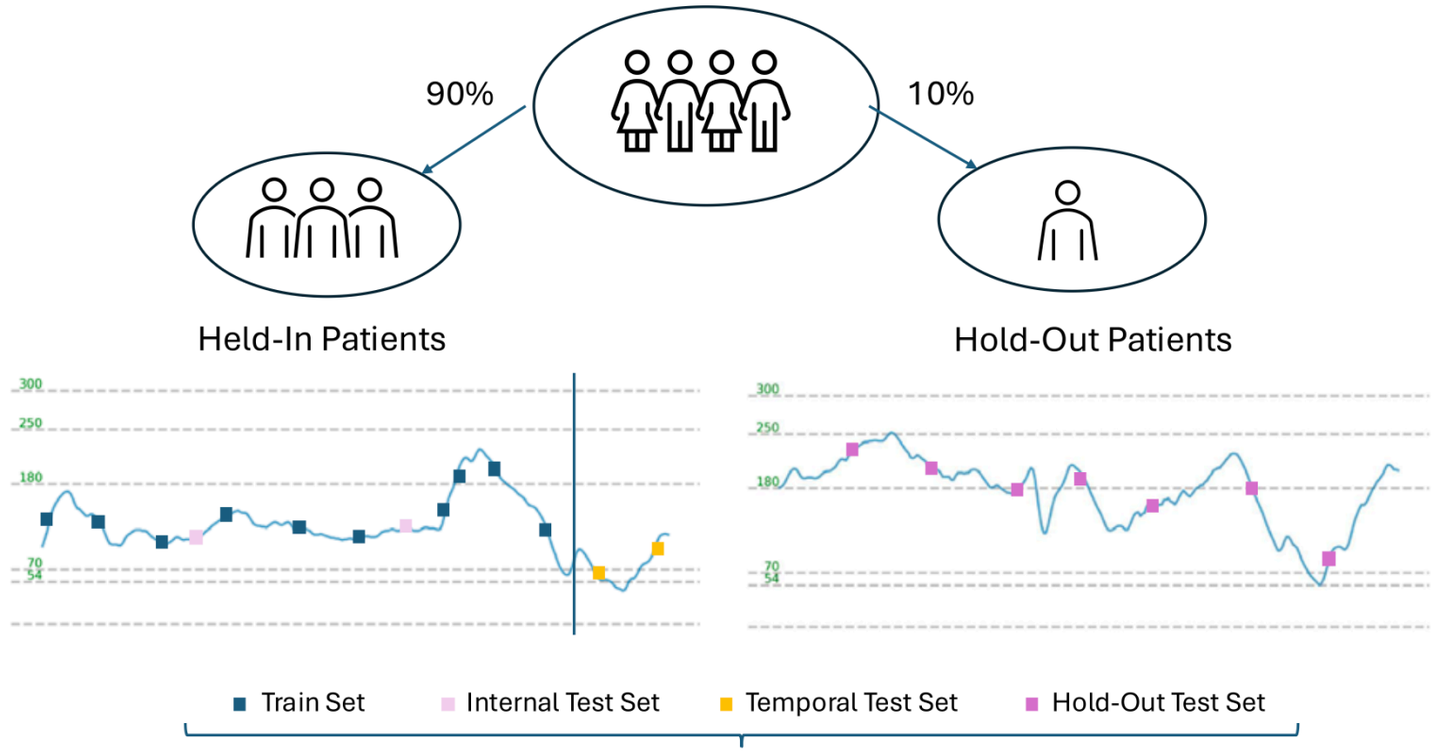
Model Evaluation

Welldoc Dataset

- Patients are first divided at the patient level:
  - Held-out test set
    - ~10% of patients
    - Completely excluded from training
    - Used to evaluate zero-shot performance on **unseen patients**
- For the remaining ~90% of patients:
  - CGM instances are ordered chronologically per patient
  - The latest 10% of instances are **reserved as a temporal test set**
  - The remaining instances are randomly split into:
    - Training set (80%)
    - Internal test set (10%)

Model Evaluation

Welldoc Dataset



- Internal test → unseen instances from known patients
- Temporal test → future periods for known patients
- Held-out test → entirely unseen patients

Model Evaluation

OhioT1DM Dataset

- The pre-trained model is additionally **evaluated** on the OhioT1DM dataset
- OhioT1DM is used as an external Hold-Out dataset.
- This dataset includes:
  - 12 T1D patients
  - CGM data sampled at 5-minute intervals
- OhioT1DM is **never used during pre-training**
- Evaluation on this dataset represents a **strict zero-shot test** against prior work
- Baseline models are trained on OhioT1DM, while CGM-LSM is evaluated directly **without retraining.**

Model Evaluation

Dataset

Dataset	Seen during training?	Purpose
Training (Welldoc)	✅ Yes	Learn model
Internal test (Welldoc)	❌ No	Known patients, random times
Temporal test (Welldoc)	❌ No	Known patients, future periods
Held-out test (Welldoc)	❌ No	Unseen patients (same dataset)
OhioT1DM	❌ No	**Unseen patients + unseen dataset**

Setup

Implementation Details

- Single NVIDIA A100 (80 GB)
- GPT-2 decoder-only Transformer
  - Transformer layers: 12
  - Attention heads: 12
  - Vocabulary:
    - 400 glucose value tokens
    - 17 special tokens
    - Embedding dimension: 768
- Batch size: 256
- Training duration: 10 epochs
- Optimizer: AdamW
  - $\beta_1 = 0.9$
  - $\beta_2 = 0.999$
- Learning rate: 5×10^{-5}
- Dropout: 0.1
- LayerNorm ε: 1×10^{-5}
- Hyperparameters selected via grid search

Model Evaluation

Evaluation Metrics

- In this study, they evaluated model performance using four metrics:

1. Root Mean Squared Error (rMSE) $$\text{rMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$
2. Mean Absolute Error (MAE) $$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$
3. MAE with a tolerance of 10 units (MAEWith10) $$\text{MAE}(\tau) = \frac{1}{n}\sum_{i=1}^{n}\min\left(\max\left(0, |y_i - \hat{y}_i| - \tau\right), 1\right)$$
4. Religion Accuracy $$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

- MAEWith10 measures the proportion of absolute errors that fall within a specified tolerance level, here set at 10 units.

- Glucose Region Accuracy evaluates the accuracy of predictions by categorizing continuous glucose values into predefined regions.

Results

Tokenization Embeddings

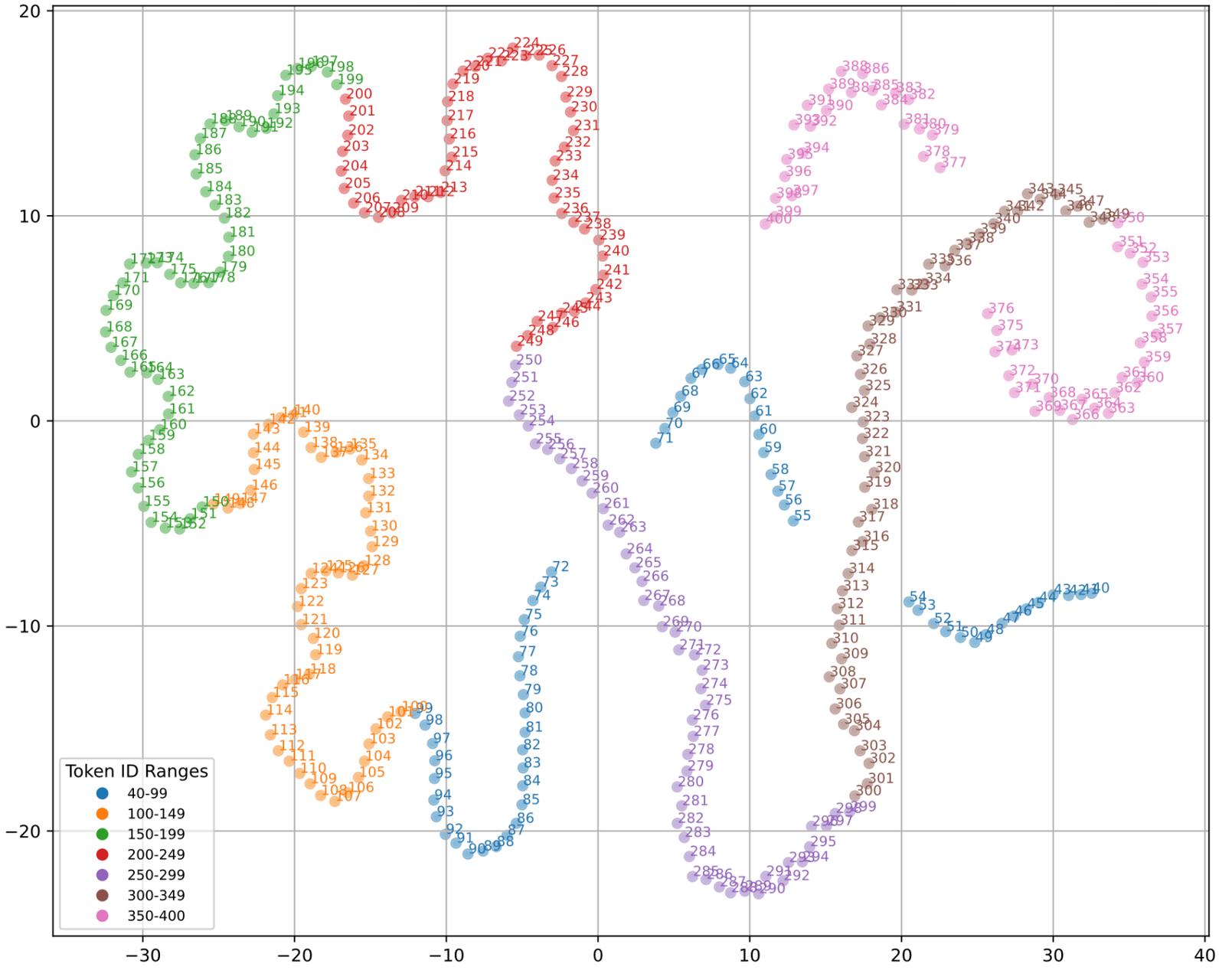


Fig. 2 | CGM-LSM token embeddings (colored by glucose range). Each point represents a token ID, which directly corresponds to a glucose value. Colors indicate glucose ranges commonly used in clinical interpretation.

Results

rMSE Report

Table 2 | Comparative performance of predictive models for future glucose levels, using Root Mean Square Errors (rMSE)

Model	Dataset		rMSE-30m	rMSE-1h	rMSE-2h
LSTM	OhioT1DM		36.022 (0.551)	37.17 (0.513)	38.703 (0.461)
RNN			36.102 (0.552)	37.344 (0.512)	38.952 (0.458)
GRU			37.555 (0.554)	38.573 (0.517)	40.108 (0.463)
Transformer			27.886 (0.463)	30.869 (0.462)	36.653 (0.47)
Informer			35.197 (0.501)	36.962 (0.485)	40.204 (0.463)
Autoformer			36.08 (0.552)	38.352 (0.542)	41.395 (0.515)
CGMLSM			9.024 (0.168)	15.895 (0.283)	26.876 (0.43)
CGM-LSM	WellDoc T1D	Internal Test	8.403 (0.066)	16.049 (0.118)	28.277 (0.188)
		Temporal Test	9.155 (0.068)	17.013 (0.118)	29.426 (0.184)
		Held-Out Test	8.926 (0.056)	16.905 (0.101)	29.812 (0.16)
	WellDoc T2D	Internal Test	7.441 (0.055)	13.418 (0.094)	22.649 (0.147)
		Temporal Test	8.025 (0.058)	14.073 (0.095)	23.216 (0.143)
		Held-Out Test	7.772 (0.055)	13.877 (0.091)	23.494 (0.143)

Each entry displays the mean rMSE followed by the confidence interval width in parentheses, indicating the range within which the true mean is expected to lie with 95% confidence.

Results

Model Performance on Different Groups

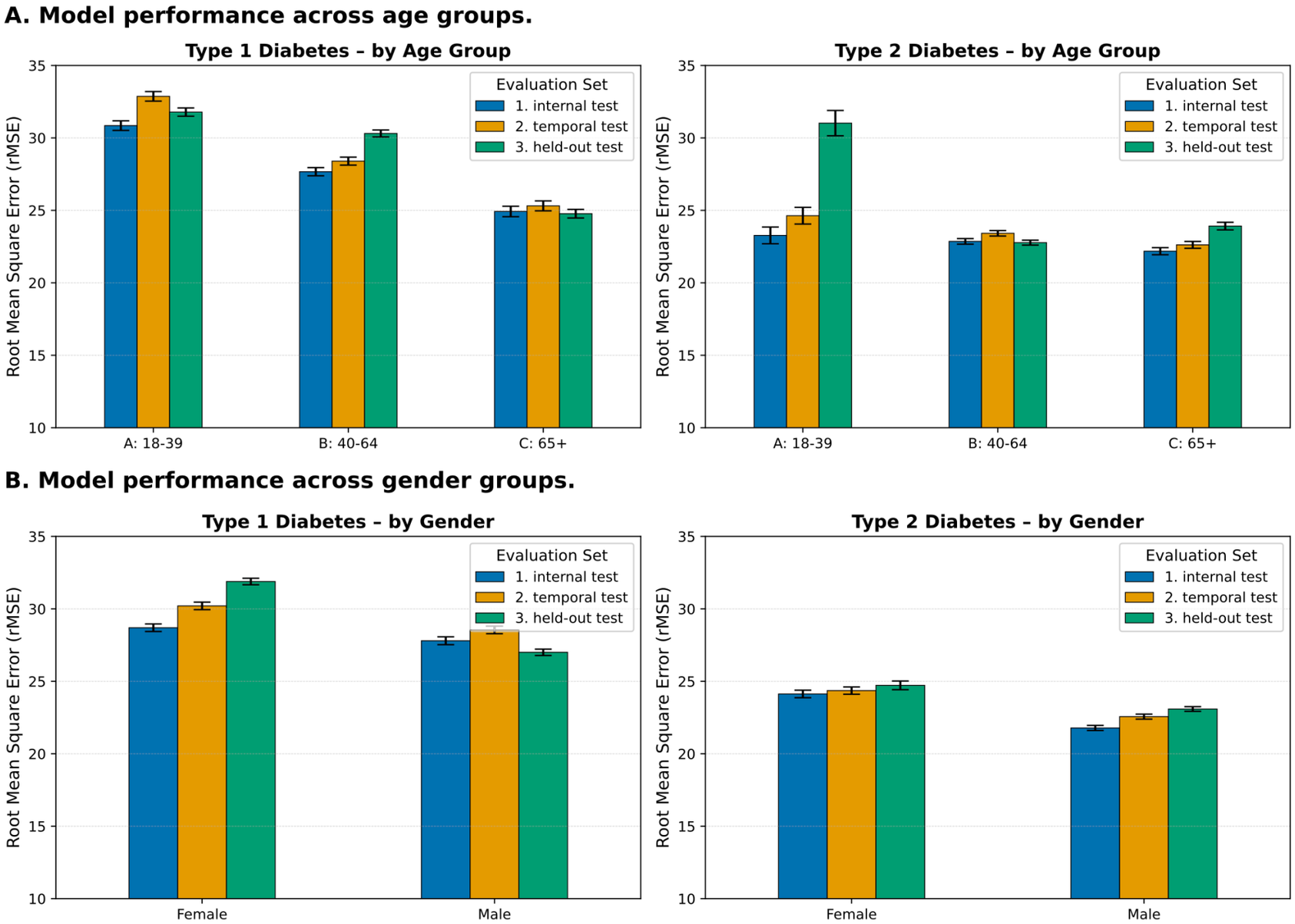


Fig. 3 | Model performance in 2-h glucose prediction across age and gender. **A** Model performance across age groups (18–39, 40–64, 65+), showing rMSE-2H values for Type 1 and Type 2 Diabetes prediction across different test sets. **B** Model performance across gender, showing rMSE-2H values for Type 1 and Type 2 Diabetes prediction across different test sets.

Thank you for your attention