SSM-CGM: Interpretable State-Space Forecasting Model of Continuous Glucose Monitoring for Personalized Diabetes Management

Shakson Isaac*
Department of Biomedical Informatics
Harvard Medical School
shakson_isaac@g.harvard.edu

Yentl Collin*
Department of Biomedical Informatics
Harvard Medical School
yentl.collin@eleves.enpc.fr

Chirag J. Patel
Department of Biomedical Informatics
Harvard Medical School
chirag_patel@hms.harvard.edu

Abstract

Continuous glucose monitoring (CGM) generates dense data streams critical for diabetes management, but most used forecasting models lack interpretability for clinical use. We present SSM-CGM, a Mamba-based neural state-space forecasting model that integrates CGM and wearable activity signals from the AI-READI cohort. SSM-CGM improves short-term accuracy over a Temporal Fusion Transformer baseline, adds interpretability through variable selection and temporal attribution, and enables counterfactual forecasts simulating how planned changes in physiological signals (e.g., heart rate, respiration) affect near-term glucose. Together, these features make SSM-CGM an interpretable, physiologically grounded framework for personalized diabetes management.

1 Introduction

Diabetes affects over 580 million adults (20–79 years) worldwide [1]. Continuous glucose monitoring (CGM) is a central tool for diabetes management, tracking interstitial glucose every 5–15 minutes for up to 2 weeks [2]. These data enable metrics such as time-in-range (TIR) which reflect risk of diabetes complications [3, 4, 5]. Yet despite these advances, translating CGM data into actionable, personalized guidance remains a central challenge.

Forecasting models predict near-future glucose trajectories, giving users a "next-hour" view. Existing approaches, including recurrent networks and transformers, demonstrate great predictive accuracy [6, 7]. However, these studies are limited, showing abilities to predict CGM forecasts but offering little interpretability, limiting usage in personalized, behaviorally-informed care.

We introduce *SSM-CGM*, an interpretable Mamba-based neural state-space model for CGM forecasting and counterfactual reasoning. Using the AI-READI cohort, SSM-CGM: (i) improves forecasting accuracy over the Temporal Fusion Transformer baseline, (ii) provides interpretability via variable selection and temporal attribution, and (iii) generates counterfactual forecasts that simulate how planned changes in physiological indicators influence glucose dynamics. Together, SSM-CGM is a framework that links predictive modeling with the needs of personalized diabetes management.

*Both authors contributed equally.

Accepted to the 39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Learning from Time Series for Health (TS4H).

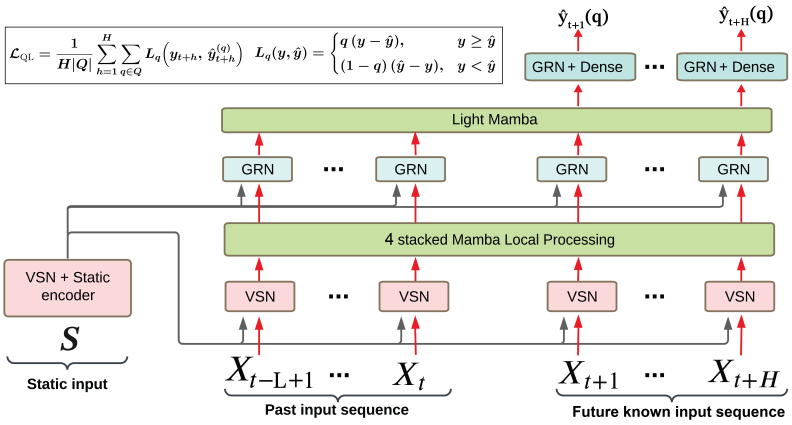

Figure 1: **SSM-CGM Architecture.**

2 Methods

Dataset. We utilized data from 741 individuals participating in AI-READI [8], each with 8-10 days of CGM (Dexcom G6) and wearable (Garmin Vivosmart 5) data sampled every 5 minutes, totaling >2M measurements (Appendix A.1). Wearable features included heart rate, respiration rate, oxygen saturation, steps, stress, calories burned, and sleep stage; demographics are in Appendix A.1. **Preprocessing.** Missing values were imputed via Last Observation Carried Forward and series clipped to intervals with overlapping CGM–wearable coverage. **Feature engineering.** (i) *static covariates*: age, diabetes status, clinical site, and participant ID; (ii) *temporal encodings*: minute-of-day and 24h sine/cosine; (iii) *historical covariates*: recent CGM lags/diffs, rolling stats, time-aligned wearable signals, and meal flags (Sec. 2); (iv) *future covariates*: time-of-day encodings, wearable signals, and meal flags which are proxies for planned daily activities (e.g., sleep, exercise, and eating).

Meal Detection. As AI-READI lacks meal annotations, we trained a meal-detection model on CGMacros [9] (Appendix A.1) using 5-min CGM along 6-h windows. Sparse, time-imprecise labels, were smoothed with a trapezoidal kernel (30-min plateau, symmetric ramps; $\tilde{y}_t \in [0, 1]$). The model, a dilated CNN + 2-layer Bi-LSTM with LayerNorm and a linear head (Appendix A.2), was trained with weighted BCE-with-logits ($\alpha = 5$ to counter class imbalance) on subject-wise, diabetes-stratified train/test splits. At inference on validation series, a sliding 6-h window (stride$= 1$) applied per-window thresholding ($\theta = 0.36$) and ratio voting ($\rho = 0.77$). Validation achieved Recall=80% and Precision=72%, sufficient for use as a predicted meal flag covariate in AI-READI.

Forecasting Objective. We predicted future CGM values $\mathbf{y}_{t+1:t+H}$ at 5-minute resolution using the static $\mathbf{s}$, historical $\mathbf{x}^{(\text{hist})}_{t-L+1:t}$, and future $\mathbf{x}^{(\text{fut})}_{t+1:t+H}$ covariates described in Section 2.

$$\hat{\mathbf{y}}_{t+1:t+H} = f_\theta\left(\mathbf{y}_{t-L+1:t},\ \mathbf{x}^{(\text{hist})}_{t-L+1:t},\ \mathbf{x}^{(\text{fut})}_{t+1:t+H},\ \mathbf{s}\right). \tag{1}$$

We used context lengths of $L = 144$ (12 h), 288 (24 h), and 576 (48 h) with a forecast horizon $H = 12$ (1 hour). As participants started and ended device use at different times of day (8-10 days total), we are able to withhold their last two hours of data for validation (penultimate) and testing (last).

SSM-CGM architecture. For our CGM forecasting objective, the *Temporal Fusion Transformer* (TFT) [10], often used for CGM forecasting, provides a strong baseline and a great interface: it separates *static*, *historical* and *future* covariates, supports multi-horizon forecasting, and offers interpretable variable importance via *Variable Selection Networks* (VSNs). In our comparisons, TFT outperformed several strong classical and deep forecasters (LightGBM, N-HiTS, LSTM, etc.) under identical preprocessing (Appendix A.3), consistent with prior CGM studies [11, 12, 13, 14].

However, CGM dynamics span hours to days, requiring long-context, low-latency models. TFT is widely used; yet sequential LSTMs and $O(L^2)$ attention can be suboptimal for long-context, online forecasting. Recent work shows that Mamba [15], a selective SSM, provides linear-time scanning, small memory footprint, and a competitive accuracy on time-series benchmarks [16, 17]. Motivated by this, we replace TFT's LSTM local processor with stacked Mamba blocks and its global self-attention

with a light Mamba layer, while retaining the original embeddings, GRNs (Appendix A.5), VSNs (Appendix A.6), and the multi-horizon head. The resulting architecture (Figure 1) preserves TFT's variable-wise fusion and leverages Mamba's state-space temporal modeling. The model uses a multi-horizon prediction head trained with a quantile loss (Figure 1) $\mathcal{L}_{\text{QL}}$ at $q \in Q = \{0.1,\, 0.5,\, 0.9\}$ to produce calibrated prediction intervals.

Interpretability. We provide two views of model interpretability in forecasting: (i) *variable importance* from VSNs ("what"), and (ii) *temporal importance* from SSM dynamics ("when"; *hidden attention*). **Variable importance.** The VSN outputs simplex weights $\alpha_{i,t}$ over time series covariates and α_k^s over static covariates c (see Appendix A.6). Aggregating these weights across participants and contexts reveals which signals drive the forecasts. **Temporal importance.** Inspired by the *Hidden Attention of Mamba* [18], we adapt hidden attention to time-series forecasting by unrolling the linear state update (Eq. 15) of SSM equations (see Appendix A.7). This yields a causal convolution with kernel $K_{\ell j}$, inducing attention-like weights over past lags: at time ℓ, with the input x, the output (readout) z and the SSM dynamics B, C, Δ, Λ (see Appendix A.7)

$$z_\ell = \sum_{j \le \ell} K_{\ell j} x_j, \quad K_{\ell j} = C_\ell \exp\big(\textstyle\sum_{t=j}^{\ell-1} \Delta t_t \, \Lambda_t\big) B_j, \quad \alpha_{\ell j} = \mathrm{softmax}_{j \le \ell}\Big(\tfrac{\log(\psi(K_{\ell j})+\varepsilon)}{\tau}\Big), \quad (2)$$

where $\alpha_{\ell j}$ are attention-like lag weights ($\varepsilon, \tau > 0$; ψ is the channel-wise ℓ_1 norm). We further employ *Mamba-2* [19], which factorizes channels into multi-head groups with stable parameterization, improving throughput and long-range capture. To make head-wise attributions comparable, taking inspiration from TFT's interpretable attention and the multi-expand SSM (MES) design in *Transformers Are SSMs* [19], we created an MES-style Mamba2: each head has its own dynamics $(B^{(h)}, \Lambda^{(h)}, \Delta t^{(h)})$, but we enforce a *shared* value projection V and readout C_{sh} across heads (see Appendix A.8); in practice, we first apply $v_j = V x_j$ *(shared across all heads)*, then specialize the dynamics per head while keeping the readout fixed to C_{sh} ($K_{\ell j} \to K_{\ell j}^{(h)}$). With this change, hidden-attention maps $\alpha_{\ell j}^{(h)}$ are on a common scale, analogous to interpretable multi-head attention in TFT, and empirically yields smoother, more stable temporal attributions.

Counterfactual Forecasting. We model counterfactual CGM forecasts as interventions on planned future covariates (e.g., "what-if heart rate increases?"). Let $\mathcal{H}_t = (\text{CGM}_{t-L+1:t}, X_{t-L+1:t}, S)$ denote history up to t, where X time-varying (e.g., activity, meals) and S static covariates. For a planned future covariate sequence $\mathbf{a}_{t+1:t+h}$, the counterfactual distribution is

$$p(\text{CGM}_{t+1:t+h}(\mathbf{a}) \mid \mathcal{H}_t) = \mathbb{P}(\text{CGM}_{t+1:t+h} \mid \mathcal{H}_t, \text{do}(X_{t+1:t+h} = \mathbf{a}))\,, \quad (3)$$

where $\text{do}(\cdot)$ fixes covariates $\mathbf{a}_{t+1:t+h}$ irrespective of natural dynamics. Under standard assumptions of consistency, sequential ignorability, and positivity, this distribution is identified by the sequential g-formula [20, 21] (Appendix A.14). In practice, *SSM-CGM* implements this directly by conditioning on the planned sequence in a single pass, rather than computing the g-formula explicitly.

3 Results

Benchmarking. We evaluated whether *SSM-CGM* improved forecasting performance compared to its baseline architecture TFT [10], specifically testing the replacement of TFT's LSTM/self-attention with our Mamba-2 (MES-style) blocks. Under identical preprocessing, splits, hyperparameter search, and training on 4 A100 GPUs (see Appendix A.9), along with matched parameter counts (SSM-CGM *1,762,714* params vs. TFT *1,793,376*), SSM-CGM *outperformed* TFT on the AI-READI dataset (Table 1). We report the training objective, MAE, and RMSE to assess sensitivity to large errors. With 12h, 24h, and 48h contexts, SSM-CGM consistently reduces MAE compared to TFT by approximately 2%, 7%, and 7%, respectively. Within SSM-CGM, we observe 5% performance gain with each doubling of context length, highlighting the value of long-range temporal information in CGM forecasting and the ability of our model to capture it.

Table 1: Benchmarking performance (1 hour horizon). Mean (95% CI).

Context Length	**SSM-CGM**			**TFT**		
	Quantile Loss ↓	**MAE ↓**	**RMSE ↓**	**Quantile Loss ↓**	**MAE ↓**	**RMSE ↓**
12 h	3.87 (3.81, 3.93)	5.97 (5.68, 6.26)	7.17 (6.83,7.52)	3.95 (3.89, 4.02)	6.07 (5.76, 6.37)	7.32 (6.96, 7.69)
24 h	3.69 (3.63, 3.75)	5.66 (5.39, 5.93)	6.86 (6.53, 7.19)	3.92 (3.86, 3.99)	6.06 (5.77, 6.36)	7.28 (6.93, 7.63)
48 h	**3.52** (3.47, 3.58)	**5.40** (5.15, 5.65)	**6.56** (6.26, 6.87)	3.70 (3.64, 3.76)	5.77 (5.50, 6.03)	6.92 (6.61, 7.24)

A

Variable	Encoder	Decoder
CGM + LAGS	0.5949	**X**
meal flag pred	0.0758	0.1077
heart rate	0.0526	0.1301
respiration rate	0.0514	0.1333
time of day sin cos	0.0410	0.1870
stress level	0.0317	0.0838
activity steps	0.0226	0.0641
oxygen saturation	0.0186	0.0526
activity calories	0.0157	0.0376
sleep stage	0.0105	0.0737

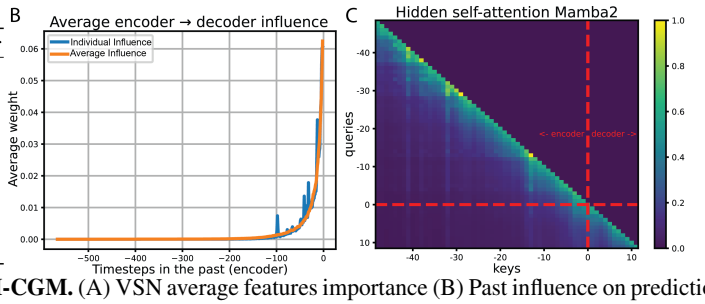

Figure 2: **Interpretability of SSM-CGM.** (A) VSN average features importance (B) Past influence on predictions (average, individual) (C) Head average self-attention map of Light Mamba (last 60-step window, one individual)

Interpretability. The VSNs report global and step-wise importances: averages over time/participants (Figure 2A) indicate that past/future wearable covariates strongly help predictions, especially meal flags, stress levels, heart/respiration rates reflecting circadian and activity patterns [22, 23], while step-wise maps (Appendix A.10) pinpoint which signals matter at each instant. Static covariates also contributed (Appendix A.12), and participant-level embeddings showed that SSM-CGM captured markers of hyperglycemia and cardiovascular risk (Appendix A.13). Averaging head-wise hidden attention maps (Section 2) yielded a stable temporal attribution profile, quantifying how past lags influence decoder steps at the cohort and individual level (Figure 2B). As expected of smooth CGM dynamics, recent history dominated, but individuals showed distinctive peaks at specific lags indicating moments the model must retain to improve forecasts. Self-hidden attention maps (Figure 2C, Appendix A.11) further demonstrated how salient past events propagate through the SSM state and shape downstream predictions. At the cohort level, averaging hidden-attention maps across layers and heads suggests a complementary division of temporal labor in differents mamba layers (Appendix A.11). Together, these analyses revealed which covariates matter and when past information is most influential, capturing both cohort-level patterns and individual-specific signals relevant for glucose management.

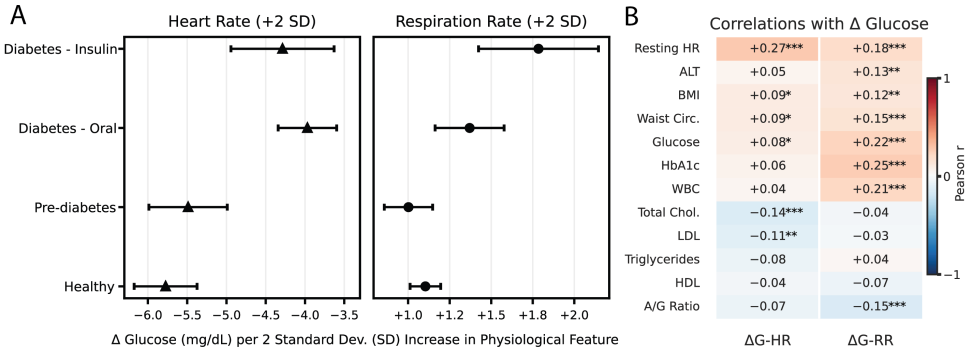

Figure 3: **Counterfactual effects and associations.** (A) Estimated change in glucose (mg/dL) from a +2 SD perturbation to heart rate (HR) or respiration rate (RR) over a 1-h horizon, stratified by diabetes status (mean $\pm$ 95% CI). (B) Pearson correlations between the counterfactual effects (ΔG-HR, ΔG-RR) and clinical laboratory measures. Stars indicate FDR-adjusted significance (p): * <0.05, ** <0.01, *** <0.001.

Counterfactuals. We next evaluated whether short-term interventions in planned activity could alter glucose trajectories. Using SSM-CGM, we computed counterfactuals (Section 2), for the two top-ranked wearable features: heart rate (HR) and respiration rate (RR) (Figure 3). Each counterfactual estimated change-in glucose (Δ glucose) when the feature is perturbed to +2 within-person SD above the individual's baseline mean value across 1-hour horizons restricted to plausible times of day (Appendix A.14). On average, the maximal short-term effect on glucose was -5.0 mg/dL for HR and $+1.2$ mg/dL for RR. Because counterfactuals are computed at the individual level, we examined their variation across diabetes status (Figure 3A). We found that ΔG-HR and ΔG-RR were uncorrelated (r = 0.01) but mapped to distinct domains of health: HR effects correlated with cardiovascular and glycemic markers, while RR effects aligned strongly with adiposity, inflammation, and hepatic dysfunction (Figure 3B). Importantly, the simulated effects reflected plausible contexts. Daytime HR increases, consistent with physical activity, were associated with improved glucose tolerance. In contrast, nighttime RR increases, mimicking impaired sleep respiration, were linked to adverse metabolic markers.

4 Conclusion

SSM-CGM advances CGM forecasting with interpretable, counterfactual, and personalized insights for diabetes management. Despite limitations (Appendix A.15), this work is an initial step toward deep models that not only forecast CGM trajectories but also guide individualized behavioral plans to improve outcomes such as time-in-range.

References

[1] International Diabetes Federation. *IDF Diabetes Atlas*. International Diabetes Federation, Brussels, Belgium, 11th edition, 2025.

[2] Dexcom, Inc. Dexcom G6 Continuous Glucose Monitoring System. Medical device, 2025.

[3] Zheng Zhou, Bao Sun, Shiqiong Huang, Chunsheng Zhu, and Meng Bian. Glycemic variability: adverse clinical outcomes and how to improve it? *Cardiovasc. Diabetol.*, 19(1):102, July 2020.

[4] Yong Yi Tan, Enhui Suan, Gerald Choon Huat Koh, Suhana Binte Suhairi, and Shilpa Tyagi. Effectiveness of continuous glucose monitoring in patient management of type 2 diabetes mellitus: an umbrella review of systematic reviews from 2011 to 2024. *Arch. Public Health*, 82(1):231, December 2024.

[5] Jingyi Lu, Xiaojing Ma, Jian Zhou, Lei Zhang, Yifei Mo, Lingwen Ying, Wei Lu, Wei Zhu, Yuqian Bao, Robert A Vigersky, and Weiping Jia. Association of time in range, as assessed by continuous glucose monitoring, with diabetic retinopathy in type 2 diabetes. *Diabetes Care*, 41(11):2370–2376, November 2018.

[6] Guy Lutsker, Gal Sapir, Smadar Shilo, Jordi Merino, Anastasia Godneva, Jerry R. Greenfield, Dorit Samocha-Bonet, Raja Dhir, Francisco Gude, Shie Mannor, Eli Meirom, Gal Chechik, Hagai Rossman, and Eran Segal. From glucose patterns to health outcomes: A generalizable foundation model for continuous glucose monitor data analysis. *arXiv preprint arXiv:2408.11876*, 2025.

[7] Renat Sergazinov, Valeriya Rogovchenko, Elizabeth Chun, Nathaniel Fernandes, and Irina Gaynanova. Glucobench: Curated list of continuous glucose monitoring datasets with prediction benchmarks. *arXiv*, 2024.

[8] AI-READI Consortium. AI-READI: rethinking AI data collection, preparation and sharing in diabetes research and beyond. *Nat. Metab.*, 6(12):2210–2212, December 2024.

[9] Ricardo Gutierrez-Osuna, David Kerr, Bobak Mortazavi, and Anurag Das. CGMacros: a scientific dataset for personalized nutrition and diet monitoring, 2025.

[10] Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

[11] Taiyu Zhu, Lei Kuang, Chengzhe Piao, Junming Zeng, Kezhi Li, and Pantelis Georgiou. Edge-based temporal fusion transformer for multi-horizon blood glucose prediction. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2023.

[12] Taiyu Zhu, Lei Kuang, Chengzhe Piao, Junming Zeng, Kezhi Li, and Pantelis Georgiou. Population-specific glucose prediction in diabetes care with transformer-based deep learning on the edge. *IEEE Transactions on Biomedical Circuits and Systems*, 18(2):236–246, 2024.

[13] Antonio J. Rodríguez-Almeida, Carmelo Betancort, Ana M. Wägner, Gustavo M. Callicó, and Himar Fabelo. Incorporating uncertainty estimation and interpretability in personalized glucose prediction using the temporal fusion transformer. *Sensors*, 25(15):4647, 2025.

[14] Renat Sergazinov, Elizabeth Chun, Valeriya Rogovchenko, Nathaniel Fernandes, Nicholas Kasman, and Irina Gaynanova. Glucobench: Curated list of continuous glucose monitoring datasets with prediction benchmarks. *arXiv preprint arXiv:2410.05780*, 2024.

[15] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[16] Xiuding Cai, Yaoyao Zhu, Xueyao Wang, and Yu Yao. Mambats: Improved selective state space models for long-term time series forecasting. *arXiv preprint arXiv:2405.16440*, 2024.

[17] Md Atik Ahamed and Qiang Cheng. Timemachine: A time series is worth 4 mambas for long-term forecasting. *arXiv preprint arXiv:2403.09898*, 2024. 27th European Conference on Artificial Intelligence (ECAI 2024).

[18] Ameen Ali, Itai Zimerman, and Lior Wolf. The hidden attention of mamba models. *arXiv preprint arXiv:2403.01590*, 2024.

[19] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

[20] James M. Robins. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modelling*, 7(9):1393–1512, 1986.

[21] Xiao Wang and Li Yin. New g-formula for the sequential causal effect and blip effect of treatment in sequential causal inference. *Annals of Statistics*, 48(1):138–160, 2020.

[22] Frank A J L Scheer, Michael F Hilton, Christos S Mantzoros, and Steven A Shea. Adverse metabolic and cardiovascular consequences of circadian misalignment. *Proc. Natl. Acad. Sci. U. S. A.*, 106(11):4453–4458, March 2009.

[23] Yue Wu, Ben Ehlert, Ahmed A Metwally, Dalia Perelman, Heyjun Park, Andrew Wallace Brooks, Fahim Abbasi, Basil Michael, Alessandra Celli, Caroline Bejikian, Ekrem Ayhan, Yingzhou Lu, Samuel M Lancaster, Daniel Hornburg, Lucia Ramirez, David Bogumil, Sarah Pollock, Frank Wong, Denver Bradley, Georg Gutjahr, Ekanath Srihari Rangan, Tao Wang, Lettie McGuire, P Venkat Rangan, Helge Ræder, Zohar Shipony, Doron Lipson, Tracey McLaughlin, and Michael P Snyder. Individual variations in glycemic responses to carbohydrates and underlying metabolic physiology. *Nat. Med.*, 31(7):2232–2243, July 2025.

[24] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017.

[25] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[26] Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler-Canseco, and Artur Dubrawski. N-hits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7027–7035, 2023.

[27] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112, 2014.

[28] Melanie J. Davies, David A. D'Alessio, Judith Fradkin, Walter N. Kernan, Chantal Mathieu, Geltrude Mingrone, Peter Rossing, Apostolos Tsapas, Deborah J. Wexler, and John B. Buse. Management of hyperglycemia in type 2 diabetes, 2018. a consensus report by the american diabetes association (ada) and the european association for the study of diabetes (easd). *Diabetes Care*, 41(12):2669–2701, 10 2018.

[29] Emma Ahlqvist, Petter Storm, Annemari Käräjämäki, Mats Martinell, Mozhgan Dorkhan, Annelie Carlsson, Petter Vikman, Rashmi B Prasad, Dina Mansour Aly, Peter Almgren, Ylva Wessman, Nael Shaat, Peter Spégel, Hindrik Mulder, Eero Lindholm, Olle Melander, Ola Hansson, Ulf Malmqvist, Åke Lernmark, Kaj Lahti, Tom Forsén, Tiinamaija Tuomi, Anders H Rosengren, and Leif Groop. Novel subgroups of adult-onset diabetes and their association with outcomes: a data-driven cluster analysis of six variables. *Lancet Diabetes Endocrinol.*, 6(5):361–369, May 2018.

A Technical Appendices and Supplementary Material

A.1 Datasets

The Artificial Intelligence Ready and Exploratory Atlas for Diabetes Insights (AI-READI) is a multimodal, curated dataset of approximately 1,000 individuals spanning glycemic states from normoglycemia to prediabetes, type 2 diabetes managed with oral medications, and type 2 diabetes managed with insulin [8]. Inclusion criteria required participants to be over 40 years old and living with or without type 2 diabetes, while exclusion criteria specified that individuals with gestational diabetes or type 1 diabetes were not eligible. Summarized statistics for AI-READI, including demographics, clinical categories, data characteristics, and CGM-derived metrics, are provided in Table 2. Sex, race, and ethnicity were not available, as these sensitive variables are excluded from the public AI-READI release to reduce re-identification risk and potential stigmatization. For our analyses, we used 741 individuals with CGM, wearable activity, and clinical laboratory features from AI-READI.

The CGMacros dataset consists of 45 adults across normoglycemia, prediabetes, and type 2 diabetes, combining CGM with detailed dietary information [9]. Participants were monitored for approximately 10 days, during which they wore a Dexcom G6 CGM and logged descriptions and photographs of their meals, enabling the estimation of caloric intake and macronutrient composition. Summarized statistics for CGMacros, including demographics, clinical characteristics, data features, and CGM-derived metrics, are also presented in Table 2.

The similarities in both datasets such as demographics, clinical features, and CGM-derived metrics gave us confidence to train a meal detection model in CGMacros that can be deployed on the AI-READI dataset.

Table 2: Cohort characteristics for AI-READI and CGMacros. Values are mean $\pm$ SD unless otherwise indicated.

Characteristic	**AI-READI**	**CGMacros**
Demographics		
Sample size (N)	741	45
Age (years)	60 ± 11	48 ± 13
BMI (kg/m^2)	30.9 ± 8.4	31.1 ± 6.7
Clinical		
Normoglycemia	275 (37%)	15 (33%)
Prediabetes	163 (22%)	16 (36%)
T2D, non-insulin	213 (29%)	14 (31%)
T2D, insulin-treated	90 (12%)	Not assessed
HbA1c (%)	6.0 ± 1.3	6.1 ± 0.9
Data characteristics		
CGM device	Dexcom G6	Dexcom G6
Wearable device	Garmin Vivosmart 5	Not collected
Sampling interval	5 min	5 min
Recording duration (days)	9.6 ± 0.8	9.7 ± 0.9
Total measurements	2,045,164	123,318
CGM-derived metrics		
TIR (70–180 mg/dL, %)	85.8 ± 22.1	85.6 ± 20.0
Mean glucose (mg/dL)	138.6 ± 40.3	140.4 ± 28.3
CV (%)	19.6 ± 5.6	20.1 ± 5.4

T2D = type 2 diabetes; CGM = continuous glucose monitoring; TIR = time in range (70–180 mg/dL); CV = coefficient of variation.

A.2 Meal Detection (details complementing Sec. 2)

To supply meal covariates for AI-READI, which lacks meal annotations, we trained a meal-detection model on the PhysioNet CGMacros dataset [9] using 5-min CGM sampled into 6-h windows.

Problem setup. We frame meal detection as a binary event detection task from CGM-only time series sampled every 5 minutes. The signal is processed in sliding 6-hour windows (L=72 samples). For each timestep t in a window, the model outputs a probability $\hat{p}_t \in [0, 1]$ that a meal occurs at that moment. Because the ground-truth meal times are time-uncertain and the data are class-imbalanced (few meal minutes vs. many non-meal minutes), we convert point annotations into smoothed targets (See Trapezoidal label smoothing section). At inference, we convert the probabilistic sequence into a binary meal-flag sequence $\hat{y}_t \in 0, 1$ by thresholding within each window and reconciling overlaps with a local consensus rule (ratio voting), yielding a single timeline of detected meal events.

Model recap. We use a dilated 1D CNN (kernel sizes $15/3/3$; dilations $1/2/4$) to capture short/medium motifs, followed by a 2-layer BiLSTM (h=64 per direction) for temporal context, LayerNorm, dropout (0.4), and a linear head producing per-step logits (Figure 4). The loss is a weighted BCE-with-logits:

$$\mathcal{L} = -\frac{1}{L}\sum_{t=1}^{L}\big[\alpha\, \tilde{y}_t \log \sigma(\hat{y}_t) + (1-\tilde{y}_t)\log\big(1-\sigma(\hat{y}_t)\big)\big]\,,$$

where α=5 upweights positives to counter class imbalance, σ is the sigmoid, and $\tilde{y}_t$ the smoothed target ($\tilde{y}_t \in [0, 1]$).

Training Setup: Sampling 5-minute CGM; window length L=72; stride s=1; CNN channels `cout`=16; 2 BiLSTM hidden h=64 per direction; dropout 0.4; LayerNorm; Adam (lr 10^{-4}, weight decay 10^{-4}); OneCycleLR; batch size 16; 10 epochs; positive class weight α=5; subject-wise splits (no leakage) stratified by diabetes status (healthy, prediabetes, diabetes).

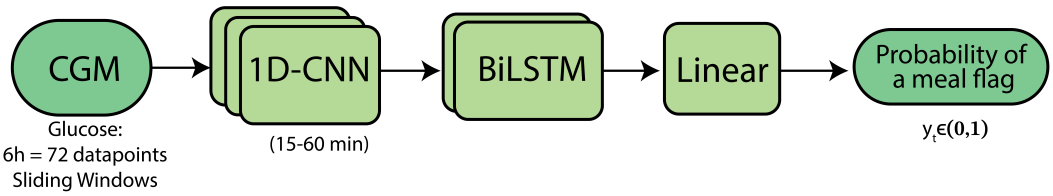

Figure 4: Meal detection model architecture

Trapezoidal label smoothing. Each annotated meal time τ_k is expanded into a trapezoid $w(t; \tau_k, P, R)$ with a central plateau (target = 1) and linear ramps (decay to 0):

$$w(t; \tau_k, P, R) = \begin{cases} 0, & |t-\tau_k| \geq P{+}R, \\ 1, & |t-\tau_k| \leq P, \\ 1 - \dfrac{|t-\tau_k| - P}{R}, & \text{otherwise.} \end{cases}$$

Intuition: the plateau (here $\approx$ 20 min total) tolerates small timing errors, while linear ramps encode decreasing responsibility farther from the reported meal. Accordingly, in our implementation we set P=2 (plateau length $\approx$ 10 min) and R=2 (ramp length $\approx$ 10 min). Within each 6 h window (L=72), we aggregate all trapezoids that overlap the window by summation and clip to keep targets bounded for BCE:

$$S_t \;=\; \sum_k w(t; \tau_k, P, R), \qquad \tilde{y}_t \;=\; \min\big(1,\, S_t\big).$$

In practice, meal overlaps are rare so the clipping is almost never activated and has negligible impact on supervision. (Alternative bounded unions such as $\max_k w(\cdot)$ behave similarly in our setting.)

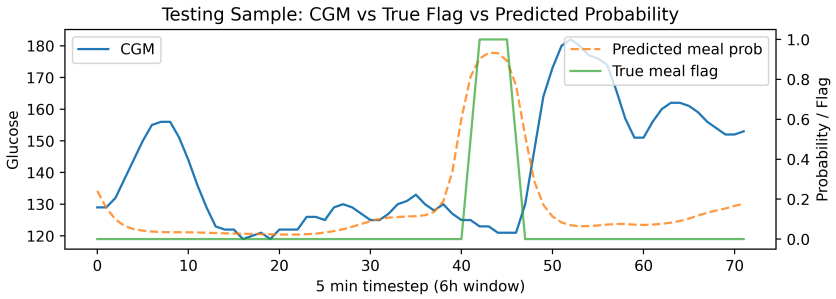

Figure 5: Predicted meal flag probability on test set window

Inference & reconstruction (thresholding + ratio voting). At test time, we slide windows with stride 1 and (i) apply a *per-window threshold* θ to convert probabilities (as shown in Figure 5) into binary predictions inside each window, then (ii) *merge overlaps* at each global index t via *ratio voting*:

$$\text{ratio}(t) \;=\; \frac{|\{\text{covering windows that predict 1 at } t\}|}{|\{\text{windows covering } t\}|}, \qquad \hat{y}_t = \mathbb{1}\big[\text{ratio}(t) \geq \rho\big].$$

Here, θ controls how strict a window is before merging; ρ controls how strong the local consensus must be across overlapping windows. Compared with averaging probabilities on overlaps, ratio voting acts as a local majority rule that (a) is robust to isolated false positives and timing jitter, and (b) better preserves short impulses that can be over-smoothed by heavy overlap.

Event-level metrics. Let $\mathcal{T}$ (truth) and $\mathcal{P}$ (predicted) be sets of maximal 1-segments; "overlap" means non-empty interval intersection. We report

$$\text{Recall}_{\text{evt}} = \frac{|\{\, e \in \mathcal{T} : \exists p \in \mathcal{P} \text{ overlap}(e,p) \,\}|}{|\mathcal{T}|}, \qquad \text{Precision}_{\text{evt}} = \frac{|\{\, p \in \mathcal{P} : \exists e \in \mathcal{T} \text{ overlap}(e,p) \,\}|}{|\mathcal{P}|}.$$

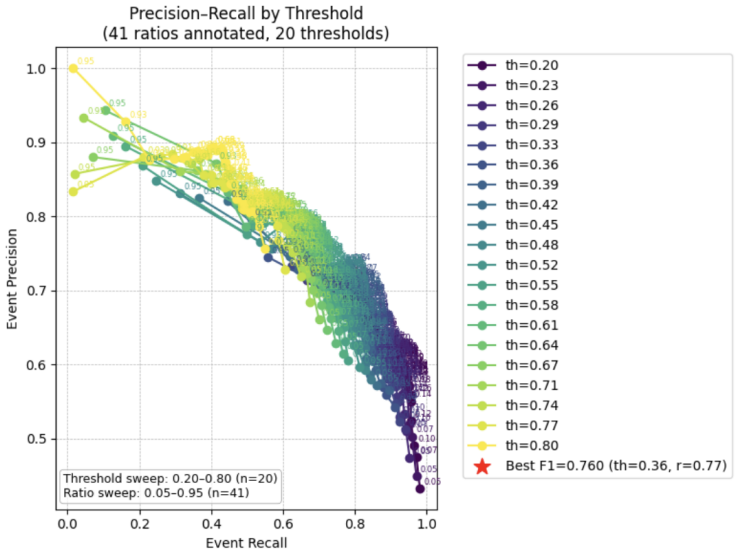

Figure 6: Validation sweep over threshold θ and ratio ρ.

Final Hyperparameters and Result **Selecting θ and ρ.** We sweep (θ, ρ) on the validation set to maximize the *event-level* F_β score (β=1) (As shown in Figure 6). The search clarifies the trade-off: lower θ increases sensitivity within windows; higher ρ demands stronger cross-window agreement to call an event. This joint tuning is crucial under heavy overlap (s=1), where naive averaging can under-detect brief events.

Selected hyperparameter (validation): θ=0.36, ρ=0.77, yielding:

$$\text{Recall}_{\text{evt}} = \mathbf{80.13}\%$$
$$\text{Precision}_{\text{evt}} = \mathbf{72.08}\%$$

Rationale for Meal Flag Deployment. We state that the meal flag prediction model is sufficient for deployment on AI-READI since its performance achieves a practical trade-off between precision (minimizing false positives, not labeling non-meal periods as meals) and recall (minimizing false negatives, capturing the majority of true meal periods). Specifically, the predicted meal flags align with expected physiological patterns of meal-induced glucose rises, providing a useful proxy for meal timing in a dataset where direct meal records are unavailable (Figure 7). Importantly, our aim is not to infer exact meal size or macronutrient composition, but to generate a coarse-grained signal for downstream modeling and interpretation. Given these considerations, we view the current level of performance as adequate to support as a predicted meal flag covariate in AI-READI.

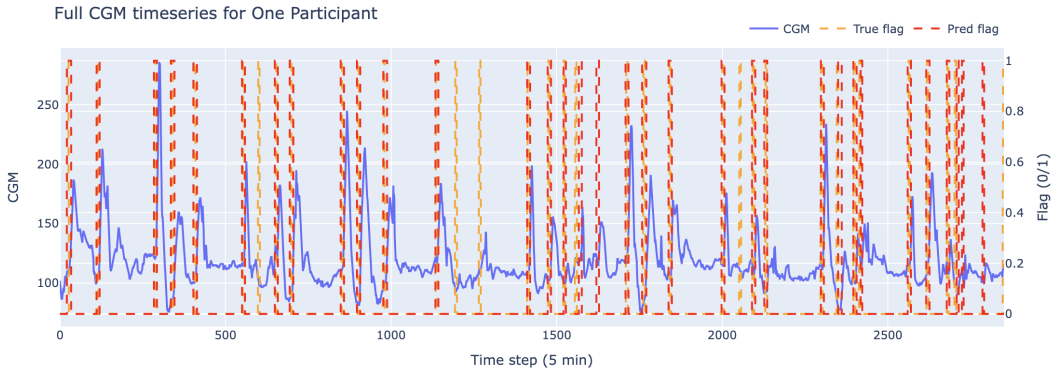

Figure 7: Full timeseries CGM meal flag predictions on one test participant after thresholding and ratio voting (CGMacros)

A.3 Quick Comparisons

We initially compared several forecasting approaches on our CGM forecasting objective, including a persistent baseline (last value carried forward), a classical gradient boosting model (LightGBM), and neural forecasting models (DeepAR, NHITS, Seq2Seq LSTM, TFT) on chosen hyperparameters and feature engineering based on prior work [24, 25, 26, 27, 10]. As shown in Table 3, all methods substantially outperformed the trivial persistent baseline, confirming the importance of modeling temporal structure.

Among the neural methods, TFT achieved the best performance. In our setting—24-hour context and a 1-hour horizon with rich known-future covariates (e.g., time-of-day encodings, planned activity) and meaningful static traits (e.g., diabetes status)—TFT is particularly well aligned with the task: variable selection and attention fuse static, past, and known-future inputs, which likely explains its better accuracy. For this reason, combined with its strong accuracy, we selected TFT as the primary comparator for detailed analyses in this study and as the architecture inspiration for SSM-CGM.

Table 3: Quick benchmarking across forecasting models. Mean absolute error (MAE; mg/dL) on AI-READI CGM data using a 24-hour context length and 1 hour horizon.

Model	**MAE (mg/dL)**
Persistent (Last Value Carried Forward)	12.30
LightGBM	9.70
DeepAR	8.70
NHITS	8.47
Seq2Seq LSTM	7.85
TFT (333k params)	**7.27**

A.4 Architecture

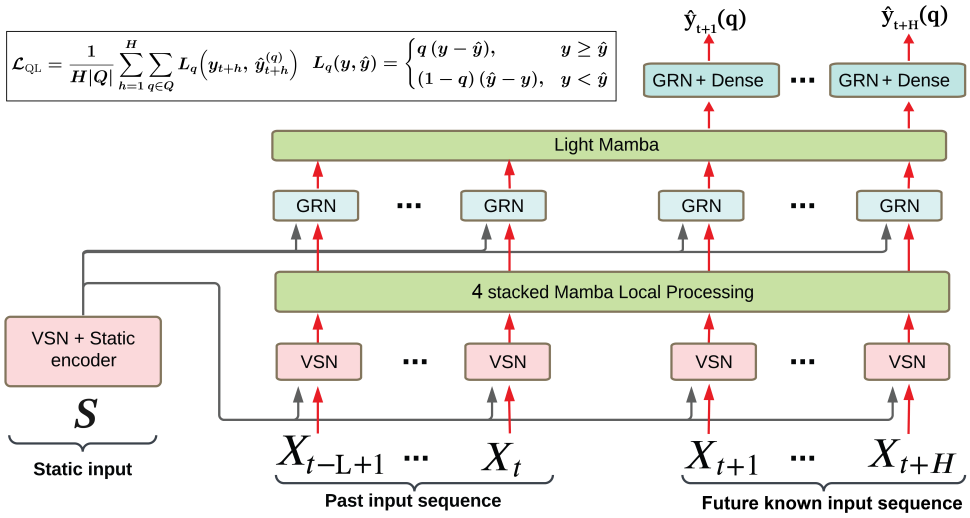

Figure 8: SSM-CGM overview: embeddings, Variable Selection Networks (static & time-varying), SSM temporal processors (Mamba-2), static enrichment, and multi-horizon quantile head.

Notation. At time t, we have historical covariates $\mathbf{x}^{(\text{hist})}_{t-L+1:t}$, known future covariates $\mathbf{x}^{(\text{fut})}_{t+1:t+H}$, static features $\mathbf{s}$, and targets $\mathbf{y}_{t+1:t+H}$. Per-variable embeddings are $e_{i,t} \in \mathbb{R}^{d_x}$.

A.5 Gated Residual Network (GRN)

Intuition and purpose. A GRN is a compact block that *stabilizes training* and *controls how much nonlinearity* is injected into a representation. It combines (i) a nonlinear transform, (ii) a learnable gate deciding how much of that transform to keep, and (iii) a residual connection with normalization. This yields better gradient flow, robustness to scale/unit mismatches, and an easy way to *condition* a variable by a *context* (e.g., static features). In **SSM-CGM**, GRNs make variable selection and fusion smoother, more stable, and person-aware.

Equations. Let $u \in \mathbb{R}^d$ be the block input (e.g., a per-variable embedding) and $c \in \mathbb{R}^{d_c}$ an optional conditioning vector (e.g., the static context). We write $[u; c]$ for concatenation.

$$\hat{z} = W_2\,\phi\big(W_1[u;c] + b_1\big) + b_2, \tag{4}$$

$$g = \sigma\big(W_g[u;c] + b_g\big), \tag{5}$$

$$z = \text{LN}\big(u + g \odot \hat{z}\big). \tag{6}$$

Here, ϕ is a nonlinearity (ELU/GELU), σ is the logistic sigmoid, $\odot$ is elementwise multiplication, and LN is LayerNorm. *Limiting behavior:* if $g \approx 0$, the block passes u through unchanged (stable linear path); if $g \approx 1$, the nonlinear transform $\hat{z}$ is fully applied. If $\dim(u) \neq \dim(\hat{z})$, replace u by a projected skip $W_{\text{skip}}u$ in (6). Dropout can be inserted after (4) without changing the form.

Why it helps in SSM-CGM. (a) *Context injection*: the gate g learns *how much* of the static context c to blend into each time step; (b) *Scale robustness*: LN + gating tame unit/scale disparities across covariates; (c) *Regularization*: the gate can suppress unnecessary nonlinearities, reducing overfitting while preserving capacity when needed.

A.6 Variable Selection Networks (VSN)

Intuition and purpose. A VSN learns, at each step, an *interpretable simplex of weights* over variables: $\alpha_{i,t} \in [0,1]$ with $\sum_i \alpha_{i,t} = 1$. This serves two goals: (i) principled *fusion* of heterogeneous covariates (different scales/reliability) into a single representation, and (ii) per-step *interpretability* ("which variables did the model rely on?"). A similar VSN on *static* features produces a *context* c that personalizes all downstream computations.

Static VSN → context c. For static features $\{s_k\}_{k=1}^{K}$:

$$e_k^{(s)} = E_k^{(s)}\, s_k, \tag{7}$$

$$\tilde{e}_k^{(s)} = \mathrm{GRN}\big([e_k^{(s)};\ \bar{s}]\big), \tag{8}$$

$$\beta_k = \mathbf{v}_s^{\top} \tilde{e}_k^{(s)} + a_s, \qquad \alpha_k^{(s)} = \frac{\exp(\beta_k)}{\sum_{j=1}^{K} \exp(\beta_j)}, \tag{9}$$

$$c = \sum_{k=1}^{K} \alpha_k^{(s)}\, P_k^{(s)}\, e_k^{(s)}. \tag{10}$$

$E_k^{(s)}$ and $P_k^{(s)}$ are small learned projections; $\bar{s}$ may be a simple summary of static variables (or omitted); and $\mathbf{v}_s \in \mathbb{R}^d$ and $a_s \in \mathbb{R}$ are learned parameters (scoring vector and bias) of the static VSN, with $d = \dim(\tilde{e}_k^{(s)})$. The simplex weights $\alpha_k^{(s)}$ reveal which static traits the model relies on to form the context c.

Temporal VSN → per-step fused representation. For V time-varying variables $\{x_{i,t}\}_{i=1}^{V}$ (CGM + time-varying covariates):

$$e_{i,t} = E_i\, x_{i,t}, \tag{11}$$

$$\tilde{e}_{i,t} = \mathrm{GRN}\big([e_{i,t};\ c]\big), \tag{12}$$

$$s_{i,t} = \mathbf{v}^{\top} \tilde{e}_{i,t} + a, \qquad \alpha_{i,t} = \frac{\exp(s_{i,t})}{\sum_{j=1}^{V} \exp(s_{j,t})}, \tag{13}$$

$$r_t = \sum_{i=1}^{V} \alpha_{i,t}\, P_i\, e_{i,t}. \tag{14}$$

E_i and P_i are small learned projections; c is the static context vector produced by the static VSN; and $\mathbf{v} \in \mathbb{R}^d$ and $a \in \mathbb{R}$ are learned parameters (scoring vector and bias) of the temporal VSN, with $d = \dim(\tilde{e}_{i,t})$. The simplex weights $\alpha_{i,t}$ identify, at each step, which time-varying variables the model conditionally relied on, and the fused vector r_t is then fed to the temporal processor.

Why it helps in SSM-CGM. (i) *Personalization*: via c, per-step mixtures adapt to participant traits (age, diabetes status, site); (ii) *Robust fusion*: softmax over GRN-conditioned scores downweights noisy or poorly scaled variables; (iii) *Interpretability*: trajectories of $\alpha_{i,t}$ (aggregated per subject/context) indicate *which* signals the model conditionally relied on (associational, not causal), supporting analysis and recommendations.

A.7 State-space temporal processor (Mamba)

Mamba performs linear-time, streaming sequence processing via a stable state-space scan:

$$h_{t+1} \;=\; A_t\, h_t \;+\; B_t\, x_t, \qquad A_t \;=\; \exp\!\big(\Delta t_t\, \Lambda_t\big), \quad \operatorname{Re}\lambda_i(\Lambda_t) \le 0, \tag{15}$$

$$z_t \;=\; C_t\, h_t \;+\; D\, x_t. \tag{16}$$

Here, x_t denotes time-varying inputs, h_t the SSM state, and z_t the readout. The *selective step* $\Delta t_t > 0$ modulates both A_t and B_t; A_t, B_t, and C_t are light input-conditioned projections. In standard Mamba, Λ_t is (block-)diagonal and therefore A_t is (block-)diagonal; $D\, x_t$ is an optional direct/skip term (omitted in later equations). A_t acts as a per-channel forget/filter; Δt_t sets the local time scale.

Hidden attention from the SSM. To attribute *when* the model relied on the past, we unroll the *state path* (omit D so the instantaneous skip at lag 0 does not dominate). For any output index ℓ:

$$z_\ell = \sum_{j \le \ell} K_{\ell j}\, x_j, \qquad K_{\ell j} \;=\; C_\ell \Big(\prod_{t=j}^{\ell-1} A_{\Delta t_t}\Big) B_j \;=\; C_\ell \exp\!\Big(\sum_{t=j}^{\ell-1} \Delta t_t\, \Lambda_t\Big) B_j. \tag{17}$$

We summarize the kernel $K_{\ell j}$ into a nonnegative scalar using the *channelwise* ℓ_1 norm, $\psi(K_{\ell j}) \equiv \|K_{\ell j}\|_{1,\mathrm{ch}}$, and define a *lag-importance* distribution with $\varepsilon = 10^{-9}$ and $\tau = 1$:

$$\alpha_{\ell j} \;=\; \operatorname*{softmax}_{j \le \ell}\Big(\log\big(\psi(K_{\ell j}) + 10^{-9}\big)\Big). \tag{18}$$

The weights $\alpha_{\ell j}$ act like a *hidden attention* over past lags j for each forecast index ℓ.

A.8 Mamba-2 MES-style

Motivation. Mamba-2 factorizes channels into multiple heads with grouped states and an explicit (A, B, C, D) parameterization, improving throughput and long-range capture. To make head-wise attributions *comparable*, we *share* the value space and the readout across heads, analogous in spirit to the *interpretable multi-head self-attention* of TFT where common value projections ease cross-head comparison.

Per-head SSM and shared projections. Let H be the number of heads. We introduce a shared value projection $V : \mathbb{R}^{d_x} \to X$ and a shared readout $C_{\mathrm{sh}} : \mathbb{R}^{d_h} \to \mathbb{R}^{d_z}$. Each head h has its own stable dynamics and expansion:

$$v_t = V\, x_t, \qquad v_t \in X, \tag{19}$$

$$h_{t+1}^{(h)} = A_{\Delta t_t}^{(h)}\, h_t^{(h)} + B^{(h)}\, v_t, \qquad A_{\Delta t_t}^{(h)} = \exp\big(\Delta t_t\, \Lambda_t^{(h)}\big), \tag{20}$$

$$z_t^{(h)} = C_{\mathrm{sh}}\, h_t^{(h)} \;+\; D\, x_t, \qquad z_t = \sum_{h=1}^{H} z_t^{(h)}. \tag{21}$$

The skip D is shared (or omitted). Unrolling the *state path* yields the head-wise kernel

$$K_{\ell j}^{(h)} \;=\; C_{\mathrm{sh}} \left(\prod_{t=j}^{\ell-1} A_{\Delta t_t}^{(h)} \right) B^{(h)} \;=\; C_{\mathrm{sh}} \exp\Big(\sum_{t=j}^{\ell-1} \Delta t_t\, \Lambda_t^{(h)} \Big) B^{(h)}, \tag{22}$$

and the output decomposition

$$z_\ell \;=\; \sum_{h=1}^{H} \sum_{j \le \ell} K_{\ell j}^{(h)}\, v_j. \tag{23}$$

Shared value projection V. We tested several ways to define V: (i) a squeeze-and-excitation (SE) mechanism across heads—first averaging over the SSM channel dimension P to obtain $s_t \in \mathbb{R}^H$, then applying a small MLP $H \to H/r \to H$ and a softmax over H to weight and aggregate heads; (ii) head-wise max pooling; and (iii) a simple head-wise mean. We ultimately adopt the head-wise mean as the shared value because it is simple, stable, and effective:

$$v_t \;=\; \tfrac{1}{H} \sum_{h=1}^{H} x_t^{(h)} \in \mathbb{R}^P.$$

This choice is a pragmatic baseline, and richer value-sharing schemes (e.g., learned head gating or data-dependent projections) are left for future work.

Head-wise hidden attention and comparability. Because *both* the value projection V and the readout C_{sh} are shared, the scalar summaries $\psi\big(K_{\ell j}^{(h)}\big)$ live on a common scale across heads. We define per-head lag-weights

$$\alpha_{\ell j}^{(h)} \;=\; \operatorname*{softmax}_{j \le \ell} \left(\frac{\log\big(\psi(K_{\ell j}^{(h)}) + \varepsilon\big)}{\tau} \right), \tag{24}$$

and, if desired, a head-aggregated map

$$\bar{\alpha}_{\ell j} \;=\; \operatorname*{softmax}_{j \le \ell} \left(\frac{1}{\tau} \log\Big(\tfrac{1}{H} \sum_{h=1}^{H} \psi(K_{\ell j}^{(h)}) + \varepsilon \Big) \right). \tag{25}$$

This mirrors TFT's interpretable multi-head attention: by fixing the *value* (V) and the *readout* (C_{sh}) across heads, differences in $\alpha_{\ell j}^{(h)}$ reflect *temporal dynamics* (via $\Lambda_t^{(h)}, B^{(h)}$), not arbitrary head-specific value/readout scalings.

Practical notes. We parameterize each $\Lambda_t^{(h)}$ to have nonpositive real parts (e.g., diagonal with $\Lambda_{kk}^{(h)} = -\exp(\theta_{kk}^{(h)})$) to ensure stability; τ tunes sharpness of the attribution maps; ψ used is the channelwise ℓ_1 norm. Hidden-attention maps (18, 24, 25) are *associational* and complement VSN importances: together they answer *what* (variables) and *when* (lags) the model relied on for CGM forecasting.

A.9 Benchmarking protocol, fairness, and reproducibility

Goal. We compared **SSM–CGM** against a strong baseline (Temporal Fusion Transformer, **TFT**) under *matched budget*, *identical data/views*, and *aligned hyperparameter axes*. Any performance difference should stem solely from the temporal processor (SSM/Mamba vs. LSTM+attention).

Setup and parity. We use the same AI–READI cohort, features, and splits (Sec. 2) with identical preprocessing. The task is a multi–horizon forecast ($H{=}12$ steps at 5 min) with the same covariate interface (static / time–varying historical / time–varying future) and context lengths of 12 h, 24 h, and 48 h. **TFT** (via `pytorch-forecasting`) uses VSNs, LSTMs, interpretable multi–head attention, and a quantile head; **SSM–CGM** keeps the exact same interface/head and replaces only the temporal stack with Mamba–2 (MES–style). Both models optimize the same QuantileLoss ($q \in \{0.1, 0.5, 0.9\}$) with Adam, early stopping (patience 15), a 30–epoch cap, and identical validation cadence(5 times per epoch) /batching/teacher forcing; we report QuantileLoss, MAE, and RMSE (Table 1).

Table 4: Best hyperparameters used in all reported results.

Setting	**TFT (baseline)**	**SSM–CGM (Mamba–2, MES)**
	Architecture	
Hidden size d_{model}	128	128
Attention heads	8	8 (expand=4, headdim=64)
Temporal depth	LSTM layers=4	Mamba depth=4
Dropout	0.2	0.2
d_{state}	—	128
d_{conv}	—	8
Expand factor	—	4
Head dimension	—	64
Groups (`ngroups`)	—	1
Parameter count	1,793,376	1,762,714
	Training	
Loss	QuantileLoss $q \in \{0.1, 0.5, 0.9\}$	
Learning rate	1×10^{-3}	1×10^{-3}
Batch size	32	32
Max epochs	30	30
Early stopping patience	15	15
Validation check interval	0.2 (per epoch)	0.2 (per epoch)
	Runtime / system	
Hardware	4×A100 (DDP), TF32 enabled	
Chunk size	—	128
dt limit	—	$(10^{-3}, 10)$
Learnable init states	—	True

Hyperparameter search. Because **SSM–CGM** is designed to match the **TFT** interface, we use the *same axes and ranges* for both models (e.g., LSTM depth for TFT mirrors Mamba depth for SSM). We run an *random search* with *∼25 configurations per model*, sampling uniformly over the shared space below. Remaining Mamba–specific settings are fixed based on prior Mamba literature (see Table 4):

$$\begin{aligned}
&\text{Hidden size } d_{\text{model}} \in \{64, 128\},\\
&\text{Heads (TFT) / head–dim config (SSM)} \in \{2, 4, 8\},\\
&\text{Depth} \in \{1, 2, 4\} \quad (\text{LSTM layers for TFT; Mamba depth for SSM}),\\
&\text{Learning rate} \in \{\,3\times10^{-3},\ 1\times10^{-3}\},\\
&\text{Batch size} \in \{32, 64\}.
\end{aligned}$$

The primary selection criterion is validation QuantileLoss; ties break by MAE, then RMSE, then parameter count. The best configuration found on the 24 h context has been transferred to 48 h and

12 h for both models. Notably, the best settings are *identical* for TFT and SSM–CGM (Table 4), consistent with the aligned axes and shared interface (that makes similar number of params: 2% less for SSM CGM).

Runs use 4×A100 40GB (DDP) on Google Cloud; TF32/mixed precision follow vendor defaults; checkpoints are saved every epoch. On the 48 h context, training time per epoch was approximately 1.5 hours on these 4 GPU for both SSM–CGM and TFT.

Limitations. We transfer hyperparameter to 12 h and 48 h context without a dedicated retuning to control compute. This could under–optimize either model for others context length. We mitigate this by enforcing strict budget symmetry and aligned axes. A targeted 48 h (and 12 h) refinement is left for future work.

A.10 VSN Interpretability

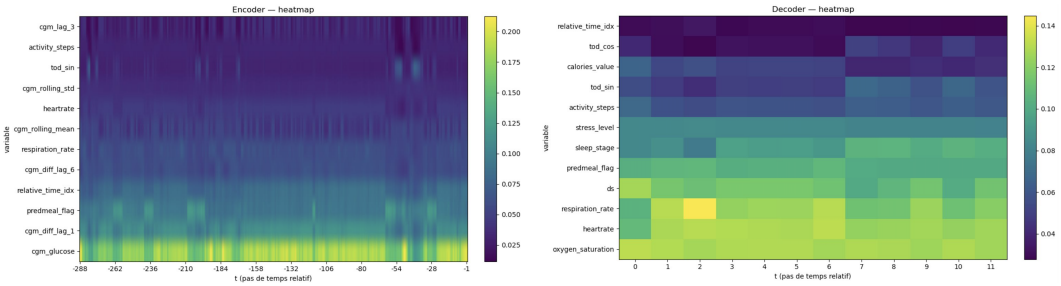

Figure 9: VSN variable by time importance for the encoder and the decoder

These VSN results indicate that wearable covariates are informative in both the encoder and the decoder. Their time-varying weights reveal when the model conditionally relies on each signal to form its forecast. In the individual example shown in Figure 9, we observe substantial within-subject variability over time in both stages. For instance, peaks in the weight assigned to the predicted-meal flag within the encoder align with when meal flags are present, indicating that when a meal is likely, the model allocates more mass to this covariate and correspondingly shifts its forecast.

A.11 Temporal attention Interpretability

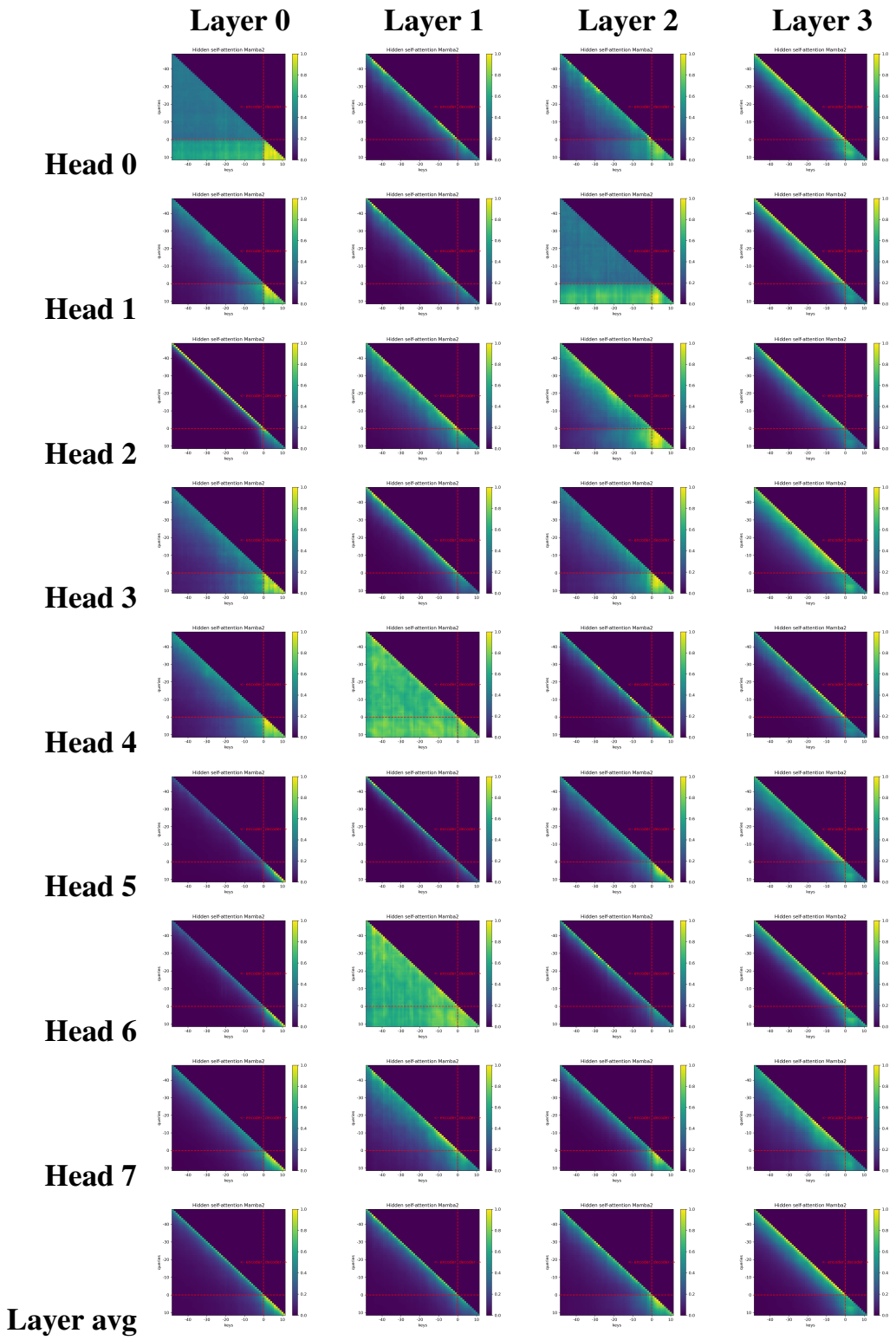

Figure 10: Average hidden self-attention maps over the last 60 time steps across all individuals, by Mamba head and layer depth (stacked Mamba).

Hidden-attention maps derived from the SSM unrolling across all Mamba layers provide a fine-grained view of *when* past information is used for each prediction step. At the individual level, these maps expose specific lag structures; at the cohort level, layer- and head-wise averages clarify complementary roles across the stack.

Analysis of the maps in Figure 10 gives intuition on the division of temporal labor across the stacked Mamba: earlier layers appear to emphasize very recent lags that stabilize short-horizon decoder steps, whereas deeper layers tend to place more mass further back in the encoder window, consistent with slower dynamics and circadian structure, before passing representations to the light-Mamba. This pattern is reflected by head-wise maps that partition the temporal context within each layer and by layer-average rows showing a progressive broadening of the effective receptive field.

Figure 2C displays the light-Mamba self–hidden-attention over the last 60 time steps. Comparing the cohort-average map with individual maps (Figure 11) shows that certain past segments are consistently retained and influence the decoder, while others are rapidly down-weighted. This pattern is consistent with CGM dynamics in which very recent history dominates, punctuated by salient events that remain influential over longer lags.

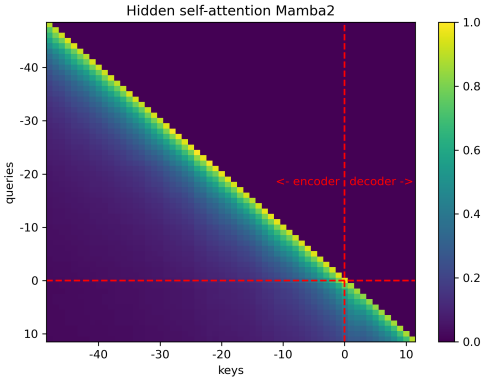

Figure 11: Average hidden self-attention, over the last 60 time steps, across all participants (light Mamba).

A.12 Interpretability - Static Covariates

Static features in SSM-CGM propagate information across sequential inputs through a static variable selection network (VSN) combined with a static encoder that handles both categorical and continuous features (Figure 1). The static VSN enables computation of feature importances for each static feature (Table 5). Participant ID had the highest feature importance, indicating that individual variation in glucose dynamics is important to encode for forecasting CGM. The next most important features were variance in the glucose levels (glucose scale) and diabetes status suggesting that capturing differences in participants with high vs low glycemic variability improves forecasting. Other static features such as age, and clinical site also show importance, though to a lesser extent.

Table 5: Static Variable Importances

Variable	Importance
participant id	0.4707
glucose scale	0.2553
diabetes status	0.0921
encoder length	0.0729
age	0.0439
clinical site	0.0338
glucose center	0.0313

In addition to static feature importances, we looked into if the forecasting errors of SSM-CGM differ across diabetes status, age, and clinical site (Figure 12). We find that on average, people with diabetes (either insulin or oral medications) have higher mean absolute error (MAE) than people who are healthy or have prediabetes. Additionally, we suspect people with diabetes taking insulin have a larger MAE in the SSM-CGM model since the more unpredictable glucose dynamics are attributed to not having information of when individuals take insulin in the AI-READI dataset. On average, we see people who are older tend to have a higher MAE but the effect is small (r = 0.09, p-val = 0.019). For clinical site, we observe that the global average MAE is within the confidence bands of the sites used participant recruitment. This is promising as clinical site is an observable batch effect feature in the AI-READI dataset.

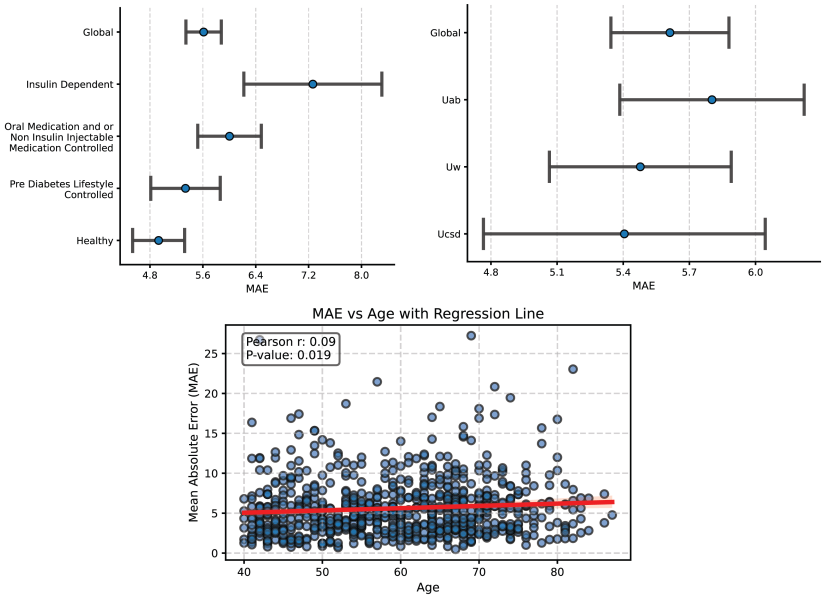

Figure 12: MAE error by category in static features.

A.13 Interpretability - Embeddings

To reason whether SSM-CGM captures glucose dynamics we utilized the static embeddings of diabetes status and participant ID (Figure 13). Cosine similarity revealed that embeddings of healthy individuals were distinct from those of individuals with diabetes, while people with diabetes on different medications (insulin vs. oral) shared similar embeddings (sim = 0.37). Individuals with prediabetes were closer to healthy participants than to those with diabetes on either oral or insulin therapy. This ordering reflects clinical knowledge: healthy and individuals with prediabetes generally maintain better glucose control than people with type 2 diabetes, with those on insulin, typically a last-line treatment, having the most difficult-to-control glucose [28].

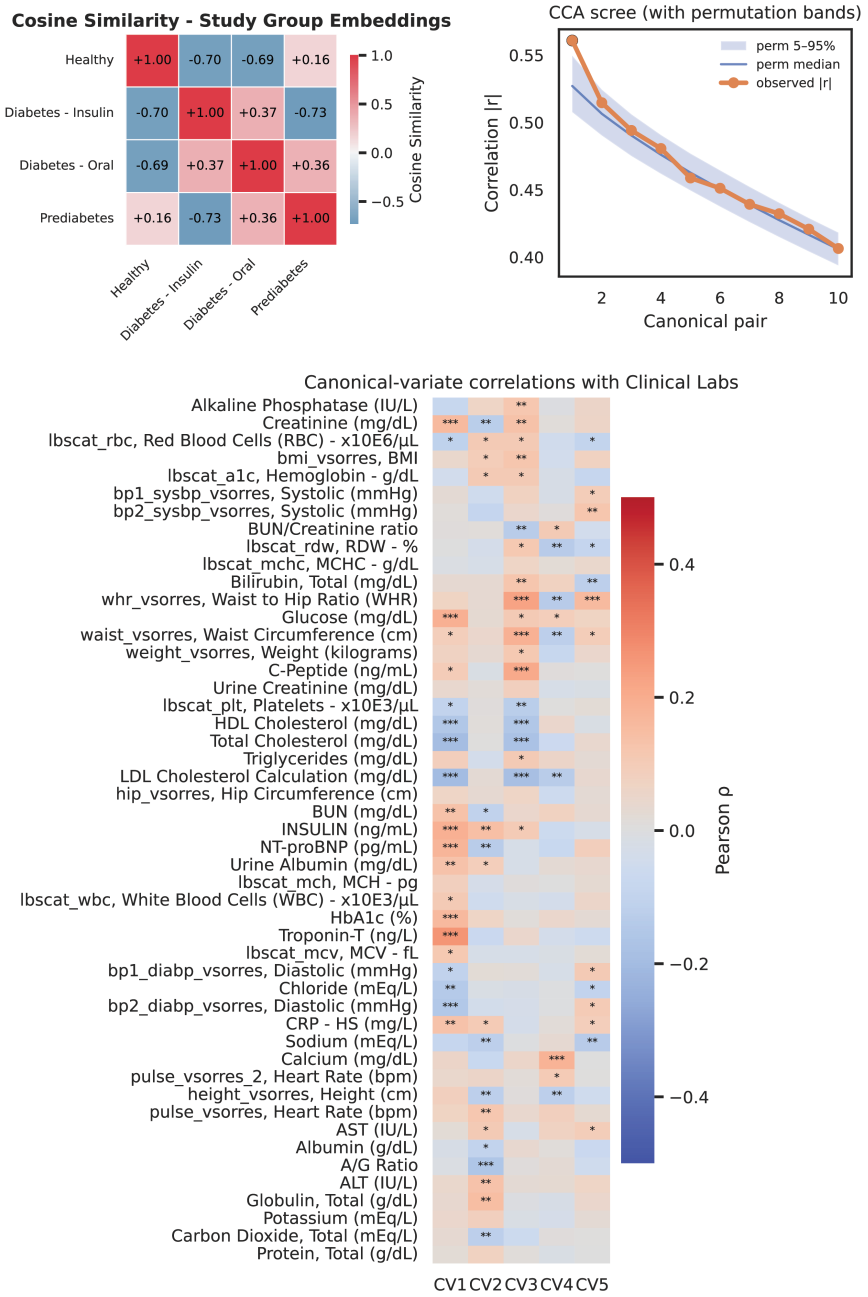

Figure 13: Interpreting participant ID embeddings of SSM-CGM.

To further understand what SSM-CGM is learning, we took the individual static embeddings and utilized canonical correlation analysis (CCA) to compare if the variation in individual embeddings share similar axes of variation with standard clinical lab values (Figure 13). For robustness, we utilized permutation testing (n = 10,000) samples to identify a null distribution of expected range of correlations under no true association. We observed that the 1st canonical pair has an observed canonical correlation above the null confidence interval suggesting a robust signal between SSM-CGM individual embeddings with clinical lab features, while the 2nd-5th canonical pairs display correlations that might be observed by random chance. Based on this finding, we correlated clinical lab values with the canonical-variates of the SSM-CGM embedding to observe which lab values are corresponding to each canonical pair CV1-5. For CV1, troponin, HbA1c, insulin, glucose, and cholesterol features tend to be explain the signals shared between the individual embeddings captured and the clinical lab values. This variation likely reflects glycemic traits and characteristic metabolic signatures associated with diabetes [29].

A.14 Identification Assumptions for Counterfactual Forecasting

Objective. Counterfactual forecasting in our work corresponds to planned interventions on future covariates (e.g., heart rate, respiration). Formally, the goal is to estimate how future glucose trajectories would change if we were to set these covariates to specified values, irrespective of their natural dynamics.

Sequential g-formula. Under assumptions of consistency, sequential ignorability, and positivity, the counterfactual distribution of a planned covariate sequence $\mathbf{a}_{t+1:t+h}$ is identified by the sequential *g-formula*:

$$p(\mathrm{CGM}_{t+1:t+h}(\mathbf{a}) \mid \mathcal{H}_t) = \prod_{s=t+1}^{t+h} p(\mathrm{CGM}_s \mid \mathcal{H}_{s-1}, X_s = a_s), \tag{26}$$

where $\mathcal{H}_{s-1}$ is updated with simulated $\mathrm{CGM}_{t+1:s-1}$ and fixed $X_{t+1:s-1} = \mathbf{a}_{t+1:s-1}$ under this recursive definition.

In our work, **SSM-CGM** directly conditions on the entire planned sequence $\mathbf{a}_{t+1:t+h}$ in a single forward pass, yielding

$$\widehat{p}_\theta(\mathrm{CGM}_{t+1:t+h} \mid \mathcal{H}_t, \mathbf{a}_{t+1:t+h}), \tag{27}$$

which is *equivalent in principle* to the g-formula factorization but avoids explicit recursive simulation.

Identification Assumptions. Identification of counterfactual CGM forecasts requires the standard assumptions of *consistency*, *sequential ignorability*, and *positivity*. While these assumptions are formally untestable from observational data, we provide supporting evidence and domain-specific reasoning suggesting they are plausible in our setting. For a deterministic future intervention plan $\mathbf{a}_{t+1:t+h}$, these can be written in sequence-level notation as follows:

Consistency. If the observed covariates equal the planned intervention sequence, then the observed outcome trajectory equals the counterfactual trajectory:

$$\text{If } X_{t+1:t+h} = \mathbf{a}_{t+1:t+h}, \quad \text{then } \mathrm{CGM}_{t+1:t+h}(\mathbf{a}) = \mathrm{CGM}_{t+1:t+h}. \tag{28}$$

Consistency requires that when the planned intervention path coincides with the observed covariates, the counterfactual glucose trajectory equals the observed trajectory. Consistency is supported by the use of objective sensor-based measurements (e.g. heart rate, respiration rate, glucose levels, etc.), which ensure that observed and planned values are defined on the same measurement scale. Our counterfactual forecasting procedure implements this assumption directly. When the planned action sequence equals the observed one, the counterfactual forecast reduces to the model's standard prediction of the observed trajectory. Although forecasts do not need to equal the realized trajectory exactly, this construction aligns the model with the formal consistency assumption.

Sequential Ignorability (No Unmeasured Confounding). Conditional on the observed history, future counterfactual glucose trajectories are independent of planned actions:

$$\mathrm{CGM}_{t+1:t+h}(\mathbf{a}) \perp\!\!\!\perp \mathbf{a}_{t+1:t+h} \mid \mathcal{H}_t. \tag{29}$$

This assumption requires that, conditional on the observed history $\mathcal{H}_t$, the planned intervention X_s is independent of future counterfactual glucose trajectories. In our setting, this assumption is more plausible because we treat HR, RR, and activity measures as the direct intervention variables. By intervening directly on these sensor-based covariates, rather than on upstream unobserved causes (e.g., fitness, hormones), we reduce the scope for hidden confounding. The inclusion of rich histories (CGM, activity, meals, HR) further mitigates omitted-variable bias, making sequential ignorability more credible in this framework.

Positivity. For any history that occurs with positive probability, all planned actions must have positive probability of being observed:

$$\Pr(\mathbf{a}_{t+1:t+h} \mid \mathcal{H}_t) > 0 \quad \text{whenever } p(\mathcal{H}_t) > 0. \tag{30}$$

We empirically assess positivity, by examining the distributions of two representative covariates, heart rate (HR; Figure 14) and respiration rate (RR; Figure 15), across times of day in the AI-READI cohort.

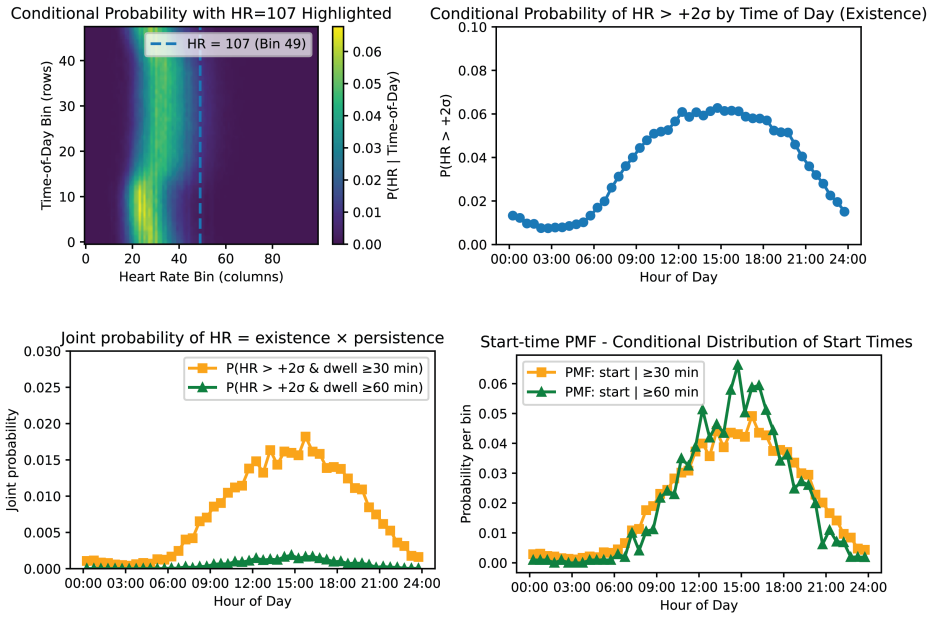

Figure 14: Positivity of elevated heart rate (HR) across the day.

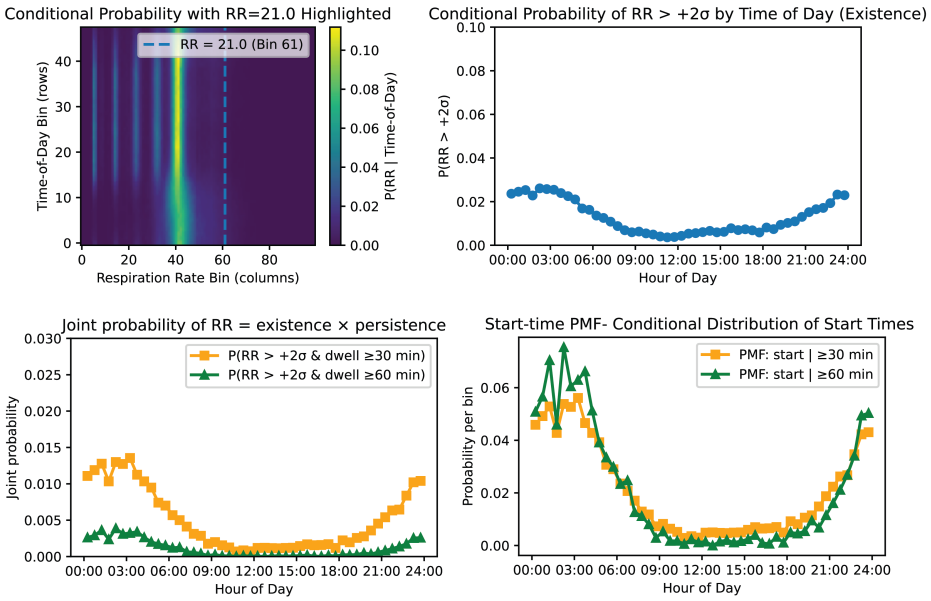

Figure 15: Positivity of elevated respiration rate (RR) across the day.

Both covariates showed support across physiologically plausible ranges, with conditional probabilities $p(a_{t+1:t+h} \mid \text{Time of Day}) > 0$ throughout. For example, a $+2$ SD HR perturbation (107 bpm) was most commonly observed between 9 AM and 9 PM, consistent with daytime activity, while a $+2$ SD RR perturbation (21 breaths/min) was most supported between 9 PM and 6 AM, reflecting nighttime respiration. These patterns demonstrate that our counterfactual interventions, restricted to times of day where such values are commonly observed, lie within empirically supported regions of covariate space. Importantly, these distributions also show persistence: elevated HR or RR lasting 30–60 minutes remains plausible (Figures 14, 15). Therefore, moderate shifts are well-represented across contexts, satisfying positivity, whereas extreme or physiologically implausible regimes (e.g., very high HR during sleep) would violate positivity and are excluded from analysis.

A.15 Limitations and Future work

Our study has several limitations spanning the dataset, modeling framework, interpretability, and clinical scope.

Dataset-related limitations. The AI-READI dataset lacks direct annotations for meals and medication use (e.g., insulin, metformin, GLP1), both of which strongly influence glucose dynamics. We addressed the absence of meal records by training a meal-detection model on an external dataset, but this provides only a coarse proxy for meal timing and does not capture meal size or composition. Improving meal detection through higher-quality annotations will be important for refining future models. Similarly, unrecorded insulin administration introduces variability that likely contributes to higher prediction error in insulin-treated participants.

Modeling and benchmarking limitations. Although SSM-CGM consistently outperformed the Temporal Fusion Transformer (TFT), our benchmarking relied on TFT as the main comparator. Additional comparisons to alternative forecasters and broader ablation studies, including the roles of future covariates, preprocessing, and alternative Mamba configurations (e.g., ablation of light vs. stacked layers) will clarify which model components drive performance gains. We also note that our runtime is substantial (1.5 hour per epoch on 4 A100 GPUs), and further work is needed to improve efficiency.

Interpretability and counterfactual reasoning. While the model provides interpretable outputs via variable selection and temporal attribution, these remain associational rather than causal. Counterfactual forecasts are identified under standard assumptions of consistency, sequential ignorability, and positivity, which cannot be fully validated from observational data. As such, interpretability and counterfactual outputs should be viewed as hypothesis-generating rather than definitive.

Forecasting scope. Our evaluation was limited to a 1-hour forecasting horizon. While this window aligns with near-term clinical decision-making, extending to forecasts of different lengths could support a wider range of use cases. Moreover, our current counterfactual interventions targeted single covariates, whereas real-world behaviors involve coordinated, multivariate activity patterns. Extending counterfactual reasoning to multivariate interventions could better support actionable, individualized recommendations.

Generalization across cohorts. Our results are specific to the AI-READI cohort, and it remains unclear how well the model generalizes across datasets with different demographics, devices, or recording conditions. The effective sample size for detecting model improvements is not well established, and replication in other large-scale CGM cohorts will be essential.

Future Directions. Beyond addressing these limitations, several extensions are possible. Incorporating planned categorical covariates (e.g., scheduled exercise, meals, or sleep) could make forecasting and counterfactuals more realistic, since individuals can plan activities but not continuous physiological values such as heart rate minute by minute. Extending the same model architecture to multivariate forecasting of both glucose and wearable signals could strengthen counterfactual reasoning by providing forecasted future inputs on learned behavior. Building streaming-capable versions of SSM-CGM, leveraging the linear-time properties of Mamba, would improve deployability. Finally, leveraging SSM-CGM forecasting models combined with reinforcement learning methods could recommend personalized, adaptive behavioral policies to maximize time-in-range (70–180 mg/dL).

A.16 Code Availability

Code is available at github.com/shakson-isaac/SSM-CGM under the Harvard license for non-commercial research use.