Project Execution Plan:
Calibrating a Causal Transformer with Aggregate RCT Data

MyWay Digital Health

June 9, 2025

Abstract

This document outlines the step-by-step instructions for implementing the Causal Transformer calibration project. The objective is to develop a system that produces personalized treatment effect estimates that are calibrated against aggregate evidence from published Randomized Controlled Trials (RCTs). The methodology combines a Causal Transformer as a base predictor with a Gaussian Process (GP) calibration model.

1 Phase 1: Base Model Preparation and Training

The first phase involves defining the data structure, preparing the observational dataset from SCI-Diabetes, and training the base Causal Transformer model.

1.1 Understanding the Causal Transformer Architecture

The Causal Transformer is a sequence model adapted for causal inference. It processes patient data as a series of temporal "tokens". The model takes three corresponding input streams for each patient:

- **Covariate Stream (X):** A vector of baseline patient characteristics measured *before* time-zero. $\mathbf{X} \in \mathbb{R}^{d_x}$.
- **Treatment Stream (A):** A vector indicating the treatment assigned at time-zero. $\mathbf{A} \in \mathbb{R}^{d_a}$.
- **Outcome Stream (Y):** A vector of the outcomes observed for the patient *after* time-zero. $\mathbf{Y} \in \mathbb{R}^{d_y}$.

The model is trained to predict the outcome stream **Y** given the baseline covariates **X** and treatment **A**. It can then be used to generate counterfactual predictions by changing the input treatment vector **A**. The Causal Transformer's full architecture can model patient data as complex time-varying sequences of covariates, interventions and outcomes. Our project will streamline the intervention data structure from a sequence of temporal tokens to a single, binary treatment decision at 'time-zero'. This is done to emulate a two-arm RCT testing the effect of a specific treatment. Covariates and outcomes will still be represented as sequences of temporal tokens. This simplification will be more stable to train as a first iteration of the project, will allow us to answer clinically relevant causal questions about the effect of any given treatment, and will allow us to calibrate against two-armed RCTs.

1.2 Step 1.1: Define Data Streams (A, X, Y)

We will define the specific variables for each stream.

- **Treatments (A):** We will create separate foundation models for different clinical questions. Our initial model will focus on the pairwise comparison of SGLT2 inhibitors versus DPP-4 inhibitors, a common clinical scenario.
  - **Treatment (A=1):** Initiation of an SGLT2 inhibitor.
  - **Active Comparator (A=0):** Initiation of a DPP-4 inhibitor.

  This defines **A** as a simple binary variable. This process can be repeated later to build a new foundation model for each set of RCTs (for example GLP-1 agonist vs placebo). At inference time, when the effect of a given medication is queried, this is routed to the relevant foundation model. It is important to note that every pairwise comparison of drug classes (SGLT2i vs DPP4i, GLP1a vs DPP4i etc) would require a separate foundation model and set of RCTs to calibrate against. This is a labour-intensive process, so the pairwise comparisons should be chosen carefully to be in areas of clinical equipoise where RCTs are available for calibration.

- **Outcomes (Y):** The outcome stream **Y** should contain outcomes that are both clinically important and commonly reported in our set of RCTs. A task is to **survey all SGLT2i vs DPP4i RCTs and identify the set of common primary and key secondary outcomes**. A proposed initial set is:
  - Change in HbA_{1c} from baseline.
  - Time-to-first Major Adverse Cardiovascular Event (MACE).
  - All-cause mortality.

  For time-to-event outcomes (e.g., MACE, mortality), the **Y** stream will be processed as a sequence of binary indicators denoting event occurrence within discrete time intervals. The final reported outcome (for calibration and deployment) will summarize this sequence as time-to-event and a status indicator (event or censored). Mean physiological data will be reported for each time interval. This can be converted to 'change from baseline' for calibration against RCTs.

- **Covariates (X):** See next step.

1.3 Step 1.2: Covariate Selection (The X stream)

This is a critical data curation task. The set of covariates in the **X** stream must be features that are available in **both** the SCI-Diabetes observational dataset and reported as baseline characteristics in our grounding RCTs. The selection process is as follows:

1. Create a master spreadsheet of all baseline characteristics reported in the "Table 1" of the SGLT2i vs DPP-4i RCTs.

2. Create a list of all available, high-quality features in the SCI-Diabetes dataset.

3. **Satisfy the Ignorability Assumption through Domain Expertise:** The Ignorability (or "no unmeasured confounding") assumption in the Potential Outcomes framework is crucial for valid causal inference. It requires that our covariate set includes all features that affect both treatment choice and the outcome. This will require careful consideration to make sure we don't miss any confounding variables.

4. The final covariate set for the **X** stream will be the **intersection of these two sets** (the available observational data and the RCT characteristics), guided by domain knowledge. This ensures that we can fully describe each RCT population using features the Transformer understands, while making a concerted effort to include all relevant confounders.

Note: For simplicity and robustness, it is advisable to start with a parsimonious set of universally available covariates (e.g., age, sex, BMI, baseline HbA1c, baseline eGFR, duration of diabetes, history of cardiovascular disease, baseline metformin use). The impact of this covariate choice can be assessed later via sensitivity analyses (e.g., by observing model output stability when adding or removing covariates).

1.4 Step 1.3: Create the Master Longitudinal Dataset

This step is critical. We will transform the raw, event-level SCI-Diabetes data into a structured longitudinal dataset that emulates a trial, using a ”new-user, active-comparator” design. This ensures that for a given pairwise comparison (e.g., SGLT2i vs DPP-4i), each patient appears only once. The final dataset will not be ”one row per patient,” but rather a collection of patient trajectories, where each trajectory contains sequences of temporal tokens for covariates and outcomes.

1. **Define Index Date (Time-Zero):** For each patient in SCI-Diabetes, scan their medication history to find the date of their *first-ever* prescription for a drug in either of our two classes of interest (e.g., SGLT2i or DPP-4i). This date becomes that patient's personal index date. Patients who have used either class before are excluded (”new-user” design).

2. **Assign Treatment (A):** Based on the drug prescribed on the index date, assign the patient their static treatment variable **A**. If an SGLT2i was initiated, $\mathbf{A} = 1$; if a DPP-4i was initiated, $\mathbf{A} = 0$. This assignment is fixed for the patient's entire follow-up period for generating counterfactuals, emulating an intention-to-treat principle.

3. **Extract Temporal Data Streams (X, Y):** For each patient, we will create parallel time-aligned sequences based on a chosen time granularity (e.g., monthly intervals).

   - **Covariate Stream (X):** Extract a sequence of all selected covariates (from Step 1.2) for each time interval, both *before* and *after* the index date. This captures the patient's dynamic health state over their entire history. For each interval, summary statistics (e.g., mean, max) or the most recent value may be used.
   - **Outcome Stream (Y):** For the follow-up period *after* the index date, extract a sequence of all target outcomes.
     - For continuous outcomes (e.g., HbA1c), this will be the value at each interval.
     - For time-to-event outcomes (e.g., MACE), this will be a binary indicator (1 if the event occurred in the interval, 0 otherwise).

   The follow-up period is censored at the earliest of: the event occurrence, loss to follow-up, or a pre-specified administrative end date (e.g., 5 years).

4. **Assemble and Finalize:** Combine the streams for all patients into a single, clean dataset format suitable for sequence models (e.g., a list of patient objects, or a ”long-format” table with patient ID and time index). Perform necessary cleaning and imputation on the sequential covariate data, paying special attention to temporal consistency.

1.5 Step 1.4: Adapt and Train the Causal Transformer

1. Download a public implementation of the Causal Transformer (e.g., from the original authors' GitHub repository).
2. **Adapt the Model for Binary Treatment:** The original Causal Transformer is designed for a multi-categorical treatment variable. We will specialize it for our simpler binary treatment case. This requires two key modifications:
   - **Treatment Input:** The model's treatment embedding layer will be simplified to handle a binary (0/1) input rather than a multi-category input.
   - **Adversarial Loss:** The treatment classifier head (`G_A`) will be changed from a multi-class classifier (with a Softmax output) to a binary classifier (with a Sigmoid output). The corresponding domain confusion loss will be adapted from multi-class cross-entropy to binary cross-entropy.

   This adaptation simplifies the model, increases training stability, and directly aligns the architecture with our pairwise causal question.
3. Adapt the data loader to read our master dataset format.
4. Configure training hyperparameters and train the adapted Causal Transformer on the entire dataset created in Step 1.3. This is a one-time, computationally intensive step for each pairwise model (e.g., one for SGLT2i vs DPP-4i, another for GLP1a vs DPP-4i).
5. Save the trained model weights. This is now your **Base Predictor**.
6. *Future Extension:* Once this pipeline is validated, a future research direction could involve implementing the full Causal Transformer with a multi-categorical treatment vector to model the simultaneous choice between SGLT2i, GLP1a, and DPP4i, and calibrating it against multi-arm trials where available.

2 Phase 2: Calibration Model Development

This phase involves using the Base Predictor to generate predictions for our RCTs and preparing the data to train the calibration model.

2.1 Step 2.1: Simulate RCT Cohorts

For each of the $i \in \{1, ..., 25\}$ RCTs:

1. **Generate a large synthetic patient pool:** Utilize the project's dedicated **synthetic data generation models** to produce a large pool of virtual patients ($M \approx 1,000,000$). Each patient j must have a complete covariate vector $\mathbf{X}_j$ that is statistically representative of the SCI-Diabetes population.
2. **Apply Exclusion Criteria:** Apply the pragmatic filters identified in the project planning phase (e.g., age range, baseline values) to this pool to create a candidate set of patients.
3. **Match Population Characteristics via Weighting:** To avoid the curse of dimensionality, use a weighting technique like **Iterative Proportional Fitting (Raking)** to assign a weight w_j to each patient in the candidate set. Adjust the weights w_j such that

the weighted summary statistics of the synthetic cohort match the target statistics from the RCT's Table 1.

$$\sum_{j=1}^{M'} w_j \cdot \text{age}_j = \text{mean\_age}_{\text{RCT}_i}$$

$$\sum_{j=1}^{M'} w_j \cdot \text{is\_male}_j = \text{proportion\_male}_{\text{RCT}_i}$$

$$\vdots$$

The result is a weighted set of virtual patients that statistically replicates the population of RCT_i.

2.2 Step 2.2: Generate Transformer Predictions

For each simulated, weighted RCT cohort i:

1. Feed the cohort's covariate vectors $\mathbf{X}$ through the Base Predictor twice:
   - Once with the treatment vector $\mathbf{A}_{\text{treat}}$ to get predicted outcomes $\hat{\mathbf{Y}}_{\text{treat}}$.
   - Once with the treatment vector $\mathbf{A}_{\text{placebo}}$ to get predicted outcomes $\hat{\mathbf{Y}}_{\text{placebo}}$.
2. Calculate the Individual Treatment Effect (ITE) for each patient j: $\text{ITE}_j = \hat{\mathbf{Y}}_{\text{treat},j} - \hat{\mathbf{Y}}_{\text{placebo},j}$.
3. Calculate the Transformer's predicted Average Treatment Effect (ATE) for that cohort by taking the weighted average of the ITEs:

$$\text{ATE}_{\text{transformer},i} = \sum_{j=1}^{M'} w_j \cdot \text{ITE}_j \tag{1}$$

2.3 Step 2.3: Feature Engineering & Error Calculation

For each RCT i:

1. **Compute `Population_Vector`$_i$:** This is the vector of summary statistics from the RCT's Table 1 that were used as targets for raking. It is the feature vector for the calibration model.

$$\texttt{Population\_Vector}_i = [\text{mean\_age}, \text{sd\_age}, \text{prop\_male}, \ldots]$$

2. **Calculate Target `Error`$_i$:** This is the target variable for the calibration model. Let $\text{ATE}_{\text{rct},i}$ be the true ATE reported in the publication for RCT i.

$$\texttt{Error}_i = \text{ATE}_{\text{transformer},i} - \text{ATE}_{\text{rct},i} \tag{2}$$

   There will be one such error value for each outcome (e.g., `Error_HbA1c`, `Error_MACE`). We will train a separate GP for each outcome.

3 Phase 3: Analysis and Calibration Model Training

3.1 Step 3.1: Visualize the Manifold Hypothesis (t-SNE)

To build intuition and visually validate our core hypothesis, we will visualize the error manifold.

1. Take the set of `Population_Vector`s we created. This is a dataset where each row is an RCT, and columns are population characteristics.
2. Use a dimensionality reduction algorithm, such as t-SNE or UMAP, to project these high-dimensional vectors into a 2D space.
3. Create a scatter plot of the 2D-projected points.
4. Color each point on the plot by its corresponding $\texttt{Error}_i$ value. Use a continuous color scale (e.g., blue for negative error, red for positive error).
5. **Interpretation:** If the manifold hypothesis holds, this plot will show structure. You should see "blobs" or gradients of color, indicating that RCTs with similar populations (and are therefore close in the t-SNE plot) also have similar prediction errors. A random-looking "salt and pepper" plot would invalidate the hypothesis.

3.2 Step 3.2: Train the GP Calibration Model

1. Create the training dataset for the GP:
   - Input Features: The `Population_Vector`s.
   - Target Variable: The `Error` values.
2. Using a library like GPyTorch, scikit-learn, or GPflow, train a Gaussian Process Regressor. The GP will learn the function:

$$\text{GP}(\texttt{Population\_Vector}) \rightarrow (\mu_{\text{error}}, \sigma^2_{\text{error}})$$

3. Choose a suitable kernel, such as the RBF (Radial Basis Function) kernel, and optimize its hyperparameters (e.g., lengthscale, variance) by maximizing the log marginal likelihood on the training data.

4 Phase 4: Deployment for Calibrated Prediction

This is the final workflow for predicting the outcome of a new target population (e.g., the SURPASS-CVOT trial that is underway).

1. **Define Target Population:** Obtain the baseline characteristics (the "Table 1") for the new target population. This forms your new `Population_Vector_new`.
2. **Generate Raw ATE:** Simulate a cohort matching `Population_Vector_new` (using the same SDG model and raking procedure) and run it through the **Base Predictor** (the Causal Transformer) to get the raw prediction, `ATE_raw`.
3. **Predict Error:** Feed `Population_Vector_new` into the trained **GP Calibration Model**. It will output the predicted error mean ($\delta_{\text{predicted}}$) and variance ($\sigma^2_\delta$).

4. **Calculate Final Estimate:** Combine the outputs to get the final calibrated ATE and its 95% confidence interval.

$$\texttt{ATE\_calibrated} = \texttt{ATE\_raw} - \delta_{\text{predicted}} \tag{3}$$

$$95\% \text{ CI} = \texttt{ATE\_calibrated} \pm 1.96 \cdot \sqrt{\sigma_{\delta}^{2}} \tag{4}$$

This provides a final, trustworthy prediction that is consistent with the body of existing clinical trial evidence, complete with a principled uncertainty estimate.