# Life Sequence Transformer: Generative Modelling of Socio-Economic Trajectories from Administrative Data

**Alberto Cabezas**
Environmental Change Institute
University of Oxford

**Carlotta Montorsi**
Department of Living Conditions
Luxembourg Institute of Socio-Economic Research (LISER)

## Abstract

Generative modelling with Transformer architectures can simulate complex sequential structures across various applications. We extend this line of work to the social sciences by introducing a Transformer-based generative model tailored to longitudinal socio-economic data. Our contributions are: (i) we design a novel encoding method that represents socio-economic life histories as sequences, including overlapping events across life domains; and (ii) we adapt generative modelling techniques to simulate plausible alternative life trajectories conditioned on past histories. Using large-scale data from the Italian social security administration (INPS), we show that the model can be trained at scale, reproduces realistic labour market patterns consistent with known causal relationships, and generates coherent hypothetical life paths. This work demonstrates the feasibility of generative modelling for socio-economic trajectories and opens new opportunities for policy-oriented research, with counterfactual analysis as a particularly promising application.

## 1 INTRODUCTION

In fields such as physics (Geneva and Zabaras, 2022), chemistry (Kim et al., 2021), and health (Yang et al., 2023), researchers successfully use Transformer-based generative models to simulate surrogate dynamical systems under varying conditions and estimate event probabilities. In addition to their predictive capabilities, generative models have the potential to explore hypothetical system evolutions under different scenarios.

We extend this line of research to the social sciences by proposing a Transformer-based generative model for sequences of human life events. Longitudinal administrative records contain rich information on individuals' trajectories across labour markets, health, education, and other socio-economic domains, yet these data are rarely treated as sequences for generative modelling. Through a novel encoding method of socio-economic events, we enable the training of a decoder-only Transformer model capable of generating plausible synthetic life trajectories.

Our model is trained on 1.6 million individual labour market histories from the Italian social security administration (INPS), described in Section 3. The representation of these histories as overlapping socio-economic events encoded into life sequences is introduced in Section 4. The architecture and training procedure of our life sequence Transformer are detailed in Section 5. The empirical analysis in Section 6 illustrates the generative potential of the framework through two benchmarking exercises: the accuracy of simulated sequences in capturing established causal nexus among life events (Section 6.2) and the predictive accuracy on out-of-sample trajectories (Section 6.3).

Once trained, the model can generate realistic continuations of partially input sequences and simulate alternative life paths. In other words, it can create counterfactuals, i.e., plausible "what if" scenarios that describe how events might have unfolded under different circumstances. This ability provides a scalable framework for constructing coherent counterfactual trajectories without the need for external control groups. In Section 7, we elaborate on how this approach can be integrated into experimental designs in the social sciences and outline promising directions for future research that build on our framework.

## 2 BACKGROUND

Our work builds on recent advances in Transformer-based sequence modelling for longitudinal labour and health data. In particularly, Life2Vec (Savcisens et al., 2024), CAREER (Vafa et al., 2024), and TransformEHR (Yang et al., 2023). Life2Vec tests the validity

of encoder-only Transformer models for modelling socio-economic administrative data from Denmark. It outperforms the predictive performance of standard models, such as mortality tables, logistic regression, or recurrent neural networks, in predicting an individual's future life outcomes (e.g., early mortality) or time-invariant characteristics (e.g., personality nuances). However, this model lacks generative capability to simulate sequences of future tokens, and the data encoding cannot be easily adapted to a generative task. In contrast, our proposed encoding approach enables the training of a generative decoder, allowing for the simulation of counterfactual life sequences on a model architecture directly inspired by Life2Vec.

The CAREER encoder-decoder Transformer model focuses on job sequence generation with annual temporal resolution and no rich occupational attributes, such as firm size, social security benefits, or multiple job episodes within the same year. We enhance this framework by introducing fine-grained temporal encoding and a richer set of characteristics that span labour and demographic dimensions. In CAREER, time is introduced with a timestep of one year, and there is no way to model the duration of events or the exact timing of their occurrence. You could increase the timestep to monthly steps, but this would restrict generation to a fixed monthly grid, offering no control over when events begin or how long they last within the year. Our method of encoding time, by contrast, enables flexible temporal generation by allowing each life event to have a defined starting month and duration in months within the year.

TransformEHR is an encoder-decoder Transformer generative model that encodes sequences of electronic health records and predicts the subsequent closest diagnoses (ICD-10 codes). It includes demographic and time embeddings to model the numerical format of the visit date. TransformEHR is pre-trained to predict a patient's future visits with ICD codes, given longitudinal codes up to the current visit. Similarly to Life2Vec, the timing of each visit must be specified at inference time, meaning the model cannot decide when events occur, but only predict diagnoses for predefined dates. In contrast, our model allows the generation process to determine when events occur and their respective durations. Crucially, our model supports variable event durations, which is essential for modelling life sequences that extend beyond instantaneous diagnoses.

## 3 WHIP DATA

Our model is trained on the Work History Italian Panel (WHIP), a 1:15 sample of individuals drawn from the INPS database (Leombruni et al., 2010; Bena et al., 2012). INPS records cover most private-sector employees, quasi-dependent and self-employed workers, as well as periods of retirement and access to social benefits (e.g., unemployment or maternity leave). The WHIP provides detailed longitudinal employment and benefit histories for over 4 million individuals.

We use the WHIP database for the period 1990–2015 and apply sample selection criteria to obtain sequences that are sufficiently long to provide meaningful information. Specifically, we retain only individuals with complete information up to 2015, at least half of their potential labour force participation period observed, non-missing birth dates, at least five records in total, plausible entry ages (between 15 and 80), and a labour force attachment of at least 10%. This resulted in a sample of around 1.65 million life trajectories. A full description of the dataset and details of the selection procedure are provided in Appendix B.

Each labour event contains timing (start month and duration) and income, which is standardised into monthly real terms and discretised into 100 quantiles. Yearly income is the principal variable for the social security system, as it forms the basis for calculating contributions and benefits. Each income record is linked to a range of demographic, workplace, and labour-force-specific attributes. Demographic information includes sex, birth year, and month, and the macro-area of birth. Workplace characteristics cover the sector of activity, enterprise size, and workplace location at the provincial level.

Labour force information is recorded in terms of labour status (e.g., employee, self-employed, pensioner, unemployed) with additional modifiers depending on status. For employees, these include job title, work intensity, number of weeks of maternity leave, number of weeks of sick leave, and whether the contract is part-time or full-time. Together, these attributes enable a detailed reconstruction of individuals' trajectories across work, unemployment, pensions, and benefit receipt. A full description of all 15 variables and encoding details is provided in Appendix B.1.

## 4 DATA ENCODING

We build on the work of Savcisens et al. (2024), where life events are concatenated in a sequential structure, with absolute and relative temporal information embedded as auxiliary attributes rather than forming the structural backbone of the sequence. Instead, our encoding method imposes an explicit calendar-like structure on each life trajectory to support the sequential generation of future tokens. The entire observed work history is divided into yearly segments, each further subdivided into twelve monthly intervals, which are

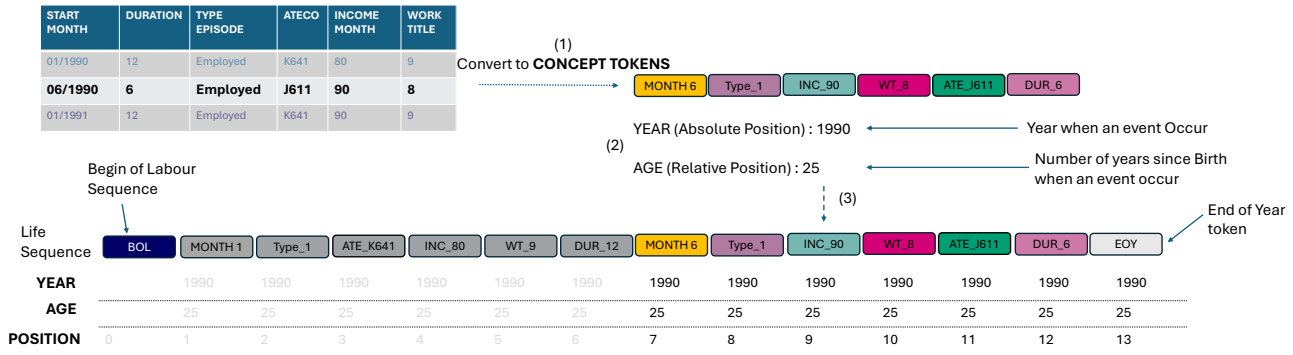



Figure 1: The steps for converting data from Tabular format to the Generative model suitable format: (1) Convert relevant features to concept tokens (2) Look up relevant temporal information such as age and year (3) Concatenate token, time, and positional embeddings to the sequence.

then populated with event information as needed. Life events are represented as discrete units with attributes such as employment type, income quantiles, work intensity, sector, qualification, and duration, forming a sequence of socio-economic states.

For instance, if an individual remains employed at the same firm for 10 uninterrupted years, we encode this as 10 events, each lasting 12 months, described using the same set of descriptive tokens, and with a likely increase in income or a change in work title if promoted over the years. If a person holds a one-year contract but experiences a one-month interruption, the year is split into two separate events with different durations. Though the same set of concept-tokens is otherwise used to describe these events. When no event occurs within a year, the structure still records an "empty" year preserving the regular yearly pace of the sequence.

Figure 1 illustrates the encoding pipeline. We consider each tabular record (i.e., each row in the administrative dataset) as an *event* and translate each column describing it into a *concept-token*. Then, we concatenate all the events for a given individual in chronological order and create an individual sequence.

Each event includes embeddings on positional information: year or *absolute* position reporting the calendar year when the event occurred, age or *relative* position with the individual's age at the time of the event, and the usual *positional* embedding used in Transformer models for the position of the token in the sequence. Notice, these three embeddings are not part of the generative task; instead, they are deterministic.

Sequences start with the special token `<BOL>` (Beginning of Life). Individual events are then encoded using a fixed token order. `<MONTH_X>` marks the calendar month the event begins and helps separate multiple events within the same year. Next comes `<TYPE_X>` to indicate the event type (e.g., employment, unemployment), optionally followed by modifiers like `<ATECO_X>` (for sector) and `<WRK_TITLE_X>` (for job title), depending on the event. `<DUR_X>` defines the event's duration in months and concludes the block describing the event. If more events follow in the same year, another `<MONTH_X>` appears; otherwise, the `<EOY>` (End of Year) token concludes the description of the events in a year. If no further events ensue, the entire life sequence ends with an `<EOL>` (End of Life) token. Each sequence maintains a consistent ordering of these concept-tokens, ensuring a structured representation and facilitating the learning of this artificial language.

While the experiments in this paper focus on labour-related life sequences, we emphasise that the proposed encoding scheme applies to other life domains, such as health and education. In the health domain, for instance, life sequences could represent concatenations of diagnostic attributes, annotated with the month of diagnosis and the duration of treatment or hospitalisation. Although a monthly resolution may appear too coarse for modelling certain health events, this limitation is mitigated by including intensity tokens `<INTENSITY_X>`, which capture the extent of engagement within each event's duration. For a detailed description of the data and a labour life sequence example, see Appendix B.

Background time-invariant information, such as gender, place, and date of birth, is also represented using concept-tokens (e.g., `<F>` or `<M>` for female and male). This information is placed at the beginning of the se-

quence, forming the entire input as a single stream of tokens. The encoding distinguishes between static demographic attributes and the temporal sequence of life events, which begins with the `<BOL>` token.

# 5 TRANSFORMER MODEL

We propose a decoder-only Transformer architecture, see Figure 7, trained autoregressively to predict the next token in the sequence, conditioned on past events and background information.

## 5.1 Model Architecture

The model follows a standard Transformer decoder-only design, consisting of three components: an embedding block, stacked Transformer-based decoder layers, and a task-specific output head. We make three key modifications to the standard architecture. First, we replace the conventional attention mechanism with Performer Attention (Choromanski et al., 2022), which reduces the number of trainable parameters and improves computational efficiency. Second, we incorporate Rotary Position Embeddings (RoPE) (Su et al., 2023), adapted to work with Performer Attention, to better capture temporal patterns in the sequence. Third, to stabilise training, we apply RMS Normalisation (Zhang and Sennrich, 2019) before the attention and feedforward layers. The decoder uses causal (or masked) self-attention, ensuring that each token can only attend to earlier positions in the sequence. During training, the model is optimised using a next-token prediction objective, learning to generate coherent event sequences conditioned on prior events and time-invariant variables.

**Token embeddings** The first transformation of the input sequence occurs in the embedding layer, where each token in the sequence $S_r$ is mapped to a continuous vector representation using an embedding matrix $\mathcal{E}_{\mathcal{V}} : \mathcal{V} \to \mathbb{R}^d$. Each row of $\mathcal{E}_{\mathcal{V}}$ corresponds to a unique token from the vocabulary $\mathcal{V}$, and each column represents one of the $d$ embedding hidden dimensions. Following Savcisens et al. (2024), we center the embeddings by removing the mean from each matrix row.

**Time embeddings** To encode temporal information, i.e., age and calendar year, we use the Time2Vec method (Kazemi et al., 2019), which allows the model to capture both linear and periodic aspects of time. This method uses trainable parameters $\omega$ and $\varphi$ to define a flexible sinusoidal function. We construct two separate time embeddings: one for age, denoted by $\mathcal{T}_A : \mathcal{A} \to \mathbb{R}^d$ with $Z := 4$, and one for year, $\mathcal{T}_Y : \mathcal{Y} \to \mathbb{R}^d$ with $Z := 2$. These embeddings are computed as:

$$\mathcal{T}(x)[z] = \begin{cases} \tanh(w_z x + \varphi_z) & \text{if } z \in [0, d//Z) \\ \sin(w_z x + \varphi_z) & \text{if } z \in [d//Z, d) \end{cases} \quad (1)$$

Initial iterations of the model used standard Time2Vec encoding, i.e., a linear component with trigonometric functions. However, we noticed that it led to poor out-of-sample performance: the model quickly lost coherence and failed to maintain the grammar of the life sequence when extrapolating beyond the training period. We hypothesised that this issue stemmed from the unbounded linear trend in the Time2Vec representation of age and year. Additionally, trigonometric components appeared to mimic linear trends, particularly for the year variable, and also for age.

Hence, we adapt the original Time2Vec by varying the proportion of linear and periodic components for different temporal signals and bounding the linear component for better out-of-sample generalisation (see Section 6.3). For year embeddings, we allocate a larger share of the hidden dimensions to the linear component (tanh) to better capture the short-range time frame, which spans only 25 years. For the Age embedding, which spans 70 years in our dataset, we expand periodicity (sin) to reflect cyclical patterns (e.g., retirement or career phases) that recur across broader age bands.

**Positional embeddings** In addition to temporal embeddings, we incorporate standard positional encodings to capture the order of tokens within each sequence. Our approach is adapted to use with local and global heads of Performer attention (Su et al., 2023): sinusoidal positional encodings combined with RoPE in the global attention heads. Both positional and time embeddings are added to the token embeddings through a ReZero gate (Bachlechner et al., 2021) before being passed to the attention layers.

## 5.2 Training Procedure

Our dataset consists of 1,652,877 individual life sequences. We divide them into training (70%), validation (15%), and test (15%) subsets. The models are trained for 15 to 20 epochs with early stopping. Each training batch contains 18 sequences, and gradients are accumulated over five steps before an optimiser step, resulting in an effective batch size of 90.

**Optimization** We employ the `AdamW` optimiser (Loshchilov and Hutter, 2019), with no weight decay applied to bias terms, embedding weights, or normalisation layers. The learning rate is controlled by the One Cycle policy (Smith and Topin, 2019) with a warm-up phase covering 30% of the training steps. Gradient

| | Accuracy | Precision | Recall | F1 Score | Perplexity |
|---|---|---|---|---|---|
| 0 known years | 0.6225 | 0.6868 | 0.6225 | 0.6427 | 0.7288 |
| 1 known year | 0.6773 | 0.7216 | 0.6773 | 0.6895 | 0.6937 |
| 5 known years | 0.6883 | 0.7285 | 0.6883 | 0.6991 | 0.6727 |
| 10 known years | 0.7039 | 0.7409 | 0.7039 | 0.713 | 0.6466 |

Table 1: Test set metrics for next-token prediction, macro-averaged across the vocabulary, conditioning on 0, 5, 1 and 10 initial years of sequence.

norms are clipped at a maximum value of 5 to prevent exploding gradients. To avoid overfitting, a dropout rate of 10% is applied to the attention, feedforward, and embedding layers. Dataloaders and projection matrices of Performer attention are reloaded every epoch to ensure robustness in sampling. We use mixed-precision with `bfloat16`, which we chose to reduce memory usage and improve training efficiency, considering that RMS Normalisation has been shown to improve numerical stability in lower-precision regimes (Zhang and Sennrich, 2019).

**Hyperparameters tuning** We sample hyperparameters using the Tree-Structured Parzen Estimator (Watanabe, 2023) and Optuna (Akiba et al., 2019). Experiments showed that deeper models performed better. Without normalisation, the model misuses the embedding space, often requiring unusually large embedding dimensions to compensate. With RMS normalisation, this effect was mitigated, although the model benefited from longer warmup phases compared to setups using only ReZero. Increasing depth proved more effective than scaling up feedforward or embedding dimensions, or the number of random features. For local attention, we set the window size to cover approximately three complete events, which empirically balanced context coverage and memory use. Ultimately, we selected hyperparameters that maximized model capacity while fitting within the 16GB of VRAM available on an RTX A4000 GPU: 10 decoder layers, with 8 attention heads, 6 local with a window size of 36 and 2 global with projection of 256 random features; embedding size of 240, and position-wise feed forward size of 960 on all layers. The model has around 10.1 million tunable parameters.

# 6 EMPIRICAL ANALYSIS

In this section, we present the next-token prediction task metrics and evaluate the model's ability to forecast out-of-sample for years beyond the training period. We then evaluate performance through a series of causal inference benchmarking exercises by comparing the generated life sequences against well-established empirical causal findings from labour economics. All experiments are run on the test subset of our data.

## 6.1 Next-Token Prediction Metrics

We begin by evaluating the model on a next-token prediction task over the test subset, which measures how accurately it generates the next element in a life sequence given prior tokens and background information. Training each model took approximately 9 hours per epoch for the training phase and 1.5 hours per epoch for validation, totalling 10.5 hours per epoch. The final model was trained for 20 epochs for a total training time of 210 hours on a single NVIDIA RTX A4000 GPU with 16GB of VRAM.

Table 1 reports a standard set of classification metrics computed across the entire sequence dimension and vocabulary, excluding padding tokens. Each row corresponds to a different length of observed history provided as input. Specifically, we generate the continuation of a life sequence after observing either zero, one, five, or ten years of past events. The metrics include macro-averaged accuracy, precision, recall, and F1 score, each evaluated using top-1 predictions, i.e., consider only the highest probability to assign the predicted token. We also report the square root cross-entropy loss, i.e., perplexity, a widely used measure in language modelling.

As expected, the more historical context the model observes, the better it can predict future tokens. This is consistent with the intuition that individuals often follow relatively stable life patterns after a certain age, e.g., remain within certain industries over time. Access to a longer sequence of past events enables the model to learn person-specific trajectories and reduces uncertainty about what is likely to happen next.

## 6.2 Causal Benchmarks

We evaluate the model's capacity for generating causal relationships by testing whether it reproduces well-established effects from the labour economics literature and national policy. To do so, we truncate observed individual life sequences at specific cutoff years and condition the model on this truncated history. The model is then tasked with generating the continuation of the sequence forward in time. We only truncate at the level of whole years, so that the conditioning context

always consists of a realistic, complete history up to the cutoff point. In each experiment, we choose the cutoff year to coincide with the onset of the event of interest (i.e., unemployment, maternity, or retirement), so that the subsequently generated trajectory can be compared to known empirical effects. Additional details of this procedure are provided in Appendix D. All generation experiments are run on batches of 8 sequences. On a GPU, generating the next token takes approximately 0.2 seconds per batch, while on a CPU it takes about 2 seconds.

First, we evaluate whether the model captures the impact of an unemployment benefit policy rule in Italy. Until 2017, Italy's mobility allowance granted income support to workers displaced due to firm closure, with benefit duration varying discontinuously by age (cutoffs at 40 and 49) and by region (South vs. North/Central) (Brunello and Miniaci, 1997). For each individual, we truncate the sequence at the year when displacement is first observed and provide this truncated history to the model. The model must then generate forward, including the timing and duration of subsequent unemployment and benefit receipt.

Figure 2 shows the simulated average duration of mobility allowance receipt as a function of age at displacement. The simulated data captures the overall upward trend in benefit duration with age, suggesting that the model has learned the general relationship between age and benefit duration. However, it falls short of reproducing the sharp discontinuity observed at the threshold, instead smoothing over the abrupt policy-induced change. One likely reason is data scarcity: fewer than 1% of sequences involve a mobility allowance spell (see Figure 6), which restricts the amount of policy-relevant signal available for training.

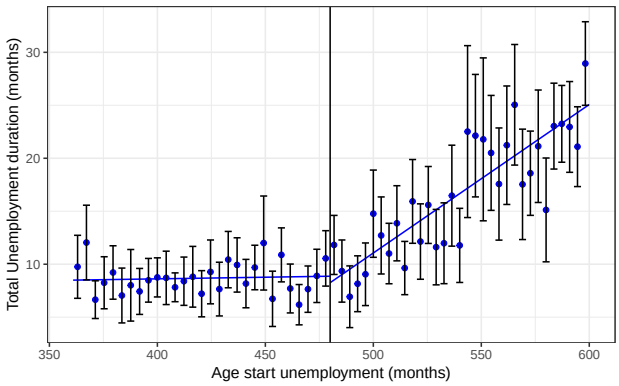



Figure 2: Model-generated average duration with 90% confidence interval of mobility allowance receipt duration as a function of worker age at job loss in months in a 10-year bandwidth around the 40-year-old threshold.

A more formal discontinuity analysis confirms this pattern. Table 2 reports local average treatment effects (ATEs) for different bandwidths around the 40-year cutoff. At narrow bandwidths (e.g., ±12 months), the model ATE fails to significantly capture the discontinuity. As the window expands to ±48 and ±96 months, however, the model's ATEs move closer to the empirical benchmark, with discrepancies shrinking to below one month and confidence intervals overlapping zero. At the widest bandwidth (±144 months), both the empirical and model estimates are around 8–9 months, with a modest and statistically insignificant difference.

The second exercise examines the motherhood penalty, i.e., the long-term career setbacks women face after having children, often including reduced earnings and labour market detachment (Casarico and Lattanzio, 2023; Kleven et al., 2019). Here, we simulate women's careers from the moment of first maternity. Specifically, we truncate sequences either (i) at the year of the first observed maternity leave (offset–0), or (ii) one year before that event (offset–1). The latter setup allows maternity leave itself to be generated or not, and enables counterfactual analysis by comparing career paths with and without maternity.

Figure 3 compares the real and simulated income trajectories of mothers following their first childbirth (see a breakdown by sector in Figure 12). Synthetic sequences are generated conditional on the real sequence up to the maternity leave year. The model successfully captures the key features of the observed trends, closely replicating the average post-birth income dynamics.

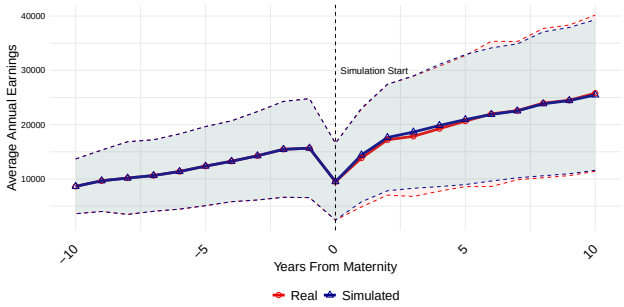



Figure 3: Real and model-generated trajectories of average annual earnings for working mothers in a 10-year window before and after their first maternity leave episode (offset–0). The solid lines represent average annual earnings, while the shaded areas denote the one standard deviation range.

To estimate the treatment effect of childbirth on earnings, we use an event-study design with controls for age and calendar year, as in Kleven et al. (2019). Because fertility is endogenous, we construct counterfactual trajectories using both propensity-score matched non-mothers and model-generated no-maternity paths

| | $\text{ATE}_{\text{emp}}$ | $\text{ATE}_{\text{model}}$ | $\Delta$ | $n$ |
|---|---|---|---|---|
| Pension (1y cutoff) | 7.47 [2.50, 12.23] | 9.10 [4.95, 13.26] | 1.62 [-0.36, 3.67] | 1165 |
| Pension (4y cutoff) | 7.47 [2.42, 12.59] | 11.02 [7.07, 15.14] | 3.55 [ 2.09, 5.07] | 1165 |
| Unemployment ±12m | 4.18 [1.46, 6.67] | 0.21 [-2.17, 2.55] | -3.98 [-6.38, -1.66] | 153 |
| Unemployment ±48m | 3.41 [1.76, 4.90] | 2.49 [1.08, 3.89] | -0.92 [-2.18, 0.54] | 556 |
| Unemployment ±96m | 7.25 [5.70, 8.70] | 7.09 [5.86, 8.39] | -0.16 [-1.36, 0.97] | 1168 |
| Unemployment ±144m | 8.27 [7.02, 9.51] | 8.92 [7.76, 10.02] | 0.65 [-0.34, 1.67] | 1502 |
| Maternity 0–3 years (%) | -25.1 [-28.3, -22.2] | -22.6 [-24.6, -20.5] | 2.45 [-1.20, 6.20] | 631 |
| Maternity 0–5 years (%) | -19.1 [-22.0, -16.5] | -18.8 [-20.7, -16.8] | 0.31 [-2.90, 3.80] | 631 |
| Maternity 0–10 years (%) | -11.0 [-13.7, -8.30] | -13.4 [-15.6, -11.2] | -2.35 [-5.90, 1.10] | 631 |

Table 2: Empirical and model-based average treatment effects (ATEs) point estimates with 95% confidence interval in brackets. $\Delta$ is the signed difference between model and empirical ATEs. $n$ denotes the number of individuals in the analysis sample. For details on how ATEs are constructed in each experiment, see Appendix D.

(offset–1 simulations). The details of this methodology are presented in Appendix D.2. These event-study coefficients are then summarised into average percentage income penalties over post-birth horizons, making them directly comparable to the ATEs reported for pension and unemployment.

The results show that the model reproduces the empirical child penalties closely across all horizons (Table 2). What is especially interesting is the pattern of differences: the model tends to understate the immediate short-run penalty but generates a more persistent effect in the medium and long run. In other words, the simulated trajectories suggest a smaller initial income drop but slower recovery, implying that the penalty lingers for longer.

Figure 10 illustrates how the model-generated simulated controls (non-maternity paths) compare to real mothers' earnings trajectories, with propensity score–matched mothers included as a comparison. The simulated controls track the counterfactual trend, reinforcing the finding that the model not only captures the observed income losses of mothers but also produces realistic no-maternity baselines.

The final exercise investigates whether the model replicates the well-documented effect of birth month on retirement timing (Ardito et al., 2020). Institutional features, such as school starting dates and labour market entry, generate systematic differences in retirement timing between individuals born in January versus those born in December. To test this, we truncate observed sequences prior to retirement. In one variant, we cut off at the year immediately before the observed retirement (so the model must decide whether retirement occurs in the following year or later). In another variant, we truncate four years before retirement, which increases the uncertainty in retirement timing and allows us to compare under different predictive horizons.

Figure 4 shows results with the one-year cutoff. Here, the model reproduces the negative effect of December birth on retirement age, indicating robustness of the signal. However, a clear upward bias emerges. This suggests that conditioning the model too close to the event may shift its internal expectations and effectively push retirement further into the future.

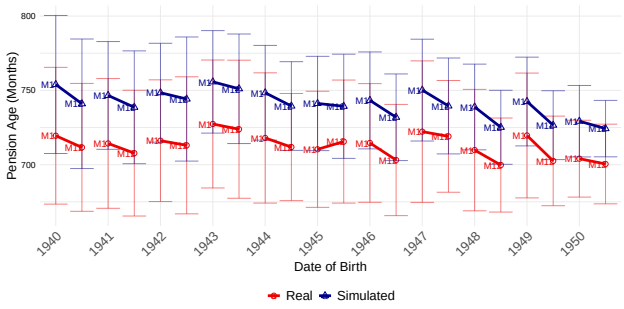



Figure 4: Real (circle dots) and model-generated (triangle dots) average retirement ages, measured in months, for male individuals born between 1940 and 1950, using a cutoff one year before the observed retirement year. Each point indicates the cohort-specific average retirement age for January (M1) and December (M12), with vertical bars representing one standard deviation.

Figure 5 reports results with the four-year cutoff. The model captures the expected empirical pattern: December-born individuals retire earlier than January-born peers, and the magnitude of this birth-month effect is well preserved across cohorts. The upward bias in retirement ages is largely absent under this earlier cutoff, indicating that conditioning further from the retirement event leads to a closer alignment with the observed timing of pension entry.

Table 2 formalises these findings. Both cutoffs generate significant differences in retirement age between treated (born in January) and untreated (born in December) individuals, consistent with the empirical benchmark.

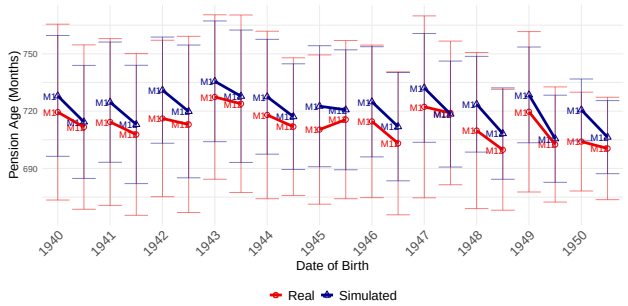



Figure 5: Real (circle dots) and model-generated (triangle dots) average retirement ages, measured in months, for male individuals born between 1940 and 1950, using a cutoff four years before the observed retirement year. Each point indicates the cohort-specific average retirement age for January (M1) and December (M12), with vertical bars representing one standard deviation.

Yet the size of the model's effect is sensitive to the conditioning horizon. Truncating the sequence further from the outcome offers a stronger test of counterfactual reasoning. Still, it tends to overstate effect sizes, illustrating how the length of the conditioning horizon shapes both predictive accuracy and the model's interpretation of the causal nexus among life events.

### 6.3 Out-of-sample Generation

We evaluate the model's out-of-sample performance by investigating the coherence of generated life sequences beyond the temporal window of the training data, i.e., after 2015. In this setting, the model is conditioned on observed sequences up to 2015 and generates tokens into the future. Good out-of-sample performance indicates robustness, a desirable property for a model expected to simulate plausible scenarios.

Given the absence of ground truth data for these later years, our evaluation focuses on syntactic and structural correctness rather than predictive accuracy. Specifically, we examine the generated sequences for internal consistency, valid token ordering, and the absence of structural violations, such as overlapping events or misaligned episode boundaries (more details in Appendix D.4). This assessment allows us to verify that the model preserves the learned "grammar" of life sequences when extrapolating beyond the observed data.

Figure 14 presents the density of out-of-sample generated years in which the first structural failure occurs, considering only sequences where a failure did occur within a 20-year generation window. The results show that sequence generation begins to fail around year 7 or 8, though a non-negligible share persists further, with successes becoming exponentially less likely as the generation horizon extends, and 5% of sequences remain structurally valid after 20 years.

## 7 DISCUSSION

We propose a novel encoding method for converting longitudinal administrative data into sequences and train a generative life sequence Transformer model. Despite its relatively small size, the model learns to reproduce realistic and causally consistent labour market histories and generates coherent synthetic life paths conditional on past trajectories, showing the potential of Transformer architectures in this domain.

Through empirical exercises, we study the strengths and limitations of our model. On the one hand, it successfully reproduces causal patterns, from birth-month effects on retirement to the motherhood penalty. On the other hand, the exercises reveal systematic sensitivities. Rule-driven discontinuities are smoothed over when the underlying events are rare in the data; treatment effects vary depending on how counterfactuals are defined, whether by comparing mothers to non-mothers or by truncating retirement histories at different horizons; and the timing of effects can shift, with the model sometimes understating short-run penalties but prolonging them over the longer term.

These findings highlight a broader challenge: counterfactual inference with generative models depends on how we define treatment, select controls, and decide where to cut the input history. These are not just technical choices but conceptual ones, raising fundamental questions about what constitutes a life sequence counterfactual. They also suggest future research on developing principled methods for introducing researcher-imposed variation into simulated life sequences, ensuring theory-guided design of counterfactual scenarios.

Finally, while our results show encouraging alignment with empirical findings, they also remind us that predictive modelling of individual life courses carries ethical and societal risks. Structural biases present in training data will be reinforced. Without appropriate human oversight, there is also the potential for misuse, for example, in automated decision-making systems that influence resource allocation or eligibility for benefits. Future applications must combine methodological advances with safeguards for fairness, privacy, and responsible use.

### Acknowledgements

The authors acknowledge European NextGenerationEU funding through the PNRR project "Age-It - Ageing well in an ageing society", code PE0000015, MUR 341/2022 tender.

## References


A. Abadie, A. Diamond, and J. Hainmueller. Synthetic control methods for comparative case studies: Estimating the effect of california's tobacco control program. *Journal of the American statistical Association*, 105(490):493–505, 2010.

D. Acemoglu, L. Fergusson, and S. Johnson. Population and conflict. *The Review of Economic Studies*, 87 (4):1565–1604, 2020.

A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

C. Alonso. Beyond labor market outcomes: The impact of the minimum wage on nondurable consumption. *Journal of Human Resources*, 57(5):1690–1714, 2022.

C. Ardito, R. Leombruni, D. Blane, and A. d'Errico. To work or not to work? the effect of higher pension age on cardiovascular health. *Industrial Relations: A Journal of Economy and Society*, 59(3):399–434, 2020.

T. Bachlechner, B. P. Majumder, H. Mao, G. Cottrell, and J. McAuley. Rezero is all you need: Fast convergence at large depth. In *Uncertainty in Artificial Intelligence*, pages 1352–1361. PMLR, 2021.

A. Bena, R. Leombruni, M. Giraudo, and G. Costa. A new italian surveillance system for occupational injuries: characteristics and initial results. *American journal of Industrial Medicine*, 55(7):584–592, 2012.

G. Brunello and R. Miniaci. Benefit transfers in italy: an empirical study of mobility lists in the milan area. *Oxford Bulletin of Economics and statistics*, 59(3): 329–347, 1997.

D. Card and A. B. Krueger. Minimum wages and employment: A case study of the fast-food industry in new jersey and pennsylvania. *The American Economic Review*, 84(4):772–793, 1994. URL `http://www.jstor.org/stable/2118030`.

N. Cartwright. *Counterfactuals in Economics: A Commentary*, pages 191–216. Massachusetts Institute of Technology Press, 2007a.

N. Cartwright. Are RCTs the gold standard? *BioSocieties*, 2(1):11–20, 2007b.

A. Casarico and S. Lattanzio. Behind the child penalty: understanding what contributes to the labour market costs of motherhood. *Journal of Population Economics*, 36(3):1489–1511, 2023.

K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller. Rethinking attention with performers, 2022. URL `https://arxiv.org/abs/2009.14794`.

M. E. Consens, C. Dufault, M. Wainberg, D. Forster, M. Karimzadeh, H. Goodarzi, F. J. Theis, A. Moses, and B. Wang. Transformers and genome language models. *Nature Machine Intelligence*, pages 1–17, 2025.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

E. M. Foster. Causal inference and developmental psychology. *Developmental Psychology*, 46(6):1454, 2010.

N. Geneva and N. Zabaras. Transformers for modeling physical systems. *Neural Networks*, 146:272–289, 2022.

J. J. Heckman. Econometric causality. *International statistical review*, 76(1):1–27, 2008.

L. S. Jacobson, R. J. LaLonde, and D. G. Sullivan. Earnings losses of displaced workers. *The American Economic Review*, pages 685–709, 1993.

S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. Brubaker. Time2vec: Learning a vector representation of time, 2019. URL `https://arxiv.org/abs/1907.05321`.

H. Kim, J. Na, and W. B. Lee. Generative chemical transformer: neural machine learning of molecular geometric structures from chemical language via attention. *Journal of chemical information and modeling*, 61(12):5804–5814, 2021.

H. Kleven, C. Landais, and J. E. Søgaard. Children and gender inequality: Evidence from denmark. *American Economic Journal: Applied Economics*, 11(4): 181–209, 2019.

R. Leombruni, R. Quaranta, and C. Villosio. Note di pubblicazione di whip v. 3.2. *WHIP Technical Report No. 1/2010*, 2010.

J. S. Levy. Counterfactuals, causal inference, and historical analysis. *Security Studies*, 24(3):378–402, 2015.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization, 2019. URL `https://arxiv.org/abs/1711.05101`.

A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

D. B. Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American statistical Association*, 100(469):322–331, 2005.

G. Savcisens, T. Eliassi-Rad, L. K. Hansen, L. H. Mortensen, L. Lilleholt, A. Rogers, I. Zettler, and S. Lehmann. Using sequences of life-events to predict human lives. *Nature Computational Science*, 4(1): 43–56, 2024.

J. S. Sekhon. Opiates for the matches: Matching methods for causal inference. *Annual Review of Political Science*, 12(1):487–508, 2009.

L. N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL `https://arxiv.org/abs/2104.09864`.

G. Thomas. After the gold rush: Questioning the "gold standard" and reappraising the status of experiment and randomized controlled trials in education. *Harvard Educational Review*, 86(3):390–411, 2016.

K. Vafa, E. Palikot, T. Du, A. Kanodia, S. Athey, and D. M. Blei. Career: A foundation model for labor sequence data, 2024. URL `https://arxiv.org/abs/2202.08370`.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Y. Wang, H. Huang, C. Rudin, and Y. Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization, 2021. URL `https://arxiv.org/abs/2012.04456`.

S. Watanabe. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance, 2023. URL `https://arxiv.org/abs/2304.11127`.

Z. Yang, A. Mitra, W. Liu, D. Berlowitz, and H. Yu. TransformEHR: transformer-based encoder-decoder generative model to enhance prediction of disease outcomes using electronic health records. *Nature Communications*, 14(1):7857, 2023.

B. Zhang and R. Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

## Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Not Applicable
   (b) Complete proofs of all theoretical results. Not Applicable
   (c) Clear explanations of any assumptions. Not Applicable

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes
   (b) The license information of the assets, if applicable. Yes
   (c) New assets either in the supplemental material or as a URL, if applicable. Yes
   (d) Information about consent from data providers/curators. Not Applicable

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable

# Supplementary Materials

Counterfactual inference is a cornerstone of scientific reasoning in disciplines such as political science Sekhon (2009), economics (Heckman, 2008), history (Levy, 2015), and psychology (Foster, 2010), and has also sparked ongoing methodological debate (Cartwright, 2007a). Counterfactual identification requires constructing hypothetical scenarios that resemble what would have happened in the absence of an event assumed to be causal (Rubin, 2005). Such a scenario allows quantifying the impacts of specific events, like treatments or policies, or the underlying causes of observed outcomes. Typical research questions include: What is the causal effect of increasing the minimum wage on household consumption (Alonso, 2022)? What are the drivers of civil conflict (Acemoglu et al., 2020)?

The gold standard approach for identifying causal effects involves randomised interventions, in which the treatment is assigned randomly to some individuals and withheld from others (Cartwright, 2007b). The difference in outcomes between the treated and control groups is interpreted as the causal effect of treatment. Randomisation ensures that characteristics that might correlate with the outcome of interest, such as gender, age, and socioeconomic status, are, on average, evenly distributed across the treatment and control groups. Thus, any systematic outcome difference can be attributed to the treatment itself.

However, randomisation rarely occurs naturally and can be challenging to implement even in controlled settings, due to ethical constraints and substantial costs (Thomas, 2016). As a result, economists and other researchers have developed strategies and mathematical models to identify suitable control groups without relying on random treatment assignment. Despite being widely adopted, these observational methods often rely on specific assumptions about the data and the underlying causal mechanisms, which are typically difficult to verify in practice.

## A Difference-in-differences

Difference-in-differences (DiD) is one of the most commonly used causal identification strategies for evaluating the effects of policy interventions by comparing changes in an outcome variable between a treatment group and a control group, both before and after the intervention. Here, the main assumption is that the treated and control groups would have followed parallel trends over time in the absence of the policy or event under evaluation. This assumption, while critical, is rarely validated and often debated in empirical applications.

Card and Krueger (1994) on minimum wage in New Jersey is considered one of the most famous DiD studies; the authors compared employment in the fast food sector in New Jersey and in Pennsylvania, in February 1992 and in November 1992, after New Jersey's minimum wage rose from $4.25 to $5.05 in April 1992. Observing a change in employment in New Jersey only, before and after the treatment, would fail to control for omitted variables such as weather and macroeconomic conditions of the region. By including Pennsylvania as a control in a difference-in-differences model, any bias caused by shared variables between New Jersey and Pennsylvania is implicitly controlled for, even when these variables are unobserved. Assuming that New Jersey and Pennsylvania have parallel trends over time, Pennsylvania's change in employment can be interpreted as the change New Jersey would have experienced had it not increased the minimum wage, and vice versa. The evidence suggested that the increased minimum wage did not lead to a decrease in employment in New Jersey, contrary to what some economic theories would suggest.

### A.1 Event Study

While powerful, DiD is not always applicable, especially in settings where no natural control group exists. Consider, for example, the question: What is the causal effect of childbirth on a woman's career trajectory? Ideally, one would randomise fertility to estimate this effect, but such an experiment is infeasible. Applying DiD would require identifying a group of women identical to mothers in both observable and unobservable characteristics, and then

assuming their career paths would have evolved similarly, absent childbirth. In practice, achieving such a match is highly challenging.

In such cases, researchers often turn to an event study design (Jacobson et al., 1993), a generalisation of DiD that exploits variation in treatment timing. Event studies are especially useful when all units are eventually treated at different times. The method compares changes in outcomes before and after treatment for each unit, leveraging variation across units treated at different points in time. In the childbirth example, this allows for estimating treatment effects by aligning individuals around the "event" (birth of a child) and comparing pre- and post-treatment trajectories. Like DiD, event studies rely on a version of the parallel trends assumption—that, absent the event/treatment, the trend of the outcome would have continued as observed in the pre-treatment period.

### A.2 Propensity Score

While Difference-in-Differences (DiD) relies on the parallel trends assumption, in many empirical settings, treatment and control groups differ substantially in observable characteristics, making the assumption less credible. One strategy to mitigate this concern is to combine DiD with Propensity Score Matching (PSM), a method introduced by Rosenbaum and Rubin (1983). The core idea of PSM is to construct a control group that is statistically similar to the treated group in terms of observed covariates, by matching treated and untreated units with comparable estimated probabilities of treatment (the "propensity score").

When PSM is combined with DiD, the matching procedure reduces bias from differences in observable characteristics, while the DiD framework accounts for unobserved time-invariant heterogeneity. This hybrid approach is particularly valuable when treatment assignment is not random and when untreated units differ systematically from treated ones. By matching first and then applying DiD, researchers ensure that the parallel trends assumption is more plausible for the reweighted or matched sample.

### A.3 Synthetic Control

The Synthetic Control Method (SCM) is another causal inference approach closely related to difference-in-differences. It is particularly well-suited to cases where a single unit (such as a country, region, or firm) undergoes treatment, and no comparable control group exists. Instead of relying on a parallel trends assumption between predefined groups, SCM constructs a synthetic version of the treated unit using a weighted combination of untreated units. These weights are chosen to closely match the treated unit's outcome trajectory and covariates in the pre-treatment period.

This method was originally proposed by Abadie et al. (2010) to study the economic impact of California's 1988 tobacco control program. Rather than comparing California to a single state, SCM used a weighted average of other U.S. states to construct a "synthetic California" that closely mirrored its economic indicators before the intervention. Any divergence between the treated unit and its synthetic control after treatment can then be attributed to the intervention, under the assumption that the synthetic control approximates the counterfactual trajectory the treated unit would have followed in the absence of treatment.

Compared to standard DiD, SCM is more flexible and data-driven. It relaxes the requirement that the treatment and control units follow identical trends and instead ensures a close match in the pre-treatment period. This makes SCM particularly valuable in comparative case studies or when few treatment units are available, though it also comes with limitations—such as sensitivity to model specification and the requirement for a rich pool of potential control units.

## B Work History Italian Panel Data

In Italy, the organisation that collects social insurance contributions and distributes social security benefits, such as pensions, maternity leave, and unemployment benefits, is the Istituto Nazionale della Previdenza Sociale (INPS). As such, INPS gathers data on significant workforce segments, both active and inactive. The reference population comprises all individuals employed by private firms, workers with quasi-dependent employment arrangements, self-employed, retirement spells, and non-working spells in which the individual received social benefits, like unemployment subsidies or mobility benefits. The workers for whom activity is not observed are those who

worked as freelancers and have an autonomous security fund. The observed group represents all manufacturing, construction, and services production sectors. However, it does not cover public employment, which involves approximately 3.5 million people, predominantly in the education and health sectors. Certain sectors, such as agriculture, are not fully represented in INPS data since its data does not cover any activity in the shadow economy. If an individual is observed working one year in a private company, and then he/she start working in the shadow economy, INPS will register only that one year the individual was legally working.

The Work History Italian Panel (WHIP) (Leombruni et al., 2010; Bena et al., 2012) consists of a systematic sample of 1 in 15 individuals drawn from the INPS database based on their date of birth. This database allows us to track the employment histories of sampled individuals, including all periods of employment recorded by INPS, retirement, and any time spent receiving social security benefits, such as unemployment subsidies or maternity leave. The WHIP currently covers the period from 1985 to 2019.

For our analysis, we limit the sample to WHIP observations recorded from 1990 to 2015 for training. We started in 1990 as the information's reliability improved significantly from that date onward. Additionally, we apply the following selection criteria to further refine our sample:

1. Have information until the end of the observed period (2015)
2. Effective period observed greater than or equal to half of the worker's potential period. Thus, we include only individuals observed for at least half of their maximum possible participation period, where labour force participation is assumed to start at age 30 and is bounded by the 1990–2015 data window and an 85-year lifespan. For example, an individual born in 1975 turns 30 in 2005, we expect to observe him/her from 2005 to 2015. If we observe him only from 2013 to 2015 (3 years), the ratio between effective (3) and potential (10) is 0.3. Therefore, we exclude this individual from the sample.
3. No missing date of birth
4. At least five records
5. Age of entry in the panel higher than 15 and lower than 80. We exclude those individuals who are too young or too old.
6. Fraction of returns or investment income over the total number of records lower than 90%. We exclude people who are fully detached from the workforce.

| Initial Number of Individuals | 4032674 |
|---|---|
| **Selection Criteria** | **% Drop** |
| 1. | 13.24 |
| 2. | 16.45 |
| 3. | 0.16 |
| 4. | 6.5 |
| 5. | 4.91 |
| 6. | 13.83 |
| Final Number of Individuals | 1652877 |

Table 3: Sample selection. Each row in the table corresponds to a selection criterion. The second column reports the percentage of individuals discarded, regardless of the other selection criteria.

### B.1 Labour Records

The primary individual information in WHIP is yearly income. This is the most important information for INPS to compute contributions and monitor social security benefits. Many other variables are attached to each yearly income. For our study, we focus on 15 income-related variables.

**Demographic information** WHIP records include an individual's demographic/background information. These are (1) sex, (2) birth date, and (3) macro-area of birth. In our case, sex is a binary variable that encodes male or female. We retrieve only the birth year from the birth date to encode the age when a labour event occurs.

The macro-area of birth specifies one of the 5 Italian groups of regions (North-East, North-West, Center, South, or Island). There is a sixth category for foreign-born people.

**Workplace information** Another set of variables pertains to features of the workplace. Notably, only for working individuals the following information is stored, e.g., residents receiving a pension, unemployment benefits or rents, do not have any of these specified, while for a self-employed person we observe only the workplace location.

(4) Economic activity/Sector denotes the services or goods that the company produces. WHIP encodes this information through the ATECO classification system[1]. ATECO is the national version of the European nomenclature, Nace Rev. 2, Regulation (EC) No 1893/2006 of the European Parliament and of the Council of 20 December 2006[2]. The ATECO has a hierarchical tree-like system, each additional digit provides a more detailed breakdown of the service, see the example in the following table. For our study, we use only the first four digits of the ATECO code to balance informativeness and sample size of each category.

| ATECO CODE | Definition |
|---|---|
| G | Wholesale and retail trade |
| 45 | Repair of motor vehicles and motorcycles |
| 45.1 | Trade in motor vehicles |
| 45.11 | Trade in passenger cars and light motor vehicles |

(5) Enterprise size is encoded in 5 ordinal categories: less than 10 employees, between 10 and 19, between 20 and 199, between 200 and 999, and more or equal to 1000. Finally, (6) Workplace location can be one of the 107 Italian provinces, plus an additional category for foreign locations.

**Individual income information** The last set of features provides details about the individual's income and its nature. (7) Labour force status is specific to WHIP, i.e., it does not follow any international classification system. It describes the type of income, i.e., in what context the income is associated with a person. Examples of these statuses include: Employee, Pensioner, Self-employed, Unemployed, encompassing work-related and other socioeconomic statuses. Employee, Self-Employed, or Para-Subordinate (e.g., sub-contractors, consultants, people with work contracts but not all the benefits) have one additional modifying information, the (8) Working title which indicates the qualification in the job and may fall into one of 11 categories: in training, Workman, Clerk, Junior Manager, Senior Manager, technical-administrative, Sportsman, Artist, Voucher, Collaborator, or Professional.

For Employee records only, we have four additional modifiers: (9) work intensity, (10) maternity leave intensity, (11) sick leave intensity, and whether the work is (12) part-time or full-time. The three intensities describe the average number of weeks in a month when an employee was working, on maternity leave, or sick. We derive the intensity by dividing the overall number of weeks spent in a given category (e.g., working) by the total number of months the employment relationship lasted in a given year. Therefore, each record can take a maximum value of around 4.3 (52 weeks divided by twelve), and a minimum value of 0. Notably, maternity leave is only recorded for female employees. Finally, we discretise the intensity into five categories based on the rounded value. If, for example, the intensity work is zero, we encode it as `<WRKINT_S0>`, if the intensity is less than 1, we encode it as `<WRKINT_S1>`, for intensities greater than 4, they are encoded into the category `<WRKINT_S4+>`.

The final attributes are the (13) month the event started, its (14) duration in months within a year, and (15) income during the event. The nature of income differs depending on the labour force status. For example, for Employees, Self-employed, Para-Subordinate, and Rentiers, it is the reported income from which the INPS calculates social security contributions. For Unemployed and Pensioner, the income is the amount that INPS provides to the individual. The raw version of the income is a yearly value; we divide it by the duration in months of a given labour force status within a year and derive a monthly income. We further adjust it for inflation, deriving a real monthly income, and discretise this value into 100 quantiles.

**Vocabulary** The collection of all concept-tokens creates the vocabulary $\mathcal{V}$. The vocabulary is made up of tokens for all categories of each variable and has size $|\mathcal{V}| = 661$. The vocabulary $\mathcal{V}$ is generated from the training

[1] `https://codiceateco.it/sezioni`

[2] `https://eur-lex.europa.eu/eli/reg/2006/1893/oj/eng`

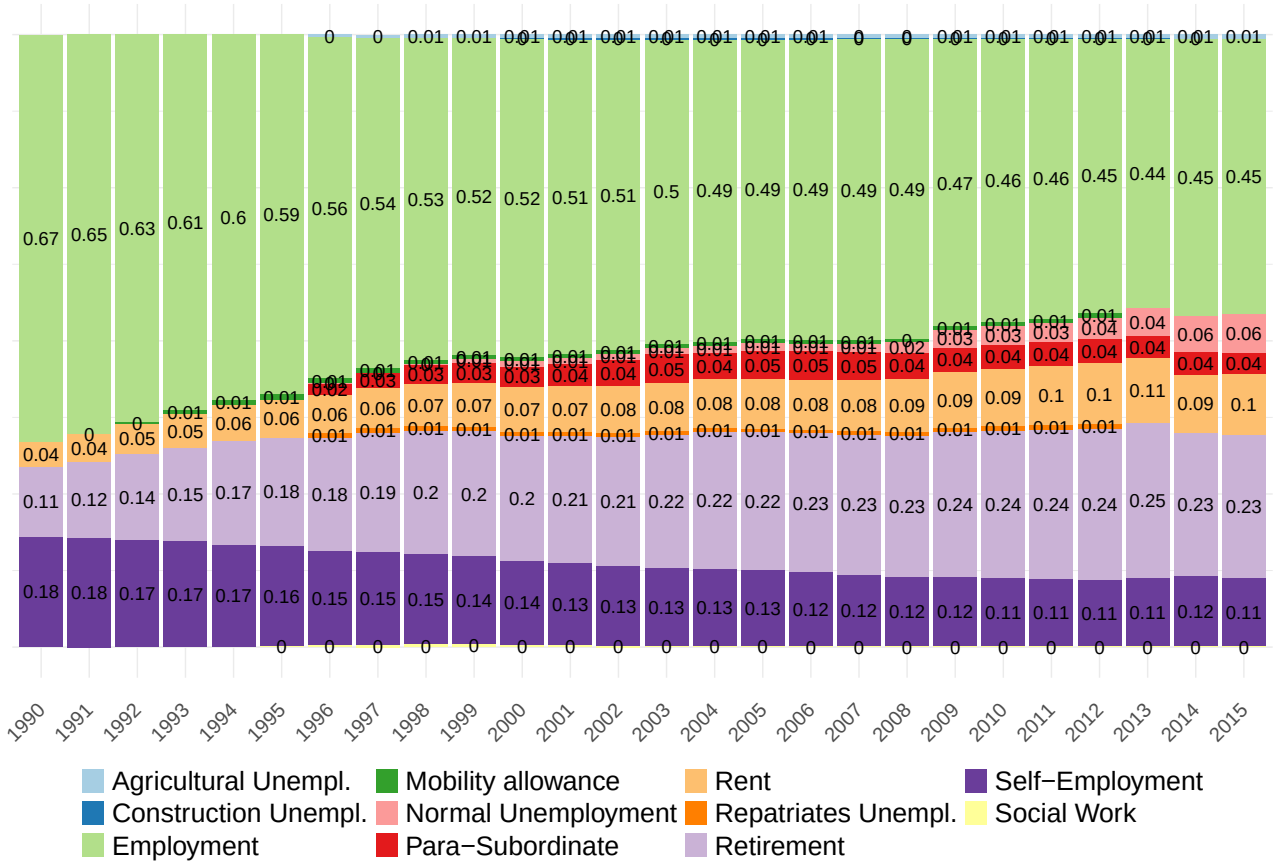



Figure 6: Labour Force Status composition for each year in the global dataset

| | | |
|---|---|---|
| General | Number of Individuals | 1652877 |
| | Average number of events | 30.48 |
| | Median Year | 2005 |
| | Median Age | 42 |
| Background | % Female | 37.1 |
| | % North-West | 22.6 |
| | % North-East | 18.4 |
| | % Center | 15.9 |
| | % South | 22.4 |
| | % Islands | 9.73 |
| | % Foreign born | 11 |
| | % Missing Area of Birth | 0.0005 |
| | Median Year of Birth | 1966 |

Table 4: Descriptive Statistics of the Dataset

subset and does not include all the possible categories in the data. Table 5 lists all variables and their possible categories.

| **Token Type** | **Variables** | **N. Categories** | **Encoding** |
|---|---|---|---|
| Background Information | Year of Birth | 86 | 1913-1998 |
| | Sex | 2 | F-M |
| | Area of Birth | 6 | 1-6 (Foreign) |
| Labour | Enterprise type | 280 | ATECO 4-digits |
| | Enterprise size | 5 | WHIP classification |
| | Work Location | 110 | Italian provinces codes |
| | Labour force status | 11 | WHIP classification |
| | Work title | 11 | WHIP classification |
| | Work Arrangements | 2 | Part time - Full time |
| | Work intensity | 5 | Threshold based |
| | Maternity leave intensity | 5 | Threshold based |
| | Sick leave intensity | 5 | Threshold based |
| | Income | 100 | Quantile based |
| | Start Month | 12 | 1-12 |
| | Duration | 12 | 1-12 |
| Special | Special | 11 | [PAD]...[UNK] |

Table 5: Vocabulary specification

**Labour life sequence** Each life sequence is represented as a list of discrete tokens, encoding time-indexed events and their characteristics. Sequences are capped at a maximum of 1,560 tokens. Any sequence shorter is padded with a padding token, and any sequence longer is truncated starting from the earliest years (i.e., from the start of the sequence), ensuring that all input tensors are of consistent shape for training and inference. Longer sequences are truncated by whole years so as not to lose the temporal structure of the sequence (i.e., all tokens in a year are deleted until the sequence is at most 1,560 tokens long).

When an individual has no recorded activity in a given year, the year is still represented in the sequence. We then insert a silent event placeholder: `<MONTH_1>` followed by `<DURATION_12>` with no attributes in between, to preserve the temporal regularity of the sequence. The ordering of tokens within a sequence is chronological. Events that begin in the same calendar month are randomly shuffled relative to each other, regardless of their duration. This is based on the assumption that events co-occurring in the same month cannot be reliably ordered based on observed data, and shuffling them prevents the model from overfitting to arbitrary patterns in event ordering. Outside of these same-month shuffles, the rest of the sequence remains ordered. Besides same-month shuffling, we introduce a small amount of token dropout, randomly removing about 1% of tokens during training. This regularisation mechanism discourages the model from overfitting to missing data and promotes robustness.

The dropout is uniformly applied and does not target any specific token type.

Inside each event, there is a meaningful and deliberate order to the tokens. Their order reflects a semantic priority derived from the downstream modelling goals. For example, income is placed before ATECO (economic activity), because for many of our tasks, we are more interested in how income can inform or predict the sector of employment, rather than the reverse. That is, we assume income to be more causally upstream in our use cases. While alternative orderings are possible, changing the token order within events would require retraining, as the model implicitly learns positional correlations. Nevertheless, this ordering can be adjusted in future work depending on the application.

The sequence lengths (measured as the number of tokens per individual life trajectory, excluding background information) have the following descriptive statistics

| | Mean | Median | Mode | Minimum | Maximum |
|---|---|---|---|---|---|
| Tokens | 232 | 220 | 128 | 21 | 1554 |

Table 6: Descriptive statistics of sequences

**This example is not based on an individual in the dataset, any resemblance to a real individual is coincidental.** Consider an example as an illustration of the encoding:

`A3 | F | MONTH_1 | YEAR_1942`,

i.e., a female born in January of 1942 in the Center region of Italy. Assume that in the first year of measurement (1990), she worked full time during the whole year, then had a year of no recorded activity, and then got another job in a different province for the entire following year:

`[BOL] (0, 0) | MONTH_1 (1, 48) | [TIPO_1] (1, 48) | INCOME_20 (1, 48) | WRKT_1 (1, 48) | WRKP_XX (1, 48) | ATE_A123 (1, 48) | FSIZE_1 (1, 48) | FULL_TIME (1, 48) | WRKINT_S3 (1, 48) | SIKINT_S0 (1, 48) | MATINT_S0 (1, 48) | DUR_12 (1, 48) | [EOY] (1, 48) | MONTH_1 (2, 49) | DUR_12 (2, 49) | [EOY] (2, 49)| MONTH_1 (3, 50) | [TIPO_1] (3, 50) | INCOME_22 (3, 50) | WRKT_2 (3, 50) | WRKP_YY (3, 50) | ATE_A124 (3, 50) | FSIZE_1 (3, 50) | FULL_TIME (3, 50) | WRKINT_S3 (3, 50) | SIKINT_S0 (3, 50) | MATINT_S0 (3, 50) | DUR_12 (3, 50) | [EOY] (3, 50)`

The numbers inside parentheses next to the tokens are the year (measured as years after 1989) and her age. Notice that the `[EOY]` token is measured with the current year and age, and once this token appears, the year and age increase by one. Hence, when the model generates an `[EOY]` token, time embeddings are updated deterministically. Each of her working spells is different because of its attributes, though for both of her working spells, she worked an average of 2 to 3 weeks each month and never took sick or maternity leave.

Her life sequence continues for every year until 2015, and when she is around 60 years of age, she retires and starts getting a pension from INPS:

`[EOY] (12, 59) | MONTH_1 (13, 60) | [TIPO_1] (13, 60) | INCOME_39 (13, 60) | WRKT_2 (13, 60) | WRKP_YY (13, 60) | ATE_A124 (13, 60) | FSIZE_1 (13, 60) | FULL_TIME (13, 60) | WRKINT_S3 (13, 60) | SIKINT_S0 (13, 60) | MATINT_S0 (13, 60) | DUR_5 (13, 60) | MONTH_5 (13, 60) | [TIPO_10] (13, 60) | INCOME_35 (13, 60) | DUR_8 (13, 60) | [EOY] (13, 60) | MONTH_1 (14, 61) | [TIPO_10] (14, 61) | INCOME_35 (14, 61) | DUR_12 (14, 61) | [EOY] (14, 61)`

Notice she still works for the first 5 months of 2005 and only starts receiving her pension in May. After that, she is likely to stay on a pension scheme until 2015.

## C Model architecture

Transformers are a class of neural network architectures initially developed for natural language processing tasks, such as machine translation or text summarization (Vaswani et al., 2017), but are now being applied to a wide variety of sequence types from different domains, including protein discovery Consens et al. (2025), health predictions (Yang et al., 2023), music generation Agostinelli et al. (2023). Their strength lies in the self-attention mechanism, which enables them to model long-range dependencies in sequential input.

One of the most influential *encoder-only* Transformer models is BERT (*Bidirectional Encoder Representations from Transformers*), introduced by Devlin et al. (2019). BERT is trained using a masked language modelling (MLM) objective, where a subset of tokens in the input sequence is randomly masked, and the model learns to predict them using bidirectional context, i.e., both left and right tokens. This bidirectional attention mechanism enables BERT to build rich, context-aware representations of entire sequences, making it highly effective for discriminative tasks such as text classification, question answering, and semantic similarity. Since BERT processes the entire input at once, it excels at tasks that require a holistic understanding of the input sequence, rather than step-by-step prediction. An *encoder-only* version of our model is similar to Savcisens et al. (2024)'s Life2Vec, which is inspired by the BERT architecture and training objective.

In contrast, GPT (*Generative Pre-trained Transformer*) exemplifies a *decoder-only* architecture. Introduced by Radford et al. (2018), GPT is trained with an auto-regressive objective, predicting the next token in a sequence based solely on the preceding tokens. GPT uses causal (masked) self-attention, which restricts each token's attention to preceding positions in the sequence, preventing information from "leaking" from future tokens. Unlike BERT, GPT does not explicitly mask input tokens, but the causal attention mechanism ensures that generation proceeds in a left-to-right manner. This makes the GPT architecture particularly well-suited for generative tasks, such as language generation, sequential modelling, and, in our case, simulating counterfactual life sequences. Figure 7 illustrates the decoder-only architecture we train in this paper.

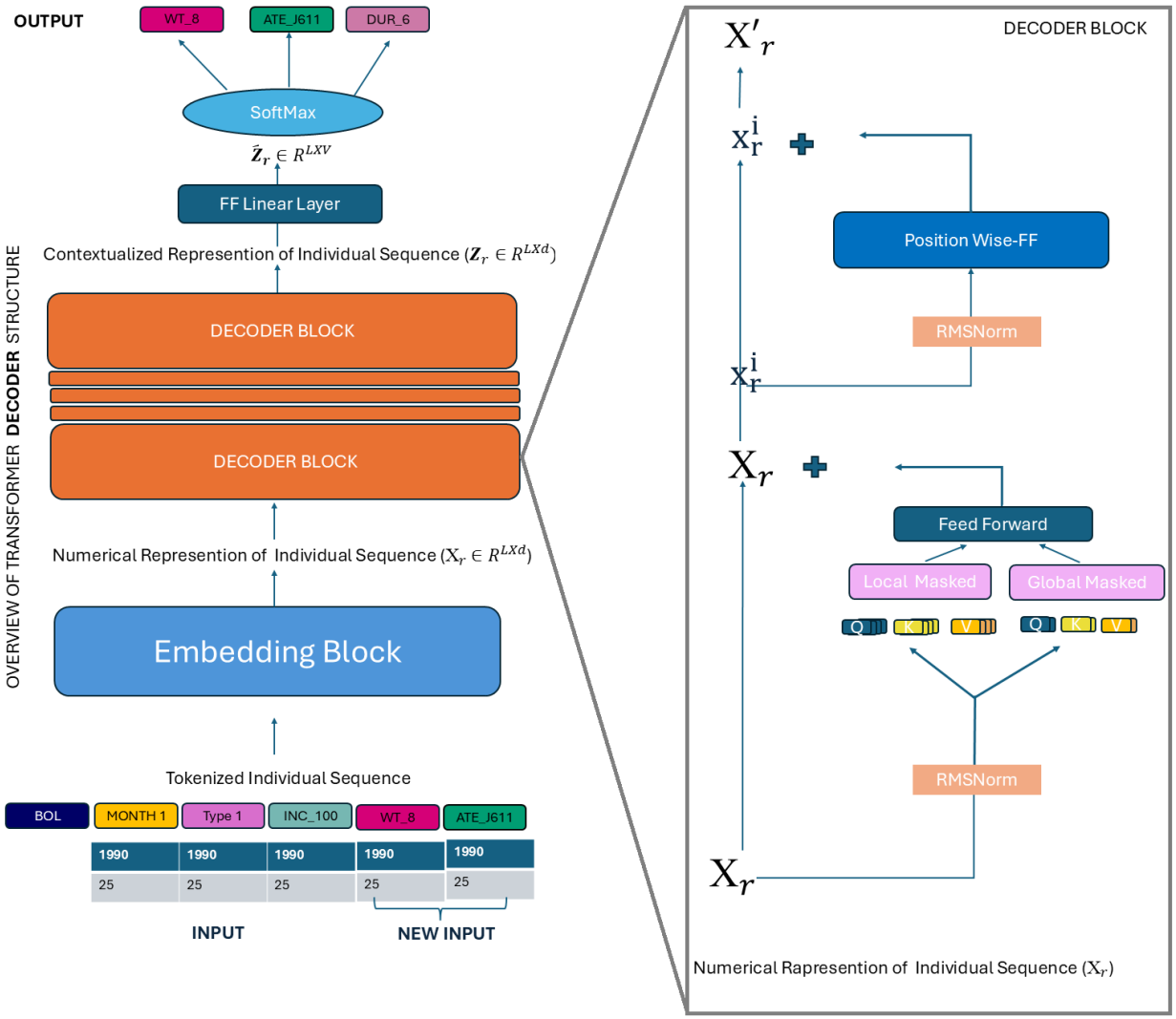



Figure 7: Overall architecture of our decoder-only model. Each tokenised life sequence is passed through an embedding block and a stack of decoder blocks. Each token, consisting of concept, age, and year information, is embedded using three components: concept embeddings retrieved from a learned lookup matrix $\mathcal{E}_\mathcal{V}$ (normalised by subtracting its mean), and temporal features (age and year) encoded via Time2Vec. These are combined with learnable positional embeddings to form the input token representations. The resulting sequence is processed through multiple decoder blocks, each structured to preserve the autoregressive nature of the task. Each block starts with RMS normalisation, followed by a masked multi-head attention layer (using the Performer mechanism with local and global heads), ensuring that each token only attends to preceding tokens. The output of the attention layer is combined with the input via a residual connection. This is followed by another RMS normalisation and a position-wise feed-forward layer, whose output is again added through a residual connection. Finally, the contextualised output representations are passed through a linear projection head and to a softmax function that assigns a probability to each token in the vocabulary. The predicted token is the one with the highest probability score.

# D Experimental details

This section outlines the experimental setup, notation, and estimands, as well as the process of converting generated simulations into treatment effects that are directly comparable to their empirical counterparts. Our goal is to place generative model-based evaluation on the same footing as standard causal inference designs in applied economics. We build on the event study and propensity-score frameworks described in Appendix A.1–A.2, adapting them to the context of sequences generated from our model.

**Common setup and notation** Let $i$ index selected individuals in the test set. Each empirical exercise is framed in terms of a binary treatment $T_i$ and a scalar outcome $Y_i$. The central estimand is the Average Treatment Effect (ATE), defined as the expected difference in outcomes between treated and untreated/control units:

$$\text{ATE} = \mathbb{E}[Y_i(1) - Y_i(0)],$$

where $Y_i(1)$ and $Y_i(0)$ denote the potential outcomes for individual $i$ under treatment and control, respectively. In practice, we never observe both potential outcomes for the same person; however, under appropriate research designs (such as random assignment, natural experiments, or valid identification strategies), we can estimate the ATE as the difference in mean outcomes between the treated and control groups. In our experiment, these quantities are:

- Pension: $T_i = 1$ if individual $i$ was born in January, $T_i = 0$ if born in December. The outcome $Y_i$ is the age at first pension claim, measured in months.
- Unemployment: $T_i = 1$ if individual $i$ experienced their first unemployment spell at or after 40 years of age (480 months), $T_i = 0$ otherwise. The outcome $Y_i$ is the number of months spent in the first unemployment spell. The estimand is local, defined within a symmetric window $h$ around the 480-month cutoff.
- Maternity: $T_i = 1$ if individual $i$ experiences at least one maternity leave episode in her sequence, $T_i = 0$ otherwise. The outcome $Y_i$ is the average post-maternity annual earnings.

For maternity, treatment is endogenous rather than rule-based. As such, the definition of a control group is non-trivial. Thus, we employ an event study design and a propensity score matching approach (Section A.1) to align careers around childbirth events and identify a suitable control group of non-mothers that matched he pre-birth characteristics of the mothers. To make results comparable across experiments, we summarise the estimated time-path into a single number: the mean effect over the birth year and the first ten post-birth years (see Section D.2).

**From simulations to outcomes** For every person $i$ we generate $K$ continuations from a fixed cutoff of their observed history. This cutoff point corresponds to the moment just before the outcome of interest takes place:

- Pension: one year before observed retirement (denoted 1y in tables) or four years before retirement (4y). An example of the former setup is shown in Figure 8.
- Unemployment: the year of the first observed unemployment spell.
- Maternity: the year of the first observed maternity leave (offset–0), as well as one year prior (offset–1), to allow counterfactual generation without maternity.

Each simulation $k$ yields a synthetic outcome $\tilde{Y}_{ik}$. We compute the person-level Monte Carlo mean

$$Y_i^{\text{sim}} \;=\; \frac{1}{K}\sum_{k=1}^{K} \tilde{Y}_{ik},$$

and then use $\{Y_i^{\text{sim}}\}_{i=1}^{n}$ to estimate treatment effects in the same way we do with real outcomes $\{Y_i^{\text{real}}\}_{i=1}^{n}$. This ensures that all ATEs are defined at the individual level, not at the simulation-draw level.




(a) Original sequence




(b) Sequence Cut




(c) Final input sequence

Figure 8: Example of the simulation setup: (a) shows the original work sequence, (b) the sequence cut applied one year before retirement occurs, and (c) the resulting input sequence used for simulation.

We report sample sizes $n$, number of simulations $K$, and uncertainty measures obtained via a paired bootstrap at the individual level. Specifically, we resample individuals (with replacement) from the fixed analysis sample, e.g., January or December retirees for pensions, or individuals within the 40-year bandwidth for unemployment. For every bootstrap sample, we recompute both the empirical and model ATEs,

$$\widehat{\mathrm{ATE}}_{\mathrm{emp}} = \mathbb{E}[Y_i^{\mathrm{real}} \mid T_i{=}1] \;-\; \mathbb{E}[Y_i^{\mathrm{real}} \mid T_i{=}0], \qquad \widehat{\mathrm{ATE}}_{\mathrm{model}} = \mathbb{E}[Y_i^{\mathrm{sim}} \mid T_i{=}1] \;-\; \mathbb{E}[Y_i^{\mathrm{sim}} \mid T_i{=}0],$$

and their difference

$$\Delta = \widehat{\mathrm{ATE}}_{\mathrm{model}} - \widehat{\mathrm{ATE}}_{\mathrm{emp}}.$$

We use $B = 1000$ resamples and form percentile intervals from the empirical distributions. In the tables we therefore report (i) point estimates of $\widehat{\mathrm{ATE}}_{\mathrm{emp}}$ and $\widehat{\mathrm{ATE}}_{\mathrm{model}}$, (ii) their respective 95% confidence intervals, and (iii) the signed discrepancy $\Delta$ with its own confidence intervals.

### D.1 Mobility allowance duration

The first exercise examines whether the model can replicate age-based differences in the duration of unemployment benefits, specifically the mobility allowance, a now-discontinued income support program for workers displaced due to firm closures in Italy. As shown in Figure 6, less than 1% of records in our global sample fall under this category.

Under the policy rules in place until January 2017, the maximum benefit duration depended on the worker's age at the time of job loss. Workers over 40 were eligible for up to 24 months of benefits, while those under 40 could receive income support for up to 12 months (Brunello and Miniaci, 1997). This sharp policy threshold makes it an ideal setting to test whether the model internalises and reflects such causal discontinuities in allowance duration.

We drew individuals with at least one registered episode of mobility allowance from the test set who also have only standard employment or retirement spells in their histories. To increase homogeneity, we exclude individuals with any episode of self-employment, artistic or athletic careers, or voucher-based contracts. Additionally, we retain only those employed in northern Italian provinces at the time of firm closure, since a worker from the south was allowed one additional year of benefits after 40 years of age. We restrict the sample to workers who began receiving the benefit between the ages of 30 and 50, and who have less than 1% missing data on key variables: economic sector (ATECO), work province, income, and job title. The resulting sample consists of 1,502 individuals.

For each individual, we provide the model with their life trajectory up to the year we first observe the receipt of unemployment benefits. The model then generates the sequence from that point onward. This setup allows us to test whether the simulated unemployment subsidy duration reflects the correct age-based discontinuity.

To quantify the effect, we exploit the policy cutoff at 40 years of age (480 months) and estimate a local average treatment effect (ATE) within symmetric windows $h \in \{12, 48, 96\}$ months around the threshold. Let $A_i$ denote age at first unemployment, and define the sample and treatment within the window, and treatment dummy $T_i$

$$\mathcal{I}(h) = \{i : |A_i - 480| \le h\}, \qquad T_i = \mathbb{I}[A_i \ge 480].$$

Because an exogenous age cutoff determines the treatment, differences in unemployment duration just above and just below the threshold can be interpreted as a causal effect of the policy rule. Accordingly, the treatment effect is defined as a local difference in means, directly comparable between the empirical data and the model-generated sequences

$$\widehat{\mathrm{ATE}}(h) = \mathbb{E}[Y_i \mid T_i{=}1,\, i \in \mathcal{I}(h)] - \mathbb{E}[Y_i \mid T_i{=}0,\, i \in \mathcal{I}(h)],$$

where $Y_i$ is the observed duration of the first unemployment spell. This corresponds to a simple difference in means above vs. below the cutoff inside the window, with bootstrap confidence intervals obtained by resampling individuals from $\mathcal{I}(h)$, and the signed discrepancy dependent on the window

$$\Delta(h) = \widehat{\mathrm{ATE}}_{\mathrm{model}}(h) - \widehat{\mathrm{ATE}}_{\mathrm{emp}}(h).$$

### D.2 Child Penalty

The second exercise evaluates the model's capacity to reproduce the income trajectory experienced by women following childbirth, a well-established finding in the causal literature on gender inequality in the labour market (Kleven et al., 2019; Casarico and Lattanzio, 2023).

From the test set, we selected a sample of women who had only held standard employment contracts throughout their recorded work histories. We identify mothers as those who experienced at least one maternity leave spell, defined by a maternity leave of at least two months in a given year. To ensure sample homogeneity, we exclude women who were ever classified as managers or professional athletes, as these careers may follow atypical trajectories. We retain only those with less than 1% missing data across key variables (ATECO sector, work province, income, and job title). Finally, we restrict to women whose first childbirth occurs between the 5th and 20th observed year, ensuring sufficient pre- and post-treatment periods, and only mothers working in the southern regions of Italy. The resulting evaluation set contains 631 mothers.

For each mother, we truncate the observed sequence at maternity leave ($t = 0$) and generate forward from that point. In addition, we construct offset-1 variants where the sequence is truncated one year before the maternity leave ($t = -1$). In these offset-1 runs, the model is not forced to generate a maternity event in the sequence, providing a natural basis for counterfactual comparison.

Unlike pension and unemployment, where treatment is determined by exogenous or quasi-exogenous rules, fertility is a choice-driven event. To address this treatment endogeneity, we adopt an event study design that compares sharp changes in labour market outcomes around the birth of the first child for mothers (M) relative to non-mothers (NM). The timing of childbirth creates a discrete shock to mothers' careers, while other determinants of labour market outcomes would be expected to evolve smoothly over time. This approach also allows us to trace the full dynamic trajectory of the effects with precision, exploiting individual-level variation in the timing of first births (Kleven et al., 2019).

For individual $i$ in year $s$ with event time $t$ (measured in years since first maternity leave) and treatment group $T \in \{\mathrm{M}, \mathrm{NM}\}$, we run the following regression separately for mothers and non-mothers:

$$Y_{ist}^{T} = \sum_{j \neq -1} \alpha_j^T \mathbf{1}[t = j] \;+\; \sum_k \beta_k^T \mathbf{1}[\mathrm{age} = k] \;+\; \sum_y \gamma_y^T \mathbf{1}[\mathrm{year} = y] \;+\; \nu_{ist}^T,$$

omitting $t = -1$ as the reference period. This yields dynamic effects $\hat{\alpha}_t^T$ relative to the year before birth, while the flexible age and calendar-year fixed effects control for lifecycle and macroeconomic trends.

To compute percentage penalties, we normalise with the counterfactual income path absent childbirth, defined by the predicted values from the age and year components,

$$\widetilde{Y}_{ist}^{T} = \sum_k \hat{\beta}_k^T \mathbf{1}[\mathrm{age} = k] + \sum_y \hat{\gamma}_y \mathbf{1}[\mathrm{year} = y], \qquad \widehat{P}_t^T = \frac{\hat{\alpha}_t^T}{\mathbb{E}[\widetilde{Y}_{ist}^T \mid t]}.$$

We construct two types of controls, i.e., non-mothers (NM), against which the post-birth path of mothers is compared:

1. *Matched non-mothers.* Using propensity scores based on pre-birth characteristics that correlated both with treatment assignment and outcomes (age, year of birth, area of birth, year start working, first sector), we match each mother to a non-mother on common support. The matched control is aligned to the mother's event time so that $t = 0$ corresponds to the mother's first birth year.

2. *Simulated no-maternity trajectories.* For each true mother, we use the offset-1 simulations where the model does not generate any maternity leave episode. We then average them per mother-year to form a synthetic counterfactual trajectory aligned to the event time of the true.

At event time $t$, we take the event-study outputs for mothers (treated, T = M) and the chosen control group (non-mothers or simulated no-maternity paths, T = NM) and define the percentage child penalty as the difference in the estimated event coefficients

$$P_t \;=\; \widehat{P}_t^{(\mathrm{T=M})} \;-\; \widehat{P}_t^{(\mathrm{T=NM})}, \tag{2}$$

where each $\widehat{P}_t^{(\cdot)}$ is already reported in percentage units by the event-time regression. We summarise the dynamic path by averaging over an event-time window

$$P_{a\text{–}b} \;=\; \frac{1}{b-a+1}\sum_{t=a}^{b} P_t. \tag{3}$$

For each pairing (empirical matching and model-based control) and each event time $t$ used in the window, we observe the estimated percentage effects and their standard errors for both arms:

$$\Big(\widehat{P}_t^{(\mathrm{T=M})},\, \widehat{\sigma}_t^{(\mathrm{T=M})}\Big) \quad \text{and} \quad \Big(\widehat{P}_t^{(\mathrm{T=NM})},\, \widehat{\sigma}_t^{(\mathrm{T=NM})}\Big).$$

We then implement a parametric (Gaussian) bootstrap:

1. For every $t \in \{a, \ldots, b\}$, draw independently

$$\widetilde{P}_t^{(\mathrm{T=M})} \sim \mathcal{N}\big(\widehat{P}_t^{(\mathrm{T=M})},\, (\widehat{\sigma}_t^{(\mathrm{T=M})})^2\big), \qquad \widetilde{P}_t^{(\mathrm{T=NM})} \sim \mathcal{N}\big(\widehat{P}_t^{(\mathrm{T=NM})},\, (\widehat{\sigma}_t^{(\mathrm{T=NM})})^2\big). \tag{4}$$

2. Form the bootstrap penalty path and its window average:

$$\widetilde{P}_t \;=\; \widetilde{P}_t^{(\mathrm{T=M})} \;-\; \widetilde{P}_t^{(\mathrm{T=NM})}, \qquad \widetilde{P}_{a\text{–}b} \;=\; \frac{1}{b-a+1}\sum_{t=a}^{b} \widetilde{P}_t. \tag{5}$$

3. Repeat the previous steps $B$ times. The 95% confidence interval for $P_{a\text{–}b}$ is the percentile interval over these runs.

Let $P_{a\text{–}b}^{\mathrm{model}}$ and $P_{a\text{–}b}^{\mathrm{emp}}$ denote the window-averaged penalties computed from the model-based control and from the empirical matched control, respectively. We report the signed difference and its absolute value:

$$\Delta_{a\text{–}b} \;=\; P_{a\text{–}b}^{\mathrm{model}} - P_{a\text{–}b}^{\mathrm{emp}}. \tag{6}$$

A confidence interval for $\Delta_{a\text{–}b}$ is obtained from *paired* parametric draws: on each bootstrap iteration $b$, we simulate $\widetilde{P}_{a\text{–}b}^{\mathrm{model}}$ and $\widetilde{P}_{a\text{–}b}^{\mathrm{emp}}$ from their respective Gaussians (using the reported standard errors at each $t$) and record $\widetilde{\Delta}_{a\text{–}b} = \widetilde{P}_{a\text{–}b}^{\mathrm{model}} - \widetilde{P}_{a\text{–}b}^{\mathrm{emp}}$. The percentile interval over these runs is reported as the 95% confidence interval.

### D.3 Retirement Timing

As a final causal benchmarking exercise, we evaluate the model's ability to replicate the well-documented finding that the month of birth influences retirement timing (Ardito et al., 2020). Specifically, individuals born earlier in the calendar year tend to retire slightly later, due to institutional and lifecycle timing mechanisms.

We select from the test set all male individuals born between 1940 and 1950, restricting the sample to those born in either January or December. To ensure a clean employment trajectory, we include only those who

have been continuously and exclusively employed in standard employment, excluding individuals who were ever self-employed, para-subordinate, unemployed, or employed under special regimes (e.g., artists, athletes, or voucher-based contracts). We further restrict the sample to those whose first pension spell occurs between ages 55 and 68, and who exhibit no long gaps in their administrative records. To reduce noise from incomplete information, we also exclude individuals with missing values in key attributes such as work province, economic sector (ATECO), income, and job title. After applying these criteria, the final evaluation set comprises 1,010 individuals.

For each of these individuals, we truncate their observed sequence prior to retirement and let the model generate forward until 2015. We consider two cutoff horizons: one year before the observed retirement (1y) and four years before retirement (4y), with $K = 8$ simulations per individual. The longer horizon increases predictive uncertainty but avoids anchoring the model too closely to the retirement event.

The empirical average treatment effect is the difference in mean retirement age (in months) between January- and December-born individuals in the evaluation set:

$$\widehat{\mathrm{ATE}} \;=\; \mathbb{E}[Y_i \mid T_i{=}1] \;-\; \mathbb{E}[Y_i \mid T_i{=}0],$$

where $T_i = 1$ if born in January and $T_i = 0$ if born in December. In the pension application, treatment assignment is effectively exogenous, since the month of birth is orthogonal to individual career choices and labour market conditions. As a result, the treatment effect can be interpreted as a clean difference in means between the two groups of retirees.

### D.4 Out-of-sample Generation

We randomly selected 1,500 individuals from the test set, specifically drawn from those used in the three causal inference benchmarking exercises. For each individual, we generated their life trajectory for 20 years beyond the last year observed in the training data (2015).

The purpose of this experiment is to evaluate the structural coherence of out-of-sample generations over an extended horizon. For each generated sequence, we identify the first year in which the model produces a structural inconsistency. Consistency means that each calendar year must begin/end with an End-of-Year `<EOY>` token, followed by a valid combination of monthly tokens and corresponding durations. These can be empty, i.e., no event, or include a valid life event, but must respect the order: month, then optional content, then duration. For each year, there must be at least one event, even if empty (so no repeated `<EOY>` tokens). A duration token must be followed either by another month token or an `<EOY>` token indicating the start of the next year, and `<EOY>` tokens must always be followed by a month token.

Once a structural failure is detected in a given sequence, we record the year of failure and disregard any subsequent tokens, even if the sequence appears to recover. We then plot the density of the failure years to analyse how long the model maintains structural validity over time.

## E Additional results

Figure 9 compares the real and simulated average duration of mobility allowance receipt as a function of age at displacement. Taken together, these results suggest that while the model does not reproduce the sharp discontinuity in unemployment benefit duration at the exact policy cutoff, it does capture the broader age gradient and approximates the average magnitude of the causal effect when evaluated in wider horizons around the treatment. This underscores both a limitation, i.e., the difficulty of learning from rare, rule-driven discontinuities, and a strength, i.e., when sufficient natural variation is present, the model's simulated treatment effects align closely with their empirical counterparts.

Figure 10 provides a first look at how mothers' average earnings trajectories compare to two types of counterfactual controls: model-generated synthetic paths without maternity events (left) and propensity-score–matched non-mothers (right). Both approaches highlight the earnings divergence that emerges around childbirth, but they differ in persistence: the simulated control yields a child penalty that appears to persist over time, whereas with propensity-score matching, the point estimates attenuate toward zero by year 10 after birth. Notably, in the two years before maternity, we already observe a slight increase followed by a decrease in earnings, even before

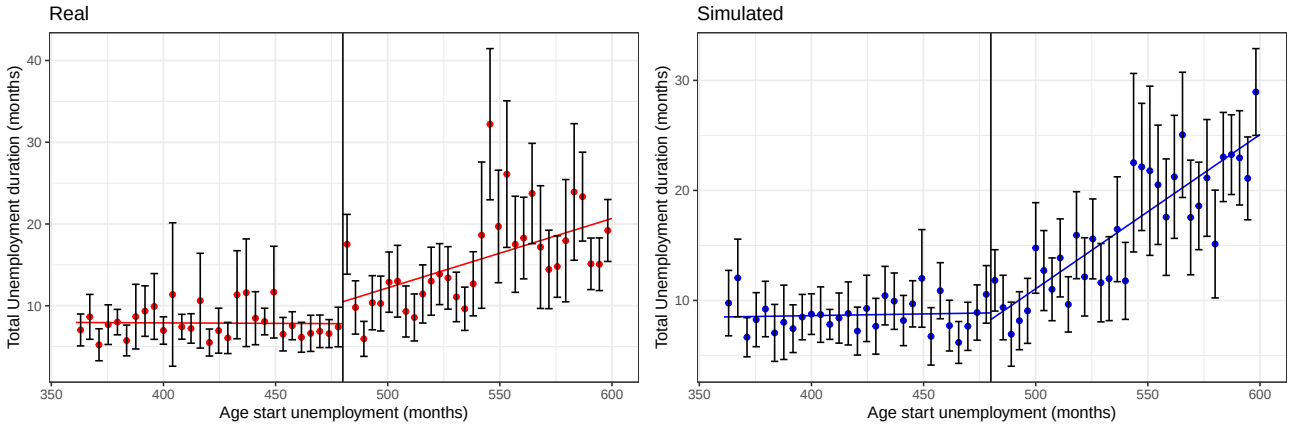



Figure 9: Real and model simulated average duration with 90% confidence interval of mobility allowance duration as a function of worker age at job loss in months in a 10-year bandwidth around the 40-year-old threshold.

childbirth occurs, highlighting how difficult it is to pinpoint when treatment truly begins and what baseline trajectory should serve as the counterfactual. This contrast raises the broader question of what constitutes a good counterfactual group and appropriate conceptual questions: at what point does a woman *become* a mother for the purpose of defining counterfactuals, and when are the relevant choices actually made, surely not at the precise year the first child arrives?

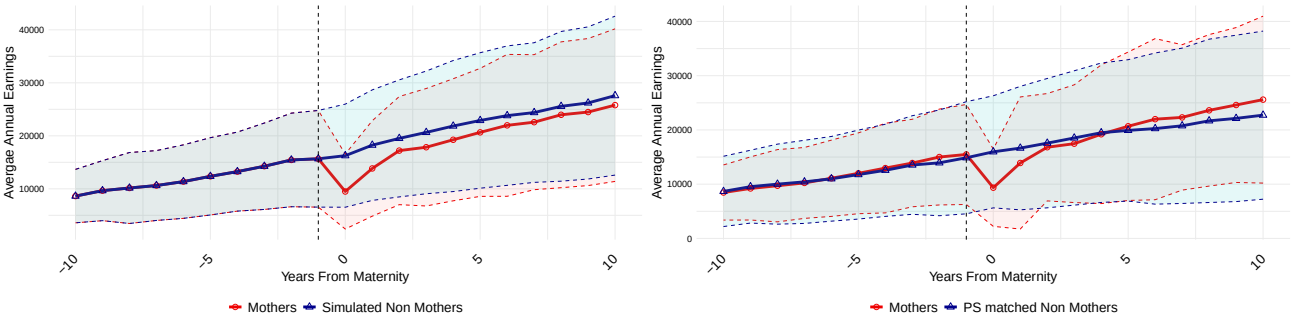



Figure 10: Average annual earnings of mothers (circle dots) compared to two types of controls (triangle dots). The left panel uses model-generated controls based on offset–1 simulations without maternity events, and the right panel uses propensity-score–matched non-mothers. Shaded areas denote one standard deviation.

Figure 11 then summarises the event time coefficients estimated as a percentage of the counterfactual outcome absence childbirth, i.e., $\widehat{P}_t^T$, relative to the year before the maternity event ($t = -1$). The model-generated simulated controls (left panel) and propensity-score–matched non-mothers (right panel) yield broadly similar patterns, though with notable differences. In the simulated control case, the immediate short-run penalty tends to be smaller, but the estimates suggest a more persistent medium- to long-run effect. By contrast, the propensity-score comparison shows a sharper initial earnings drop and a quicker recovery. Taken together, these results show that the model reproduces empirical child penalties quite closely across horizons, while also revealing how alternative counterfactual constructions can shift the interpretation of the penalty's timing and persistence.

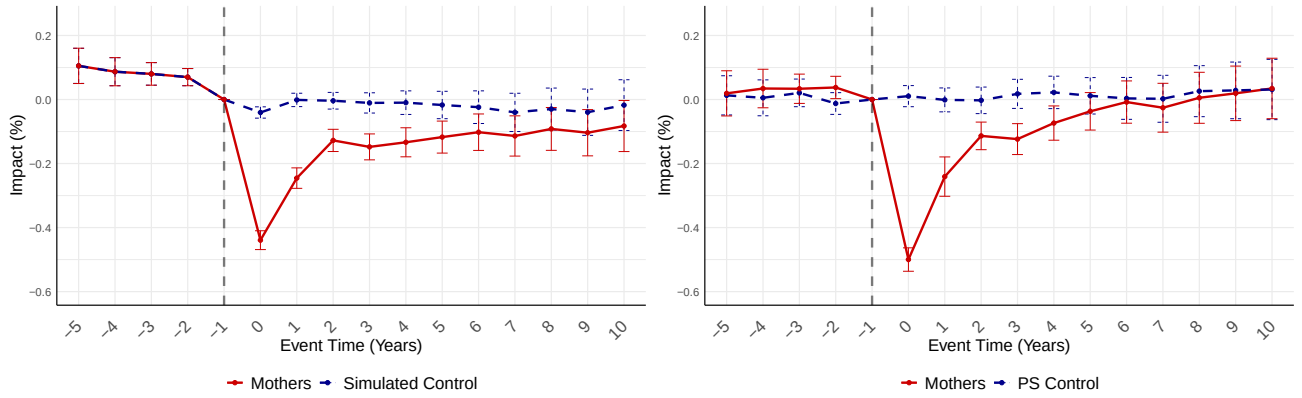



Figure 11: Event time coefficients estimated as a percentage of the counterfactual outcome $\widehat{P}_t^T$ relative the year before the maternity event ($t = -1$). The left panel uses model-generated synthetic controls based on offset–1 simulations without maternity events, while the right panel uses propensity-score–matched non-mothers as controls. The horizontal axis shows event time $t$ (years since birth), and vertical bars denote 95% confidence intervals.

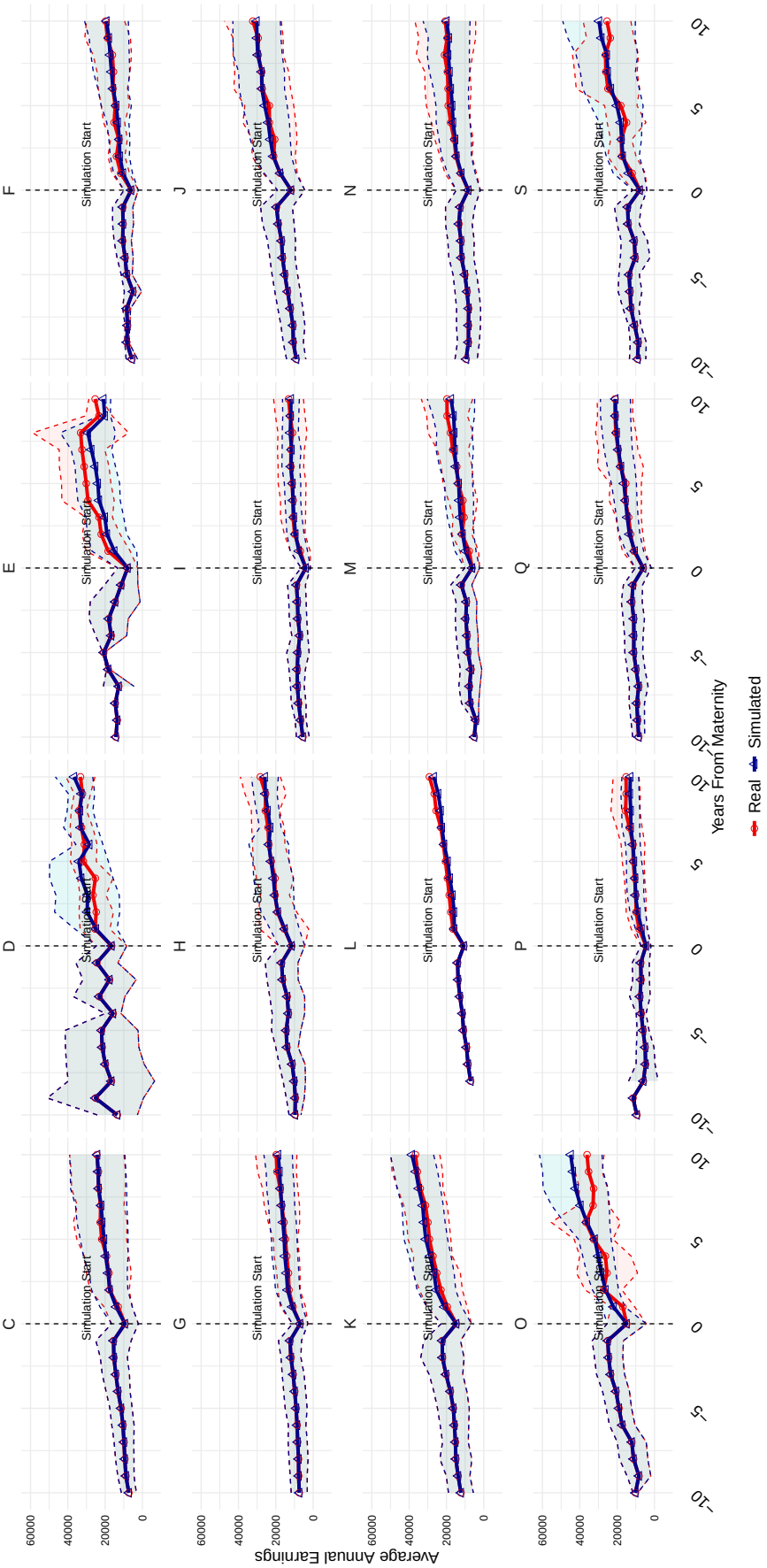



Figure 12: ATECO grouped real and model-generated trajectories of average annual earnings for working mothers in a 10-year window before and after their first maternity leave episode (Years From Maternity = 0). The solid lines represent the average monthly income, while the shaded areas denote the one standard deviation range.

Figure 13 illustrates how the estimated retirement age varies with the choice of conditioning horizon. When the model is conditioned four years before retirement (left panel), it produces a clearer discontinuity between January- and December-born cohorts, whereas conditioning only one year before retirement (right panel) yields much smaller differences. This contrast underscores how sensitive the results are to where the observed sequence is cut: cutting further away forces the model to generate more of the trajectory, which amplifies the birth-month effect, while cutting closer reduces this reliance on simulated paths. These patterns highlight not only the model's ability to reproduce well-documented empirical regularities but also the conceptual challenge of deciding how much information to condition on when constructing counterfactuals in retirement settings.

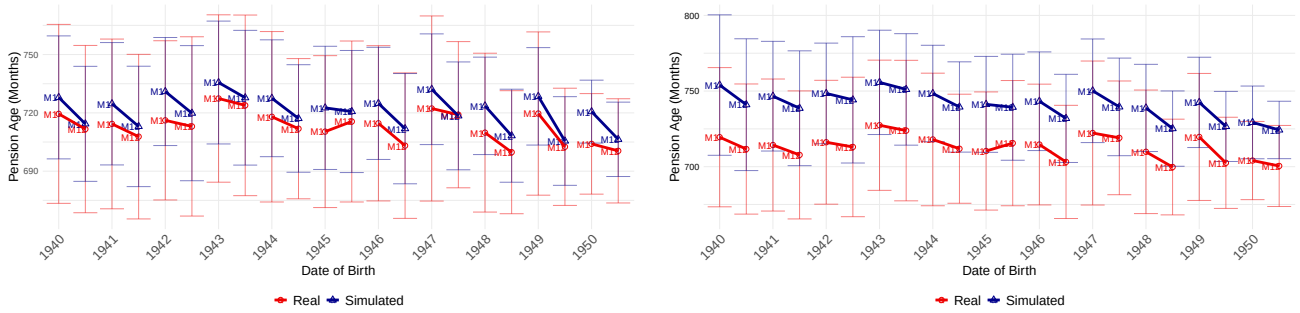



Figure 13: Real and model-generated (blue) average retirement ages, measured in months, for male individuals born between 1940 and 1950. The left panel shows results using a cutoff four years before the observed retirement year, while the right panel shows results using a cutoff one year before. Each point indicates the cohort-specific average retirement age for January (M1) and December (M12), with vertical bars representing one standard deviation.

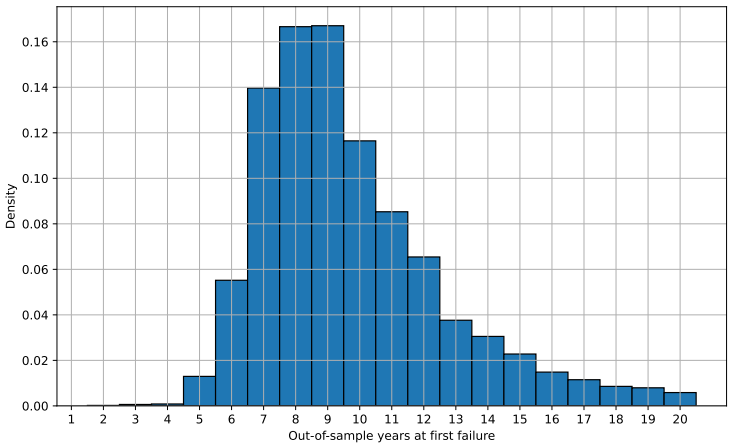



Figure 14: Density plot for the out-of-sample year when the model generates incorrect sequences, 5% of the observations were able to complete the 20 years of generation without failures.

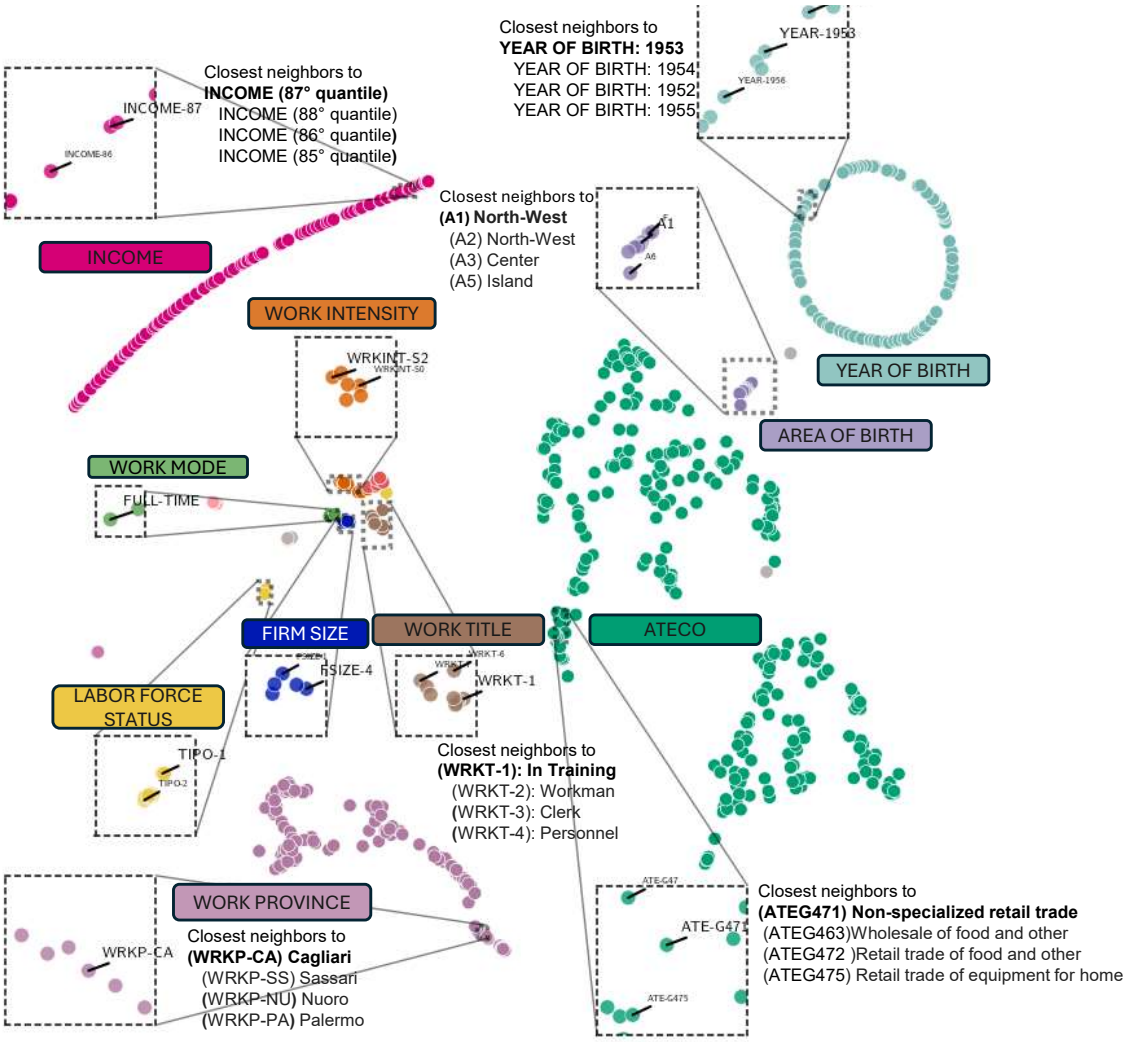



Figure 15: Two-dimensional projection of the learned embedding space from the model, visualised using the PaCMAP dimensionality reduction technique (Wang et al., 2021). Each point represents a token from the vocabulary ($|\mathcal{V}| = 661$), with different colours denoting different token categories. Several regions of interest are highlighted with zoomed-in views, each showing the top three nearest neighbours of a specific token. Neighbour relationships are computed using cosine distance in the original high-dimensional embedding space ($d = 240$).