Submitted to *Quantitative Economics*

# LABOR-LLM: Language-Based Occupational Representations with Large Language Models

SUSAN ATHEY
Graduate School of Business, Stanford University

HERMAN BRUNBORG
Institute for Computational and Mathematical Engineering, Stanford University

TIANYU DU
Institute for Computational and Mathematical Engineering, Stanford University

AYUSH KANODIA
MoveUp AI

KEYON VAFA
Harvard Data Science Initiative, Harvard University

Vafa et al. (2024) introduced a transformer-based econometric model, CAREER, that predicts a worker's next job as a function of career history (an "occupation model"). CAREER was initially esti-

---

Susan Athey: athey@stanford.edu
Herman Brunborg: brunborg@stanford.edu
Tianyu Du: tianyudu@stanford.edu
Ayush Kanodia: kanodiaayush@gmail.com
Keyon Vafa: kvafa@g.harvard.edu
We gratefully acknowledge financial support from the Golub Capital Social Impact Lab, the Stanford Institute for Human-Centered Artificial Intelligence, and the Business, Governance, and Society Initiative at Stanford's Graduate School of Business. Herman Brunborg thanks Aker Scholarship Foundation for financial support. Author order is alphabetical. Tianyu Du was the student leading the project and would be considered first author in the engineering tradition.

mated ("pre-trained") using a large, unrepresentative resume dataset, which served as a "foundation model," and parameter estimation was continued ("fine-tuned") using data from a representative survey. CAREER had better predictive performance than benchmarks. This paper considers an alternative where the resume-based foundation model is replaced by a large language model (LLM). We convert tabular data from the survey into text files that resemble resumes and fine-tune the LLMs using these text files with the objective to predict the next token (word). The resulting fine-tuned LLM is used as an input to an occupation model. Its predictive performance surpasses all prior models. We demonstrate the value of fine-tuning and further show that by adding more career data from a different population, fine-tuning smaller LLMs surpasses the performance of fine-tuning larger models.

KEYWORDS. Occupation Transitions, Large Language Model, Foundation Model.

JEL CLASSIFICATION. J24, J62, C55.

## 1. INTRODUCTION

This paper introduces a new approach to making predictions about the evolution of worker careers that builds on the "foundation model" approach recently popularized in generative artificial intelligence. The application we focus on is the problem of predicting a worker's next job as a function of the worker's prior history. This problem is challenging because of the high dimensionality of the feature space: When there are 335 possible occupations, there are $335^t$ possible sequences of occupations in $t$ periods of observation. In addition, the prediction space is large. Given a history of jobs, a predictive model produces 335 probabilities corresponding to the possible next jobs.

Historically, the economics literature has addressed these challenges in a few ways. In terms of simplifying the outcomes, the literature has typically collapsed the space of occupations into a much smaller number of high level categories (Boskin (1974)), or it has taken a "hedonic" approach, describing jobs by their characteristics, such as skills requirements (e.g., Cortes (2016)).[1] In terms of reducing the dimensionality of the covariates, economic models typically use heuristic approaches such as focusing on the most recent previous job and summary statistics that describe the rest of history, such as years of experience (e.g., Hall et al. (1972)). However, we will show in this paper that these approaches have limitations: using heuristics to reduce dimensionality limits the set of applications of the model and hurts predictive power. For example, we might wish to characterize job transitions granularly in order to identify those that have become less common over time, or transitions that are particularly likely after layoffs; an occupation model that incorporates career history may also contribute to analyses of transitions in and out of the labor force, or in and out of poverty (e.g., Stevens (1994)). Accurate predictions often play a supporting role in answering causal economic questions; predictive models are used to estimate counterfactual outcomes that would occur in the absence of treatment, and predictive models must account for covariates (here, history) that may be correlated with treatment assignment to avoid omitted variable bias. Predictive models also play a supporting role in estimating treatment effect heterogeneity (Athey et al. (2023)). In the context of recommendation systems or automated job advice (de Ruijt and Bhulai (2021)), accurate estimates of conditional transition probabilities may be a key building block.

In this paper, we develop a novel approach to this problem where dimensionality reduction of outcomes (the next job) and career history is data-driven. Our approach improves upon previous approaches in terms of predictive power in held-out data. We start from the observation that the problem of predicting the next job in a worker's career is analogous to the problem of predicting the next word in a sequence of text, suggesting that approaches that have recently been highly successful for predicting the next word may also be applicable here. Previous research (Vafa et al. (2024)) took language modeling as an inspiration and built a custom model for occupation prediction; in this paper, we introduce an

[1]The hedonic approach has also been used in related literature in industrial organization where consumers select among many products.

approach that directly uses the next-word probability models associated with popular open source Large Language Models (LLMs).

To understand how we use LLMs for the discrete choice problem of predicting job transitions, consider how LLMs are commonly developed and used today. The empirical model (most commonly, a transformer neural network) reduces the dimensionality of covariates through the use of "embeddings" or "representations" which are lower-dimensional latent variables estimated from data. In the case of text, an embedding function is an (estimated) mapping from a sequence of words into a real-valued vector. Estimation of the model makes use of variants of stochastic gradient descent, where each observation (instance of a next-word prediction) is ordered randomly and then observations are processed sequentially. The parameters of the model are updated in the direction of the gradient of the objective function evaluated at the relevant observation. Stochastic gradient descent is applied to two distinct datasets in sequence. The first dataset is usually very large and may not be representative of the population of interest, and estimation of model parameters on this dataset is referred to as "pre-training," while the resulting estimated model is referred to as a "foundation model" (Bommasani et al. (2022)). For some applications, the foundation model is used "off-the-shelf" and estimation ends at this step, but in other applications a second dataset is used. The second dataset is usually a randomly selected "training" subsample of the dataset of primary interest, and it is usually much smaller than the first dataset. Estimation of model parameters using stochastic gradient descent picks up where the pre-training left off, processing only observations from the training dataset.

Several observations about the approach of pre-training and fine-tuning shed light on why it can be effective. First, the pre-training step may identify structure in the prediction problem (in the case of language, the meaning of words, grammar, and facts) that may be relevant across different contexts. With a very large pre-training corpus, it is possible to estimate a large number of parameters (generally billions or more), enabling a substantial amount of information to be encoded in the model. Second, it is not necessary to have access to the pre-training dataset in order to carry out the fine-tuning step. All that is needed is access to the model parameters and an understanding of the functional form of the embedding function. A third advantage that we will not fully exploit in this paper is that the objective can be modified (e.g., predict a different outcome variable) in fine-tuning. See, e.g., Bommasani et al. (2022) for further discussion.

An open question about the fine-tuning approach is whether the fact that the pre-training dataset is not representative of the target implies that the final estimated model will exhibit bias relative to the true conditional transition probabilities in the population of interest. There may be a tradeoff between using a large, non-representative dataset to better learn underlying structure (e.g. meaning of language), and getting a model that makes conditional predictions that are representative of a target dataset of interest. In this paper, we show that if such biases are important, the advantages of the foundation model approach outweigh them in our application.

The foundation model approach has been applied in many settings beyond text (Savcisens et al. (2024), Wu et al. (2021), Radford et al. (2021)). For the problem of next-job prediction, Vafa et al. (2024) built CAREER. CAREER relies on a "custom" econometric model based on the same transformer architecture popular in LLMs, but modified so that the vocabulary of the transformer is limited to the space of jobs, and customized to give special treatment to staying in a job. The pre-training data was a set of about 23 million resumes of U.S. workers acquired from Zippia, Inc., where the resumes are not representative of the U.S. population. Vafa et al. (2024) then fine-tuned the model using data from U.S. government surveys (the Panel Study of Income Dynamics (PSID) (Survey Research Center, Institute for Social Research, University of Michigan (2024)) and two cohorts from the National Longitudinal Survey of Youth (NLSY79 and NLSY97) (Bureau of Labor Statistics, U.S. Department of Labor (2023, 2024)), showing that predictive performance was significantly better than existing benchmarks from the literature. Further, the paper shows that the underlying structure identified by the foundation model has predictive power for related tasks; when the model is fine-tuned to predict wages, which are not available in the pre-training resume dataset, it improves the predictive power for wages above popular regression models relied upon in labor economics. CAREER used an embedding space of 768 dimensions, and the model had about 5.6 million parameters.

In this paper, we propose an alternative to CAREER, which we refer to as the **LA**nguage-**B**ased **O**ccupational **R**epresentations with **L**arge **L**anguage **M**odels (LABOR-LLM) framework. This framework incorporates several approaches to leveraging LLMs for modeling labor market data and producing representative predictions. LABOR-LLM uses a similar approach to CAREER with several modifications. Most importantly, the foundation model we use is an LLM, so it is trained on natural language. We focus on Llama-2, the open-weight model provided by Meta. Second, in our preferred LABOR-LLM approach, which we call Fine-Tuned LABOR-LLM or FT-LABOR-LLM, instead of fine-tuning the model on tabular data as constructed from government surveys, we fine-tune it on a textual version of the government survey (or combinations of government surveys). In particular, we transform the survey data into what we call a "text template" that looks similar to the text of a resume, and fine-tune the language model on a dataset consisting of one document (sequence of words resembling a resume) for each worker in a government survey dataset. The objective of the fine-tuning is next-word prediction for the text resume.

The fine-tuned model can, in principle, be used in a variety of ways. One approach would be to use it to create data-driven low-dimensional embeddings of history, and use those embeddings as if they were observed covariates in a predictive model such as a multinomial logistic regression. We explore such an approach in the paper, but we show that it does not work as well as FT-LABOR-LLM.

The FT-LABOR-LLM approach involves adapting an LLM that generates an estimate of the probability of the next word (conditional on that word being preceded by a particular sequence of words) to an occupation model that predicts the job in a particular year as a function of career history. To do so, we use the probability model associated with the fine-tuned LLM to evaluate the probability that the next text in our text template is the text corresponding to a particular job, conditional on the preceding text being equal to the text of the text template truncated at the year of interest, recalling that the text template was automatically generated from the worker's history recorded in the tabular survey data.

We show that the performance of FT-LABOR-LLM is better than that of CAREER, despite CAREER being custom-designed for the problem and pre-trained on a very relevant corpus of documents, resumes of U.S. workers. Recalling that CAREER in turn substantially outperformed alternatives from the literature, FT-LABOR-LLM is established to be the state of the art in terms of predictive performance. We highlight the importance of the fine-tuning step by showing that, without fine-tuning, off-the-shelf Llama-2 makes plausible-sounding predictions of jobs, but it is not as accurate in terms of the next job probability distributions conditional on history, and it "hallucinates" invalid job titles because it is not fine-tuned exclusively on labor sequence data. The latest LLM available from OpenAI has similar challenges.

In the remainder of the paper, we assess the sources of the performance benefits. We begin by assessing the role of model size (number of parameters) and the volume of data. We show that using a larger LLM as the foundation model, in particular the version of Llama-2 with 13 billion parameters rather than the version with 7 billion parameters, improves predictive performance. However, we show that adding in data from different government surveys (even though they are drawn from different time periods) quickly improves the performance of the smaller model, matching and then surpassing the performance of the larger model. Thus, data is a substitute for model size.[2] Since smaller models are less expensive to estimate, and especially cheaper to make predictions from, working with a smaller model has distinct advantages.

We next assess whether FT-LABOR-LLM is making use of information embedded in the text of the job title. To do so, we replace the job titles with numeric codes in the training data and show that this approach degrades predictive performance substantially. We further establish that demographics, most notably gender, but also the interaction of gender, ethnicity, and region, play an important role in predicting job transitions. Finally, we show that predictive perfor-

[2]Other papers have shown that more data improves model performance for both pre-training (Vafa et al. (2024), Kaplan et al. (2020)) and fine-tuning (Dong et al. (2023), Bucher and Martini (2024)) data.

mance is degraded unless at least 10 periods of worker history is included; truncating the history degrades performance.

Overall, the success of FT-LABOR-LLM provides an example of how LLMs can be used as foundation models for an economic problem that was traditionally studied using categorical, discrete-choice prediction models. In addition to providing superior predictive performance, the LABOR-LLM approach has some advantages because the pre-training step does not have to be carried out by the individual researcher; rather open, general purpose LLMs can be used (or closed models can be used through paid API access, although with less control on the part of the analyst).

## 2. Related Work

*Career Trajectory Modeling and Next Job Prediction* In the economics literature, when studies of worker transitions analyze the relationship between worker characteristics and career histories to career transitions, they have traditionally relied on fairly simple predictive models and considered only a few occupation categories. For example, Boskin (1974) use a conditional logistic regression model to analyze the factors affecting workers' transitions among 11 occupational groups, where the factors included estimated earnings, training expenses, and costs due to unemployment. Schmidt and Strauss (1975) use a multinomial logistic regression to analyze the impact of race, sex, educational attainment, and labor market experience on the probability that individuals transition into one of five different occupational categories, revealing significant effects of these variables on occupational outcomes. Hall et al. (1972) examine the dynamics of labor force turnover in the U.S., analyzing the influences of demographic factors, labor demand fluctuations, and job stability on unemployment. To study turnover, the authors consider factors such as race, counts and ages of children, estimated wage, income, age, marital status, location, and employment category (including private wage or salary, government roles, self-employment, and unpaid family work). Blau and Riphahn (1999) model labor force transitions among older married couples, showing that one spouse's employment status significantly impacts the employment status of the other, with financial incentives and preferences for shared leisure influencing these transitions. In addition to demographic characteristics, the authors incorporate human capital and education variables, including the tenure on the current job and retirement benefits in their models.

*Machine Learning Methods for Next Job Prediction* For the problem of predicting worker job transitions, our paper is the first to use LLMs as a foundation model. As discussed in the introduction, the most closely related paper to ours is Vafa et al. (2024), which builds CAREER, a custom foundation model that is a modified version of the transformer models used in language models, and restricts attention to predicting numerically encoded jobs. CAREER has fewer parameters than FT-LABOR-LLM, and the pre-training dataset, while highly relevant, is much smaller than the corpus used for Llama-2. CAREER does not make use of the textual descriptions of job titles.

Prior to CAREER, other authors (e.g., Li et al. (2017), Meng et al. (2019), Zhang et al. (2021)) made use of various versions of neural networks for the next job prediction problem, sometimes training on large datasets. For example, Li et al. (2017) use a Long Short-Term Memory (LSTM) neural network to predict job transitions, where the embedding dimension is 200, and the training set incorporates more than a million individuals. He et al. (2021) build a model to predict the next job position out of 32 frequent position names, as well as job salary and firm size for that position, using a dataset of 70,000 resumes. These papers do not make use of foundation models.

Another approach taken by Zhang et al. (2019) seeks to predict aggregate transition probabilities between pairs of job titles within the same firm. Their approach, which generates embeddings for each job title, does not attempt to condition on individual worker history.

*Adapting LLMs to Build Domain-Specific Models* Adapting pre-trained models to specific domains via fine-tuning has become a prevalent approach for improving the performance of LLMs for specific tasks. The (full parameter) fine-tuning approach involves further updating all weights of a pre-trained model using domain-specific data and optimization techniques such as gradient descent (Wei et al. (2022)). The pre-training and fine-tuning paradigm has produced state-of-the-art models for dialogue systems (Yi et al. (2024)), code generation (Chen et al. (2021)), music generation (Agostinelli et al. (2023)), scientific knowledge (Taylor et al. (2022)), protein structure prediction (Rives et al. (2021)), chemistry (Zhang et al. (2024)), medicine (Singhal et al. (2022)), and other settings. The literature on the adaptation of LLMs for recommendation systems is also closely related. Geng et al. (2022) introduce a general paradigm to adapt the recommendation task to language processing.

Our paper compares our fine-tuning approach to one where LLM embeddings are extracted and treated as covariates in a multinomial logistic regression. This type of approach has been popular in language analysis for a long time; for example, it is used by sentiment classifiers (Reimers and Gurevych (2019)).

Finally, prompt engineering and in-context learning are alternative approaches to fine-tuning LLMs that require minimal computation and avoid the need for direct model access (Brown et al. (2020)). Prompt engineering involves designing specific queries, instructions, or examples within the prompt to direct the model's response. By tuning the language and structure of prompts, researchers can shape the model's output for different applications (Maharjan et al. (2024)). Researchers can also use in-context learning by providing relevant example data within the prompt itself, priming the model to continue the pattern and apply similar logic to new inputs (Yin et al. (2024), Bao et al. (2023)). In this paper, we consider an approach in which we prompt off-the-shelf LLMs for a prediction of the next job using a textual representation of worker career history as the prompt. We show that including example resumes in the prompt helps improve performance of off-the-shelf pre-trained LLMs, although performance is still worse than FT-LABOR-LLM.

*Other Applications of LLMs to Sequential Prediction Problems in Economics*
LLMs have also been used to model time series data (Jin et al. (2024)) and in forecasting. For instance, Faria-e Castro and Leibovici (2024) investigate the ability of LLMs to produce in-sample conditional inflation forecasts during the 2019–2023 period.

*The Biases and Representativeness of LLMs* A recent literature has emerged that aims to assess whether the outputs of foundation models are representative of larger populations, for example, whether the answers to opinion survey questions are representative of the population (Santurkar et al. (2023), Argyle et al. (2023)). One proposed approach is to query LLMs with survey responses from long-standing opinion surveys and see how aligned their responses are with the survey average. Our question differs in that we want to know whether the (fine-tuned) LLM can make predictions about job transitions that are representative of real-world transitions, conditional on history, which is a more complicated question to answer, as the population conditional probabilities are unknown due to the high dimensional space of potential histories.

## 3. Occupation Models

### 3.1 *Notation for Occupation Models.*

We refer to a model that predicts an individual's next occupation as a function of career history and other individual characteristics as an **occupation model**. Our paper focuses on a specific type of occupation model, which predicts the occupation in the next time period conditional on the previous occupations and covariates.

In this section, we develop notation for occupation models. Let $t \in \{1, .., T_i\}$ correspond to a year in which an individual $i$ was surveyed and otherwise met our filtering requirements, where $T_i$ denotes the total number of individual-year observations of this individual. Note that our cleaned survey datasets do not, in general, have observations in every calendar year. We refer to an observation of an individual's occupation as a "transition," with some abuse of terminology since we use the term for the first observation and also even when the individual stays in the same occupation. Let $\text{year}_{i,t}$ denote the calendar year corresponding to transition $t$ for individual $i$. We represent occupations as discrete variables, in particular following the `occ1990dd`, a variant of the OCC occupational classification system of Autor and Dorn (2013), as described in Appendix N. Let $\mathcal{Y}$ denote the set of all occupations, and let $y_{i,t} \in \mathcal{Y}$ represent the occupation that an individual $i$ has at transition index $t$. We let $y_{i,<t} = (y_{i,1}, \ldots, y_{i,t-1})$ denote an individual's job sequence prior to their $t$'th observation (for $t \leq 1$, define $y_{i,<t} = \emptyset$). Let $\mathcal{X}_{\text{inv}}$ be the support of time-invariant covariates (in our application, race, ethnicity, region, and sometimes birth year, denoted by $x_i$), while $\mathcal{X}_{\text{var}}$ is the support of time-varying covariates (in our application, education and calendar year, denoted by $x_{i,t}$). Let $x_{i,\leq t} = (x_i, x_{i,1}, \ldots, x_{i,t}) \in \mathcal{X}_{\text{inv}} \times \mathcal{X}_{\text{var}}^t$ denote the time-invariant covariates and time-varying covariates up to and including $t$. We refer to $(x_{i,\leq t}, y_{i,<t},)$ as the worker's career history at transition $t$.

The probability that the worker's next job is $y_{i,t}$, conditional on the worker's history, is written $P(y_{i,t} \mid x_{i,\leq t}, y_{i,<t})$.

### 3.2 *Assessing Predictive Performance of Occupation Models*

We evaluate an occupation model's performance by comparing its predictions of an individual's next job to their actual next job. Specifically, we evaluate models by computing their perplexity, a commonly used metric in Natural Language Processing (NLP). The perplexity is a negative monotonic transformation of the sample log-likelihood, with lower perplexity indicating that a model's predictions are more accurate. Formally, the perplexity of an occupation model $\hat{P}$ on a set of transitions (individual-year observations) for units $i = 1, .., I$ is given by

$$\text{perplexity} = \exp\left\{ -\frac{1}{\sum_i T_i} \sum_{i=1}^{I} \sum_{t=1}^{T_i} w_{it} \left[ \log \hat{P}(y_{i,t} \mid x_{i,\leq t}, y_{i,<t}) \right] \right\},$$

where $w_{it}$ denotes the sampling weight for the individual relative to a population of interest. In this paper, for simplicity, we set $w_{it} = 1$. Note that a completely uninformative model that assigns uniform mass to each possible occupation would achieve a perplexity of $|\mathcal{Y}|$. We consider additional evaluation metrics (such as calibration) in Section 10.

### 3.3 *Quantifying Uncertainty in Performance Metrics*

When comparing the performance of alternative occupation models, we wish to quantify the uncertainty about estimates of performance. The randomness in measured perplexity for a given model arises from several sources: sampling variation in the training data, randomness in the fine-tuning pipeline (e.g., data shuffling for a stochastic gradient descent optimizer), and sampling variation of the test data.

To estimate the uncertainty arising from the first two sources, we bootstrap the training set used for fine-tuning (sampling at the individual level) and estimate the variation in measures of the performance of models across bootstrap samples. We refer to the resulting standard errors as "training-set-bootstrapped." To capture sampling variation of the training set, we sample with replacement.

According to the support team of Together AI, the platform we use to fine-tune LLMs, the randomness in their fine-tuning process arises mainly from randomizing the order of observations in the process of optimizing via stochastic gradient descent; each instance of re-tuning a bootstrap sample will include randomization of this type.[3] Because fine-tuning is very expensive to carry out, we conduct an experiment for three of the models, as described below in Section 8.2 and Appendix A.

To estimate the uncertainty arising from sampling variation in the test set, we bootstrap the test set and refer to the resulting standard errors as "test-set-bootstrap." We sample at the individual level with replacement and, in the analysis we report below, use 100 bootstrap replications. We employ a similar bootstrapping approach to calculate the test-set-bootstrap standard errors for the differences in perplexities between the two models. We select bootstrap samples at the individual level, compute the perplexities for both models on the bootstrap sample, and calculate the standard deviation of the difference in perplexities. See

[3]Unfortunately, at the time of this writing, there is no way to specify the random seed for reproducibility. The support team also mentioned "adapter weight initialization" as another source of randomness in the fine-tuning pipeline, which is only relevant if one is fine-tuning using the Low-Rank Adaptation (LoRA) technique. We are doing full-parameter fine-tuning instead.

Appendix A for more details on both the training- and test-set-bootstrap standard errors.

## 4. Large Language Models as Foundation Models

### 4.1 *LLM Notation*

The LLMs we use in this paper are trained to perform next-word prediction. Let $\mathcal{W}$ be the allowable set of words and punctuation, while $\cup_{j=1}^{\infty}\mathcal{W}^j$ is the set of sequences of words.

In practice LLMs work in the space of "tokens," where words can be transformed into a sequence of tokens using a process called "tokenization." We let $\mathcal{V}_{\text{LLM}}$ be the set of all possible tokens (also known as the vocabulary set) for a particular LLM. Popular commercial-scale LLMs typically use vocabulary sets with 10,000 to 100,000 tokens; for example, $|\mathcal{V}_{\text{Llama-2}}| = 32,000$. Let $\textsc{Tok}\colon \cup_{j=1}^{\infty}\mathcal{W}^j \to \cup_{j=1}^{\infty}\mathcal{V}^j$ denote the function mapping a sequence of words to a sequence of tokens.

The LLMs we consider can be viewed as estimating the probability that the next token equals $v_{k+1}$, conditional on a sequence of $k$ tokens (i.e., the prompt). LLMs also impose restrictions on the "context length," or the maximum length of a sequence that can be conditioned on, which we denote $C$, with particular values $C_{\text{LLM}}$ imposed by different LLMs. Then, we let $\hat{P}^{\mathcal{V}}_{\text{LLM}} : \mathcal{V}_{\text{LLM}} \times \cup_{k\le C_{\text{LLM}}}\mathcal{V}^k_{\text{LLM}} \to [0,1]$ denote the LLM's estimate of the probability of the next token conditional on the input sequence, which for particular values of $v_1,\ldots,v_{k+1}$ is written $\hat{P}^{\mathcal{V}}_{\text{LLM}}(v_{k+1} \mid v_1,\ldots,v_k)$.

The conditional probability $\hat{P}^{\mathcal{V}}_{\text{LLM}}(v_{k+1},\ldots,v_{k+k'} \mid v_1,\ldots,v_k)$ for $k+k' \le C_{\text{LLM}}+1$ can be derived from individual next-token predictions using the chain rule.

### 4.2 *Text Templates*

In this section, we describe the **text template** function we use to convert a worker's history of jobs and covariates into a sequence of words and punctuation. This text template can be combined with a tokenizer and an LLM to make next job predictions, as described in the next section.

We let $\text{TITLE} : \mathcal{Y} \rightarrow \cup_{j=1}^{\infty} \mathcal{W}^j$ denote the mapping from an occupation to its English-language title. Note that this mapping should be bijective (one-to-one). For example, the title of the occupation with `occ1990dd` code 95 is "nurse practitioners."[4] The number of tokens needed to represent a job title depends on the tokenizer; using the Llama-2 tokenizer, the number ranges from 2 to 28 in the survey datasets we analyze, with an average length of 8.3 tokens (and a standard deviation of 4.8).[5]

To represent covariates as text, we express an individual's educational status using values such as `graduate degree`. Online Appendix A provides the full mapping between `occ1990dd` codes and their job titles.

Building on this strategy, we define a **text template** function, $\text{TMPL}(x_{i,\leq t}, y_{i,<t})$, that transforms an individual's career history into a textual summary. The text template incorporates additional punctuation, line breaks, and meta-data, as detailed in Appendix C, and illustrated in the following example.

```
<A worker from the PSID dataset>
The following information is available about the work history of a female
↪  black or african american US worker residing in the south region.
The worker was born in 1963.
The worker has the following records of work experience, one entry per
↪  line, including year, education level, and the job title:
1984 (some college): Cooks
1985 (some college): Cooks
1987 (some college): Food servers, nonrestaurant
1989 (some college): Cleaners of vehicles and equipment
<END OF DATA>
```

[4]Even though the `occ1990dd` system does not include job titles directly, one can crosswalk it to, for example, the Standard Occupational Classification (SOC) system, and use the list of job titles attached to each SOC code provided by the Bureau of Labor Statistics (https://www.bls.gov/OES/CURRENT/oes_stru.htm).

[5]"Grinding, Lapping, Polishing, and Buffing Machine Tool Setters, Operators, and Tenders, Metal and Plastic" (28 tokens) and "Cutting, punching, and press machine setters, operators, and tenders, metal and plastic" (24 tokens) are the two longest job titles. The shortest job tiles include "Cooks", "Bakers", "Tellers", and "Designers".

The example above is defined as the text representation of the **complete career history** of the individual, denoted $\textsc{Tmpl}(x_{i,\le t}, y_{i,\le T_i})$, where $T_i$ represents the number of transitions for individual $i$. These complete career histories are used for model fine-tuning, as discussed in Section 8.

Note that the individual can stay in the same job for multiple records (e.g., 1984 and 1985 in the example); the text representation explicitly reflects this information. Additionally, the dataset could miss an individual for certain years in her career trajectory; in this case, the text template will skip those years (e.g., 1987 and 1995 in the example).

We also create the text representation of the **career history** of the same individual prior to the $t^{\text{th}}$ job, denoted $\textsc{Tmpl}(x_{i,\le t}, y_{i,<t})$, by truncating the complete career history. For example, to obtain an LLM's predictions of an individual's job in 1989 given the covariates and job history, we would use as input the text above, removing the text "Cleaners of vehicles and equipment" and everything afterward (i.e., the underlined part in the example). That is, we apply the text template function to all previous job and covariate information, and conclude with a partial row for the occupation to be predicted.

On average in the survey datasets we consider, the text representation of workers' complete career histories contains around 250 to 500 tokens using the Llama-2 tokenizer, which fits well within the context window of Llama-2 models for fine-tuning. For inference tasks, the prompt encoding of an individual's career history, i.e., $\textsc{Tok}(\textsc{Tmpl}(x_{i,\le t}, y_{i,<t}))$, consists of 200 to 300 tokens on average. Detailed summary statistics on the number of tokens can be found in Online Appendix B.

### 4.3 *Using LLMs for Occupation Modeling*

In this paper, we use LLMs in three ways. First, we use an LLM to directly produce a "predicted job" in response to a "prompt." More precisely, if we first map job codes to text (the English language job title) using the text template function described in the previous subsection, and then use a tokenizer to translate the resulting sequences of past jobs into a sequence of tokens, an LLM will produce a textual "response" that is a sequence of tokens. That sequence may or may not correspond to a valid occupation, but we can, in principle, further transform the output in various ways to interpret it as an occupation. Of course, a textual response or a single predicted occupation is not an estimate of the probability of a sequence of tokens. Some commercial LLMs allow the user to set a "temperature" parameter when submitting a prompt, where a particular setting is designed to approximate sampling from the distribution of responses. Probabilities can then be estimated by repeatedly prompting the LLM. We do not follow this approach in this paper; instead, we restrict attention to LLMs where probabilities (or, where relevant, embeddings) can be directly obtained by the analyst.

Second, for those LLMs for which it is possible, we directly obtain the probability assigned to a given token. This functionality may be enabled in the setup of an open model such as Llama-2, or it may be exposed through an API in the case of a closed model such as ChatGPT-4.[6] For example, for the LLM Llama-2 7 billion parameter model, denoted Llama-2-7B, the estimated probability that "Engineer" follows the single-token sequence "Software" is written $\hat{P}^{\mathcal{V}}_{\text{Llama-2-7B}}(\text{"Engineer"}|\text{"Software"})$. To obtain the probability of the next job given a sequence of prior jobs, we first use the text template function and the tokenizer to translate the job history into a sequence of tokens; similarly, we translate the title of a particular next job $y_{i,t+1}$ into a sequence of tokens. The estimated next-token probability model associated with the LLM, denoted by $\hat{P}^{\mathcal{V}}_{\text{LLM}}(\cdot \mid v_1, \ldots, v_k) : \mathcal{V}_{\text{LLM}} \to [0,1]$, can be applied several times to determine (using the chain rule of probability) an estimate of the probability that the sequence of tokens induced by $y_{i,t+1}$ follows the sequence of tokens induced by $(y_{i,1}, \ldots, y_{i,t})$. A language-based next-token prediction model thus induces an associated occupation model, as follows:

$$
\begin{aligned}
&\hat{P}_{\text{LLM}}(y_{i,t} \mid x_{i,\leq t}, y_{i,<t}) \\
&\stackrel{\text{def}}{=} \hat{P}^{\mathcal{V}}_{\text{LLM}}(\textsc{Tok}(\textsc{Title}(y_{i,t})) \mid \textsc{Tok}(\textsc{Tmpl}(x_{i,\leq t}, y_{i,<t}))).
\end{aligned} \tag{1}
$$

[6]For example, https://cookbook.openai.com/examples/using_logprobs explains how to use the logprobs parameter in OpenAI API requests to evaluate token probabilities, allowing analysis of model confidence and alternative predictions for improved understanding of text generation.

More details are discussed in Appendix D.

Third, some LLMs make it possible to extract a lower-dimensional "embedding" or "representation" of text, where any sequence of tokens is associated with a real-valued vector. For example, for the Llama-2-7BLLM that we use in this paper, input text is represented as a vector of 4,096 floating point numbers. Formally, we let $\mathcal{E}_{\text{LLM}} : \cup_{j \leq C_{\text{LLM}}} \mathcal{V}_{\text{LLM}}^j \to \mathbb{R}^{d_{\text{LLM}}}$ be the "embedding function", where $d_{\text{LLM}}$ denotes embedding dimension. The composite function $\mathcal{E}_{\text{LLM}} \circ \textsc{Tok}$ generates the embedding of any input string of words (i.e., the "prompt").

## 5. Benchmark Occupation Models

### 5.1 *Empirical Transition Frequencies*

The empirical transition frequency is a simple baseline. Let $\#^{\text{(train)}}\{y\}$ denote the number of times occupation $y$ appears in the training data, and $\#^{\text{(train)}}\{y \to y'\}$ denote the number of times the transition from occupation $y$ to $y'$ appears in the training data. In order to avoid the challenge of dividing by 0, we add a constant (here, 1) to each occupation and each transition. The model then estimates the probability of transitioning from occupation $y$ to $y'$ (where all individuals are in the "null" occupation when $t = 0$) as

$$\hat{P}_{\text{Empirical}}(y_{i,t} \mid x_{i,\leq t}, y_{i,<t}) = \frac{\#^{\text{(train)}}\{y_{i,t-1} \to y_{i,t}\} + 1}{\#^{\text{(train)}}\{y_{i,t-1}\} + 1}.$$

The empirical model does not use any covariates or other information beyond the immediately preceding occupation to make predictions.

### 5.2 *Multinomial Logistic Regression*

Another natural approach to occupational modeling is to build a multinomial logistic regression model, where $\mathcal{Y}$ is the set of alternatives. Researchers often use a fixed number of covariates summarizing information in $(x_{i,\leq t}, y_{i,<t})$ as features. Formally, we let $z_{i,t} = g(x_{i,\leq t}, y_{i,<t})$ be the vector of covariates for predicting $y_{i,t}$, where the length of $z_{i,t}$ is fixed for all $(i,t)$. For example, $g$ might map history into a set of indicator variables for whether the previous occupation $y_{i,t-1}$ is equal to

each possible occupation, and then build a multinomial logistic regression model on top of that; in this case, $z_{i,t}$ is a vector of length $|\mathcal{Y}|$ with a single non-zero entry. With such a specification, the multinomial logistic regression model reduces to the model using empirical transition frequencies. For each occupation $y \in \mathcal{Y}$, the logistic regression model estimates a parameter $\beta_y$ with the same length as $z_{i,t}$, and the conditional distribution of next occupation is given by

$$\hat{P}_{\text{MNL}}(y_{i,t} \mid x_{i,\le t}, y_{i,<t}) = \frac{\exp(z_{i,t}^\top \beta_{y_{i,t}})}{\sum_{y' \in \mathcal{Y}} \exp(z_{i,t}^\top \beta_{y'})}. \tag{2}$$

The set of parameters $\{\beta_y\}_{y \in \mathcal{Y}}$ is estimated using maximum likelihood estimation, with optional regularization.

In our paper, we use LLMs to build an embedding vector of the career history $x_{i,\le t}, y_{i,<t}$ and use it as the vector of covariates in the logistic regression. We discuss more details in Section 7.1.

### 5.3 *CAREER*

Researchers have also proposed using transformer-based models to predict the next occupation of an individual given their covariates and history (Vafa et al. (2024)). CAREER by Vafa et al. (2024) is a transformer-based model that is trained to predict the next occupation of an individual given their covariates and history; that is, the prediction space is $\mathcal{Y}$. Compared to empirical transition frequency models and multinomial logistic regression, the CAREER model has two key differences. First, it builds a much richer functional form mapping history to predictions, making use of a custom-designed transformer neural network. Second, the model is estimated sequentially on two data sets, following the foundation model and fine-tuning approach described in the introduction. That is, first the model is pre-trained on large-scale resume data, and subsequently it is fine-tuned using representative survey data.

Consider the $t^{\text{th}}$ record of worker $i$ with $(x_{i,\le t}, y_{i,<t})$ as predictors and $y_{i,t}$ as the ground truth next occupation. CAREER estimates an embedding function $\mathcal{E}_{\text{CAREER}} : \mathcal{X}_{\text{inv}} \times \mathcal{X}_{\text{var}}^t \times \mathcal{Y}^{t-1} \to \mathbb{R}^{d_{\text{CAREER}}}$, where the value of the embedding is de-

noted $h_{i,t}$ and $d_{\text{CAREER}}$ denotes the embedding dimension. The embedding function is parameterized by an $L$-layer transformer neural network, where each layer processes the previous one to generate increasingly complex representations. Here, we provide a slightly simplified description of the transformer architecture; see Vafa et al. (2024) for more details. The first layer embedding, denoted by $h_{i,t}^{(1)} \in \mathbb{R}^{d_{\text{CAREER}}}$, only incorporates an individual's most recent job and covariates:

$$h_{i,t}^{(1)} = e_{\text{occupation}}\,(y_{i,t-1}) + e_{\text{static}}(x_i) + e_{\text{dynamic}}(x_{i,t}) + e_{\text{time}}(t),$$

where each $e$ is an embedding function with output in $\mathbb{R}^{d_{\text{CAREER}}}$. Then, CAREER constructs subsequent layers $h_{i,t}^{(\ell)}$ as described in Equation (3); for simplicity, the notation omits the dependencies on covariates and previous occupations in $h_{i,t}$.

$$\begin{aligned}
\pi_{i,t,t'}^{(\ell)} &\propto \exp\left\{\left(h_{i,t}^{(\ell)}\right)^{\top} W^{(\ell)} h_{i,t'}^{(\ell)}\right\} \qquad \text{for all } t' \leq t \\
\tilde{h}_{i,t}^{(\ell)} &= h_{i,t}^{(\ell)} + \sum_{t'=1}^{t} \pi_{i,t,t'}^{(\ell)} * h_{i,t'}^{(\ell)} \\
h_{i,t}^{(\ell+1)} &= \text{FFN}^{(\ell)}\left(\tilde{h}_{i,t}^{(\ell)}\right),
\end{aligned} \tag{3}$$

where $W^{\ell} \in \mathbb{R}^{d_{\text{CAREER}} \times d_{\text{CAREER}}}$ is a trainable model parameter and $\text{FFN}^{(\ell)} : \mathbb{R}^{d_{\text{CAREER}}} \to \mathbb{R}^{d_{\text{CAREER}}}$ is a two-layer feed-forward network specific to the $\ell^{\text{th}}$ layer. The final layer $h_{i,t}^{(L)}(x_{i,\leq t}, y_{i,<t}) \in \mathbb{R}^{d_{\text{CAREER}}}$ is a fixed-length representation summarizing the individual's career history up to the $t^{\text{th}}$ observation.

Because many individuals do not change their occupation from time $t$ to $t+1$, CAREER is designed as a two-stage model that first predicts whether an individual will switch occupations and, if so, the probability that they will switch to each occupation. It uses the representation $h_{i,t}^{(L)}$ to make this two-stage prediction:

**Stage 1.** Letting $\eta \in \mathbb{R}^{d_{\text{CAREER}}}$ be a vector of regression coefficients:

$$\hat{P}_{\text{CAREER}}(\text{move}_{i,t} \mid x_{i,\leq t}, y_{i,<t}) = \frac{1}{1 + \exp(-\eta \cdot h_{i,t}^{(L)}(x_{i,\leq t}, y_{i,<t}))},$$

**Stage 2.** Letting $\beta \in \mathbb{R}^{d_{\text{CAREER}}}$ be a matrix of regression coefficients:

$$\hat{P}_{\text{CAREER}}(y_{i,t} \mid x_{i,\leq t}, y_{i,<t}, \text{move}_{i,t} = 1) = \frac{\exp\{\beta_{y_{i,t}} \cdot h^{(L)}_{i,t}(x_{i,\leq t}, y_{i,<t}))\}}{\sum_{y' \neq y_{i,t-1}} \exp\{\beta_{y'} \cdot h^{(L)}_{i,t}(x_{i,\leq t}, y_{i,<t}))\}},$$

$$\hat{P}_{\text{CAREER}}(y \mid x_{i,\leq t}, y_{i,<t}, \text{move}_{i,t} = 0) = \mathbf{1}\{y = y_{i,t-1}\}.$$

Finally, the $\hat{P}_{\text{CAREER}}(y \mid x_{i,\leq t}, y_{i,<t})$ can be computed using quantities above.

$$\hat{P}_{\text{CAREER}}(y \mid x_{i,\leq t}, y_{i,<t}) =$$
$$\begin{cases} 1 - \hat{P}_{\text{CAREER}}(\text{move}_{i,t} \mid x_{i,\leq t}, y_{i,<t}) & \text{if } y = y_{i,t-1} \\ \hat{P}_{\text{CAREER}}(\text{move}_{i,t} \mid x_{i,\leq t}, y_{i,<t})\hat{P}_{\text{CAREER}}(y \mid x_{i,\leq t}, y_{i,<t}, \text{move}_{i,t} = 1) & \text{if } y \neq y_{i,t-1} \end{cases}.$$

In practice, the CAREER model makes predictions by marginalizing over the latent variable in the first stage.

To estimate the parameters of the model, the CAREER model is first pre-trained using 24 million career trajectories from a large dataset of resumes. Then, the pre-trained model weights are further updated in the fine-tuning step, but the gradient is computed using career trajectories from survey datasets of interest. Additional details on the CAREER model are provided in Appendix B.

## 6. Data

### 6.1 *Representative Survey Datasets.*

In this paper, our primary sources of data are three surveys of workers in the U.S. population. These surveys follow samples of individual workers, where workers are interviewed at regular intervals. The survey samples are constructed to be representative of the U.S. population at particular points in time.

The first dataset we consider is the Panel Study of Income Dynamics (PSID), which began in 1968. The sample of this dataset is intended to be representative of the United States as a whole, and new participants are added to the sample over time. Occupation information is consistently available starting in 1981, so we restrict our attention to survey years starting then. We further analyze data from two waves of the National Longitudinal Survey of Youth (NLSY). The NLSY 1979 follows a cohort of people aged 14-22 in 1979 throughout these workers' careers. The NLSY 1997 began in 1997 and followed a cohort of individuals aged 12-16 at that time throughout their careers. We use extracts from these surveys to build longitudinal datasets for individual workers. Details of our dataset construction are reported in Appendix N.

We encode occupations using the `occ1990dd` system (Autor and Dorn (2013)) to map different versions of Census OCC occupational codes to a harmonized set of codes. In addition to the 331 occupations from the `occ1990dd` taxonomy, we include three special categories: "education," "out of labor force," and "unemployed." We extract demographic characteristics, specifically gender, ethnicity, region of the country, and sometimes birth year. To simplify our analysis, we assign each worker a single, unchanging value for each demographic characteristic, typically the first valid value, and we do not allow it to change even if the original survey specifies different values in different survey years. We do not impute occupations for years without survey responses and focus on a single main occupation reported by the subject.

We refer to the cleaned versions of the three survey datasets as PSID81, NLSY79, and NLSY97. Table 1 summarizes the total number of workers and transitions (individual-year survey observations, denoted by $\sum_i T_i$) in each survey dataset. The PSID81 dataset has 10.1 transitions per individual on average (median is 8 and maximum is 29), and the NLSY79 and NLSY97 track relatively fewer workers but have more transitions per individual, with 20.82 (median is 25 and maximum is 29) and 16.56 (median is 19 and maximum is 20), respectively, observations per worker on average.

As previously discussed, we convert individuals' complete career trajectories into natural language paragraphs using a text template. The total number of tokens ranges from 3.1 million to 7.9 million. The average length of career history $\textsc{Tmpl}(x_{i,\leq t}, y_{i,<t})$ ranges from 210 to 280 tokens depending on the dataset, while the average length of complete career history $\textsc{Tmpl}(x_{i,\leq T_i}, y_{i,\leq T_i})$ ranges from 250

TABLE 1. Description of datasets.

| | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| Number of Individuals | 31,056 | 12,479 | 8,984 |
| Tokens in $\cup_i \text{TMPL}(x_{i,\leq T_i}, y_{i,\leq T_i})$ | 7,902,511 | 5,406,412 | 3,135,367 |
| Number of Transitions $\sum_i T_i$ | 313,622 | 259,778 | 148,795 |
| Tokens in $\cup_i \cup_{1\leq t\leq T_i} \text{TMPL}(x_{i,\leq t}, y_{i,<t})$ | 69,139,450 | 72,639,496 | 32,368,253 |
| "First observation" transitions $t=1$ | 9.9% | 4.8% | 6.0% |
| "Moving" transitions with $y_{i,t-1} \neq y_{i,t}$ | 38.5% | 44.5% | 37.0% |
| "Staying" transitions with $y_{i,t-1} = y_{i,t}$ | 51.6% | 50.6% | 57.0% |

*Note*: The top panel reports counts of individuals, transitions, and tokens. Token counts are reported separately for text representations used in fine-tuning $\text{TMPL}(x_{i,\leq T_i}, y_{i,\leq T_i})$ and text representations used for inference $\text{TMPL}(x_{i,\leq t}, y_{i,<t})$. The bottom panel reports the proportion of transitions corresponding to three transition types: first observation, moving, and staying.

tokens to 430 tokens; all of these templates fit well within Llama-2 model's context window of 4,096 tokens.

Transitions between two observations can be categorized into three types: *first observation*, *moving* (i.e., the current occupation is different from the occupation reported in the previous observation), and *staying* (i.e., the current occupation is the same as the occupation reported in the previous observation). Table 1 shows the number of transitions by transition type, highlighting that between 50% and 60% of transitions involve staying in the same occupation.

We divide each of the three survey datasets into training (70%), validation (10%), and test (20%) samples, where the allocation is performed at the individual level so that all of an individual's transitions are in the same set. All results in this paper about the performance of models are presented for the test set. Online Appendix C provides more details about each dataset, while Appendix Table N.1 provides summary statistics by dataset for the demographic variables we consider.

Birth cohort is an important factor affecting workers' career trajectories (Wachter (2020), Lersch et al. (2020)); however, neither birth year nor age information was used in the CAREER model. When comparing our models to CAREER, we present results about models trained without incorporating birth year, but we include this valuable information when training models for comparisons that do not in-

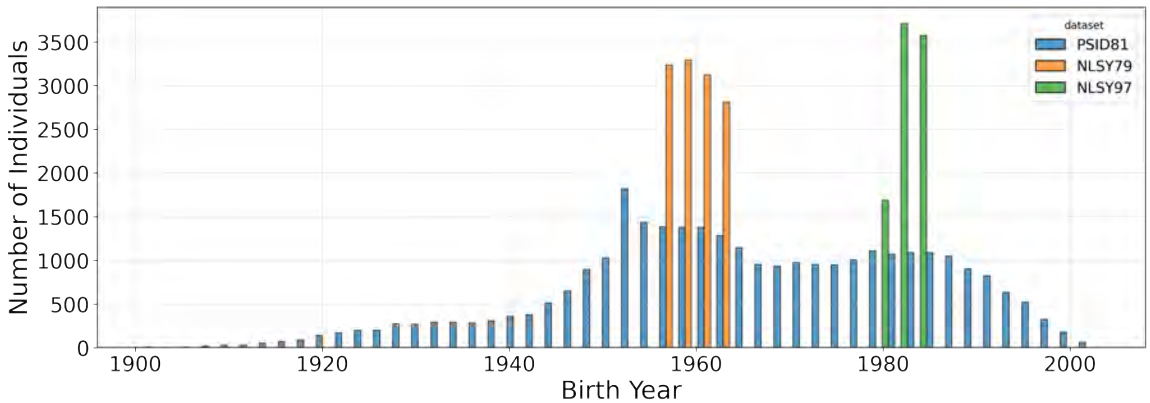



FIGURE 1. Distributions of individuals' birth years by survey dataset.

volve CAREER. Figure 1 shows distributions of individuals' birth years in each of the three survey datasets. While birth years of PSID81 individuals span the range covered by NLSY79 and NLSY97, birth years of NLSY individuals are clustered within a small range due to the design of NLSY surveys.

Table 2 presents the top ten occupations in each dataset, highlighting commonalities and variations across datasets. Notable trends include "Not in labor force" ranking highest in all datasets, while occupations like "In education" show substantial variation, ranking $9^{th}$ in PSID but $2^{nd}$ and $1^{st}$ in NLSY79 and NLSY97, respectively.

Figure 2 illustrates the distribution of individual ages across calendar years for each dataset. In the NLSY datasets, the age distribution increases steadily over time, reflecting the longitudinal design that follows the same cohort of individuals. In contrast, the PSID dataset allows for dynamic changes in its subject pool, with individuals entering (e.g., upon becoming the head of a household) and exiting the study. Consequently, the PSID81 dataset exhibits a broader but more temporally stable age distribution. This figure highlights the degree of overlap in age distributions across the three datasets, suggesting potential opportunities for transfer learning between them.

TABLE 2. Top occupations by dataset.

| Occupation | PSID81 Proportion | PSID81 Rank | NLSY79 Proportion | NLSY79 Rank | NLSY97 Proportion | NLSY97 Rank |
|---|---|---|---|---|---|---|
| Not in labor force | 0.192 | 1 | 0.177 | 1 | 0.122 | 2 |
| Unemployed | 0.067 | 2 | 0.040 | 4 | 0.034 | 3 |
| Postmasters and mail superintendents | 0.058 | 3 | 0.045 | 3 | 0.019 | 5 |
| Coin, vending, and amusement machine servicers and repairers | 0.025 | 4 | 0.025 | 5 | 0.013 | 9 |
| Secretaries and administrative assistants | 0.022 | 5 | 0.020 | 6 | 0.008 | 16 |
| Phlebotomists | 0.021 | 6 | 0.017 | 8 | 0.020 | 4 |
| Telemarketers | 0.016 | 7 | 0.005 | 41 | 0.016 | 7 |
| Maids and housekeeping cleaners | 0.014 | 8 | 0.011 | 13 | 0.004 | 31 |
| In education | 0.013 | 9 | 0.136 | 2 | 0.343 | 1 |
| Elementary and middle school teachers | 0.013 | 10 | 0.008 | 25 | 0.007 | 19 |
| Painting workers | 0.013 | 11 | 0.015 | 10 | 0.005 | 28 |
| Sales Representatives Services All Other | 0.011 | 13 | 0.017 | 9 | 0.006 | 25 |
| Septic tank servicers and sewer pipe cleaners | 0.010 | 15 | 0.018 | 7 | 0.012 | 10 |
| Cashiers | 0.009 | 21 | 0.011 | 11 | 0.018 | 6 |
| First-line supervisors/managers of retail sales workers | 0.008 | 28 | 0.005 | 40 | 0.014 | 8 |

*Note*: We take the union of the top ten occupations from each dataset separately (15 occupations in total) and report the proportion of transitions to each occupation in each dataset, as well as the rank of the proportion compared to other occupations in the same dataset. Readers can refer to Appendix Figure N.1 for a word cloud of job titles.

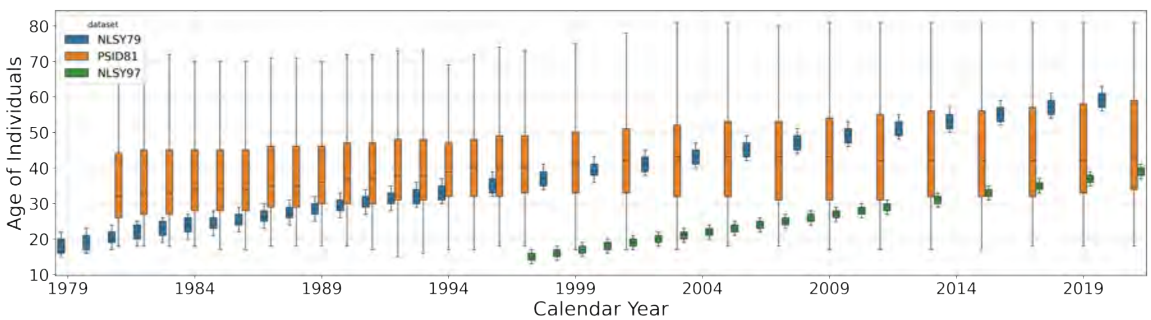



FIGURE 2. Distribution of individuals' ages by calendar year of observation.

### 6.2 *Large-Scale Resume Data*

In this paper, we re-implement the full pre-training and fine-tuning pipeline of the CAREER model so that we can carry out the fine-tuning step on identical survey datasets. Pre-training CAREER involves using a proprietary resume dataset of 23.7 million resumes acquired from Zippa Inc.[7] As described in Appendix B, we

[7]Zippia is a data-driven career intelligence platform that leverages analytics to provide personalized job recommendations, salary insights, and career development resources. The com-

follow the approach of Vafa et al. (2024) to prepare and clean the data from Zippia Inc. This pre-training resume data represents resumes from the Zippia data as annual sequences of `occ1990dd` occupations, with tie-breaking rules for multiple jobs per year. Covariates include the year of each job, last educational degree, and location, standardized following the approach we use for cleaning the survey datasets. Missing covariates are replaced by a special token, and missing occupational years are dropped. The final dataset comprises 245 million transitions (that is, individual-year observations).

## 7. Comparing Performance of Occupation Models

In this section, we explore different approaches to leveraging LLMs to build occupation models, comparing the performance of each to CAREER.

### 7.1 *LLM Embeddings as Features in Multinomial Logistic Regression Models*

This section implements and evaluates the embedding-based approach introduced in Section 4.3 to exploit LLMs for occupational modeling.. To predict an individual's next job from their embedding, we train a multinomial logistic regression model, where the outcome is the occupation codes, as described in Section 5.2.

We first convert the career history $(x_{i,\leq t}, y_{i,<t})$ to natural language using the text template described in Section 4.2. We then pass the text to an LLM and extract the model's embedding, $\mathcal{E}_{\text{LLM}}(\textsc{Tmpl}(x_{i,\leq t}, y_{i,<t})) \in \mathbb{R}^{d_{\text{LLM}}}$. This approach requires that the researcher has access to the embeddings from the LLM either through an API or by using an open-weight model. We consider a wide range of off-the-shelf models to embed career histories into embedding vectors, including Llama-2-7B/13B, Llama-3.1-8B, Llama-3.2-1B/3B, as well as the latest `text-embedding-3-large` text embedding model provided by OpenAI.

---

pany aggregates labor market data to offer tailored guidance for job seekers, aiming to optimize their career decisions and employability. Other vendors providing similar data include Kaggle https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset and Revelio https://www.data-dictionary.reveliolabs.com/index.html.

We then train a multinomial logistic regression on top of these embeddings for the next occupation prediction task.[8] Appendix E provides additional technical details on our embeddings-based approach. Table 3 compares performance across models. The previous state-of-the-art CAREER model outperforms the embedding-based multinomial logistic regression approach.[9] The embeddings in Table 3 are constructed using text templates that incorporate birth year information, whereas the CAREER model does not utilize birth year information, meaning that the CAREER model outperformed these embedding-based approaches in predictive performance despite relying on less information.

TABLE 3. Test set perplexity for embedding-based approaches vs. CAREER.

| | **Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|---|
| | **Number of Transitions** $(\sum_{i\in\textbf{test}} T_i)$ | 61,759 | 51,593 | 29,949 |
| **Model** | **Embedding Dimension** $d_{\textbf{LLM}}$ | | | |
| OpenAI Text Embedding | 3,072 | 11.18 (0.191) | 12.06 (0.245) | 9.28 (0.189) |
| OTS Llama-2-7B | 4,096 | 10.18 (0.169) | 10.76 (0.216) | 8.22 (0.164) |
| OTS Llama-2-13B | 5,120 | 10.17 (0.169) | 10.70 (0.203) | 7.99 (0.152) |
| OTS Llama-3.1-8B | 4,096 | 9.92 (0.162) | 10.52 (0.203) | 7.89 (0.151) |
| OTS Llama-3.2-1B | 2,048 | 9.92 (0.164) | 10.38 (0.200) | 7.88 (0.146) |
| OTS Llama-3.2-3B | 3,072 | 9.79 (0.156) | 10.28 (0.199) | 7.66 (0.141) |
| CAREER (Vafa et al. (2024)) | – | 8.60 (0.132) | 8.64 (0.158) | 6.41 (0.101) |

*Note*: Test-set-bootstrap standard errors are reported in parentheses.

[8]Note that the embedding-based approach cannot predict occupations that are not in the training set; therefore, we drop transitions of occupations that are present in the test set, but not the training set in Table 3. The train/test split that we use to report results in this paper has 13 transitions in the test set for PSID81 and two for NLSY97 that are dropped due to having occupation codes that are not in the training set. These few observations have a negligible impact on our perplexity metric, as it is inherently robust to individual data points. The language model-based approach addresses this issue by producing predictive probabilities that are inherently valid for all job titles, including those not represented in the training set. In later tables, there will be 13 more transitions in the PSID81 and two more transitions in NLSY97.

[9]For CAREER, predictions were made directly; we do **not** use CAREER as an embedding engine and build multinominal logistic regression on top of the embeddings.

TABLE 4. Test-set perplexity for off-the-shelf LLMs vs. CAREER.

| **Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| **Number of Transitions** $(\sum_{i \in \textbf{test}} T_i)$ | 61,772 | 51,593 | 29,951 |
| **Model** | | | |
| OTS Llama-2-7B | 361.74 (12.615) | 292.34 (9.179) | 219.56 (6.686) |
| OTS Llama-2-13B | 149.86 (5.333) | 133.32 (4.738) | 113.77 (3.539) |
| OTS Llama-3.1-8B | 140.09 (5.083) | 116.45 (4.227) | 93.73 (2.913) |
| OTS Llama-3.2-1B | 475.25 (22.660) | 404.73 (19.807) | 258.65 (11.283) |
| OTS Llama-3.2-3B | 180.55 (7.067) | 145.59 (5.607) | 115.99 (3.856) |
| CAREER (Vafa et al. (2024)) | 8.60 (0.132) | 8.64 (0.158) | 6.41 (0.101) |

*Note*: Test-set-bootstrap standard errors are reported in parentheses.

### 7.2 *Using Off-The-Shelf Large Language Models as Occupation Models*

In this section, we report results about the performance of occupation models based on off-the-shelf LLMs, applying Equation (1) to estimate $\hat{P}_{\text{LLM}}$ for several alternative LLMs.[10] Because evaluating perplexity requires accessing a model's assigned probabilities, we restrict attention to open-source LLMs where it is possible to obtain predicted probabilities directly, with the exception of Section 7.4, where we evaluate the ability of OpenAI gpt-4o-mini to produce valid job titles in response to a prompt. In particular, we study open-source LLMs from the Llama family of models: Llama-2, Llama-3.1, and Llama-3.2. For example, Llama-2 models were trained by Meta on approximately 2 trillion tokens of text, much of it from the Internet, and are among the most capable open-source LLMs currently available (Touvron et al. (2023)). We do not study bigger models such as Llama-2-70B and Llama-3.1-405B because fine-tuning and evaluating this model across many variations requires substantial cost and computational resources.

Table 4 contains the perplexity of off-the-shelf LLMs. As a comparison, we also include the perplexity of CAREER by Vafa et al. (2024), a non-language model

[10]To improve computational efficiency for prediction, we quantize all LLMs in this paper to 8-bit precision while running model inference. We perform full-precision inference on a subset of our experiments, and the difference in performance was small. See Appendix F for more details on full-precision versus quantized model experiments.

developed solely to predict nationally representative occupational trajectories. For a fair comparison to CAREER, we do not include the birth year information in LLMs' prompt $\textsc{Tmpl}(x_{i,\leq t}, y_{i,<t})$ because CAREER does not use birth year or age information either. The LLMs consistently make predictions with higher levels of perplexity.[11]

The unsatisfactory performance of off-the-shelf LLMs can be attributed to two factors: off-the-shelf LLMs are not adapted to the career trajectory distributions in our survey dataset, and these LLMs do not know the set of valid job titles to predict. To better understand the poor performance of the model based on off-the-shelf LLMs, we assess the responses that the LLMs provide when prompted with examples of tokenized text templates. Online Appendix D provides some examples. While the responses appear plausible, the LLMs also assign mass to strings that are not valid job titles. In the next section, we explore alternative prompting strategies designed to encourage the LLMs to consider only valid occupations when estimating the probability of a given occupation.

### 7.3 *Improving Off-the-Shelf LLMs using Prompting Strategies*

Table 4 shows that off-the-shelf pre-trained LLMs perform worse at predicting next occupations compared to the state-of-the-art CAREER model. In this section, we show that we can improve their performance by adding additional information into the prompt to facilitate in-context learning. We explore two types of information: (1) the list of job titles and (2) additional resume examples from other workers. A limiting factor in our ability to use such prompting strategies is the maximum context length of the models. For most models, we cannot include both the full list of job titles and example resumes. See Appendix G for details on the constraints and more granular results.

*Job Titles in the Prompt* We prepend the list of all 335 job titles, one per line, to the text representation of career history $\textsc{Tmpl}(x_{i,\leq t}, y_{i,<t})$, which informs the off-the-shelf model about the prediction space. With this modification, the prompt

[11]For reference, a completely uninformative model that assigns uniform mass to each possible occupation would achieve a perplexity of $|\mathcal{Y}|$, which is 335.

passed into the LLM becomes [List of Job Titles] $\oplus$ $\text{TMPL}(x_{i,\leq t}, y_{i,<t})$, where $\oplus$ denotes string concatenation.

*Example Resumes in the Prompt* We prepend example resumes randomly sampled (without replacement) from workers in the training set to the text representation of career history $\text{TMPL}(x_{i,\leq t}, y_{i,<t})$, which informs the off-the-shelf model about our data structure. The prompt fed into the model becomes $\text{TMPL}(x_{j_1,\leq T_{j_1}}, y_{j_1,\leq T_{j_1}}) \oplus \cdots \oplus \text{TMPL}(x_{j_K,\leq T_{j_K}}, y_{j_K,\leq T_{j_K}}) \oplus \text{TMPL}(x_{i,\leq t}, y_{i,<t})$ if we add $K$ individuals $j_1, \cdots, j_K$ where $\text{TMPL}(x_{j,\leq T_j}, y_{j,\leq T_j})$ means the complete resume for individual $j$.

Since the main models we study in this paper, Llama-2-7B and Llama-2-13B, only have enough context length for either job titles or a few examples resumes (both have 4k context length), we study the open-sourced Llama-2-7B-32k model provided by Together AI, the Llama-3.1-8B model (with a 128k context window), and the Llama-3.2-1B/3B model (with a 128k context window) to assess the benefits of combining the two prompting approaches. These models with longer context windows allow us to fit significantly more example resumes in our prompt. The average length of prompts in our experiments is much longer than the TMPL representation of career history we use in the previous section, leading to a significant increase in the computational cost of processing each prompt. As a result, we randomly sample 10% of workers from the test set of each survey dataset in this exercise.

Table 5 shows that when we use ten example resumes and job titles at the same time, the best-performing model reduces perplexity by a factor of 10 to 20, depending on the dataset. However, this approach to occupation modeling is still substantially worse than that of CAREER. We also observe that adding ten example resumes to the prompt reduces perplexity more than adding job titles for all models in Table 5. Appendix G provides results for including one, three, or five example resumes that show adding job titles to the prompt outperforms adding up to three to five example resumes.

TABLE 5. Test-set perplexity for off-the-shelf models with in-context learning examples (resumes) and/or job titles.

| | **Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|---|
| | **Number of Transitions** $(\sum_{i \in \textbf{test}} T_i)$ | 6,177 | 5,159 | 2,995 |
| **Models Without Job Titles in Prompt** | **# Resumes** | | | |
| OTS Llama-2-7b-32k | 0 | 241.04 (22.812) | 182.75 (16.373) | 173.94 (22.880) |
| OTS Llama-2-7b-32k | 10 | 36.53 (2.131) | 26.20 (1.495) | 17.52 (1.510) |
| OTS Llama-3.1-8B | 0 | 127.79 (10.564) | 110.87 (8.973) | 99.16 (11.408) |
| OTS Llama-3.1-8B | 10 | 25.08 (1.385) | 19.41 (1.009) | 13.68 (1.034) |
| OTS Llama-3.2-1B | 0 | 456.09 (51.012) | 371.33 (38.769) | 277.73 (40.961) |
| OTS Llama-3.2-1B | 10 | 52.90 (3.740) | 36.04 (2.409) | 24.99 (2.631) |
| OTS Llama-3.2-3B | 0 | 165.11 (14.493) | 134.39 (11.186) | 122.58 (14.671) |
| OTS Llama-3.2-3B | 10 | 29.92 (1.726) | 22.95 (1.306) | 16.21 (1.334) |
| **Models With Job Titles in Prompt** | **# Resumes** | | | |
| OTS Llama-2-7b-32k | 0 | 42.01 (2.522) | 45.72 (2.678) | 47.95 (4.127) |
| OTS Llama-2-7b-32k | 10 | 20.73 (0.918) | 18.04 (0.732) | 11.74 (0.736) |
| OTS Llama-3.1-8B | 0 | 30.85 (1.633) | 26.98 (1.309) | 21.91 (1.394) |
| OTS Llama-3.1-8B | 10 | 16.45 (0.763) | 15.20 (0.631) | 10.49 (0.672) |
| OTS Llama-3.2-1B | 0 | 62.23 (3.885) | 53.31 (3.068) | 45.25 (3.518) |
| OTS Llama-3.2-1B | 10 | 22.95 (1.130) | 20.25 (0.913) | 14.02 (0.990) |
| OTS Llama-3.2-3B | 0 | 39.81 (2.199) | 39.24 (2.227) | 35.44 (2.700) |
| OTS Llama-3.2-3B | 10 | 17.81 (0.824) | 16.39 (0.683) | 11.52 (0.749) |

*Note*: Perplexity on a 10% random sample of the test set, with test-set-bootstrap standard errors in parentheses.

### 7.4 *Likelihood of Generating Valid Job Titles*

As mentioned in Section 4, we can feed a LLM with a prompt and repeatedly sample from the LLM's output distribution to generate a sequence of tokens as the continuation of the prompt. Specifically, in the settings considered in the last subsection, we assess whether the model generates a continuation that starts with a valid job title:

$$\exists y \in \mathcal{Y} \text{ s.t., LLM.generate(prompt).startswith}(\textsc{Title}(y)) \tag{4}$$

Figure 3 summarizes the empirical probability that off-the-shelf Llama models generate valid job titles (i.e., the event in Equation (4) occurs) on a 10% subsample of the test set, where the figure illustrates how the results vary with different prompting strategies. We find that the probabilities range from 0.68 to greater than 0.99, the latter performance obtained from combining job titles and example resumes in the prompt.

We then conduct the same exercise using the `gpt-4o-mini-2024-07-18` model provided by OpenAI.[12] As illustrated in Figure 3, the OpenAI patterns and results are similar to those of the Llama models, with slightly larger probabilities of correct job titles and a maximum of 0.97 on the NLSY97 dataset with both job titles and ten sample resumes included in the prompt.

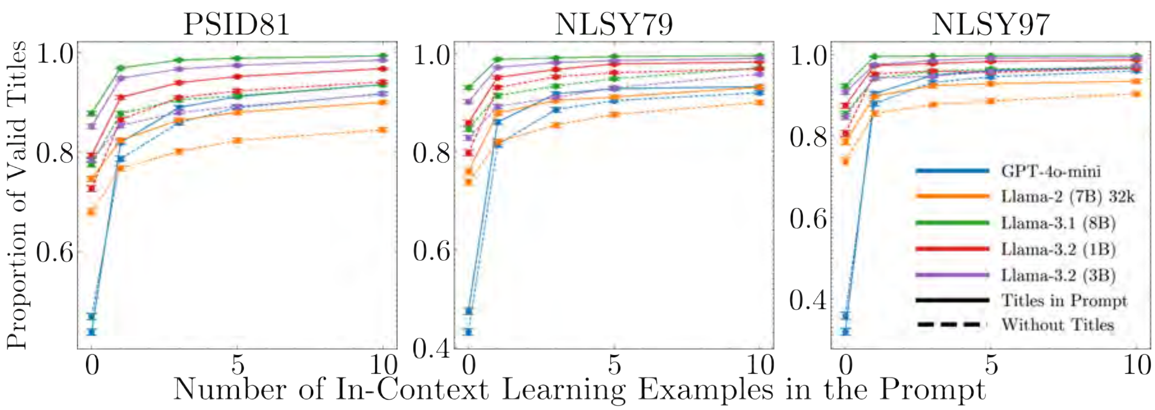



FIGURE 3. Likelihoods of generating valid job titles given different numbers of in-context learning examples (resumes) and job titles in the prompt for off-the-shelf LLMs.

## 8. Fine-Tuning LLMs to Improve Predictive Performance on Surveys

### 8.1 *Occupation Models Derived From Fine-Tuned Language Models*

In this section, we analyze the performance of occupational models based on LLMs that have been fine-tuned on text templates created from our survey datasets. We use the term FT-LABOR-LLM to refer to the combination of a base model (either Llama-2-7B or Llama-2-13B) and fine-tuning data, as well as to refer to the union of the fine-tuned models we evaluate in this paper.

[12]We use OpenAI's chat completion batch API to generate those continuations. We set the temperature to be 1, the seed to be 42, the maximum number of generated tokens to be 20, and the stop word to be the new line symbol (i.e., "\n") for these generations.

The fine-tuning process proceeds in several steps. For each individual $i$, we use the text template discussed in Section 4.2 to build a text representation of their entire career, denoted as $\textsc{Tmpl}(x_{i,\leq T_i}, y_{i,\leq T_i})$. We then fine-tune the two Llama-2 models (7B and 13B) separately on each of the three training set text templates, resulting in three sets of fine-tuned models. We refer to the occupation models derived from these fine-tuned models as FT-7B and FT-13B, dropping the "Llama-2" nomenclature because we only fine-tune Llama-2 models. Since LLMs make predictions at the token level, we fine-tune models to predict each token of a textual summary of worker careers, including punctuation and meta-data, so that the FT-LABOR-LLM learns the structure of the text template as well as the conditional probabilities of tokens corresponding to jobs. The fine-tuning procedure is illustrated in Figure 4. The resulting FT-LABOR-LLMs are themselves LLMs, and we create estimates of $\hat{P}_{\text{LLM}}$ based on each of them.

The fine-tuning process itself is carried out by maximizing the log-likelihood of next-token prediction model using a form of stochastic gradient descent. Note that Section 5.3 provides an overview of the functional form of a transformer model, recalling that the "vocabulary" of that model is the set of 335 occupations rather than the set of text tokens used by language models, and that CAREER adds a few additional features to a standard transformer model. Appendix H provides details of the estimation, which we carry out using a hosted service provided by Together AI.

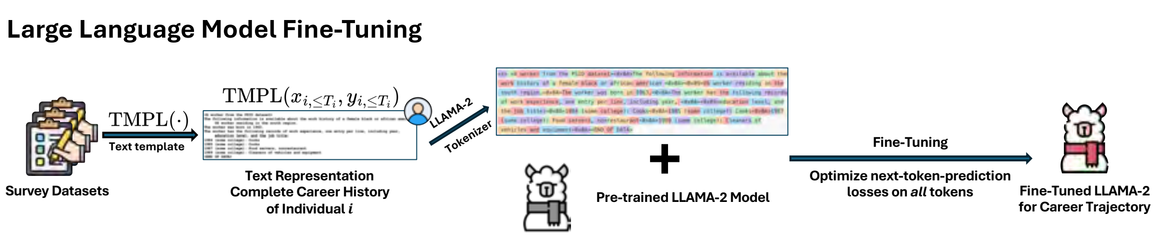



FIGURE 4. Illustration of the model fine-tuning procedure.

### 8.2 *Comparing Performance Across Foundation Models: LLM Models versus CAREER*

Table 6 reports the test set perplexity of the FT-LABOR-LLM occupation models along with the baselines described in Section 5. For a fair comparison, we explore the performance difference between CAREER (which does not use any birth year information) and the Llama-2-7B model fine-tuned and evaluated using prompts *without* the birth year information. We refer to these models as FT-7B-NBY and FT-13B-NBY to indicate the omission of birth year. We see that the perplexities are substantially lower than those based on the off-the-shelf LLMs reported in Table 4. FT-7B-NBY and FT-13B-NBY also achieve higher predictive accuracy than CAREER, which was pre-trained on 23.7 million resumes and fine-tuned for occupation modeling on survey data. The differences between CAREER and FT-7B-NBY are about ten times larger than the test-set-bootstrap standard errors (defined in Section 3.3) for PSID81 and NLSY79, while they are similar in size to the standard error for NLSY97, a substantially smaller dataset. FT-13B-NBY exhibits even larger performance improvements. Appendix I shows that both FT-7B-NBY and FT-13B-NBY also have similar or better performance than CAREER within subgroups defined by education.

As previewed in Section 3.3, one question that naturally arises is whether sampling variation in the training set and randomness in the fine-tuning estimation algorithm lead to substantial variation in estimates of performance difference. In Appendix A, we carry out a small experiment with training-set-bootstrapping. The training-set-bootstrap standard errors for perplexity of FT-7B are 0.051, 0.058, and 0.020 for PSID81, NLSY79, and NLSY97, respectively (where to facilitate other comparisons, we included birth year in the estimation and these models are fine-tuned using pooled training set). These standard errors are smaller than those reported in Table 6. We calculate the training-set-bootstrap standard error for the difference between FT-7B and FT-13B only for PSID81, and found a standard error of 0.029, larger than that corresponding test set standard error. This exercise suggests that variation due to training is not negligible, and small performance differences that appear to be statistically distinguishable from zero using test-set-bootstrap standard errors could in fact arise due to training uncertainty. Due to the large cost of training-set-bootstrapping, we report test-set-bootstrap standard errors in the rest of the paper, but we are cautious in interpreting marginally significant results.

TABLE 6. Test-set perplexity and perplexity improvement for fine-tuned vs. baseline models.

| **Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| **Number of Transitions** $(\sum_{i\in\textbf{test}} T_i)$ | 61,772 | 51,593 | 29,951 |
| **Perplexity** | | | |
| Empirical Transition Frequency | 14.65 (0.224) | 14.26 (0.271) | 10.05 (0.169) |
| CAREER (Vafa et al. (2024)) | 8.60 (0.132) | 8.64 (0.158) | 6.41 (0.101) |
| FT-7B-NBY | 8.36 (0.129) | 8.39 (0.148) | 6.40 (0.102) |
| FT-13B-NBY | 8.31 (0.127) | 8.35 (0.146) | 6.34 (0.100) |
| **Perplexity Improvement** | | | |
| PPL(CAREER)-PPL(FT-7B-NBY) | 0.24 (0.020) | 0.25 (0.023) | 0.02 (0.018) |
| PPL(CAREER)-PPL(FT-13B-NBY) | 0.29 (0.021) | 0.28 (0.023) | 0.07 (0.016) |
| PPL(FT-7B-NBY)-PPL(FT-13B-NBY) | 0.05 (0.012) | 0.04 (0.013) | 0.05 (0.011) |

*Note*: Test-set-bootstrap standard errors are in parentheses.

Next, we compare performance for the task of predicting the binary outcome of whether workers move to a different job $\text{move}_{i,t} = \mathbf{1}\{y_{i,t} \neq y_{i,t-1}\}$; and separately, we analyze performance conditional on a transition involving a move.

A standard way to evaluate the performance of alternative prediction models for binary outcomes is to compare the area under the ROC curve (AUC-ROC) in the test set, which ranges from 0 (the worst possible model) to 1 (the best possible model). Table 7 shows that the FT-7B-NBY model has AUC-ROC of 0.781, slightly greater than CAREER at 0.775. The empirical transition frequency benchmark has AUC-ROC of 0.639.

To assess how well-calibrated each model is, we split observations into ten groups based on deciles of predicted probability of changing jobs $\hat{P}(\text{move}_{i,t})$ (i.e., the next occupation $y_{i,t}$ is different from the previous one $y_{i,t-1}$), denoted as $G_1, G_2, \ldots, G_{10}$. Then, for each group, we compute the empirical percentage

TABLE 7. Area Under the ROC Curve (AUC-ROC).

| | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| Empirical | 0.653 | 0.636 | 0.604 |
| OTS Llama-2-7B-NBY (with title) | 0.713 | 0.714 | 0.677 |
| CAREER | 0.778 | 0.777 | 0.760 |
| FT-7B-NBY | 0.784 | 0.786 | 0.758 |

*Note*: For the off-the-shelf model, we use the Llama-2-7B model with the list of job titles included in the prompt.

of movers. If a model is well-calibrated, the average predicted $\hat{P}(\text{move}_{i,t})$ should match the actual proportion of movers within the corresponding group in the test set. We further calculate the average (over deciles) of the calibration error $\sqrt{\frac{1}{10}\sum_{i=j}^{10}\left[\left(\sum_{(i,t)\in G_j}\mathbb{1}\{\text{move}_{i,t}\}\right)-\left(\sum_{(i,t)\in G_j}\hat{P}_{\text{model}}(\text{move}_{i,t})\right)\right]^2}$.

Figure 5 illustrates calibration plots of the empirical transition frequency baseline, CAREER model, FT-7B-NBY model, as well as their corresponding calibration errors. The diagonal line in the plot represents a perfectly calibrated model.

We observe that our FT-7B-NBY model is better calibrated in predicting staying versus moving than the CAREER model, which underestimates moving in some groups and overestimates it in others. The CAREER model has a two-stage prediction design (i.e., predict staying versus moving, then next occupation sequentially), and the training process of CAREER pays special attention to enforcing the model calibration. In contrast, our LLM fine-tuning does not give special treatment to matching the empirical probability of staying, so it is somewhat surprising that it is better calibrated in this dimension than CAREER; with its extremely large parameter space, the FT-7B-NBY model appear to learn these probabilities without special treatment in the model.

Table 8 reports perplexity conditional on moving for alternative models, while the bottom panel reports differences between FT-LABOR-LLM models and CAREER. We see that the FT-7B-NBY and FT-13B-NBY models outperform all other models. Note that job transitions conditional on moving are inherently harder to predict.

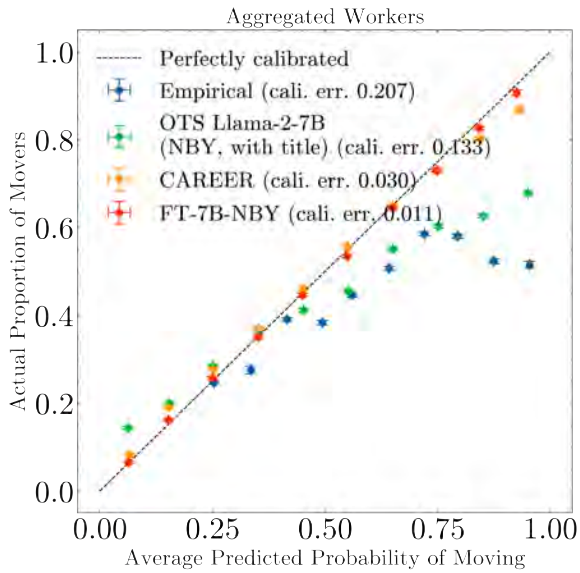



FIGURE 5. Calibration plots of baseline and fine-tuned models on the task of predicting staying in a job vs. moving jobs.

TABLE 8. Test-set perplexity and perplexity improvement for fine-tuned vs. baseline models, conditional on moving.

| **Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| **Number of Transitions** $\left(\sum_{i \in \textbf{test}} T_i\right)$ | 24,030 | 23,023 | 10,960 |
| **Perplexity** | | | |
| Empirical Transition Frequency | 59.98 (0.925) | 66.15 (0.800) | 72.31 (1.498) |
| CAREER | 24.38 (0.419) | 30.49 (0.437) | 36.43 (0.766) |
| FT-7B-NBY | 23.27 (0.402) | 29.38 (0.432) | 36.44 (0.814) |
| FT-13B-NBY | 22.97 (0.401) | 29.02 (0.427) | 35.88 (0.800) |
| **Perplexity Improvement** | | | |
| PPL(CAREER)-PPL(FT-7B-NBY) | 1.11 (0.129) | 1.14 (0.137) | 0.01 (0.214) |
| PPL(CAREER)-PPL(FT-13B-NBY) | 1.42 (0.125) | 1.50 (0.138) | 0.57 (0.193) |
| PPL(FT-7B-NBY)-PPL(FT-13B-NBY) | 0.31 (0.073) | 0.36 (0.097) | 0.56 (0.138) |

*Note*: Estimated conditional probabilities are calculated using Bayes' rule. Test sets restricted to include individual-year observations that satisfy $y_{i,t} \neq y_{i,t-1}$. Test-set-bootstrap standard errors are in parentheses.

Because the surveys we study were typically conducted every other year, our model typically needs to make predictions about transitions separated in time by two years. However, it is possible to use FT-LABOR-LLM to make predictions about transitions for years that we did not directly observe, including the years between surveys. A potential limitation of FT-LABOR-LLM is that it may not be internally consistent when predicting transitions when survey observations are separated in time; the prediction that comes out of the LLM is not constrained to be equal to the result if we were to make sequential predictions for each year and combine them via Bayes' rule. In particular, the probability FT-LABOR-LLM models assign to $y_{i,t}$ when $\text{year}_{i,t} = \text{year}_{i,t-1} + 2$ is not necessarily equal to the estimated probability by applying the model year-by-year, composing its predictions about $\text{year}_{i,t} = \text{year}_{i,t-1} + 1$ and predictions about $\text{year}_{i,t}$ conditional on potential jobs in $\text{year}_{i,t-1} + 1$. In Appendix J, we compare the model's direct predictions about a transition across two years with those constructed based on a sequence of two one-year-ahead predictions and show that the correlation between the two predictions is 0.93, meaning that the model appears to correctly account for the gap in calendar time when making predictions. We leave further exploration of this issue for future work.

## 9. Value of Data and Model Size

In this section, we analyze the roles of model complexity (number of parameters) and of quantity of data in determining performance. As discussed in the introduction, analysts using fine-tuned LLMs will need to consider costs of computation in the fine-tuning process, as well as when making predictions from the model, costs which increase with model complexity. These costs may be traded off against improved accuracy from more complex models. Another tradeoff arises when acquiring more data: more data may be available that is from a different context and thus may correspond to a different data generating process. Incorporating non-representative data in fine-tuning may or may not improve performance.

In this section, we empirically evaluate these tradeoffs by varying the datasets used for fine-tuning, for example, by combining datasets, while holding the three test sets fixed. To facilitate our discussion, we use $\mathcal{D}^{\text{(split)}}_{\text{data}}$ to denote a particular split of the dataset $\omega$, for example, $\mathcal{D}^{(\text{train})}_{\text{PSID81}}$ represents the training split of the PSID81 dataset. We explore the consequences of fine-tuning based on a different survey dataset, or combinations of survey datasets, than the survey from which the test set is drawn. Recall that all three of the survey datasets we analyze are approximately representative of the U.S. population, but as shown in Section 6.1, they incorporate different distributions of calendar year, as well as different conditional distributions of birth year for each calendar year, where we include both of these variables in the text templates for the analyses in this section.

In our first exercise, reported in Table 9, we evaluate models fine-tuned using the training split of one survey data, $\mathcal{D}^{(\text{train})}_{\omega}$, and the test split from another dataset, $\mathcal{D}^{(\text{test})}_{\omega'}$, with $\omega \neq \omega'$. This exercise shows how training from data with very different distributions of birth year and calendar year affects performance; since FT-7B and FT-13B are trained using information about both of these variables and the transformer neural network allows for rich interactions, in principle, the model could be flexible enough to predict well across distributions. In particular, the PSID81 dataset has overlap in terms of calendar year and birth year with both NLSY datasets, and it is substantially larger overall. However, the results illustrate significantly degraded predictive performance when the training data and test data are from different survey datasets.

TABLE 9. Fine-tuning on training set of dataset $\omega$ and evaluating on test split of dataset $\omega'$.

| | **Evaluation Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|---|
| | **Number of Transitions** $(\sum_{i\in\textbf{test}} T_i)$ | 61,772 | 51,593 | 29,951 |
| **Foundation Model** | **Fine-tuning Dataset** | | | |
| FT-7B | PSID81 | 8.18 (0.126) | 10.70 (0.198) | 10.52 (0.154) |
| FT-7B | NLSY79 | 9.93 (0.154) | 8.33 (0.147) | 7.96 (0.123) |
| FT-7B | NLSY97 | 12.64 (0.213) | 11.27 (0.228) | 6.35 (0.101) |
| FT-13B | PSID81 | 8.14 (0.126) | 10.16 (0.190) | 9.25 (0.135) |
| FT-13B | NLSY79 | 10.07 (0.154) | 8.28 (0.145) | 7.60 (0.114) |
| FT-13B | NLSY97 | 12.85 (0.211) | 10.93 (0.217) | 6.33 (0.100) |

*Note*: Test-set-bootstrap standard errors in parentheses.

Next, we evaluate the value of data by first pooling all training data from survey datasets together, so that $\mathcal{D}_{\text{all}}^{(\text{train})} = \bigcup_{\omega \in \{\text{PSID81, NLSY79, NLSY97}\}} \mathcal{D}_{\omega}^{(\text{train})}$. Then, we sample $P\%$ of individuals from $\mathcal{D}_{\text{all}}^{(\text{train})}$ and use the sample to fine-tune a Llama-2-7B model. Finally, we evaluate the FT-LABOR-LLM on the test split of each survey dataset separately.

Table 10 summarizes the performance of these models. The model's performance improves as we increase the amount of training data (i.e., raise the value of $P$), and the returns to data are diminishing. On the test split of dataset $\omega$, models fine-tuned on the aggregated dataset eventually outperform the model fine-tuned on the corresponding training set $\mathcal{D}_{\omega}^{(\text{train})}$, when $P \geq 80$. In addition, the models fine-tuned on the pooled data with FT-7B eventually outperform FT-13B trained on the individual baseline training sets, showing that adding data, even data from different distributions, can substitute for model complexity. Note, however, that the improvement on PSID81 is small enough (0.06) relative to the test-set-bootstrap standard error that the uncertainty derived from training may be large enough to overturn the statistical significance of the result. Indeed, in Appendix A we find a training-set-bootstrap standard error of 0.055 for this improvement, which together with test-set uncertainty would render the improvement not statistically significant.

TABLE 10. Fine-tuning on $P\%$ of the mixture of training splits of three datasets.

| **Evaluation Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| **Number of Transitions** $(\sum_{i \in \textbf{test}} T_i)$ | 61,772 | 51,593 | 29,951 |
| **Perplexity** | | | |
| FT-7B with Corresponding Training Set | 8.18 (0.126) | 8.33 (0.147) | 6.35 (0.101) |
| FT-13B with Corresponding Training Set | 8.14 (0.126) | 8.28 (0.145) | 6.33 (0.100) |
| FT-7B with 20% of Pooled Data | 8.77 (0.137) | 8.83 (0.162) | 6.53 (0.103) |
| FT-7B with 40% of Pooled Data | 8.39 (0.130) | 8.48 (0.152) | 6.34 (0.100) |
| FT-7B with 60% of Pooled Data | 8.26 (0.127) | 8.34 (0.149) | 6.26 (0.098) |
| FT-7B with 80% of Pooled Data | 8.15 (0.126) | 8.26 (0.147) | 6.21 (0.097) |
| FT-7B with 100% of Pooled Data | 8.08 (0.124) | 8.21 (0.146) | 6.19 (0.097) |
| **Perplexity Improvement** | | | |
| PPL(FT-13B)-PPL(FT-7B-20%) | -0.63 (0.024) | -0.55 (0.026) | -0.20 (0.016) |
| PPL(FT-13B)-PPL(FT-7B-40%) | -0.25 (0.017) | -0.20 (0.016) | -0.02 (0.013) |
| PPL(FT-13B)-PPL(FT-7B-60%) | -0.12 (0.014) | -0.05 (0.015) | 0.07 (0.012) |
| PPL(FT-13B)-PPL(FT-7B-80%) | -0.01 (0.013) | 0.02 (0.014) | 0.11 (0.012) |
| PPL(FT-13B)-PPL(FT-7B-100%) | 0.06 (0.014) | 0.07 (0.015) | 0.13 (0.013) |

*Note*: Test-set-bootstrap standard errors in parentheses.

In Appendix K, we consider another variation of the analysis, incrementally adding pooled data to the full baseline training set for a given survey. We find that adding the data from other surveys to the full baseline training set immediately improves performance, and increasing the training set size by 30% allows FT-7B to match or surpass the performance of FT-13B.

## 10. Sources of Performance Gains

Our experiments demonstrate that our best-performing approach, directly predicting jobs through text tokens using FT-LABOR-LLM, achieves superior perplexity scores compared to the previous state-of-the-art CAREER model. This section delves deeper into the sources of performance differences.

### 10.1 *Language Models using Numeric Job Titles*

One key difference between LLMs and baseline models, besides the number of parameters, is that the LLMs have an understanding of textual data. This section examines whether LLMs' performance is driven by their rich, deep neural network architecture or their advanced understanding of the meaning of jobs based on textual data. To do so, we create an alternative prediction space with "numeric job titles" only. We assign each occupation $y \in \mathcal{Y}$ a randomly chosen numeric job titles (in contrast to their original literal job title) from `job_000`, `job_001`, ....(e.g., `Cashiers` is mapped to `job_045`); all numeric job titles have three digits. Then, we replace all original literal job titles in the text representation with their corresponding numeric job titles, denoted as $\textsc{Tmpl}^{(\text{numeric})}(x_{i,\leq t}, y_{i,<t})$. Appendix C.1 provides an example of career history text representations with numeric job titles.

For each survey dataset, we fine-tune the Llama-2-7B model using the training corpus with numeric job titles only, and denote that fine-tuned model as FT-7B-NUMERIC; then, we compare FT-7B-NUMERIC to FT-7B fine-tuned on corresponding training split of a single survey data. While evaluating performance, we use the conditional probability of numeric job titles assigned by the LLM. For example, the predicted conditional probability of the next job being cashier is $P_{\text{LLM}}(\texttt{job\_045} \mid \textsc{Tmpl}^{(\text{numeric})}(x_{i,\leq t}, y_{i,<t}))$ instead of $P_{\text{LLM}}(\text{Cashier} \mid \textsc{Tmpl}(x_{i,\leq t}, y_{i,<t}))$, where historical jobs are also replaced with job titles.

Table 11 shows the performance of FT-7B-NUMERIC, which performed much worse than the FT-7B model using literal job titles. Our results indicate that an important contributor to the LLM's performance comes from LLM's prior knowledge about occupations; using numeric job titles disassociates this knowledge from the prediction task and hurts performance significantly.

TABLE 11. Test-set perplexity and perplexity improvement on literal vs. numeric job titles.

| **Evaluation Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| **Number of Transitions** $(\sum_{i \in \textbf{test}} T_i)$ | 61,772 | 51,593 | 29,951 |
| **Perplexity** | | | |
| PPL(FT-7B) | 8.18 (0.126) | 8.33 (0.147) | 6.35 (0.101) |
| PPL(FT-7B-NUMERIC) | 8.83 (0.141) | 9.13 (0.168) | 6.72 (0.105) |
| **Perplexity Improvement** | | | |
| PPL(FT-7B-NUMERIC)-PPL(FT-7B) | 0.64 (0.027) | 0.81 (0.031) | 0.37 (0.021) |

*Note*: Test-set-bootstrap standard errors are in parentheses.

### 10.2 *Sensitivity to Input Features*

In this section, we evaluate the importance of demographic variables for predictive performance. This exercise is not straightforward for complex, nonlinear models. If we find that including a covariate in the estimation of a model improves predictive quality on a test set, that implies that the covariate both matters in the true (unknown) data generating process, and that the predictive model makes use of the covariate in prediction. However, if excluding a covariate does not affect predictive quality, we cannot be sure whether something in the estimation process failed to capture a relationship that is present in the true data generating process (e.g., mis-specification or noise), or whether the covariate is simply not important once other covariates are incorporated. Although it is straightforward to assess whether an individual covariate has predictive power in isolation using very simple models, understanding whether it has predictive power conditional on other covariates relies on modeling. Thus, negative results about the importance of a covariate require additional analysis to confirm whether, in fact, that covariate has predictive power. Here, we do not explore the latter question.

We evaluate the importance of demographic variables for FT-7B fine-tuned on $\mathcal{D}_{\text{all}}^{(\text{train})}$, our best-performing predictive model. We apply an approach common in the machine learning literature, which entails holding fixed the estimated model, and replacing covariates with randomly assigned values in the test set, then assessing the impact on predictive performance of the model when the model is applied to the modified test set. [13]

We explore the importance of three static variables in our text representations: gender, ethnicity, and indicators for four regions of the country. To implement the randomization of the test set demographics, we create an alternative version of the test set in which, for each unit, we replace the vector of demographics with a randomly drawn vector of demographics from units in the validation set and assign the unit those demographics. We repeat this exercise with alternative combinations of variables.

Table 12 presents the results. Randomly modifying gender hurts the performance of FT-LABOR-LLM significantly. For PSID81, randomizing gender labels increases perplexity by 1 (about 12% above baseline), while ethnicity has about a quarter of the effect. For NLSY79 and NLSY97 test sets, gender has a similar impact, but ethnicity has a much lower effect. For PSID81, there is substantial additional degradation in performance from the interaction of gender and ethnicity, while NLSY79 sees ethnicity and region having larger effects when randomized jointly rather than individually. For all three survey datasets, the three-way interaction of gender, ethnicity, and region results in the largest impact, with the incremental effect of including all three covariates over two of them is substantial for PSID81 and NLSY79. These findings should be interpreted in light of the historical trends in the labor market participation relevant to the time periods covered by the different survey. Overall, these results suggest that complex interactions are important to consider when building predictive models of occupation, suggesting that simple, additive regressions of the type commonly used in labor market applications may omit important predictors.

[13] Note that this exercise is imperfect; an alternative would be to re-estimate the model omitting a covariate, since a model might increase the loadings on correlated covariates when a particular covariate is omitted. However, re-estimating the model comes with computational cost. Thus, we focus here on exercises that can be carried out without re-estimation.

TABLE 12. Test-set perplexity and perplexity improvement on actual vs. randomized demographic characteristics.

| **Evaluation Dataset** / **Number of Transitions** $(\sum_{i \in \textbf{test}} T_i)$ | PSID81 61,772 | NLSY79 51,593 | NLSY97 29,951 |
|---|---|---|---|
| **Perplexity** | | | |
| No modification / Actual | 8.18 (0.126) | 8.33 (0.147) | 6.35 (0.101) |
| Randomized ethnicity | 8.45 (0.130) | 8.40 (0.148) | 6.39 (0.100) |
| Randomized gender | 9.22 (0.151) | 9.18 (0.167) | 6.90 (0.117) |
| Randomized region | 8.20 (0.126) | 8.38 (0.148) | 6.36 (0.101) |
| Randomized gender and ethnicity | 9.37 (0.152) | 9.23 (0.167) | 6.94 (0.117) |
| Randomized gender and region | 9.29 (0.152) | 9.28 (0.169) | 6.90 (0.117) |
| Randomized ethnicity and region | 8.43 (0.129) | 8.44 (0.149) | 6.39 (0.100) |
| Randomized all variables | 9.44 (0.153) | 9.33 (0.170) | 6.93 (0.117) |
| **Perplexity Improvement** | | | |
| PPL(Randomized ethnicity)-PPL(Actual) | 0.27 (0.012) | 0.07 (0.007) | 0.04 (0.006) |
| PPL(Randomized gender)-PPL(Actual) | 1.04 (0.033) | 0.85 (0.032) | 0.54 (0.027) |
| PPL(Randomized region)-PPL(Actual) | 0.02 (0.004) | 0.05 (0.005) | 0.01 (0.002) |
| PPL(Randomized gender and ethnicity)-PPL(Actual) | 1.19 (0.034) | 0.90 (0.034) | 0.58 (0.027) |
| PPL(Randomized gender and region)-PPL(Actual) | 1.11 (0.034) | 0.95 (0.036) | 0.55 (0.027) |
| PPL(Randomized ethnicity and region)-PPL(Actual) | 0.25 (0.011) | 0.11 (0.008) | 0.04 (0.006) |
| PPL(Randomize all)-PPL(Actual) | 1.25 (0.036) | 1.00 (0.037) | 0.58 (0.027) |

*Note*: The foundation model is FT-7B fine-tuned on the union of the training sets of the surveys without any modification of demographic features. Test-set-bootstrap standard errors are in parentheses.

### 10.3 *The Value of Longer Career Histories*

In this section, we assess the predictive value of observing a worker's full history as recorded in the survey, relative to trucating the history to include only more recent observations. This question helps shed light on the sources of model performance with respect to the ability of the transformer model to capture relevant information from long histories; it also informs survey design, since following individuals over long time periods is expensive.

We proceed by evaluating how the predictive quality of FT-7B fine-tuned on $\mathcal{D}_{\text{all}}^{(\text{train})}$, our best-performing predictive model, changes when we make predictions about $y_{i,t}$ using time-invariant covariates, $x_i$, time-varying covariates and jobs reported in the $k$ *most recent* observations of $\{x_{i,\tau}\}_{\tau=t-k}^{t}$, $\{y_{i,\tau}\}_{\tau=t-k}^{t-1}$, reporting

$$\hat{P}_{\text{LLM}}\left(y_{i,t} \mid x_i, \{x_{i,\tau}\}_{\tau=t-k}^{t}, \{y_{i,\tau}\}_{\tau=t-k}^{t-1}\right).$$

With $k = t-1$, the model has access to all available history. We first create different subsets of individual-year observations from the *test set* of each dataset, defining the following non-overlapping subsets of individual-year observations $S_{t_{\text{min}}<t\leq t_{\text{min}}+5}^{(\text{test})} = \{(i,t) \in \mathcal{D}^{(\text{test})} \mid t_{\text{min}} < t \leq \text{min} + 5\}$ for $t_{\text{min}} \in \{5, 10, 15, 20, 25\}$. The NLSY97 dataset covers a shorter time span, therefore, $S_{20<t\leq 25}^{(\text{test})}$ and $S_{25<t\leq 30}^{(\text{test})}$ are defined as empty sets for NLSY97. Given a $S_{t_{\text{min}}<t\leq \text{min}+5}^{(\text{test})}$, for each observation $(i,t) \in S_{t_{\text{min}}<t\leq \text{min}+5}^{(\text{test})}$, we create text templates consisting of only the $k$ *most recent* observations of individual $i$ prior to her $t^{\text{th}}$ observation: $\text{TMPL}(x_i, \{x_{i,\tau}\}_{\tau=t-k}^{t}, \{y_{i,\tau}\}_{\tau=t-k}^{t-1})$. For values of $k$, we consider multiples of five such that $k \leq t_{\text{min}}$ (e.g., $k \in \{5, 10, 15, 20\}$ if $t_{\text{min}} = 20$). A greater value of $k$ exposes the model to more information about the individual's career history and should lead to an improved prediction accuracy.

We then assess perplexity in the test set for different subsets of constructed test data defined by values of $(k, t_{\text{min}})$:

$$\tilde{S}_{t_{\text{min}}<t\leq t_{\text{min}}+5,k}^{(\text{test})} = \left\{\left(\text{TMPL}(x_i, \{x_{i,\tau}\}_{\tau=t-k}^{t}, \{y_{i,\tau}\}_{\tau=t-k}^{t-1}), y_{i,t}\right)\right\}_{(i,t)\in S_{t_{\text{min}}<t\leq t_{\text{min}}+5}^{(\text{test})}}$$

where each element of $\tilde{S}_{t_{\text{min}}<t\leq t_{\text{min}}+5,k}^{(\text{test})}$ is a pair of (1) a prompt containing $k$ past observations prior to the $t^{\text{th}}$ record of individual $i$ and (2) the ground truth occupation individual $i$ has in her $t^{\text{th}}$ record.

We evaluate our models using the prompt-label pair in *each* $\tilde{S}_{t_{\text{min}}<t\leq t_{\text{min}}+5,k}^{(\text{test})}$ *separately*. Within each $\tilde{S}$ group, we query the likelihood that the language model assigns to the ground truth job title as the continuation of the text prompt, $\hat{P}_{\text{LLM}}(\text{TITLE}(y_{i,t}) \mid \text{TMPL}(x_i, \{x_{i,\tau}\}_{\tau=t-k}^{t}, \{y_{i,\tau}\}_{\tau=t-k}^{t-1}))$, and compute the perplexity

using all predictions within that $\tilde{S}$. Readers can refer to Appendix L for more details and examples.

Finally, we build a matrix of perplexity metrics assessing model's performance under different levels of exposure to past information. Table 13 summarizes model performance when it only has access to a limited number of past observations while predicting the next occupation. To better illustrate the result, we compute the perplexity difference between predictions made using prompts with $k \in \{10, 15, 20, 25\}$ and the baseline predictions made using prompts with $k = 5$. For example, for PSID81, the data in row $t \in (15, 20]$ and column $k = 10$ indicates that predictions made on those observations using $k = 10$ past observations in prompts for transitions indexed between 15 and 20 achieve a perplexity that is 0.19 (with a test-set-bootstrap standard error of 0.026) lower than the perplexity of predictions using $k = 5$ past observations. Truncating the career history thus leads to a significant decrease in predictive performance, although for transitions at the end of a worker's career, most of the predictive benefit is achieved with 10 or 15 years of history.

### 10.4 *Additional Analyses*

In this section, we describe several additional analyses that shed light on the sources of performance improvements. First, Appendix Table M.1 shows the extent to which the embeddings created by FT-7B fine-tuned using PSID81, the largest dataset, incorporate more information about the meaning of job titles. One way to approach this analysis is to assess the predictive power of these embeddings on a task that relates to the interpretation of the titles. We consider a particular task that requires such an understanding: predicting which part of the occupation code hierarchy a particular occupation falls into (this information was not used in LABOR-LLM, although it may have been one part of the enormous pre-training corpus for the original Llama models). We compare the predictions derived from a multinomial logistic regression using as features embeddings extracted from each of the following: FT-7B, off-the-shelf Llama-2-7B, and CAREER. We show that the embeddings from FT-7B have a test-set accuracy of

TABLE 13. Test-set perplexity improvement from increasing number of historical periods used for prediction.

| **PSID81** | $\sum_{i\in\text{test}} T_i$ | $k=10$ | 15 | 20 | 25 |
|---|---|---|---|---|---|
| $t\in(10,15]$ | 9,180 | 0.18 (0.020) | - | - | - |
| $t\in(15,20]$ | 5,288 | 0.19 (0.026) | 0.24 (0.032) | - | - |
| $t\in(20,25]$ | 3,214 | 0.07 (0.015) | 0.08 (0.018) | 0.10 (0.019) | - |
| $t\in(25,30]$ | 1,008 | 0.05 (0.015) | 0.05 (0.017) | 0.05 (0.019) | 0.05 (0.019) |
| **NLSY79** | $\sum_{i\in\text{test}} T_i$ | $k=10$ | 15 | 20 | 25 |
| $t\in(10,15]$ | 9,078 | 0.31 (0.035) | - | - | - |
| $t\in(15,20]$ | 8,051 | 0.37 (0.035) | 0.44 (0.042) | - | - |
| $t\in(20,25]$ | 6,719 | 0.13 (0.017) | 0.17 (0.018) | 0.18 (0.020) | - |
| $t\in(25,30]$ | 2,617 | 0.08 (0.026) | 0.12 (0.028) | 0.13 (0.029) | 0.13 (0.028) |
| **NLSY97** | $\sum_{i\in\text{test}} T_i$ | $k=10$ | 15 | | |
| $t\in(10,15]$ | 7,151 | 0.19 (0.031) | - | - | - |
| $t\in(15,20]$ | 4,112 | 0.11 (0.030) | 0.18 (0.039) | - | - |

*Note*: Each row corresponds to a group of individual-year observations $S^{(\text{test})}_{t_{\min}<t\leq\min+5}$, each column corresponds to a value of $k$, and each cell corresponds to the perplexity improvement due to increasing the number of past observations from 5 to $k$. Test-set-bootstrap standard errors are in parentheses.

78% for predicting the correct SOC group for an occupation, which is somewhat larger than that from off-the-shelf Llama-2-7B and CAREER (76%).

In a second exercise detailed in Appendix M, we characterize the types of transitions in which FT-13B performs better than CAREER for "mover" transitions in the test split of the PSID81 dataset by using features of a transition to predict the gap in the test-set difference in log-likelihood between FT-13B and CAREER. We find that, relative to the quintile of transitions with the smallest performance gain of FT-13B over CAREER, the quintile of transitions with the highest performance gain has the following characteristics: twice as likely to be a transition within the same detailed SOC group; more likely to be a transition between jobs that are similar according to skill descriptions given by O*NET; more likely to have many tokens in both the previous occupation and the target occupation for the transition; more likely to have textually similar job titles; and have a larger average transition index (implying the transition probabilities are conditioned on a longer history).

## 11. Conclusion

This paper proposes a novel approach, LABOR-LLM, to the problem of predicting a worker's next job conditional on history. The best-performing version of this approach, FT-LABOR-LLM, translates the tabular data about a worker's history from publicly available U.S. surveys (PSID and NLSY) into text files that resemble resumes, and then fine-tunes the Llama-2 open-weight foundation models on that corpus. Then, to estimate the probability that the next job is a particular job, say "engineer," given worker history, the approach prompts the fine-tuned LLM with the textual version of the worker's history, and extracts the probability that the LLM assigns to the text "engineer" as the next word. We show that off-the-shelf, without fine-tuning, this approach performs poorly even when OpenAI's API is used. However, the fine-tuning leads this approach to outperform all existing benchmarks. The fine-tuning eliminates the problem of occupation title "hallucinations," but more importantly, it leads the model to make accurate predictions about conditional probabilities in held-out test data. Accurate, fine-grained predictions enable economists to ask and answer more nuanced questions, and to improve the quality of causal inference analyses that rely on accurate predictions.

The paper explores some of the sources of the strong performance of FT-LABOR-LLM, showing that representative data is important, but that adding more data (even non-representative data) can lead a smaller model (in terms of number of parameters) to outperform a larger one. The paper also shows that FT-LABOR-LLM makes use of many years of history, even a worker's early career history, to improve prediction quality. Our results illustrate that the approach can be effective in datasets of moderate size (tens of thousands of transitions), leveraging the general information about jobs embedded in the open-weight LLM's representations of the text of job titles and resumes.

An advantage of the FT-LABOR-LLM approach is that all data and software necessary to apply this approach is available publicly, including the weights of the LLM, so that the main cost in practice is the cost of the computing for fine-tuning and making predictions. Low-cost cloud-based services are available (we used the service provided by Together AI) that enable fine-tuning by simply uploading documents; with these services, no coding is required for the training step, and minimal original coding is required to obtain predictions from the fine-tuned LLM. Thus, researchers can focus on analyzing the results and performing downstream empirical exercises. However, a limitation to this approach is that fine-tuning can become expensive as the dataset size grows, and repeatedly fine-tuning (for example, to bootstrap standard errors) can be prohibitively expensive.

An approach based on publicly available foundation models may also be useful in other settings, for example, any economic prediction problems that involve discrete outcomes with many alternatives and where the alternatives may be associated with meaningful textual descriptions. A sequence of purchases made by a consumer may have a similar structure. Our paper also illustrates the importance of fine-tuning: off-the-shelf LLMs may make plausible sounding predictions, but without fine-tuning they are unlikely to give accurate conditional probabilities for any particular dataset of interest.

# APPENDIX A: DETAILS FOR QUANTIFYING UNCERTAINTY IN PERFORMANCE METRICS

In this appendix, we provide the details of the bootstrapping procedures to create the test-set-bootstrap and the training-set-bootstrap discussed in Section 3.3.

## A.1 *Test Set Bootstrap for Test Set Variations*

The purpose of our bootstrapping approach for the test set is to estimate the sensitivities of our metrics (e.g., perplexities and differences in perplexities) to changes in the test set distribution. In the main paper, we first report the metric (e.g., perplexity) computed using all observations in our test set. Then, we create $B$ bootstrap samples of the test set, sampled on the individual level, to estimate the standard error of the metric. For each bootstrap iteration $b$, we sample individuals in the test set with replacement, then we collect all individual-year observations associated with these sampled individuals and the log-likelihood values assigned to these observations by the model. We use these log-likelihood values to compute the $b^{\text{th}}$ bootstrap value of the metric (e.g., perplexity). After repeating the process above $B$ times, we estimate the standard error using the standard deviation of the $B$ bootstrap values.

We call this procedure the test-set-bootstrap, and report the standard error estimation from the test-set-bootstrap along with our metrics in this paper.

## A.2 *Training Set Bootstrap for Uncertainty Training Set and Training Pipeline*

Similarly, the purpose of our bootstrapping approach for the training set is to quantify the uncertainty in model performance due to training set variation and randomness in the training pipeline, primarily due to data shuffling (according to the support team at Together AI). We create 12 bootstrapped training sets by sampling the pooled training set (i.e., the union of the training splits of PSID81, NLSY79, and NLSY97) with replacement. Let $\mathcal{D}^{\text{(train, s)}}_{\text{mixture}}$ denote the bootstrapped training data of the mixture dataset (sampled with replacement), generated using the random seed $s \in \{0, 1, ..., 11\}$. We fine-tune 12 versions of the Llama-2-7B models using these mixture dataset bootstrapped training sets and evaluate their performance using the *complete* test split of each dataset. We call this procedure the training-set-bootstrap.

To better understand how uncertainty from the training set impacts the comparisons we make in this paper, we also fine-tune 12 versions each of our Llama-2-7B and Llama-2-13B models using the PSID81, our largest survey dataset, subset of each $\mathcal{D}^{(\text{train, s})}_{\text{mixture}}$, denoted by $\mathcal{D}^{(\text{train, s})}_{\text{PSID81}}$.[14] We fine-tune these additional models to make two comparisons. First, we compare the FT-7B fine-tuned on the mixed data to the same model fine-tuned on only one dataset to understand how the training-set-bootstrap impacts our value of information analysis. Second, we compare the Llama-2-7B model fine-tuned on PSID81 to the Llama-2-13B model fine-tuned on the same data to learn how our analysis of smaller versus larger models is impacted by the training-set-bootstrap. The results for all three models and the two comparisons are shown in Table A.1.

We observe that uncertainty from the training-set-bootstrap is lower than the uncertainty from the test-set-bootstrap for perplexity; however, the training-set-bootstrap uncertainty is relatively higher than the test-set-bootstrap uncertainty when it comes to the perplexity differences. Because the computational cost is prohibitive since it involves multiple rounds of large language model fine-tuning, we did not perform the training-set-bootstrap in the main paper.

[14] We did not perform the same exercise for NLSY79 and NLSY97 due to the computational cost.

TABLE A.1. Test-set perplexity for models fine-tuned 12 times, with training-set-bootstrap standard errors.

| **Evaluation Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| **Number of Transitions** $\left(\sum_{i\in\textbf{test}} T_i\right)$ | 61,772 | 51,593 | 29,951 |
| **Perplexity** | | | |
| FT-7B with $\mathcal{D}^{(\text{train, s})}_{\text{mixture}}$ | 8.37 (0.051) | 8.49 (0.058) | 6.36 (0.020) |
| FT-7B with $\mathcal{D}^{(\text{train, s})}_{\text{PSID81}}$ | 8.49 (0.034) | - | - |
| FT-13B with $\mathcal{D}^{(\text{train, s})}_{\text{PSID81}}$ | 8.46 (0.021) | - | - |
| **Perplexity Improvement** | | | |
| PPL(FT-13B with $\mathcal{D}^{(\text{train, s})}_{\text{PSID81}}$) - PPL(FT-7B with $\mathcal{D}^{(\text{train, s})}_{\text{mixture}}$) | 0.09 (0.055) | - | - |
| PPL(FT-7B with $\mathcal{D}^{(\text{train, s})}_{\text{PSID81}}$) - PPL(FT-13B with $\mathcal{D}^{(\text{train, s})}_{\text{PSID81}}$) | 0.03 (0.029) | - | - |

*Note*: Numbers in parentheses show the standard deviation of metrics (i.e., perplexity or perplexity difference) computed using 12 random seeds, i.e., the training-set-bootstrap standard error. These standard deviations measure *solely* uncertainties due to training set variation and randomness in the training pipeline.

## APPENDIX B: DETAILS OF THE CAREER MODEL

The CAREER model leverages a large-scale dataset of online resumes, which covers 24 million workers. The data is passively collected from online resume platforms, ensuring a broad and diverse representation of career paths across various industries and job roles.

*Resume Dataset for Pre-training* We use the exact same data processing as Vafa et al. (2024) to construct the resume dataset for pre-training CAREER.[15] First, we convert each resume in the dataset into a chronological sequence of entries with the occupation (Standard Occupational Code, or SOC), starting year, and ending year. If an individual worked in multiple occupations in a single year (i.e., there are overlapping records on the resume), we select the one in which they spent the most time; in cases of equal time spent, we choose the occupation that started earlier in their career. We convert the SOC codes to `occ1990dd` codes using a crosswalk from Destin Royer to match the occupation codes in our survey datasets. The survey datasets also distinguish between non-employed categories

[15] Readers can refer to Appendix F in Vafa et al. (2024) for additional details on dataste construction.

(unemployed, out of labor force, or student), but these categories were absent in the resumes data. When the year associated with an occupation was missing, we exclude it from the dataset as we cannot determine its position in an individual's career timeline. We link each occupation to the individual's most recent educational degree, categorized into one of eight groups: high school diploma, some college, bachelor's degree, graduate degree, certificate, license, and diploma.

In addition to the dynamic variables, we use two static variables imputed based off other data in the resume: gender and location. Locations are classified into the 50 U.S. states, Puerto Rico, Washington D.C., and an "unknown" category for cases where the location could not be imputed; however, we grouped states into four regions (northeast, north central, south, west), with a fifth "other" region for Puerto Rico and missing states, to match the survey datasets. We replaced any missing values for these static variables with a special "missing" token.

This pre-processing results in a dataset containing 23.7 million resumes and 245 million individual-year observations (i.e., transitions).

*Survey Datasets for Fine-Tuning* We construct our own copies of the survey datasets to fine-tune CAREER models in this paper. Appendix N provides the details on how we process our survey datasets. We do not use birth year information in survey datasets while fine-tuning the CAREER model because the CAREER model was not designed to handle continuous variables.

*Model Architecture* Following Vafa et al. (2024), we deploy a CAREER transformer model with 12 layers, 192 embedding dimensions, 3 attention heads, and 768 hidden units in this paper. In total, this resulted in 5,553,984 parameters for the full CAREER model.

*Model Estimation* The estimation procedure of CAREER consists of two stages: (1) pre-training using resume datasets, and (2) fine-tuning using survey datasets. We use CAREER's official repository for model estimation; a copy of the repository has been included in our replication material.

The pre-training uses the Adam optimizer with $\beta$ parameters (0.9, 0.98), weight decay of 0.01, and no gradient clipping. The learning rate starts at $10^{-7}$ with a scheduler that follows an inverse square root decay, warming up over 4,000 updates to a peak of 0.0005. Training samples have a maximum token length of 512, using end-of-sequence (EOS) tokens to define breaks. Each batch contains up to 16,000 tokens, with updates performed every batch, targeting 85,000 updates in total. Model checkpoints are saved every 1,000 updates to a specified directory, and the model checkpoint with the best validation loss is recorded. We fine-tune the pre-trained model checkpoints with the lowest validation loss using a survey dataset. The Adam optimizer is used with $\beta$ parameters (0.9, 0.98), a weight decay of 0.01, and no gradient clipping. The learning rate starts at $10^{-7}$ and warms up over 500 updates to 0.0001, following an inverse square root decay scheduler. Each sample contains a maximum of 512 tokens, with end-of-sequence (EOS) tokens used for defining breaks, and each batch can include up to 16,000 tokens. When fine-tuning the survey dataset, we train the model until it overfits according to the validation loss. Finally, we evaluate the model performance using the checkpoint with the best validation performance. Both the pre-training and fine-tuning use mixed precision (FP16) for computational efficiency.

The model estimation pipeline is performed for each survey dataset separately with the random seed fixed.

## Appendix C: Details for Text Template

This appendix describes the text template in more detail. The text template starts with a preamble that describes the individual's static covariates, then lists the individual's education level and occupation for each calendar year. Specifically:

1. The first line describes the source of the data, e.g., `<A worker from the PSID dataset>`.

2. The second line describes the individual's demographic characteristics, e.g., `The following information is available about the work history of a female black or african american US worker residing in the south region.`

3. The third line describes the individual's birth year, e.g., The worker was born in 1963. Recall that the original CAREER model does not incorporate age or birth year information. Therefore, we do not include this line of information in the text template while comparing it to the CAREER model.

4. The fourth line describes the structure of the resume, i.e., The worker has the following records of work experience, one entry per line, including year, education level, and the job title:. This line is constant for all individuals and is useful for the LLM to understand the format of the subsequent rows of work experience.

5. Starting from the fifth line, each line summarizes the information of the worker from a wave of the survey she participated in, including the calendar year, education level, and title of her main occupation reported in that survey year. Specifically, it is in the format YEAR (EDUCATION): JOB TITLE, e.g., 1984 (some college): Cooks.

6. The template ends with the line <END OF DATA>.

The following example shows a complete text template of an individual worker. For more examples, see Online Appendix E.

```
<A worker from the PSID dataset>
The following information is available about the work history of a female
↪ black or african american US worker residing in the south region.
The worker was born in 1963.
The worker has the following records of work experience, one entry per
↪ line, including year, education level, and the job title:
1984 (some college): Cooks
1985 (some college): Food servers, nonrestaurant
1986 (some college): Cleaners of vehicles and equipment
1988 (some college): Food servers, nonrestaurant
1989 (some college): Bus drivers
1990 (some college): Food servers, nonrestaurant
1991 (some college): Unemployed
1992 (some college): Painting workers
1993 (some college): Painting workers
```

```
1994 (some college): Court, municipal, and license clerks
1996 (some college): Septic tank servicers and sewer pipe cleaners
<END OF DATA>
```

The survey dataset may have missing data for certain individuals in some years, as described in Appendix N. This missingness can occur if a worker did not respond to a particular wave of the survey but participated in later waves. Additionally, some surveys, such as the NLSY and PSID, have transitioned from annual to biennial surveys in recent years, resulting in gaps for certain years. The text template only has rows corresponding to the years when the individual was observed.

### C.1 *Template with Numerical Job Titles*

In Section C.1, we use a version of the text template that represents career trajectories with numerical job titles. Instead of using the actual job title such as `Cashiers`, the numerical template uses job titles like `job_144`. Here is an example:

```
<A worker from the PSID dataset>
The following information is available about the work history of a female
↪  white US worker residing in the west region.
The worker was born in 1985.
The worker has the following records of work experience, one entry per
↪  line, including year, education level, and the job title:
2007 (college): job_144
2009 (college): job_169
2011 (college): job_089
2013 (college): job_304
2015 (college): job_304
2017 (college): job_304
2021 (college): job_169
<END OF DATA>
```

## APPENDIX D: DETAILS FOR OBTAINING THE PROBABILITY ASSIGNED TO A TOKEN

In this appendix, we explain the details of to directly leverage LLMs' next token prediction capabilities to predict future occupations using job titles described in Section 4.3. To obtain the predicted probability of the next occupation, we first tokenize each job title, $\text{title}_y$, into a sequence of tokens. Suppose the string $\text{title}_y$ is tokenized into $n$ tokens $\{\text{token}_y^{(1)}, \text{token}_y^{(2)}, \ldots, \text{token}_y^{(n)}\}$. Then, the unnormalized probability of predicting $y$ is the likelihood the language model assigns to the token sequence $\{\text{token}_y^{(1)}, \text{token}_y^{(2)}, \ldots, \text{token}_y^{(n)}\}$ as the continuation of the text representation $\text{TMPL}(x_{i,\leq t}, y_{i,<t})$. The predicted probability can further be expanded using the chain rule of probability, as shown in Equation (5).

$$
\begin{aligned}
&\hat{P}^{\mathcal{V}}_{\text{LLM}}(\text{TOK}(\text{TITLE}(y)) \mid \text{TMPL}(x_{i,\leq t}, y_{i,<t})) \\
&= \hat{P}^{\mathcal{V}}_{\text{LLM}}(\{\text{token}_y^{(1)}, \text{token}_y^{(2)}, \ldots, \text{token}_y^{(n)}\} \mid \text{TMPL}(x_{i,\leq t}, y_{i,<t})) \\
&= \prod_{j=1}^{n} \hat{P}^{\mathcal{V}}_{\text{LLM}}(\text{token}_y^{(j)} \mid \text{TMPL}(x_{i,\leq t}, y_{i,<t}), \text{token}_y^{(1)}, \text{token}_y^{(2)}, \ldots, \text{token}_y^{(j-1)})
\end{aligned}
\tag{5}
$$

The $\hat{P}^{\mathcal{V}}_{\text{LLM}}(\text{token}_y^{(j)} \mid \text{TMPL}(x_{i,\leq t}, y_{i,<t}), \text{token}_y^{(1)}, \text{token}_y^{(2)}, \ldots, \text{token}_y^{(j-1)})$ is operationalized by (1) appending all tokens $\text{token}_y^{(1)}, \text{token}_y^{(2)}, \ldots, \text{token}_y^{(j-1)}$ to the text representation $\text{TMPL}(x_{i,\leq t}, y_{i,<t})$ and (2) querying the likelihood the language model assigned to $\text{token}_y^{(j)}$ as the next token conditioned on all the previous tokens.

For example, the title "software engineer" may be tokenized into two tokens, one for "software" $\in \mathcal{V}_{\text{LLM}}$ and one for "engineer" $\in \mathcal{V}_{\text{LLM}}$.[16] Equation (6) illustrates how to obtain the conditional probability assigned to "software engineer".

$$
\begin{aligned}
&\hat{P}^{\mathcal{V}}_{\text{LLM}}(\text{“software engineer”} \mid \text{prompt tokens}) \\
&= \hat{P}^{\mathcal{V}}_{\text{LLM}}(\text{“software”} \mid \text{prompt tokens}) \hat{P}^{\mathcal{V}}_{\text{LLM}}(\text{“engineer”} \mid \text{prompt tokens}, \text{“software”})
\end{aligned}
\tag{6}
$$

It is worth noting that we cannot guarantee that the model only assigns positive probabilities to valid job titles. In fact, given the presence of the softmax function in our language model, $\hat{P}^{\mathcal{V}}_{\text{LLM}}(\cdot \mid \text{TMPL}(x_{i,\leq t}, y_{i,<t}))$ is strictly positive for any sequence of tokens of any length. Therefore, the sum of all possible job titles' probabilities is not necessarily one. We would need the following normalization

[16]This is for illustration purposes only, how the LLM's tokenizer splits the phrase "software engineer" depends on the exact LLM used.

to calculate the probability of predicting $y_t$ so that predicted probabilities on all job titles sum to one.

$$\hat{P}_{\text{LLM}}^{\text{normalized}}(y_{i,t} \mid x_{i,\le t}, y_{i,<t}) = \frac{\hat{P}_{\text{LLM}}^{\mathcal{V}}(\textsc{Tok}(\textsc{Title}(y)) \mid \textsc{Tmpl}(x_{i,\le t}, y_{i,<t}))}{\sum_{y' \in \mathcal{Y}} \hat{P}_{\text{LLM}}^{\mathcal{V}}(\textsc{Tok}(\textsc{Title}(y')) \mid \textsc{Tmpl}(x_{i,\le t}, y_{i,<t}))} \tag{7}$$

The normalization operation in Equation (7) is computationally expensive, since we need to perform LLM inference $|\mathcal{Y}|$ times. In this paper, we do not perform this normalization and we use the predicted probability from Equation (5) directly. It is worth noting that since the denominator in Equation (7) is less than one (since the total probability mass on the subset of job title tokens is less than the total probability mass on all tokens), $\hat{P}_{\text{LLM}}^{\mathcal{V}} \le \hat{P}_{\text{LLM}}^{\text{normalized}}$. As a result, test perplexity for LLMs reported in the paper *under-estimates* the performance of these LLMs.

## Appendix E: Details on Embedding-Based Approach

This appendix provides the details of the embedding-based approach reported on in Section 7.1. To extract embeddings from the Llama models (fine-tuned and off-the-shelf), we use the final-layer model representation of each model. For OpenAI embeddings, we used the latest `text-embedding-3-large` model at the time the analysis was conducted (November $12^{\text{th}}$, 2024); details are available at https://platform.openai.com/docs/guides/embeddings.

We estimate the multinomial logistic regression using Bayesian Optimization to find the optimal learning rate in the log-uniform space $[10^{-6}, 10^{-2}]$. The embeddings are high-dimensional with thousands of dimensions. We also explore using embeddings of 16, 64, or 256 dimensions, using PCA to reduce our embeddings, in addition to the full-dimensional embeddings, and pick the best-performing model from our validation set.[17]

[17] We explore random forest with 50 Bayesian Optimization calls and uniform parameters $[20, 400]$ estimators, $[5, 50]$ maximum depth, $[0.01, 0.9]$ minimum samples split, $[0.01, 0.9]$ minimum samples leaf. Performance is significantly worse than multinomial logistic regression.

## APPENDIX F: DETAILS ON FULL-PRECISION VERSUS QUANTIZATIZED MODELS

Model quantization is a technique for improving models' computational efficiency and decreasing memory usage by reducing the numerical precision of model parameters (e.g., from 32-bit to 8-bit or 4-bit). Existing research has shown that LLMs with quantization can achieve similar performance to full-precision models Dettmers et al. (2023). We fine-tune the Llama-2-7B model under full precision using Together AI's platform, but we quantize model weights to 8-bit before conducting experiments for LLM inference in the main paper to save computational resources.

In this appendix, we compare the performance of the full-precision and 8-bit quantization versions of the FT-7B. Specifically, we take the FT-7B that was fine-tuned under full precision; then, we query predicted probabilities of future job titles using the two variants of the fine-tuned model, one in full precision and the other quantized to 8-bit. Table F.1 compares models' performance on different datasets. These results suggest no significant difference between the full-precision and quantized models in terms of predictive performance.

It is extremely challenging for an individual researcher to obtain the hardware for full-precision fine-tuning (e.g., >112GiB of GPU memory for 7B). Fine-tuning on quantized models would require additional tricks like LoRA because one cannot run back-propogation on quantized parameters directly. Different LoRA techniques lead to different model performance, but exploration of these techniques is beyond the scope of this paper. We highly recommend researchers to out-source the model fine-tuning part to a third-party due to the engineering complexity (e.g., training on multiple GPUs). We quantize the model during inference to speed up the inference and save GPU memory.

TABLE F.1. Test-set perplexity of full-precision versus quantized (8-bit) FT-7B.

| **Evaluation Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| **Number of Transitions** $(\sum_{i \in \textbf{test}} T_i)$ | 61,772 | 51,593 | 29,951 |
| FT-7B 8-bit Quantized Inference | 8.18 (0.126) | 8.33 (0.147) | 6.35 (0.101) |
| FT-7B Full Precision Inference | 8.16 (0.126) | 8.31 (0.147) | 6.34 (0.100) |

*Note*: FT-7B was fine-tuned using full precision. Test-set-bootstrap standard errors are in parentheses. Prompts of LLMs include birth year information in this table.

## APPENDIX G: ADDITIONAL RESULTS FOR IMPROVING OFF-THE-SHELF LLMS USING PROMPT ENGINEERING

As described in Section 7.3, we evaluate the value of adding example resumes (i.e., in-context learning examples) versus job titles to inform the off-the-shelf LLM model of either our data structure or the prediction space, respectively. Because the Llama-2 model family has a context length of 4,096, meaning the model can only effectively process prompts shorter than 4,096 tokens, there is a limit to how many example resumes can be included in our enriched prompts with in-context learning information. In our dataset, one resume is up to 900 tokens, and the list of job titles is more than 3,200 tokens long, so we cannot include even one resume in combination with all job titles for some models (Llama-2-7B and Llama-2-13B). To evaluate the performance of the job titles combined with example resumes, we additionally deploy a variant of the Llama-2 model with a 32k context length, Llama-3.1 with a 128k context length, and Llama-3.2 models with a 128k context length. Online Appendix B provides more details on the token counts of prompts in our datasets.

We expand the results in Table 5 by showing the results from including one, three, and five example resumes, either with or without job titles, in addition to the results for zero and ten example resumes, in Table G.1. Prompts in this table include birth year information to help pre-trained models better understand the population of workers in our survey datasets. The inclusion of job titles in the prompt performs as well or better than the inclusion of up to three to five resumes for all models.

TABLE G.1. Test-set perplexity for off-the-shelf models with in-context learning examples and/or job titles - expanded.

| | **Evaluation Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|---|
| | **Number of Transitions** $(\sum_{i \in \text{test}} T_i)$ | 6,177 | 5,159 | 2,995 |
| **Models Without Job Titles in Prompt** | **# Resumes** | | | |
| OTS Llama-2-13b | 0 | 137.68 (10.699) | 122.73 (9.676) | 107.08 (10.788) |
| OTS Llama-2-13b | 1 | 49.30 (3.430) | 44.80 (3.352) | 25.17 (2.480) |
| OTS Llama-2-13b | 3 | 35.99 (2.267) | 27.70 (1.785) | 18.91 (1.772) |
| OTS Llama-2-7b-32k | 0 | 241.04 (22.812) | 182.75 (16.373) | 173.94 (22.880) |
| OTS Llama-2-7b-32k | 1 | 81.78 (6.048) | 65.45 (4.990) | 34.83 (3.844) |
| OTS Llama-2-7b-32k | 3 | 53.50 (3.561) | 38.25 (2.539) | 24.86 (2.634) |
| OTS Llama-2-7b-32k | 5 | 45.27 (2.753) | 31.64 (1.993) | 21.88 (2.153) |
| OTS Llama-2-7b-32k | 10 | 36.53 (2.131) | 26.20 (1.495) | 17.52 (1.510) |
| OTS Llama-2-7b | 0 | 356.33 (27.380) | 293.28 (21.387) | 252.70 (27.979) |
| OTS Llama-2-7b | 1 | 60.96 (4.290) | 48.85 (3.467) | 28.13 (2.924) |
| OTS Llama-2-7b | 3 | 40.20 (2.532) | 29.36 (1.818) | 20.18 (2.016) |
| OTS Llama-3.1-8B | 0 | 127.79 (10.564) | 110.87 (8.973) | 99.16 (11.408) |
| OTS Llama-3.1-8B | 1 | 53.39 (3.744) | 43.27 (3.013) | 25.44 (2.346) |
| OTS Llama-3.1-8B | 3 | 35.29 (2.173) | 26.44 (1.567) | 17.93 (1.569) |
| OTS Llama-3.1-8B | 5 | 30.07 (1.725) | 22.43 (1.246) | 16.11 (1.324) |
| OTS Llama-3.1-8B | 10 | 25.08 (1.385) | 19.41 (1.009) | 13.68 (1.034) |
| OTS Llama-3.2-1B | 0 | 456.09 (51.012) | 371.33 (38.769) | 277.73 (40.961) |
| OTS Llama-3.2-1B | 1 | 165.56 (15.246) | 133.29 (12.842) | 72.93 (10.011) |
| OTS Llama-3.2-1B | 3 | 92.49 (7.515) | 62.27 (5.065) | 41.71 (5.218) |
| OTS Llama-3.2-1B | 5 | 71.80 (5.350) | 47.56 (3.620) | 34.38 (4.023) |
| OTS Llama-3.2-1B | 10 | 52.90 (3.740) | 36.04 (2.409) | 24.99 (2.631) |
| OTS Llama-3.2-3B | 0 | 165.11 (14.493) | 134.39 (11.186) | 122.58 (14.671) |
| OTS Llama-3.2-3B | 1 | 64.36 (4.575) | 54.94 (3.970) | 31.22 (3.152) |
| OTS Llama-3.2-3B | 3 | 44.06 (2.808) | 33.63 (2.156) | 22.27 (2.125) |
| OTS Llama-3.2-3B | 5 | 37.32 (2.236) | 28.05 (1.729) | 19.89 (1.815) |
| OTS Llama-3.2-3B | 10 | 29.92 (1.726) | 22.95 (1.306) | 16.21 (1.334) |
| **Models With Job Titles in Prompt** | **# Resumes** | | | |
| OTS Llama-2-13b | 0 | 33.35 (1.913) | 33.78 (1.935) | 28.35 (1.987) |
| OTS Llama-2-7b-32k | 0 | 42.01 (2.522) | 45.72 (2.678) | 47.95 (4.127) |
| OTS Llama-2-7b-32k | 1 | 28.28 (1.459) | 26.04 (1.225) | 16.25 (1.118) |
| OTS Llama-2-7b-32k | 3 | 24.00 (1.128) | 20.78 (0.868) | 13.52 (0.897) |
| OTS Llama-2-7b-32k | 5 | 22.57 (1.046) | 19.58 (0.822) | 12.74 (0.839) |
| OTS Llama-2-7b-32k | 10 | 20.73 (0.918) | 18.04 (0.732) | 11.74 (0.736) |
| OTS Llama-2-7b | 0 | 36.91 (2.135) | 33.14 (1.760) | 31.46 (2.400) |
| OTS Llama-3.1-8B | 0 | 30.85 (1.633) | 26.98 (1.309) | 21.91 (1.394) |
| OTS Llama-3.1-8B | 1 | 22.12 (1.102) | 20.43 (0.921) | 13.90 (0.912) |
| OTS Llama-3.1-8B | 3 | 19.17 (0.912) | 16.95 (0.726) | 11.86 (0.769) |
| OTS Llama-3.1-8B | 5 | 17.86 (0.828) | 16.02 (0.676) | 11.35 (0.742) |
| OTS Llama-3.1-8B | 10 | 16.45 (0.763) | 15.20 (0.631) | 10.49 (0.672) |
| OTS Llama-3.2-1B | 0 | 62.23 (3.885) | 53.31 (3.068) | 45.25 (3.518) |
| OTS Llama-3.2-1B | 1 | 35.44 (1.880) | 31.67 (1.663) | 20.85 (1.630) |
| OTS Llama-3.2-1B | 3 | 28.72 (1.431) | 24.56 (1.163) | 17.00 (1.248) |
| OTS Llama-3.2-1B | 5 | 26.03 (1.280) | 22.70 (1.057) | 15.98 (1.155) |
| OTS Llama-3.2-1B | 10 | 22.95 (1.130) | 20.25 (0.913) | 14.02 (0.990) |
| OTS Llama-3.2-3B | 0 | 39.81 (2.199) | 39.24 (2.227) | 35.44 (2.700) |
| OTS Llama-3.2-3B | 1 | 24.78 (1.204) | 23.28 (1.091) | 14.84 (0.987) |

## APPENDIX H: DETAILS ON FINE-TUNING

This section discusses the details of fine-tuning LLMs in this paper and additional results showing how the number of epochs, i.e., complete passes through the entire training dataset during the training process, impacts model performance.

For each individual $i$ in the training split, we construct a text representation of her complete career history $\text{TMPL}(x_{i,\leq T_i}, y_{i,\leq T_i})$ as described in Section 4.2. We use these text representations as the corpus to fine-tune the language models. During the fine-tuning process, the model is trained to predict the next token in each $\text{TMPL}(x_{i,\leq T_i}, y_{i,\leq T_i})$ in the training corpus conditioned on the previous tokens. The loss function not only considers the model's prediction on tokens corresponding to job titles, but also on tokens corresponding to everything else in the text representation to improve models' understanding of our text template data structure. We use $\text{TMPL}(x_{i,\leq T_i}, y_{i,\leq T_i})$ from individuals in the validation split to evaluate the performance of the fine-tuned models after each fine-tuning epoch.

For each model reported in the paper, we deploy two different training strategies: full-parameter automated mixed precision fine-tuning for three epochs (where in the context of fine-tuning, an epoch is a single complete pass through a dataset) and the same for five epochs. During the fine-tuning, we evaluate the model's validation loss after each training epoch, and keep the model checkpoint (saved snapshot of a model's parameters) that attains the lowest validation loss for evaluation. All models in this paper were fine-tuned using the two strategies mentioned, and we always report the model from the better-performing strategy.

Consider now some additional details about fine-tuning, which mirrors the pre-training process. First, note that our description of CAREER in Section 5.3 gives a high-level overview of the functional form of a transformer model, where the "vocabulary" of CAREER is jobs instead of tokens from English words. Now consider estimation details. In current practice, LLMs are usually trained so that the parameters of the transformer neural network maximize log-likelihood, which in the case of language models, where outcomes are encoded as indicator variables for tokens, is equivalent to minimizing cross-entropy loss (Touvron et al. (2023)). In stochastic gradient descent, observations are grouped into small batches. Given parameter estimates from prior batches, within each new batch, the gradient of the loss with respect to the parameters is evaluated for each observation in the batch (where the gradient is evaluated at the previous parameter estimates). The parameters are then updated using an adaptive version of stochastic gradient descent where updates are made using moving averages; see Touvron et al. (2023) for details.

In our fine tuning, we use a batch size of 32, the initial learning rate of $10^{-5}$ (which determines the step size for each update of model parameters), and a linear learning rate decay (which determines how the learning rate changes across epochs, see e.g., Jin et al. (2023)) from the initial learning rate to zero learning rate. Such learning rate scheduling of linear decaying is enforced by Together AI, and we do not have control over it at the time of fine-tuning. It is worth noting that given the linear learning rate decay, the checkpoints corresponding to the first three epochs in the three epoch settings are different from the first three epochs in the five epoch settings.

We also experiment with fine-tuning the model for more epochs while taking the checkpoint corresponding to the lowest validation loss. We observed escalating validation loss (i.e., over-fitting) after four to five epochs. Due to the prohibitive computational cost, we only fine-tuned Llama-2-7B models using the pooled training data for five (reported in the main paper), six, eight, and ten epochs. Table H.1 summarizes the perplexities of the best model checkpoint, according to the validation loss, in these settings. We do not observe significant improvement in model performance, if any, while fine-tuning the model for more epochs.

## Appendix I: Model Performance by Different Education Groups

In this appendix, we explore how models perform on different subgroups defined by educational backgrounds to evaluate whether the main results of our paper are consistent across subpopulations. First, Table I.1 presents the perplexity dif-

TABLE H.1. Test-set perplexity of FT-7B fine-tuned for 5, 6, 8, or 10 epochs.

| **Evaluation Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| **Number of Transitions** $(\sum_{i \in \textbf{test}} T_i)$ | 61,772 | 51,593 | 29,951 |
| FT-7B Best Checkpoint of 5 Epochs | 8.08 (0.124) | 8.21 (0.146) | 6.19 (0.097) |
| FT-7B Best Checkpoint of 6 Epochs | 8.12 (0.125) | 8.22 (0.147) | 6.21 (0.096) |
| FT-7B Best Checkpoint of 8 Epochs | 8.10 (0.124) | 8.19 (0.145) | 6.19 (0.098) |
| FT-7B Best Checkpoint of 10 Epochs | 8.14 (0.124) | 8.24 (0.147) | 6.22 (0.098) |

*Note*: FT-7B model is trained on the union of the three survey datasets. Test-set-bootstrap standard errors are in parentheses.

ferences between FT-7B-NBY, FT-13B-NBY, and CAREER on different subgroups and datasets. Specifically, we group individual-year observations $(i, t)$ based on education level, then compare perplexities of FT-LABOR-LLM and CAREER on these subsets of observations separately. Note that education level can change throughout an individual's career history so different observations of the same individual can belong to different education subgroups. Table I.1 indicates that our language-based approach consistently outperforms the previous state-of-the-art model for different subpopulations.

TABLE I.1. Test-set perplexity by different education groups.

| Dataset | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| **Subgroup with College Degree** | $\sum_{i \in \text{test}} T_i = 30,920$ | $\sum_{i \in \text{test}} T_i = 19,204$ | $\sum_{i \in \text{test}} T_i = 5,898$ |
| PPL(CAREER)-PPL(FT-7B-NBY) | 0.25 (0.026) | 0.29 (0.040) | -0.14 (0.075) |
| PPL(CAREER)-PPL(FT-13B-NBY) | 0.29 (0.027) | 0.35 (0.042) | -0.04 (0.074) |
| **Subgroup without College Degree** | $\sum_{i \in \text{test}} T_i = 30,852$ | $\sum_{i \in \text{test}} T_i = 32,389$ | $\sum_{i \in \text{test}} T_i = 24,053$ |
| PPL(CAREER)-PPL(FT-7B-NBY) | 0.22 (0.024) | 0.22 (0.026) | 0.04 (0.015) |
| PPL(CAREER)-PPL(FT-13B-NBY) | 0.28 (0.025) | 0.24 (0.026) | 0.08 (0.014) |

*Note*: Test-set-bootstrap standard errors are in parentheses.

Next, we consider measures of performance based on the problem of predicting whether a worker changes occupations. Figure I.1 depicts the calibration plots for FT-7B-NBY, OTS-7B-NBY, CAREER, and empirical transition probability of predicting moving from different education subgroups and datasets. Our experiment results indicate that FT-LABOR-LLM is consistently better calibrated than CAREER across subpopulations.

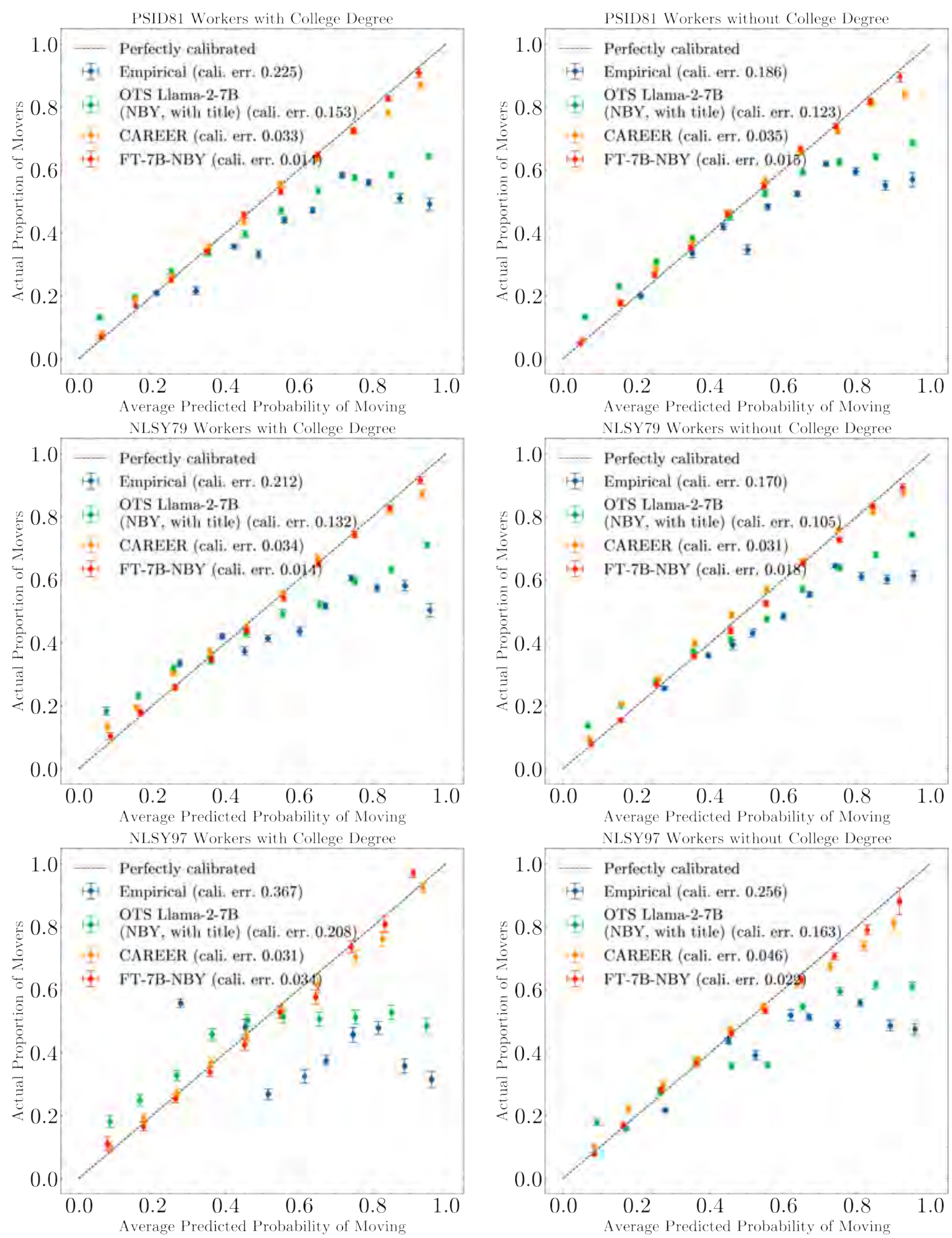



FIGURE I.1. Calibration plots for predicting moving by different education groups.

Finally, Table I.2 presents the AUC-ROC performance metric for the empirical transitions frequency model, off-the-shelf Llama-2-7B-NBY with job titles included in the prompt, FT-7B-NBY model, and CAREER model from predicting moving in different education subgroups and datasets. Again, our results indicate that FT-LABOR-LLM consistently outperforms or achieves comparable performance to CAREER across subpopulations.

TABLE I.2. Area Under the ROC Curve (AUC-ROC) by different education groups.

| | PSID81 | | NLSY79 | | NLSY97 | | Aggregated | |
|---|---|---|---|---|---|---|---|---|
| Has College Degree | Yes | No | Yes | No | Yes | No | Yes | No |
| Empirical | 0.640 | 0.667 | 0.588 | 0.662 | 0.441 | 0.648 | 0.599 | 0.663 |
| OTS Llama-2-7B-NBY (with job titles) | 0.709 | 0.719 | 0.689 | 0.729 | 0.617 | 0.694 | 0.693 | 0.717 |
| CAREER | 0.778 | 0.778 | 0.762 | 0.785 | 0.749 | 0.762 | 0.770 | 0.778 |
| FT-7B-NBY | 0.783 | 0.784 | 0.772 | 0.794 | 0.741 | 0.762 | 0.776 | 0.784 |

## APPENDIX J: ADDITIONAL RESULTS ON GAP YEAR PREDICTION

This section provides additional analyses of FT-LABOR-LLM's prediction behavior when there is a gap between the calendar years of the current job and the previous job in a transition. Let $\text{year}_{i,t}$ denote the calendar year of the $t^{\text{th}}$ transition of individual $i$, where $\text{year}_{i,t-1}$ is the calendar year of the previous transition (only defined for $t > 1$). Specifically, we focus on transitions with $t > 1$ such that $\text{year}_{i,t} = \text{year}_{i,t-1} + 2$, i.e., the gap size is exactly one calendar year, to reduce computational resource requirements. To create the dataset, we randomly sample 500 transitions from the test split of each survey dataset, resulting in a total of 1,500 transitions.

Using the FT-LABOR-LLM model fine-tuned on the mixture training data, we compute the predicted probability of landing at job $y_{i,t}$ in calendar year $\text{year}_{i,t}$ as:

$$\hat{P}(y_{i,t} \text{ in } \text{year}_{i,t} \mid y_{i,t-1} \text{ in } \text{year}_{i,t} - 2),$$

where covariates $x_{i,\leq t}$ and past jobs $y_{i,<t-1}$ are omitted in the conditional part for simplicity. This is referred to as the **direct prediction**. We also compute the

**compound prediction** as:

$$\sum_{y' \in \mathcal{Y}} \hat{P}(y_{i,t} \text{ in year}_{i,t} \mid y' \text{ in year}_{i,t}-1 \wedge y_{i,t-1} \text{ in year}_{i,t}-2) \times \hat{P}(y' \text{ in year}_{i,t}-1 \mid y_{i,t-1} \text{ in year}_{i,t}-2).$$

Computing the compound prediction for a single transition requires approximately $2 \times |\mathcal{Y}| \approx 700$ model inferences, making this experiment computationally expensive.

Finally, we compare the agreement between the direct prediction and the compound prediction using the 1,500 transitions; the log probabilities are found to be highly correlated, with a correlation coefficient of $0.93$.

## APPENDIX K: ADDITIONAL RESULTS FOR THE VALUE OF INFORMATION

In this section, we report on a complementary exercise to that conducted in Table 9 of the main paper. Instead of either fine-tuning on a single dataset or the union of all datasets, we start from each baseline survey training dataset and create new training datasets that mix in additional data from the other two surveys. Specifically, we take the training split of dataset $\omega$, $\mathcal{D}_{\omega}^{(\text{train})}$ and mix it with $P\% \times |\mathcal{D}_{\omega}^{(\text{train})}|$ additional training samples from training splits of the other two datasets $\mathcal{D}_{\omega'}^{(\text{train})} \cup \mathcal{D}_{\omega''}^{(\text{train})}$. We fine-tune Llama-2-7B models using the merged training data, and then evaluate the model's performance on the test split $\mathcal{D}_{\omega}^{(\text{test})}$.

Table K.1 summarizes the performance of these models fine-tuned with additional training data; adding sufficient non-representative data leads to improvements over the models fine-tuned with only data representative of the test set.

TABLE K.1. Test-set perplexity of fine-tuning model on full training split plus $P\%$ training data from other sources.

| **Evaluation Dataset** | PSID81 | NLSY79 | NLSY97 |
|---|---|---|---|
| **Number of Transitions** $\left(\sum_{i \in \textbf{test}} T_i\right)$ | 61,772 | 51,593 | 29,951 |
| **Perplexity** | | | |
| FT-7B with $P = 0$ | 8.18 (0.126) | 8.33 (0.147) | 6.35 (0.101) |
| FT-13B with $P = 0$ | 8.14 (0.126) | 8.28 (0.145) | 6.33 (0.100) |
| FT-7B with $P = 10$ | 8.18 (0.1272) | 8.32 (0.147) | 6.33 (0.099) |
| FT-7B with $P = 30$ | 8.11 (0.1242) | 8.29 (0.147) | 6.29 (0.099) |
| FT-7B with $P = 50$ | 8.09 (0.1232) | 8.28 (0.148) | 6.28 (0.098) |
| FT-7B with $P = 70$ | 8.09 (0.1242) | 8.27 (0.146) | 6.26 (0.099) |
| **Perplexity Improvement** | | | |
| PPL(FT-13B)-PPL(FT-7B with $P = 10$) | -0.04 (0.014) | -0.03 (0.013) | -0.01 (0.010) |
| PPL(FT-13B)-PPL(FT-7B with $P = 30$) | 0.03 (0.014) | -0.01 (0.012) | 0.03 (0.010) |
| PPL(FT-13B)-PPL(FT-7B with $P = 50$) | 0.05 (0.013) | 0.00 (0.013) | 0.05 (0.010) |
| PPL(FT-13B)-PPL(FT-7B with $P = 70$) | 0.05 (0.014) | 0.02 (0.013) | 0.07 (0.010) |

*Note*: Test-set-bootstrap standard errors are in parentheses.

## APPENDIX L: DETAILS ON THE VALUE OF LONGER CAREER HISTORIES

In this appendix, we provide additional details on our experiment evaluating the value of longer career histories in Section 10.3.

For this experiment, we limit the length of career history to the $k$ *most recent* observations of $\{x_{i,\tau}\}_{\tau=t-k}^{t}$, which includes both time-varying and time-invariant covariates, and $\{y_{i,\tau}\}_{\tau=t-k}^{t-1}$, $P\left(y_{i,t} \mid \{x_{i,\tau}\}_{\tau=t-k}^{t}, \{y_{i,\tau}\}_{\tau=t-k}^{t-1}\right)$. When $k = \infty$ (equivalently, $k = t - 1$) the model has access to all previous observations. Consider the following prompt that would be fed into the LLM to predict the fifth occupation using the first four observations.

```
<A worker from the PSID dataset>
The following information is available about the work history of a female
↪ white US worker residing in the west region.
The worker was born in 1985.
The worker has the following records of work experience, one entry per
↪ line, including year, education level, and the job title:
2007 (college): Postmasters and mail superintendents
```

```
2009 (college): Athletes, coaches, umpires, and related workers
2011 (college): Education administrators
2013 (college): Child care workers
2015 (college):
```

If we set $k = 2$ most recent previous observations, we would drop the first two observations in the years 2007 and 2009 and feed the following prompt into the LLM to predict the fifth occupation using only the two most recent observations instead of the full prompt above.

```
<A worker from the PSID dataset>
The following information is available about the work history of a female
↪  white US worker residing in the west region.
The worker was born in 1985.
The worker has the following records of work experience, one entry per
↪  line, including year, eaducation level, and the job title:
2011 (college): Education administrators
2013 (college): Child care workers
2015 (college):
```

Formally, define the following non-overlapping subsets of individual-year observations from the test set:

- $S^{\text{(test)}}_{5<t\leq 10} = \{(i,t) \in \mathcal{D}^{\text{(test)}} \mid 5 < t \leq 10\}$,
- $S^{\text{(test)}}_{10<t\leq 15} = \{(i,t) \in \mathcal{D}^{\text{(test)}} \mid 10 < t \leq 15\}$,
- $S^{\text{(test)}}_{15<t\leq 20} = \{(i,t) \in \mathcal{D}^{\text{(test)}} \mid 15 < t \leq 20\}$,
- $S^{\text{(test)}}_{20<t\leq 25} = \{(i,t) \in \mathcal{D}^{\text{(test)}} \mid 20 < t \leq 25\}$,
- $S^{\text{(test)}}_{25<t\leq 30} = \{(i,t) \in \mathcal{D}^{\text{(test)}} \mid 25 < t \leq 30\}$.

The NLSY97 dataset covers a shorter time span, therefore, $S^{\text{(test)}}_{20<t\leq 25}$ and $S^{\text{(test)}}_{25<t\leq 30}$ are defined as empty sets for NLSY97.

Given a $S^{\text{(test)}}_{t_{\text{min}}<t\leq \text{min}+5}$, for each observation $(i,t) \in S^{\text{(test)}}_{t_{\text{min}}<t\leq \text{min}+5}$, we create a text templates consisting of only $k$ *most recent* observations of individual $i$ prior to her $t^{\text{th}}$ observation: $\textsc{Tmpl}(x_i, \{x_{i,\tau}\}_{\tau=t-k}^{t}, \{y_{i,\tau}\}_{\tau=t-k}^{t-1})$ for various values of $k$. Specifically,

- $k \in \{5\}$ if $t_{\text{min}} = 5$.
- $k \in \{5, 10\}$ if $t_{\text{min}} = 10$.
- $k \in \{5, 10, 15\}$ if $t_{\text{min}} = 15$.
- $k \in \{5, 10, 15, 20\}$ if $t_{\text{min}} = 20$.
- $k \in \{5, 10, 15, 20, 25\}$ if $t_{\text{min}} = 25$.

After this procedure, we create an array of prediction tasks (i.e., pairs of text prompt and ground truth job) with different combinations of $t_{\text{min}}$ and $k$:

$$\tilde{S}^{\text{(test)}}_{t_{\text{min}}<t\le t_{\text{min}}+5,k} = \{\textsc{Tmpl}\left((x_i, \{x_{i,\tau}\}_{\tau=t-k}^{t}, \{y_{i,\tau}\}_{\tau=t-k}^{t-1}), y_{i,t}\right)\}_{(i,t)\in S^{\text{(test)}}_{t_{\text{min}}<t\le \text{min}+5}}$$

where each element of $\tilde{S}^{\text{(test)}}_{t_{\text{min}}<t\le t_{\text{min}}+5,k}$ is an pair of (1) a prompt containing $k$ past observations prior to the $t^{\text{th}}$ record of individual $i$ and (2) the ground truth occupation individual $i$ has in her $t^{\text{th}}$ record (i.e., the label).

We evaluate our models using the prompt-label pair in *each* $\tilde{S}^{\textit{(test)}}_{t_{min}<t\le t_{min}+5,k}$ *separately*. Within each $\tilde{S}$ group, we query the likelihood that the language model assigns to the ground truth job title as the continuation of the text prompt, $\hat{P}_{\text{LLM}}(\textsc{Title}(y_{i,t}) \mid \textsc{Tmpl}(x_i, \{x_{i,\tau}\}_{\tau=t-k}^{t}, \{y_{i,\tau}\}_{\tau=t-k}^{t-1}))$, and compute the perplexity using all predictions within that $\tilde{S}$. Finally, we build a matrix of perplexity metrics assessing model's performance under different levels of exposure to past information, the results of which are reported in Table 13 in the main text.

## Appendix M: Details for Additional Analyses

In this appendix, we provide additional details on the two exercises we perform in Section 10.4. First, we report the details of our analysis to learn the extent to which the embeddings created by FT-7B incorporate information about the meaning of job titles by assessing the predictive power of these embeddings on a task related to the interpretation of job titles. Specifically, we use different transformer models to generate embedding vectors for all occupations $y \in \mathcal{Y}$, and set up a prediction task to explore how much SOC occupational hierarchy these embeddings encode. Since we only have around 300 occupations and embedding dimensions are much higher (e.g., 4,096), we apply PCA dimension reduction to reduce all embeddings to 32 dimensions. Then, we build a multinomial logistic regression (with an elastic-net regularization) to predict which of the following six SOC groups an occupation belongs to: "Alternate aggregations", "Management, Business, Science, and Arts Occupations", "Service Occupations", "Sales and Office Occupations", "Natural Resources, Construction, and Maintenance Occupations Production, Transportation, and Material Moving Occupations", and "Military Specific Occupations". We regularize the multinomial regression using a convex combination of L1 and L2 regularization (i.e., the elastic-net, $\frac{\alpha||\beta||_1+(1-\alpha)||\beta||_2}{C}$); and we use five-fold cross-validation to choose the best regularization strength $C$ and weight $\alpha$.

Table M.1 shows that LLM embeddings can capture meaningful patterns in occupational hierarchies, highlighting the importance of prior knowledge in the predictions.

TABLE M.1. Test-set accuracy of predicting correct SOC-group given embeddings.

| **Embedding Method** | **Test Set Accuracy** |
|---|---|
| FT-7B | 78.21% (0.063) |
| CAREER | 76.42% (0.257) |
| OTS Llama-2-7B | 75.82% (0.049) |

*Note*: Test-set-bootstrap standard errors are in parentheses. All models are PCA-ed to 32 dimensions.

Second, we report the details of our analysis to learn for which types of transitions FT-13B outperforms CAREER in predicting whether an individual "moves" jobs. Specifically, we ask the question: for what kind of mover observations $(i,t)$ with characteristics $(y_{i,t}, x_{i,\leq t}, y_{i,<t})$ do language models outperform the previous specialized transformer? We focus on "mover" transitions in the test split of the PSID81 dataset since it is our largest dataset.

To begin, we define our prediction target as the difference in the log-likelihood of the ground truth between predictions from FT-13B and CAREER, as follows:

$$\Delta\hat{P}_{\text{job}} = \log\hat{P}_{\text{LLM}}(y_{i,t} \mid y_{i,t} \neq y_{i-1,t}, x_{i,\leq t}, y_{i,<t}) \\ - \log\hat{P}_{\text{CAREER}}(y_{i,t} \mid y_{i,t} \neq y_{i-1,t}, x_{i,\leq t}, y_{i,<t}) \tag{8}$$

where $\Delta\hat{P}_{\text{job}}$ quantifies the improvement of FT-13B over the CAREER model for a particular transition $(i,t)$ (i.e., individual-year observation).

We build a predictive generalized random forest (which embeds sample splitting to avoid overfitting as described in Athey et al. (2018)) to predict this difference using as covariates the variables in Table M.2. We assign each realization of covariates to a quintile based on the resulting estimates of the difference between the models (i.e., $\Delta\hat{P}_{\text{job}}$). The presence of heterogeneity in the quintile-level test set mean differences in log-likelihood indicates that the intensity of differences in performance between FT-13B and CAREER vary as a function of the features of the individual-year observation, denoted $\Phi_{i,t}(y_{i,t}, x_{i,\leq t}, y_{i,<t})$. Note that logged variables are computed as $\log(x+1)$ to avoid $\log(0)$.

Then, we show the values of several features in each quintile, allowing us to understand the factors that vary systematically between higher and lower quintiles. The corresponding heat map is shown in Figure M; for example, Figure M shows that fine-tuned Llama-2-13B performs better for movers as the transition index increases and the number of tokens in the career history prompt increases. This improvement can again be attributed to the attention mechanism and pre-training.

## Appendix N: Data Appendix

The paper uses three nationally representative survey datasets from the United States to assess the performance of occupation models in predicting career trajectories: the Panel Study of Income Dynamics (PSID81), the National Longitudinal Survey of Youth 1979 (NLSY79), and the National Longitudinal Survey of Youth 1997 (NLSY97). In addition, the paper uses occupational information from O*Net to create a job similarity feature in the data. This data appendix details

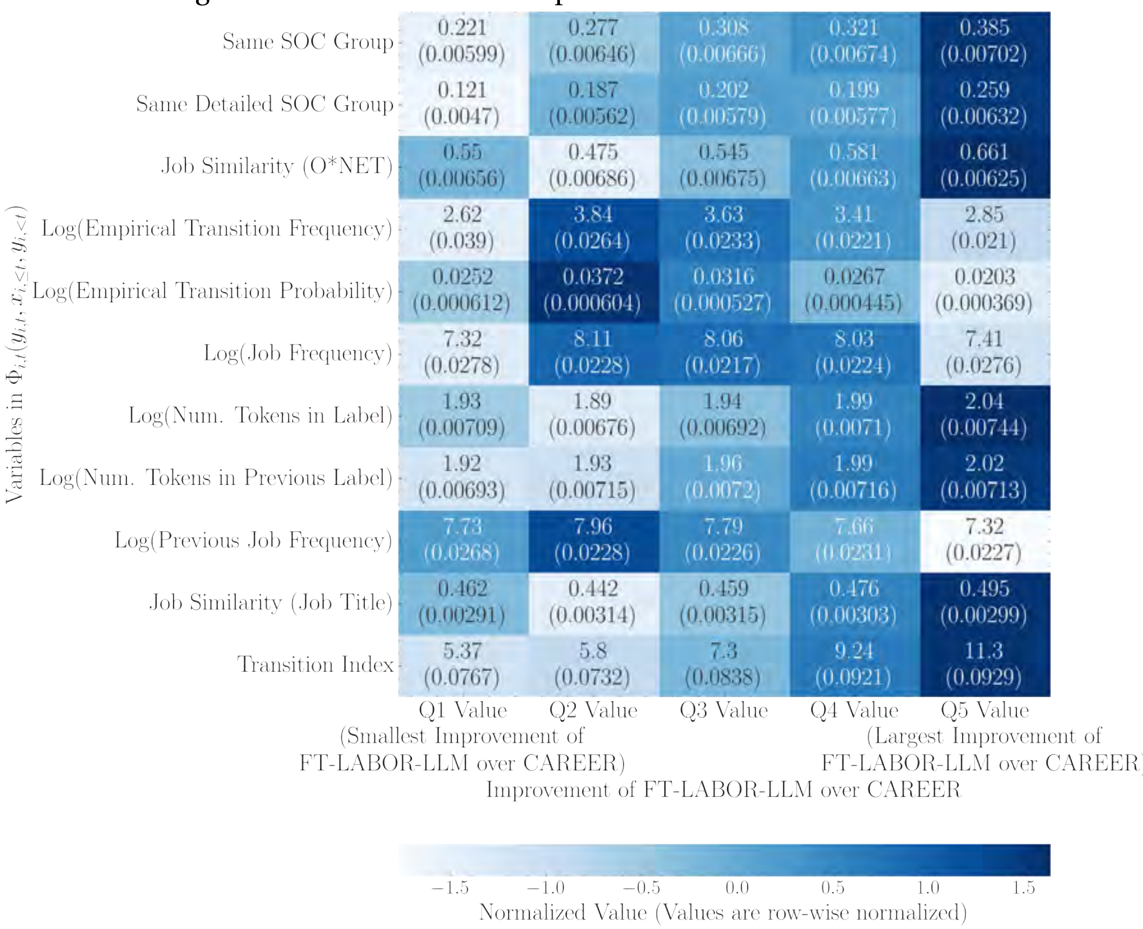
TABLE M.2. Description of features used in the heterogeneous advantage analysis.

| Feature | Description |
|---|---|
| **Transition index** | The transition index $t$ of the job $y_{i,t}$, which is the number of prior observations in the dataset. With a higher $t$, the models have access to a longer career history while making the prediction. |
| **(Logged) job frequency** | The number of occurrences of occupation $y_{i,t}$ in the dataset. |
| **(Logged) previous job frequency** | The number of occurrences of occupation $y_{i,t-1}$ in the dataset. |
| **(Logged) empirical transition frequency** | The empirical number of transitions $y_{i,t-1} \to y_{i,t}$, calculated as $\#^{\text{(train)}}\{y_{i,t-1} \to y_{i,t}\}$. |
| **(Logged) empirical transition probability** | The empirical probability of transition $y_{i,t-1} \to y_{i,t}$, calculated as $\frac{\#^{\text{(train)}}\{y_{i,t-1} \to y_{i,t}\}}{\#^{\text{(train)}}\{y_{i,t-1}\}}$. |
| **(Logged) number of tokens in job title** | The number of tokens in the job title of occupation $y_{i,t}$. |
| **(Logged) number of tokens in previous job title** | The number of tokens in the previous job title $y_{i,t-1}$. |
| **Same SOC group** | Using the SOC hierarchy to cluster $y_{i,t-1}$ and $y_{i,t}$ into SOC-group$(y_{i,t-1})$ and SOC-group$(y_{i,t})$. Indicators measure the magnitude of job transition: $\mathbf{1}\{\text{SOC-group}(y_{i,t-1}) = \text{SOC-group}(y_{i,t})\}$. |
| **Same detailed SOC group** | Using the SOC hierarchy to cluster $y_{i,t-1}$ and $y_{i,t}$ into SOC-detailed-group$(y_{i,t-1})$ and SOC-detailed-group$(y_{i,t})$. Indicators measure the magnitude of job transition: $\mathbf{1}\{\text{SOC-detailed-group}(y_{i,t-1}) = \text{SOC-detailed-group}(y_{i,t})\}$. |
| **Occupational Similarity based on O*NET** | We compute cosine similarities between job $y_{i,t-1}$ and $y_{i,t}$ on eight aspects in the O*NET dataset: "Abilities", "Composite Attributes", "Interests", "Knowledge", "Skills", "Work Activities", "Work Styles", and "Work Values", separately; then, include the average cosine similarity. |
| **Similarity between job titles** | Cosine similarity of embeddings for job titles $y_{i,t-1}$ and $y_{i,t}$, generated using the off-the-shelf Llama-2-7B. |
| **Embedding of career history** $\textbf{T}\text{MPL}(x_{i,\leq t}, y_{i,<t})$ | Embedding of text representation $\text{TMPL}(x_{i,\leq t}, y_{i,<t})$ generated using the off-the-shelf Llama-2-13B model. The embedding space is reduced from 5,120 to 32 dimensions via PCA for faster GRF estimation. |

each data source, how it was retrieved, and the data pre-processing steps we took for each dataset. We also provide descriptive statistics on the static variables in this appendix, and describe the process of combining the datasets.

For each survey dataset, we construct a group of static and dynamic variables. Static variables that remain consistent over time are "personal id," "gender," "birth year," "race/ethnicity," and "region." We also construct two dynamic variables for each survey year, "occupation" and "education level," that employ two input variables "education enrollment status" (for NLSY datasets only) and "employment status." The sections below describe how the listed variables are constructed using each dataset.

FIGURE M.1. Average covariate values within each quintile as defined by the predicted difference in log-likelihood on conditional prediction.




*Note*: Each cell depicts the corresponding feature's values for each quintile by the estimated difference. Standard errors of feature values are shown in parentheses.

### N.1 *The Panel Study of Income Dynamics (PSID81)*

The Panel Study of Income Dynamics (PSID) is a longitudinal U.S. household survey tracking families and their individual members (Survey Research Center, Institute for Social Research, University of Michigan (2024)). The first annual wave from 1968 included approximately 4,800 households. Since then, the PSID has traced all individuals from those households and their descendants, collecting information on individuals and their co-residents on an annual basis through 1997, then biennially starting in 1999. Each member of the original PSID study and their descendants continue to be surveyed, even after leaving the household of origin. This is true for children, other adult members, and ex-spouses forming new family units. The original PSID study was focused on the dynamics of poverty, so the 1968 wave oversampled low-income households and had a relatively large sub-sample of Black respondents. A representative sample of 2,043 Latino households (of Mexican, Cuban and Puerto Rican origin) was added in 1990, but was dropped by the PSID in 1995, so we drop this sample from our final dataset.

To replicate the results in our study, researchers can download the data file that we used from the PSID data center at https://simba.isr.umich.edu/DC/c.aspx. After creating an account, the researcher can use the "Previous Cart" option, search for the email tianyudu@stanford.edu, and select Job "339649" The raw data file used for the analysis in this paper was created and downloaded on November 2$^{\text{nd}}$, 2024 at 10:52:52 PM. If the above dataset cannot be successfully retrieved, our replication notebook also provides a complete list of variables we used and the instruction to obtain these data from the PSID data server at https://simba.isr.umich.edu/DC/l.aspx.

In this project, we restrict our attention to survey years between 1981 and 2021 (inclusive) because occupation code was originally recorded with only one or two digits in 1979 and 1980, and retrospective updating to three-digit codes was missing for many individuals. We also restrict our sample to individual-year observations that are household heads or spouses because we observe occupation and race/ethnicity information only for these family members. After the pre-processing described below, our resulting final dataset, which we refer to as PSID81, has 31,056 individuals and 313,622 total individual-year observations of occupations.

We use five static covariates for each individual, dropping individuals for whom this information is missing: personal id, gender, race/ethnicity, region, and birth year. We construct each individual's personal id by combining the PSID identifiers for family and individual. We use the main PSID variable for gender, classifying individuals as "male" or "female." Race/ethnicity is recorded each survey year by the PSID, with definitions varying slightly from year to year.[18] We collapse all definitions into either "white" (consistent category across years), "black," or "other/unknown." Then, we take the first non-other/unknown observation of race/ethnicity for our static variable, or classify the individual as other/unknown if their race/ethnicity is never classified as white or black. We base the region variable on the state in which a family lives, which is recorded each survey year by the PSID. First, we construct region as a 4-category variable that takes the values "northeast," "south," "west," and "northcentral" based on state. Then, we take the first non-missing observation as our static variable.

We construct birth year based on the age variable recorded each survey year by the PSID. To compute birth year, we take the mode of the difference between the survey year and the individual's age for each individual-year observation. When there is more than one mode, we take the average of the two most frequent birth years. Two modes, which we observe for 1,702 individuals, are likely the result of variation in the timing of a survey within the calendar year. Three and four modes, which we observe for 32 and 3 individuals, respectively, are likely due to measurement error.

We construct two dynamic variables for each individual-year observation in addition to the calendar year of survey: education level and occupation. We construct education level based on the years of education recorded each survey year in the PSID81. We categorize years of education into "less than high school," "high school," "some college," "college," and "any graduate" each year, then forward-fill education to replace missing values and impose the restriction that education level be non-decreasing.

We construct our main variable of interest, occupation, using the same preprocessing steps applied by Vafa et al. (2024) to facilitate comparisons, combining information from multiple variables recorded each survey year by the PSID81. First, we crosswalk individual-year observations of occupation that are recorded as either 1970 or 2000 census codes to the occ1990dd scheme for uniformity throughout the dataset (Autor and Dorn (2013)). We then collapse the employ-

[18] Race/ethnicity for spouse was collected by the PSID starting in 1985.

ment status variable into four categories: "employed," "out of labor force" (defined as "Retired," "Permanently disabled," or "Housewife"), "unemployed" (defined as "Only temporarily laid off" or "Looking for work, unemployed"), and "student." All other original values that do not fit into these categories are treated as missing for employment status. Lastly, we replace individual-year observations of occupation with employment status when employment status is non-employed (out-of-labor-force, unemployed, or student). So employment status replaces missing values of occupation, but it also replaces valid occupation codes when employment status is one of the three non-employed statuses, meaning that non-employed statuses take priority over occupation.

After constructing our dynamic variables of interest, we filter individuals and individual-year observations with invalid values for these variables. Our data filtering process starts with 35,516 individuals with 360,373 individual-year observations after the 1981 survey (inclusive), when the individual was either the household head or the spouse of the head.

We start with restricting our dataset to individual-year observations that have "sequence number" values between 1 and 20, meaning the individual lives in the household, leading to 35,298 individuals and 352,191 individual-year observations. We then restrict individual-year observations with age between 18 and 80 (inclusive), resulting in 344,682 individual-year observations from 35,068 unique individuals. Then, we drop 2,999 individuals whose occupation status is not in the labor force across all years, resulted in 32,069 unique individuals and 323,420 individual-year observations. After combining occupation and employment status into our final occupation variable, we drop 5,037 individual-year observations with missing or invalid values for occupation, leading to 31,795 individuals and 318,383 individual-year observations. We drop 632 individuals with 4,512 individual-year observations with missing educational information even after the forward filling, which corresponds to individuals whom we never observe years of education and individual-year observations that occur before the first non-missing observation of years of education. The filtering on educational level leads to 31,163 individuals and 313,871 individual-year observations. Finally, 107 individual (249 individual-year observations) with no observation of family state (for the region variable), resulted in 31,056 individuals and 313,622 individual-year observations. After the processing above, we have no missing values for personal id or gender, or birth year, and race/ethnicity has no missing values by construction (other/unknown category). The sequential filtering steps lead to the final PSID81 dataset used in this study.

### N.2 *National Longitudinal Survey of Youth (NLSY)*

The National Longitudinal Survey of Youth of 1979 (NLSY79) and 1997 (NLSY97) are two cohort-based surveys sponsored by the U.S. Bureau of Labor Statistics that follow individuals born in the United States.

*NLSY79* The NLSY79 includes individuals born between 1957 and 1964 who were between 14 and 22 years old at the time data collection started in 1979. The original cohort contained 12,686 respondents. These individuals were interviewed annually from 1979 through 1994, and biennially thereafter. We use data from surveys conducted 1979 through 2020. To replicate the results in our study, researchers can download the NLSY79 data file at https://www.nlsinfo.org/investigator/pages/sea After creating an account, the researcher can search and select the variables listed, and download the data file. After the pre-processing described below, our resulting dataset, which we refer to as NLSY79, has 12,479 individuals and 259,778 total individual-year observations of occupations.

As in the PSID81 dataset, we use five static covariates for each individual, dropping individuals for whom this information is missing: personal id, gender, race/ethnicity, region, and birth year. Personal id requires no processing. We use the main NLSY variables for gender, race/ethnicity, and birth year. There are no missing values for these variables and the only processing is descriptive labeling. Gender has two values: “male” or “female.” Race/ethnicity has three values: “Hispanic,” “black,” or “non-Hispanic/non-black.” Birth year has eight values from “1959” to “1964.”

The region variable is recorded each survey year by the NLSY as one of four values: “northeast,” “south,” “west,” and “northcentral.” We take the first non-missing observation as our static variable. We drop 2 individuals with no region information in any year.

We construct two dynamic variables for each individual-year observation, dropping observations for which either variable is missing: education level and occupation. We construct education level based on the years of education recorded each survey year in the NLSY through 2016.[19] For 2018 and 2020, we use the same educational level as in 2016. When we compare the highest degree obtained in 2016 to the highest degree ever obtained, we have a $99.59\%$ match. We categorize years of education into "less than high school," "high school," "some college," "college," and "any graduate" each year, then forward-fill education to replace missing values and impose the restriction that education level be non-decreasing. We drop 12 individual-year observations because of invalid skip and 12 individual-year observations because of non-interview that occur prior to the first valid observation of education for an individual. We also dropped x individuals for whom we never observe years of education.

We again construct our main variable of interest, occupation, using similar pre-processing steps applied by Vafa et al. (2024) to facilitate comparisons, combining information from multiple variables recorded each survey year by the NLSY. For the occupation variable, we crosswalk individual-year observations, which are recorded as either 1970 or 2000 census codes, to 1990 census codes for consistency across datasets (Autor and Dorn (2013)). The educational enrollment status variable requires no processing beyond descriptive labels and has two values: "yes" or "no," where yes means the individual is a student that year.

Employment status is recorded on a weekly basis, with retrospective updating. To create employment status at the year level, we take the most frequent informative response (i.e., not the "no information" or "not working" status, where the latter does not differentiate unemployed from out of labor force, or other missing values). We then collapse the employment status variable into three categories: "employed" (defined as "active miliary service," "associated with employment,"

[19]This variable is labeled "highest degree obtained" by NLSY, but captures years of education rather than just completed degrees.

or any value that corresponds to a "job number"), "out of labor force" (defined as "not associated with employment" or "out of labor force") and "unemployed." All other original values that do not fit into these categories are treated as missing for employment status.

To combine the occupation, educational enrollment status, and employment status variables into our final processed occupation variable, we do the following for each individual-year observation: We use "student" when educational enrollment status is yes. If not, we use "out of labor force" or "unemployed" if employment status is one of those values. If the occupation is still undecided, we use occupational code if it is specified. After combining occupation, educational enrollment status and employment status into our final occupation variable, we drop 108,034 individual-year observations with missing or invalid values for occupation.

*NLSY97* The NLSY97 includes individuals born between 1980 and 1984 who were between 12 and 17 years old at the time data collection started in 1997. The original cohort contained 8,984 respondents. These individuals were interviewed annually from 1997 through 2011, and biennially thereafter. We use data from surveys conducted 1997 through 2021. To replicate the results in our study, researchers can download the the NLSY97 data file at https://www.nlsinfo.org/investigator/pages/se After creating an account, the researcher can search and select the variables listed, and download the data file. One can find official tutorials of accessing NLSY data at https://www.nlsinfo.org/content/getting-started/introduction-to-the-nls/tutorials-and-videos. After the pre-processing described below, our resulting dataset, which we refer to as NLSY97, has 8,984 individuals and 148,795 total individual-year observations of occupations.

As in the other two datasets, we use five static covariates for each individual, dropping individuals for whom this information is missing: personal id, gender, race/ethnicity, region, and birth year. Personal id requires no processing. We use the main NLSY variables for gender, race/ethnicity, and birth year. There are no missing values for these variables and the only processing is descriptive labeling. Gender has two values: "male" or "female." Differing from NLSY79, race/ethnicity has four values: "Hispanic or Latino," "black or African-American," "mixed race non-Hispanic," or "non-Hispanic/non-black." Birth year has five values from "1980" to "1984."

As in the NLSY79, the region variable is recorded each survey year as one of four values: "northeast," "south," "west," and "northcentral;" however, there are no missing values for the first year 1997, so we download only the variable for 1997 and use it as our static variable.

The construction of the two dynamic variables, education level and occupation, for each individual-year observation also follows our process for NLSY79. Unlike the NLSY79, the education variable we use records highest *degree* achieved each survey year, so we do not need to convert years of education to degree. We do some aggregation to achieve the same levels as other datasets: "less than high school" (defined as "none" or "GED"), "high school," "some college," "college," and "any graduate" (defined as "Master's," "PhD," or "Professional Degree"). As in the other datasets, we forward-fill education to replace missing values and impose the restriction that education level be non-decreasing. There are no individual-year observations that occur before the first non-missing observation of years of education and no individuals for whom we never observe years of education.

We again construct our main variable of interest, occupation, using the same pre-processing steps applied by Vafa et al. (2024) to facilitate comparisons, combining information from multiple variables recorded each survey year by the NLSY. For the occupation variable, we crosswalk individual-year observations from the 2000 census codes to 1990 census codes for consistency across datasets (Autor and Dorn (2013)).[20] There are many "non enrolled" and "enrolled" values for the educational enrollment status variables, which we aggregate.

As in the NLSY79, employment status is recorded on a weekly basis, with retrospective updating. To create employment status at the year level, we take the most frequent informative response (i.e., not the "no information" or "not working" status). We then collapse the employment status variable into three cate-

[20]To have the right number of digits for the cross-walk, we divide each occupation code by ten.

gories: “employed” (defined as “active miliary service,” “associated with employment,” or any value that corresponds to a “job number”), “out of labor force” (defined as “not associated with employment” or “out of labor force”) and “unemployed.” All other original values that do not fit into these categories are treated as missing for employment status.

To combine the occupation, educational enrollment status, and employment status variables into our final processed occupation variable, we do the following for each individual-year observation: We use “student” when educational enrollment status is enrolled. If not, we use “out of labor force” or “unemployed” if employment status is one of those values. If the occupation is still undecided, we use occupational code if it is specified. After combining occupation, educational enrollment status and employment status into our final occupation variable, we drop 30,885 individual-year observations with missing or invalid values for occupation.

### N.3 *O*NET*

The O*NET dataset is the main occupational information database in the United States, developed by the U.S. Department of Labor. For each occupation, it includes the following occupational characteristics, encoded as text: Tasks, Technology Skills, Tools Used, Work Activities, Detailed Work Activities, Work Context, Job Zone, Skills, Knowledge, Abilities, Interests, Work Values, Work Styles, Related Occupations. The O*NET data is publicly available and can be accessed at online.

We match O*NET data for 335 job titles from career trajectories we built on survey data to further train LABOR-LLM models. O*NET variables included in this matching process are Skills, Knowledge, Abilities, Tasks, Interests, Work Styles, Work Activities, Work Values, and Related Job Titles. We use these variables to build textual representations based on the job description (which includes up to five descriptions from the closest matching SOC codes), categorical data (Skills through Work Values, calculating the average importance score for each variable across all matching SOC and selecting the top five), and Related Job Titles (sam-

TABLE N.1. Share of observations with different demographic characteristics.

| | PSID81 | | NLSY79 | | NLSY97 | |
|---|---|---|---|---|---|---|
| | Individual | Transition | Individual | Transition | Individual | Transition |
| **Gender** | | | | | | |
| Female | 50.5% | 54.2% | 49.7% | 51.6% | 48.8% | 50.3% |
| Male | 49.5% | 45.8% | 50.3% | 48.4% | 51.2% | 49.7% |
| **Ethnicity** | | | | | | |
| Black | - | - | 25.1% | 28.1% | - | - |
| Black or African-American | 34.5% | 32.1% | - | - | 26.0% | 26.9% |
| Hispanic | - | - | 16.0% | 17.9% | - | - |
| Hispanic or Latino | - | - | - | - | 21.2% | 21.3% |
| Mixed-Race Non-Hispanic | - | - | - | - | 0.9% | 0.9% |
| Non-Black Non-Hispanic | - | - | 58.9% | 54.1% | 51.9% | 50.9% |
| Other or Unknown | 6.1% | 3.1% | - | - | - | - |
| White | 59.4% | 64.7% | - | - | - | - |
| **Region** | | | | | | |
| Northcentral | 24.2% | 25.8% | 23.8% | 25.2% | 22.8% | 22.8% |
| Northeast | 13.7% | 15.4% | 20.4% | 19.2% | 17.6% | 17.3% |
| South | 43.9% | 41.8% | 36.7% | 37.0% | 37.4% | 37.8% |
| West | 18.2% | 17.1% | 19.1% | 18.6% | 22.2% | 22.1% |

ple up to five specific job titles from the closest matching SOC codes). We generate one text file for each job title in our dataset.

## N.4 *Summary Statistics*

Table N.1 provides summary statistics by dataset for the demographic variables we use in our analysis. Recall that the demographics are assigned to be constant within our cleaned dataset even if they changed over time in the original survey data. Note further that the ethnicity encoding across datasets are slightly different.

Figure N.1 presents example job titles in a word cloud, weighted by their popularity. Each job title's font size is scaled proportionally to its frequency in the test sets of the three datasets (PSID81, NLSY79, NLSY97) combined, measured by the number of individual-year observations; thus, more prevalent occupations appear larger, highlighting their distribution within our labor market data.

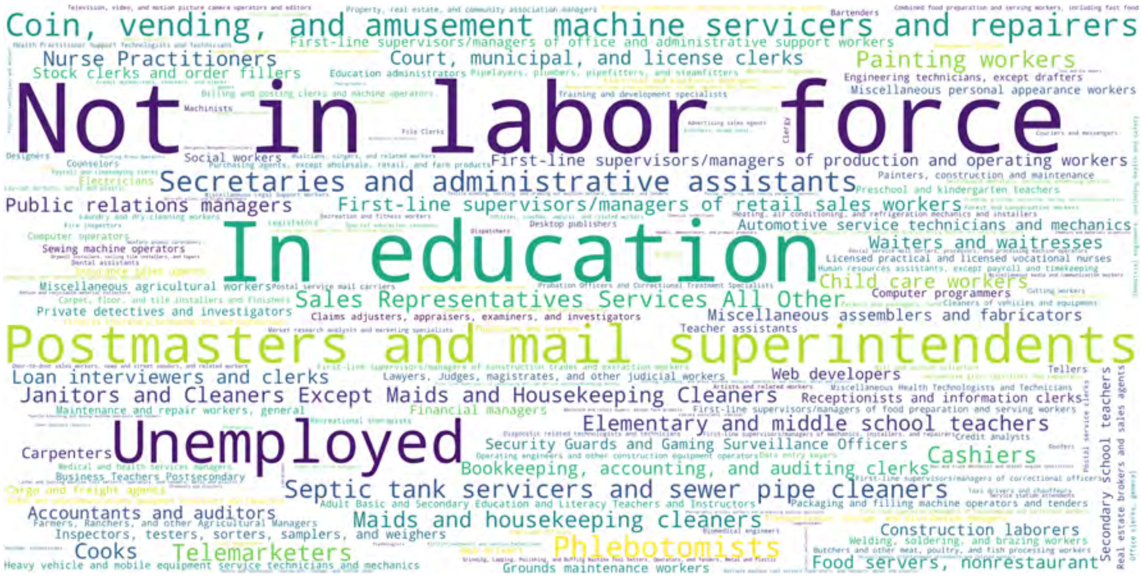



FIGURE N.1. Word cloud of job titles, scaled by title popularity.

## N.5 *Combined data sources*

Once pre-processed, the PSID81, NLSY79, and NLSY97 datasets are used to construct the input files to fine-tune all predictive models included in this project on their career trajectory data and covariates. For this purpose, each dataset is divided into three subdatasets: training, validation and test. The construction of the datasets for this stage follows Vafa et al. (2024). The resumes or sequences of jobs are prepared into individual data files for the split they correspond to. That is, the resume data resulting from PSID81, NLSY79, and NLSY97 is structured as “train.job,” “valid.job,” and “test.job.” In each file, each row corresponds to one individual in the sample, and jobs are designated using a classification code, such as O*NET or occ1990dd. Each covariate included in the dataset follows the same structure, and it should have the same number of rows as the job file associated, corresponding to the same group of individuals. Note that the covariate “birth years” is not included to fine-tune the CAREER-LLM model.

## References


Agostinelli, Andrea, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank (2023), "MusicLM: Generating Music From Text." URL http://arxiv.org/abs/2301.11325. ArXiv:2301.11325 [cs, eess]. [10]

Argyle, Lisa P., Ethan C. Busby, Nancy Fulda, Joshua R. Gubler, Christopher Rytting, and David Wingate (2023), "Out of One, Many: Using Language Models to Simulate Human Samples." *Political Analysis*, 31 (3), 337–351, URL https://www.cambridge.org/core/product/identifier/S1047198723000025/type/journal_article. [11]

Athey, Susan, Niall Keleher, and Jann Spiess (2023), "Machine learning who to nudge: causal vs predictive targeting in a field experiment on student financial aid renewal." *arXiv preprint arXiv:2310.08672*. [3]

Athey, Susan, Julie Tibshirani, and Stefan Wager (2018), "Generalized Random Forests." URL http://arxiv.org/abs/1610.01271. ArXiv:1610.01271 [econ, stat]. [73]

Autor, David H and David Dorn (2013), "The growth of low-skill service jobs and the polarization of the us labor market." *American economic review*, 103 (5), 1553–1597. [11, 22, 77, 80, 82]

Bao, Keqin, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He (2023), "Tallrec: An effective and efficient tuning framework to align large language model with recommendation." In *Proceedings of the 17th ACM Conference on Recommender Systems*, 1007–1014. [10]

Blau, David M. and Regina T. Riphahn (1999), "Labor force transitions of older married couples in Germany." *Labour Economics*, 6 (2), 229–252, URL https://www.sciencedirect.com/science/article/pii/S0927537199000172. [8]

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma

Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang (2022), "On the Opportunities and Risks of Foundation Models." URL http://arxiv.org/abs/2108.07258. ArXiv:2108.07258 [cs]. [4, 5]

Boskin, Michael J. (1974), "A Conditional Logit Model of Occupational Choice." *Journal of Political Economy*, 82 (2, Part 1), 389–398, URL https://www.journals.uchicago.edu/doi/abs/10.1086/260198. Publisher: The University of Chicago Press. [2, 8]

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020), "Language models are few-shot learners." URL https://arxiv.org/abs/2005.14165. [10]

Bucher, Martin Juan José and Marco Martini (2024), "Fine-tuned'small'llms (still) significantly outperform zero-shot generative ai models in text classification." *arXiv preprint arXiv:2406.08660.* [7]

Bureau of Labor Statistics, U.S. Department of Labor (2023), "National longitudinal survey of youth 1979 cohort, 1979-2020 (rounds 1-29)." [5]

Bureau of Labor Statistics, U.S. Department of Labor (2024), "National longitudinal survey of youth 1997 cohort, 1997-2021 (rounds 1-20)." [5]

Chen, Mark, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba (2021), "Evaluating Large Language Models Trained on Code." URL http://arxiv.org/abs/2107.03374. ArXiv:2107.03374 [cs]. [10]

Cortes, Guido Matias (2016), "Where have the middle-wage workers gone? a study of polarization using panel data." *Journal of Labor Economics*, 34 (1), 63–105. [3]

de Ruijt, Corné and Sandjai Bhulai (2021), "Job Recommender Systems: A Review." URL http://arxiv.org/abs/2111.13576. ArXiv:2111.13576 [cs]. [3]

Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer (2023), "QLoRA: Efficient Finetuning of Quantized LLMs." URL http://arxiv.org/abs/2305.14314. ArXiv:2305.14314 [cs]. [60]

Dong, Guanting, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou (2023), "How abilities in large language models are affected by supervised fine-tuning data composition." *arXiv preprint arXiv:2310.05492*. [7]

Faria-e Castro, Miguel and Fernando Leibovici (2024), "Artificial intelligence and inflation forecasts." *Federal Reserve Bank of St. Louis Working Paper 2023*, 15. [10]

Geng, Shijie, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang (2022), "Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5)." In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, 299–315, Association for Computing Machinery, New York, NY, USA, URL https://dl.acm.org/doi/10.1145/3523227.3546767. [10]

Hall, Robert E., Aaron Gordon, and Charles Holt (1972), "Turnover in the Labor Force." *Brookings Papers on Economic Activity*, 1972 (3), 709, URL https://www.jstor.org/stable/2534130?origin=crossref. [3, 8]

He, Miao, Xiaoming Zhan, Dayong Shen, Yuanyuan Zhu, Hua Zhao, and Renjie He (2021), "What about your next job? predicting professional career trajectory using neural networks." In *Proceedings of the 2021 4th International Conference on Machine Learning and Machine Intelligence*, 184–189. [9]

Jin, Hongpeng, Wenqi Wei, Xuyu Wang, Wenbin Zhang, and Yanzhao Wu (2023), "Rethinking learning rate tuning in the era of large language models." In *2023 IEEE 5th International Conference on Cognitive Machine Intelligence (CogMI)*, 112–121, IEEE. [64]

Jin, Ming, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen (2024),

"TIME-LLM: TIME SERIES FORECASTING BY REPROGRAMMING LARGE LANGUAGE MODELS." [10]

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei (2020), "Scaling laws for neural language models." *arXiv preprint arXiv:2001.08361*. [7]

Lersch, Philipp M., Wiebke Schulz, and George Leckie (2020), "The variability of occupational attainment: How prestige trajectories diversified within birth cohorts over the twentieth century." *American Sociological Review*, 85 (6), 1084–1116, URL https://doi.org/10.1177/0003122420966324. [23]

Li, Liangyue, How Jing, Hanghang Tong, Jaewon Yang, Qi He, and Bee-Chung Chen (2017), "NEMO: Next Career Move Prediction with Contextual Embedding." In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, 505–513, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, URL https://dl.acm.org/doi/10.1145/3041021.3054200. [9]

Maharjan, Jenish, Anurag Garikipati, Navan Preet Singh, Leo Cyrus, Mayank Sharma, Madalina Ciobanu, Gina Barnes, Rahul Thapa, Qingqing Mao, and Ritankar Das (2024), "Openmedlm: prompt engineering can out-perform fine-tuning in medical question-answering with open-source large language models." *Scientific Reports*, 14 (1), 14156. [10]

Meng, Qingxin, Hengshu Zhu, Keli Xiao, Le Zhang, and Hui Xiong (2019), "A Hierarchical Career-Path-Aware Neural Network for Job Mobility Prediction." In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, 14–24, Association for Computing Machinery, New York, NY, USA, URL https://dl.acm.org/doi/10.1145/3292500.3330969. [9]

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. (2021), "Learning transferable visual models from natural language supervision." In *International conference on machine learning*, 8748–8763, PMLR. [5]

Reimers, Nils and Iryna Gurevych (2019), “Sentence-BERT: Sentence embeddings using Siamese BERT-networks.” In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, eds.), 3982–3992, Association for Computational Linguistics, Hong Kong, China, URL https://aclanthology.org/D19-1410. [10]

Rives, Alexander, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus (2021), “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences.” *Proceedings of the National Academy of Sciences of the United States of America*, 118 (15), e2016239118. [10]

Santurkar, Shibani, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto (2023), “Whose Opinions Do Language Models Reflect?” URL http://arxiv.org/abs/2303.17548. ArXiv:2303.17548 [cs]. [11]

Savcisens, Germans, Tina Eliassi-Rad, Lars Kai Hansen, Laust Hvas Mortensen, Lau Lilleholt, Anna Rogers, Ingo Zettler, and Sune Lehmann (2024), “Using sequences of life-events to predict human lives.” *Nature Computational Science*, 4 (1), 43–56. [5]

Schmidt, Peter and Robert P. Strauss (1975), “The Prediction of Occupation Using Multiple Logit Models.” *International Economic Review*, 16 (2), 471–486, URL https://www.jstor.org/stable/2525826. Publisher: [Economics Department of the University of Pennsylvania, Wiley, Institute of Social and Economic Research, Osaka University]. [8]

Singhal, Karan, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan (2022), "Large Language Models Encode Clinical Knowledge." URL http://arxiv.org/abs/2212.13138. ArXiv:2212.13138 [cs]. [10]

Stevens, Ann Huff (1994), "The dynamics of poverty spells: Updating bane and ellwood." *The American Economic Review*, 84 (2), 34–37. [3]

Survey Research Center, Institute for Social Research, University of Michigan (2024), "Panel study of income dynamics, public use dataset." Produced and distributed by the Survey Research Center, Institute for Social Research, University of Michigan, Ann Arbor, MI. [5, 75]

Taylor, Ross, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic (2022), "Galactica: A Large Language Model for Science." URL http://arxiv.org/abs/2211.09085. ArXiv:2211.09085 [cs, stat]. [10]

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom (2023), "Llama 2: Open Foundation and Fine-Tuned Chat Models." URL http://arxiv.org/abs/2307.09288. ArXiv:2307.09288 [cs]. [28, 64]

Vafa, Keyon, Emil Palikot, Tianyu Du, Ayush Kanodia, Susan Athey, and David M. Blei (2024), "CAREER: A Foundation Model for Labor Sequence Data." URL http:

//arxiv.org/abs/2202.08370. ArXiv:2202.08370 [cs, econ]. [1, 3, 5, 7, 9, 19, 20, 26, 27, 28, 35, 53, 54, 77, 80, 82, 85]

Wachter, Till Von (2020), "The persistent effects of initial labor market conditions for young adults and their sources." *Journal of Economic Perspectives*, 34 (4), 168–194. [23]

Wei, Jason, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le (2022), "Finetuned Language Models Are Zero-Shot Learners." URL http://arxiv.org/abs/2109.01652. ArXiv:2109.01652 [cs]. [10]

Wu, Mike, Noah Goodman, Chris Piech, and Chelsea Finn (2021), "Prototransformer: A meta-learning approach to providing student feedback." *arXiv preprint arXiv:2107.14035*. [5]

Yi, Zihao, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen (2024), "A Survey on Recent Advances in LLM-Based Multi-turn Dialogue Systems." URL http://arxiv.org/abs/2402.18013. ArXiv:2402.18013 [cs]. [10]

Yin, Qingyu, Xuzheng He, Luoao Deng, Chak Tou Leong, Fan Wang, Yanzhao Yan, Xiaoyu Shen, and Qiang Zhang (2024), "Deeper insights without updates: The power of in-context learning over fine-tuning." *arXiv preprint arXiv:2410.04691*. [10]

Zhang, Denghui, Junming Liu, Hengshu Zhu, Yanchi Liu, Lichen Wang, Pengyang Wang, and Hui Xiong (2019), "Job2vec: Job title benchmarking with collective multi-view representation learning." In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2763–2771. [9]

Zhang, Di, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, Dongzhan Zhou, Shufei Zhang, Mao Su, Han-Sen Zhong, and Yuqiang Li (2024), "ChemLLM: A Chemical Large Language Model." URL http://arxiv.org/abs/2402.06852. ArXiv:2402.06852 [cs]. [10]

Zhang, Le, Ding Zhou, Hengshu Zhu, Tong Xu, Rui Zha, Enhong Chen, and Hui Xiong (2021), "Attentive Heterogeneous Graph Embedding for Job Mobility Prediction." In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2192–2201, ACM, Virtual Event Singapore, URL https://dl.acm.org/doi/10.1145/3447548.3467388. [9]