Estimation of Dynamic Discrete Choice Models with Unobserved State Variables Using Reinforcement Learning*

Yingyao Hu[†]
Johns Hopkins University

Fangzhu Yang [‡]
Johns Hopkins University

March 9, 2025

Abstract

Dynamic discrete choice models (DDCs) present significant computational challenges, particularly with high-dimensional state spaces and unobserved heterogeneity. This paper introduces a unified estimation framework that combines policy gradient methods from reinforcement learning with the indirect inference approach. By directly estimating the policy function in the model's inner loop, the method maps deep structural parameters to optimal policy parameters, greatly reducing computational costs. The framework is adaptable to models with partially observed state variables, making it effective for estimating models with various unobserved state variables. Notably, discretizing unobserved state variables is unnecessary, enabling the estimation of DDCs with continuous, time-varying unobserved variables. Empirical validation shows that the proposed method produces estimates that closely align with true parameter values, enhancing both computational efficiency and accuracy for analyzing dynamic decision-making processes.

Keywords: dynamic discrete choice model, unobservable heterogeneity, machine learning, reinforcement learning, policy gradient method

[*] This paper is based on the second chapter of Fangzhu Yang's doctoral dissertation at Johns Hopkins University (Yang (2024)).

[†] Department of Economics, Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218. Email: yhu@jhu.edu.

[‡] Department of Economics, Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218. Email: Fangzhu517@hotmail.com.

1 Introduction

An important family of economic problems involves solving the dynamic discrete choice model (DDC), where an agent chooses the optimal decision in each period to maximize her lifetime utility. This type of model brings significant computational challenges for those with high-dimensional state space that contains many continuous state variables. Discretizing continuous state variables results in an inaccurate approximation of the true modeling process and posts a non-negligible coding burden. At the same time, looping over the huge amount of grid points for the state variables to calculate the value function makes computation infeasible for solving complicated models. In addition, many interesting dynamic models involve unobservable heterogeneity. Unsurprisingly, it is even more challenging to handle a complicated dynamic model with unobserved state variables, especially when these unobserved variables are time-varying and continuous.

To deal with this problem, this paper proposes a unified estimation framework, embedding policy gradient methods (Sutton et al., 1999) from the reinforcement learning literature into the well-established indirect inference method (Gourieroux et al., 1993) in economics to estimate dynamic discrete choice models with potentially a large state space and various types of unobserved state variables. We use the full model solution method that contains two layers of loops. In the outer loop, we use the Simulated Method of Moments (SMM) to search over the deep parameter space to minimize the distance between the simulated moments and data moments. In the inner loop, we solve for the optimal policy given the current structural parameters. Instead of using the traditional dynamic programming method to solve for the optimal policy function, we take advantage of the policy gradient methods by directly parametrizing the optimal policy function and adopting the policy gradient theorem (Sutton et al., 1999) to update the policy parameters until convergence. By doing so, we obtain a mapping from the structural parameters to the optimal policy function parameters, which allows us to estimate the structural parameters with significantly reduced computational burden, since it allows us to avoid looping over all the grid points of the state variables, discretizing the state variables, or calculating the value function.

In addition to estimating dynamic discrete choice models with fully observed state space, our method can also be used for estimating models with partially observed state variables. Building on the non-parametric identification results in Hu and Shum (2012), we show that our proposed algorithm provides a convenient way to estimate dynamic discrete choice models with continuous and time-varying unobserved state variables for those that satisfy the invertibility assumption required in their paper. Our method is suitable for models that satisfy the identification assumptions in Hu and Shum (2012), since we can directly adopt their non-parametric identification results that the full model is identified using four consecutive periods of data by targeting relevant moments in the outer loop, while in the inner using policy gradient method to solve for the optimal policy function and simulating the lifetime path for individuals to generate targeted moments. Therefore, our proposed algorithm serves as an easy-to-implement framework to conduct inference for dynamic discrete choice models with various types of unobserved state variables, where discretization of unobserved state variables is no longer necessary as this algorithm is able to handle continuous unobserved state variables.

In this paper, we start by proposing the algorithm for estimating dynamic discrete choice models with fully observed state space. After introducing our method, we then compare it with the traditional dynamic programming backward induction method by looking at a toy version of the Rust bus engine replacement model (Rust, 1987). We show that the policy gradient method is able to generate an optimal policy function that is similar to the one obtained using dynamic programming. Moreover, the simulation results show that our proposed algorithm can recover the true parameters reasonably well. Building upon this simple model, we then study a more complicated dynamic discrete choice model with fully observed high-dimensional state space that contains continuous state variables and stochastic transitional models. We conduct simulation studies to demonstrate that our method produces estimates that are centered around the true values, while also significantly reducing computational time compared to the dynamic programming method. In addition, the precision loss is minimal compared to the traditional method. Therefore, this Monte Carlo study provides evidence that our method is able to estimate complicated dynamic discrete choice models with high-dimensional state space at a much faster speed than traditional methods while maintaining similar levels of precision.

In the main results section, we focus on dynamic discrete choice models with partially observed state space, introducing an algorithm rooted in reinforcement learning for estimation. Leveraging the identification framework outlined in Hu and Shum

(2012), our algorithm simultaneously updates structural and transitional model parameters by targeting four consecutive data periods using the Simulated Method of Moments in the outer loop. Within the inner loop, we employ the policy gradient method to estimate optimal policy function parameters conditioned on the current structural and transitional model parameters. In essence, the algorithm contains three parameter sets: optimal policy functions, structural parameters, and transitional model parameters. By establishing a mapping from the combined set of structural and transitional model parameters to optimal policy parameters, our approach effectively identifies these parameters for models that satisfy the assumptions outlined in Hu and Shum (2012).

To show that our proposed model works well for estimating dynamic discrete choice models with continuous and time-varying unobserved state variables, we run a Monte Carlo study by looking at a model that satisfies the assumptions required in Hu and Shum (2012). We then show that our proposed method is able to derive estimates that are closely centered around the true values at a reasonably fast computational speed in the presence of continuous and time-varying unobserved heterogeneity. To compare with existing algorithms, we turn to a special case of DDCs with time-invariant and discrete unobservables. We use the same model as the one in Arcidiacono and Miller (2011) and show that our method performs as well as the method proposed by Arcidiacono and Miller (2011) with a shorter time of computation. Finally, we conduct an empirical study using the dynamic household bargaining fertility model with time-varying and continuous unobserved Pareto weights that follow an AR(1) process in the model. We show that the estimation results make intuitive sense in terms of model implications. In summary, we show that our method is applicable to different dynamic models and results in good estimates with reduced computational burden.

Our Contributions We consider this paper to have mainly three contributions. Firstly, this method proposed a convenient inference framework for dynamic models with continuous and time-varying unobserved variables, solving the issue that it is nontrivial to estimate this type of models [1]. Current popular methods estimate the reduced-form CCP using the EM algorithm for dynamic models with unobserved

[1] Although this paper mainly focuses on dynamic discrete choice models, our proposed method can be easily extended to dynamic models with continuous choice variables by choosing the proper policy function parametrization.

state space. However, this type of method is not suitable for dealing with continuous and time-varying unobserved state variables. Our proposed method fills this gap by parametrizing the optimal policy as a function of both observed and unobserved state variables, either discrete or continuous, and identifying the structural parameters by targeting moments satisfying the identification requirements in Hu and Shum (2012). Our method makes discretization of unobserved state variables unnecessary, able to accommodate a richer set of models with a flexible structure of unobserved state space.

Secondly, our proposed method significantly reduces the computational burden for complicated dynamic models with large state space and many continuous variables, while suffering from limited loss of precision in estimation results. Unlike the traditional methods that require looping over all possible state space to calculate value functions in order to derive the policy function, our proposed method directly parametrizes the policy function and leverages the policy gradient theorem to update the policy parameters, avoiding the iterations over huge state space, resulting in significant time-saving in computation. In addition, different from the traditional methods where discretization of continuous state variables is required, our proposed method can directly work with continuous state variables, avoiding approximation through interpolation over grid points. Lastly, our method can handle complicated dynamic models. For instance, when the model has no finite dependence assumption (Hotz et al., 1994), the proposed method is still able to handle this situation.

Lastly, we provide a unified framework for estimating dynamic discrete choice models of various types by adopting the popular policy gradient method from the reinforcement learning literature and combining it with the simulated method of moments. If using traditional methods, there will be big differences in whether the model has unobservables or not. Different estimation techniques will be used for different scenarios. Using our proposed method, on the other hand, the same set of algorithms can be adapted to different groups of models, making the inference structure easy to implement.

Related Literature This paper relates to the large literature of estimating dynamic discrete choice models (Rust, 1987; Hotz and Miller, 1993; Hotz et al., 1994; Aguirregabiria and Mira, 2007; Arcidiacono and Jones, 2003; Arcidiacono and Miller, 2011; Gallant et al., 2018) and the literature on the identification of models with unobserved heterogeneity (Kasahara and Shimotsu, 2009; Hu and Shum, 2012; Hwang, 2024). Our paper differs in that we focus on the estimation of the dynamic discrete choice models with unobserved state variables, combining the policy gradient method from the reinforcement learning literature with indirect inference. While Gallant et al. (2018) also can estimate models with continuous and serially correlated unobserved state variables, they do not have any identification arguments. On the contrary, we build on the non-parametric identification results in Hu and Shum (2012) and show that our method can estimate models with very flexible conditions, including those with time-varying continuous and endogenous unobserved state variables.

Our paper is also related to the big stream of literature on reinforcement learning. Reinforcement learning is a sub-field under machine learning, where the goal of reinforcement learning is to find an optimal behavior strategy for the agent to obtain optimal rewards. This method has been widely adopted in artificial intelligence and operations research. However, reinforcement learning has received limited attention in economics. In particular, we focus on offline reinforcement learning (RL) (Lange et al., 2012), in which a policy (a sequence of actions) model is reinforced, by the feedback from the offline (previously collected) data including individuals' longitudinal observations and choices, to optimize sequential decisions that maximize a reward.

Finally, our paper is mostly related to the literature on the policy gradient method. Since proposed by Sutton et al. (1999), policy gradient methods in reinforcement learning (RL) have been extensively studied and utilized across many tasks due to their adaptability and straightforward implementation schemes (Kakade, 2001; Silver et al. (2014), Silver et al. (2014)). This method has been successfully applied to many fields, such as robotics and artificial intelligence (Peters and Schaal, 2006; Yu et al., 2017), auto driving (Li et al., 2022), and sequential medicine decisions (Jin et al., 2023). The policy gradient methods aim at modeling and optimizing the policy directly. The policy is usually modeled with a parameterized function with respect to $\theta, \pi_\theta(a|s)$. The value of the reward (objective) function depends on this policy and then various algorithms can be applied to optimize for the best reward. Using the policy gradient theorem by Sutton et al. (1999), the gradient of the value function can be transferred into the gradient of the policy function, thus providing a closed form for updating the policy parameters. In addition, our paper is related to the growing literature on estimating partially observable Markov decision processes (POMDPs) using policy gradient methods (Hong et al., 2023).

The rest of the paper is organized as follows. Section 2 introduces our proposed algorithm for dynamic discrete choice models with fully observed state space and provides Monte Carlo evidence to validate the effectiveness of our proposed method. Section 3 contains the main result of estimating dynamic discrete choice models with unobserved state space using our proposed method. In Section 4, we focus on a special case of a dynamic discrete choice model with fixed and discrete unobserved state variables and compare our method with existing methods. We present an empirical study in Section 5 for a dynamic model with time-varying and continuous unobserved state variables. Finally, Section 6 concludes this paper.

2 Reinforcement Learning Estimation of DDCs with Fully Observed State Space

In this section, we discuss our proposed algorithm for estimating the dynamic discrete choice models with fully observed state space. Consider a structural dynamic model with a process $\{\boldsymbol{X}_t, J_t\}$, where $\boldsymbol{X}_t \in \mathbb{R}^d$ stands for the observed state variables that evolve over time and $J_t \in \{0, 1\}$ is the agent's choice variable in period t [2]. We assume that the researcher observes a panel dataset consisting of an i.i.d random sample of $\{(\boldsymbol{X}_1, J_1), (\boldsymbol{X}_2, J_2), ..., (\boldsymbol{X}_T, J_T)\}$ for many agents i. Assume that the transitional model $f_{\boldsymbol{X}_t|\boldsymbol{X}_{t-1},J_{t-1}}$ is known or can be pre-estimated using the data. The flow utility of the agent in period t is denoted by $U(\boldsymbol{X}_t, J_t; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the set of parameters that determines the period-reward for the agent. The agent's objective is to choose a sequence of $\{J_1, J_2, ..., J_T\}$ to maximize her lifetime value.

Intuitively, our algorithm mainly contains two loops. In the outer loop, we use the Simulated Method of Moments and search over the structural parameter space to find the set of parameters that can match the simulated moments with the data moments. In the inner loop, we condition on the structural parameters in the current outer loop and use the Policy Gradient Method from reinforcement learning to update the policy function parameters. After the convergence of the inner loop, we obtain a mapping from the deep parameters in the outer loop to the optimal policy function parameters in the inner loop. After the convergence of the outer loop, we can then obtain the estimates for the structural parameters.

[2] We assume J_t is a binary choice variable for simplicity here. Our method can be used for estimating dynamic models with multi-level categorical or even continuous choice variables.

The key step of the algorithm, as suggested by the policy gradient method, is to parametrize the choice variable J_t as a function of the state variables, denoted by $\pi_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(J_t|\boldsymbol{X}_t)$. Specifically, we assume:

$$\Pr(J_t = 1|\boldsymbol{X}_t; \boldsymbol{\gamma}(\boldsymbol{\theta})) = \frac{\exp(\boldsymbol{X}_t\boldsymbol{\gamma}(\boldsymbol{\theta}))}{1+\exp(\boldsymbol{X}_t\boldsymbol{\gamma}(\boldsymbol{\theta}))}, \tag{2.1}$$

where $\boldsymbol{\gamma}(\boldsymbol{\theta})$ stands for the policy function parameters that depend on the structural parameters $\boldsymbol{\theta}$. We use a logistic function to parametrize the policy function for a binary choice variable. For a continuous choice variable, a regression function with a normal error term can be used instead. We assume a linear relationship between the state variables $\boldsymbol{X}t$ and parameters $\boldsymbol{\gamma}$, which is common in practice, but the parametrization $\pi\boldsymbol{\gamma}(J_t|\boldsymbol{X}_t)$ can be flexible. For example, higher-order terms of $\boldsymbol{X}t$ or historical state variables $Xt-k$ can be included. As long as the gradient with respect to the parameters has a closed-form solution, this approach will work. Recent research in deep reinforcement learning (François-Lavet et al., 2018) uses deep neural networks to parametrize the policy function, enabling it to handle large state spaces in dynamic models.

Let $V(\boldsymbol{\gamma}(\boldsymbol{\theta}))$ denote the reward function that depends on the policy parameter $\boldsymbol{\gamma}(\boldsymbol{\theta})$. Using gradient ascent, we can move $\boldsymbol{\gamma}$ toward the direction suggested by the gradient $\nabla_{\boldsymbol{\gamma}}V(\boldsymbol{\gamma}(\boldsymbol{\theta}))$ to find the best for that produces the highest return:

$$\boldsymbol{\gamma}_{q+1} = \boldsymbol{\gamma}_q + s_q\nabla_{\boldsymbol{\gamma}}V(\boldsymbol{\gamma}_q(\boldsymbol{\theta})),$$

where q denotes the current iteration number, and s_q is a hyper-parameter that governs the step size of the update. It is nontrivial to calculate $\nabla_{\boldsymbol{\gamma}}V(\boldsymbol{\gamma}_q(\boldsymbol{\theta}))$ because it involves the action and the stationary distribution of states following the target section behavior, both are indirectly or indirectly determined by $\pi_{\boldsymbol{\gamma}(\boldsymbol{\theta})}(J_t|\boldsymbol{X}_t)$. Luckily, we can use the policy gradient theorem to simplify the derivative of the objective function to a function that does not involve the derivative of the state distribution. As proved in Sutton and Barto (2018), the policy gradient theorem states that:

$$\nabla_{\boldsymbol{\gamma}}V(\boldsymbol{\gamma}) = E_X\Big[E_J\big[Q^{\pi_{\boldsymbol{\gamma}}}(\boldsymbol{X}, J)\nabla_{\boldsymbol{\gamma}}\log\pi_{\boldsymbol{\gamma}}(J|\boldsymbol{X})\big]\Big],$$

where $Q^{\pi_{\boldsymbol{\gamma}}}(\boldsymbol{X}, J)$ is the state-action value function of the policy $\pi_{\boldsymbol{\gamma}}$. In other words, the gradient of the value function can be transferred to the gradient of the policy function, $\nabla_{\boldsymbol{\gamma}}\log\pi_{\boldsymbol{\gamma}}(J|\boldsymbol{X})$, which becomes tractable now. The expectations are taken over the choice probability of J and the transitional probability of the state variables. In the algorithm of searching for the optimal $\boldsymbol{\gamma}$, we use Monte Carlo simulation to calculate the expectations.

Building on the policy gradient method, we propose an algorithm that combines it with indirect inference to estimate structural parameters in dynamic discrete choice models. The policy gradient method aims to find the optimal policy, which is an intermediate step in our goal of estimating structural parameters. Our algorithm has two layers: the inner loop uses the policy gradient method to solve for the optimal policy, while the outer loop estimates the structural parameters by matching data moments with simulated moments using SMM. Figure 1 presents the flow chart for the inner loop of our proposed method. The policy function in the inner loop depends on the structural parameter in the outer loop, creating a mapping from structural parameters to optimal policy parameters. In the inner loop, we parameterize the policy function $\gamma(\theta)$, simulate individual trajectories, and calculate the policy gradient. The value function gradient is then used to update the policy function parameters. In the outer loop, we use the optimal policy parameters $\gamma^*(\theta)$ to simulate trajectories and estimate the structural parameters by matching moments. The policy gradient method and SMM complement each other as both involve simulating individual life-cycle paths.

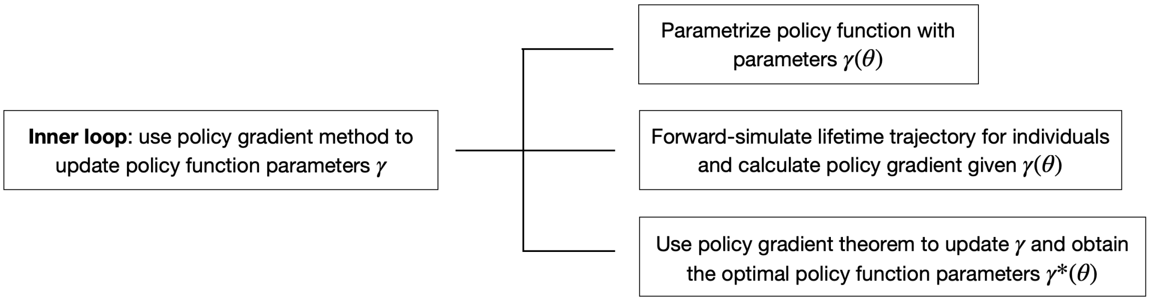
Parametrize policy function with parameters $\gamma(\theta)$
Inner loop: use policy gradient method to update policy function parameters γ
Forward-simulate lifetime trajectory for individuals and calculate policy gradient given $\gamma(\theta)$
Use policy gradient theorem to update γ and obtain the optimal policy function parameters $\gamma^*(\theta)$

Figure 1: Flow chart for the Inner Loop of the RL + Indirect inference Algorithm for Models With Fully Observed State Space

Algorithm A.1–Algorithm A.3 in Appendix A summarize our proposed method in detail. After presenting our proposed algorithm for estimating DDCs with fully observed state space, we perform two sets of Monte Carlo simulations. First, we validate the effectiveness of our method in accurately recovering the true parameters. Second, we demonstrate the computational advantages of our method over full-model

solution methods when estimating models with large state spaces.

In the first Monte Carlo simulation, we estimate a toy version of the Rust bus engine replacement model (Rust, 1987) using our RL + indirect inference method and compare the results with the NFXP algorithm. Model details and simulation results are in Section 1 of the Supplementary Appendix. We estimate the structural parameters (θ_1, θ_2) using Algorithm A.1–Algorithm A.3. First, we show rapid convergence of the lifetime reward during the inner loop iterations, indicating effective policy updates. Second, the policy table compares the RL-based optimal policy with the exact solution from backward induction, demonstrating similar engine replacement probabilities and lifetime values. Finally, the estimation table shows results closely aligned with the true values, comparable to the NFXP method. This simulation demonstrates the effectiveness of our method in estimating DDCs with a fully observed state space.

In the second Monte Carlo simulation, we extend our study to dynamic discrete choice models (DDCs) with large state spaces and continuous variables, focusing on a dynamic collective fertility model with a fully observed state space. Model setup, simulation details, and results are in Section 2 of the Supplementary Appendix. The model tracks various state variables for both the husband and wife each period and includes discrete fertility and continuous leisure and consumption choices. Unlike traditional methods that discretize continuous variables like wages, our method handles them directly. We follow Algorithm A.1–Algorithm A.3 and target 59 moments, including working hours and fertility outcomes. We compare results with full-model solutions using backward induction. The estimation results show that our method provides estimates close to the true parameters while reducing computational burden, demonstrating the effectiveness of our method in estimating complex DDCs.

In summary, by conducting the two sets of Monte Carlo simulations, we have demonstrated the effectiveness of our proposed method in recovering the true parameters in DDCs with fully observed state space, with results comparable to the traditional methods such as the NFXP algorithm. Most importantly, we show that using this method can significantly reduce the computational burden of DDCs with large state space and at the same time avoid the discretization of any continuous state variables. In the next section, we move to our main results, where we show that our proposed method is able to be accommodated for estimating DDCs with unobserved state variables.

3 Reinforcement Learning Estimation of DDCs with Unobserved State Variables

Dynamic models with unobserved heterogeneity have long been an important topic in economics. Estimation of dynamic models with unobserved state variables can be tricky, especially with time-varying unobservables. Motivated by these empirical needs, in this section, we apply our method one step further and focus on dynamic discrete choice models with time-varying continuous unobserved state variables. We show that when the model satisfies the assumptions listed in Hu and Shum (2012), our proposed RL + indirect inference method is able to recover the true parameters. We first discuss the model setup and assumptions required for identification, and we focus on a Rust model (Rust, 1987) with extensions and verify that it satisfies all the required assumptions. Then we discuss our proposed method for estimating these dynamic discrete models with continuous unobserved state variables and show how we apply this method to estimate the Rust model with this kind of unobserved variable. Finally, we present the estimation results for our Monte Carlo study and demonstrate that our method works well for estimating this type of model.

3.1 Model Identification

Consider a structural dynamic model with a process $\{X_t, S_t^*, J_t\}$, where X_t stands for the observed state variable(s) and S_t^* denotes the time-varying unobserved state variables (USVs), which are potentially observed by agents but not available to econometrician. $J_t \in \{0, 1\}$ is the agent's choice variable in period t. We assume that researchers observe a panel dataset consisting of an i.i.d random sample of $\{(X_1, J_1), (X_2, J_2), ..., (X_T, J_T)\}$ for many agents. For each agent i,

$$\{(X_1, J_1, S_1^*), (X_2, J_2, S_2^*), ..., (X_T, J_T, S_T^*)\}$$

is independently and randomly drawn from a bounded continuous distribution

$$f_{(X_1,J_1,S_1^*),(X_2,J_2,S_2^*),...,(X_T,J_T,S_T^*)}.$$

The assumptions are as follows:

Assumption 1. *(i) First-order Markov :* $f_{X_t,J_t,S_t^*|X_{t-1},J_{t-1},S_{t-1}^*,\Omega_{<t-1}} = f_{X_t,J_t,S_t^*|X_{t-1},J_{t-1},S_{t-1}^*}$, *where* $\Omega_{<t-1} = \{X_{t-2}, ..., X_1, J_{t-2}, ..., J_1, ..., S_{t-2}^*, ..., S_1^*\}$.

(ii) Limited feedback: $f_{X_t,J_t|X_{t-1},J_{t-1},S^*_t,S^*_{t-1}} = f_{X_t,J_t|X_{t-1},J_{t-1},S^*_t}$.

Assumption 2. *Invertibility. Let* $V_t \equiv g_t(W_t)$*, where* $W_t = \{X_t, J_t\}$*. The function* $g_t : \mathbb{R}^2 \to \mathbb{R}$*. Denote the supports of* V_t *and* W_t *as* $\mathcal{V}_t$ *and* $\mathcal{W}_t$*, respectively. Let* $L_{V_{t-2},\bar{w}_{t-1},\bar{w}_t,V_{t+1}}$ *denote the linear operator that maps from the* $\mathcal{L}^P$ *space of functions of* V_{t+1} *to the* $\mathcal{L}^P$ *splace of functions of* V_{t-2}*. There exists variable(s)* V_t *such that*
(i) for any $w_t \in \mathcal{W}_t$ *, there exists a* $w_{t-1} \in \mathcal{W}_{t-1}$ *and a neighborhood* $\mathcal{N}^2$ *around* (w_t, w_{t-1}) *such that, for any* $(\bar{w}_t, \bar{w}_{t-1}) \in \mathcal{N}^2$, $L_{V_{t-2},\bar{w}_{t-1},\bar{w}_t,V_{t+1}}$ *is one-to-one.*
(ii) for any $w_t \in \mathcal{W}_t$, $L_{V_{t+1}|w_t,S^*_t}$ *is one-to-one.* x *(iii) for any* $w_{t-1} \in \mathcal{W}_{t-1}$, L_{V_{t-2},w_{t-1},V_t} *is one-to-one.*

Assumption 3. *Uniqueness of spectral decomposition. For any* $w_t \in \mathcal{W}_t$ *and any* $\overline{s}^*_t \neq \widetilde{s}^*_t$*, there exists a* $w_{t-1} \in \mathcal{W}_{t-1}$ *and corresponding neighborhood* $\mathcal{N}^r$ *with* $\overline{w}_t \neq w_t, \overline{w}_{t-1} \neq w_{t-1}$*: (i)* $0 < k(w_t, \overline{w}_t, w_{t-1}, \overline{w}_{t-1}, s^*_t) < C < \infty$ *for any* $s^*_t \in \mathcal{S}^*_t$ *and some constant* C*;*
(ii) $k(w_t, \overline{w}_t, w_{t-1}, \overline{w}_{t-1}, \overline{s}^*_t) \neq k(w_t, \overline{w}_t, w_{t-1}, \overline{w}_{t-1}, \widetilde{s}^*_t)$*, where*

$$k(w_t, \overline{w}_t, w_{t-1}, \overline{w}_{t-1}, s^*_t) = \frac{f_{W_t|W_{t-1},S^*_t}(w_t|w_{t-1}, s^*_t) f_{W_t|W_{t-1},S^*_t}(\overline{w}_t|\overline{w}_{t-1}, s^*_t)}{f_{W_t|W_{t-1},S^*_t}(\overline{w}_t|w_{t-1}, s^*_t) f_{W_t|W_{t-1},S^*_t}(w_t|\overline{w}_{t-1}, s^*_t)}. \tag{3.1}$$

Assumption 4. *Monotonicity and normalization. For any* $w_t \in \mathcal{W}_t$*, there exists a known functional* G *such that* $G[f_{V_{t+1}|W_t,S^*_t}(\cdot|w_t, s^*_t)]$ *is monotonic in* s^*_t*.*
We normalize $s^*_t = G[f_{V_{t+1}|W_t,S^*_t}(\cdot|w_t, s^*_t)]$*.*

Applying Theorem 1 in Hu and Shum (2012), we have that

Lemma 1. *Under Assumptions 1-4, we have that:*
$f_{X_t,S^*_t|J_{t-1},X_{t-1},S^*_{t-1}}$ *is identified from* $f_{J_{t+1},X_{t+1},J_t,X_t,J_{t-1},X_{t-1},J_{t-2},X_{t-2},J_{t-3},X_{t-3}}$.
Initial condition $f_{X_{t-3},S^*_{t-3},J_{t-3}}$ *is identified from* $f_{J_{t+1},X_{t+1},J_t,X_t,J_{t-1},X_{t-1},J_{t-2},X_{t-2},J_{t-3},X_{t-3}}$.

Lemma 1 states that, a total of five consecutive periods of observed state variables and choice variables are needed to non-parametrically identify the conditional distribution $f_{X_t,S^*_t|J_{t-1},X_{t-1},S^*_{t-1}}$. At the same time, with the assumption that the initial distributions of X and S^* are independent, we can also identify the initial condition $f_{X_{t-3},S^*_{t-3},J_{t-3}}$ using the same set of observed state and choice variables.

3.1.1 A Rust Model with Extensions

Consider Rust (1987) bus engine replacement problem, where a bus operator periodically replaces buses' engines to minimize the expected cost of operating the buses. The choice variable $J_{it} \in \{0, 1\}$ is whether to replace the engine or not in each period t for bus company i. The state variable $X_{it} \in \mathbb{R}$ is the accumulated mileage for bus

i. We extend the original model to incorporate an unobserved state variable $S_{it} \in \mathbb{R}$, which is the condition of the bus i at time t. A higher S_{it} implies a better bus condition, which results in a lower cost for doing maintenance work for the bus and a faster accumulation in mileage. In summary, the condition of the bus is an unobservable state variable that affects both the expected operating cost and the transition of the mileage for the bus. For notational simplicity, the individual index i will be suppressed when understood from the context. We introduce the data-generating model in detail below.

We assume the transitional process for the bus condition S_t^* is fully exogenous and follows an AR(1) process:

$$S_t^* = \lambda_1 S_{t-1}^* + \nu_t; \quad \nu_t \sim N(0, \sigma_2^2) \tag{3.2}$$

The transitional process for the mileage X_t if $J_t = 0$ is:

$$X_{t+1} = X_t\big[1 + \lambda_3 \exp(\eta_{t+1} + \lambda_2 S_{t+1}^*)\big], \quad f_{\eta_{t+1}}(\eta) = \exp(\eta - e^{\eta}) \tag{3.3}$$

When the engine is not replaced ($J_t = 0$), X_{t+1} accumulates based on last period's mileage X_t and the increment depends on the parameters (λ_2, λ_3), a random shock $\eta_{t+1} \in \mathbb{R}$ that follows an extreme value distribution, and the condition of the bus, S_{t+1}^*, at time period of $t+1$. Equation 3.3 implies that the condition of the bus raises the accumulation of mileage. The higher S_t^* is, the faster the mileage will accumulate since the bus will be driven more often for longer trips. Notice that we assume S_t^* is realized before X_t, so that X_t depends on S_t^*. Moreover, we also assume the initial mileage $X_1 > 0$, such that $X_t > 0$ for all t, and we have that for given X_t, $X_{t+1} \in (X_t, +\infty)$.

The transitional process for mileage X_t if $J_t = 1$ is:

$$p(X_{t+1}|X_t, J_t, \beta_2) = \beta_2 \exp(-\beta_2 X_{t+1}) \tag{3.4}$$

When the bus engine is replaced ($J_t = 1$), X_{t+1} does not depend on the previous mileage or the condition of the bus anymore and instead follows an exponential distribution with the parameter β_2. This assumption is to capture the fact that the Rust bus engine model has limited dependence. When engine replacement happens, the accumulation of the mileage in the next period restarts, without depending on the historical path anymore. To satisfy the assumption that $X_t > 0$ for all t, we assume the mileage follows an exponential distribution with a parameter β_2 after the engine replacement, instead of directly becoming zero after the engine is replaced.

The initial values of state variables (S_1^*, X_1) are independent and have the following distributions:

$$S_1^* \sim N(\alpha_1, \sigma_1^2); \quad X_1 \sim \exp(\beta_1), \tag{3.5}$$

where we assume the condition S_t^* in the initial period $t = 1$ follows a Normal distribution with mean α_1 and variance σ_1^2; whereas the mileage X_t in the initial period $t = 1$ follows a exponential distribution with the parameter of β_1.

The flow utility in period t is specified as follows:

$$\begin{aligned} U(X_t, J_t, S_t^*, \epsilon_{jt}) &= u(X_t, J_t, S_t^*; \theta_1, \theta_2, \theta_3) + \epsilon_{jt} \\ &= -\theta_1 X_t \exp(-S_t^*) - \theta_2 \mathbb{I}(J_t = 1) + \theta_3 S_t^* + \epsilon_{jt}, \end{aligned} \tag{3.6}$$

where ϵ_{jt} is a discrete-choice-specific idiosyncratic shock that follows the type I extreme value distribution, which affects the engine-change decision.

The utility function reflects that a lower value of S_t indicates worse bus conditions, leading to higher maintenance costs. To capture this, we include an interaction term between X_t and $\exp(-S_t)$, representing the impact of bus condition on maintenance costs at mileage X_t. Additionally, the cost of replacing the engine is represented by $-\theta_2 \times \mathbb{I}(J_t = 1)$ in the utility function. Lastly, we assume that bus condition directly influences the utility, with better bus condition providing positive utility to the company, along with reducing maintenance costs for a given mileage.

Each period, a bus company chooses whether to replace the bus engine or not to maximize its discounted future value. The value function of the bus company is:

$$V(X_t, S_t^*, \epsilon_{jt}) = \max_{J_t \in \{0,1\}} \left\{ u(X_t, J_t, S_t^*; \theta_1, \theta_2, \theta_3) + \epsilon_{jt} + \beta E[V(X_{t+1}, S_{t+1}^*, \epsilon_{jt+1}) | X_{it}, S_{it}, J_t] \right\} \tag{3.7}$$

In order to adopt our algorithm to estimate this model, we first need to show that the model satisfies Assumption 1–Assumption 4 in Section 3.1. When $J_t = 1$, the transitional model of X_t can be directly estimated using $X_{t+1} | X_t, J_t = 1$, and the transitional model of S_t^* does not involve J_t. Therefore, when proving that each assumption is satisfied, we focus on the case when $J_t = 0$. As long as we can show that when $J_t = 0$ the model is identified, we can conclude that the whole model is identified. Detailed proof for the four assumptions is shown in Appendix B.

Having proved that all the four assumptions are satisfied, we can apply Lemma 1 to reach identification, where $f_{X_t,S_t^*|J_{t-1},X_{t-1},S_{t-1}^*}$ and the initial condition $f_{X_{t-3},S_{t-3}^*,J_{t-3}}$ are identified from the joint distribution of five consecutive periods of observed state variables and choice variables $f_{J_{t+1},X_{t+1},J_t,X_t,J_{t-1},X_{t-1},J_{t-2},X_{t-2},J_{t-3},X_{t-3}}$.

3.2 RL + Indirect Inference Estimation

In this section, we discuss how our proposed method can be accommodated to estimate DDCs with unobserved state variables. Similar to the case without unobserved state variables, the first step is to parametrize the choice variable as a function of both the observed and unobserved state variables:

$$\Pr(J_t = 1|X_t, S_t^*, t; \boldsymbol{\gamma}) = \frac{\exp(\gamma_0 + \gamma_1 t + \gamma_2 X_t + \gamma_3 S_t^*)}{1 + \exp(\gamma_0 + \gamma_1 t + \gamma_2 X_t + \gamma_3 S_t^*)}. \tag{3.8}$$

We use the logistic model as the link function due to the binary choice variable, and apply a simple linear model to describe how state variables affect the intermediate value. More complex models can be used to capture the nonlinear relationship between state variables and choice variables.

Figure 2 shows the flow chart of our proposed method. Similar to Figure 1, the algorithm contains two layers of loops, where the outer loop searches over the parameter space by matching the data moments and the simulated moments and, given the set of parameters in the current outer loop, the inner loop solves for the optimal policy using the policy gradient method, by forward simulating the lifetime trajectory of each individual and update the policy parameters using stochastic gradient descent method. Different from the algorithm in Section 2 for fully observed dynamic models, now we both have unknown structural parameters $\boldsymbol{\theta}$ and unknown transitional model parameters $\boldsymbol{\xi}$ to estimate. To summarize, we have three sets of parameters now: $\{\boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{\xi}\}$, which stands for the policy function parameters, the structural parameters, and the transitional model parameters. The algorithm in the inner loop updates $\boldsymbol{\gamma}$, conditional on not only $\boldsymbol{\theta}$ but also $\boldsymbol{\xi}$. The key of our method is that, in the outer loop of the Simulated Method of Moments, we search over the combined parameter space for $\{\boldsymbol{\theta}, \boldsymbol{\xi}\}$ by targeting moments consistent with the identification results in Lemma 1. Because of the identification results in Hu and Shum (2012), we only need to match moments of observed state variables and choice variables to identify $\boldsymbol{\theta}$ and

$\boldsymbol{\xi}$ separately. At the same time, however, we need to match the moments of four periods' observed data to identify the parameters.

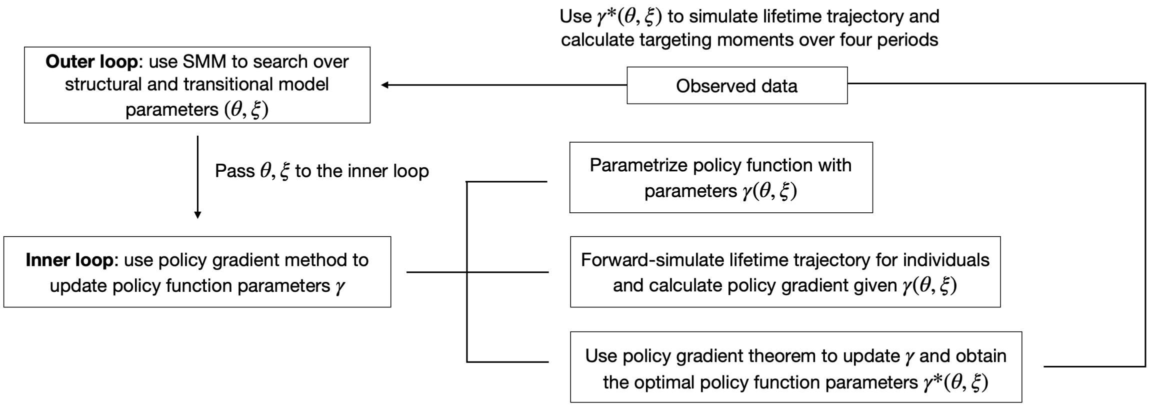
Use $\gamma^*(\theta,\xi)$ to simulate lifetime trajectory and calculate targeting moments over four periods
Outer loop: use SMM to search over structural and transitional model parameters (θ,ξ)
Observed data
Pass θ,ξ to the inner loop
Parametrize policy function with parameters $\gamma(\theta,\xi)$
Inner loop: use policy gradient method to update policy function parameters γ
Forward-simulate lifetime trajectory for individuals and calculate policy gradient given $\gamma(\theta,\xi)$
Use policy gradient theorem to update γ and obtain the optimal policy function parameters $\gamma^*(\theta,\xi)$

Figure 2: Flow chart for the RL + Indirect inference Algorithm with Unobserved State Variables

We present detailed algorithms for our method in Algorithm 1–Algorithm 3. Similar to Algorithm A.1–A.3, the algorithm has two loops: the outer loop searches the parameter space by matching data and simulated moments, while the inner loop solves for the optimal policy using the policy gradient method. This is done by forward simulating individual trajectories and updating policy parameters via stochastic gradient descent. Unlike the algorithm in Section 2 for fully observed dynamic models, we now estimate both structural parameters $\boldsymbol{\theta}$ and transitional model parameters $\boldsymbol{\xi}$. Thus, we have three sets of parameters: $\boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{\xi}$, representing policy function, structural, and transitional model parameters. In the outer loop of the Simulated Method of Moments, we search over the combined parameter space for $\boldsymbol{\theta}, \boldsymbol{\xi}$, targeting moments consistent with the identification results in Lemma 1. The inner loop updates $\boldsymbol{\gamma}$, conditional on both $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$. The reason why our proposed algorithm can separately estimate $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ relies on the identification results in Hu and Shum (2012).

Algorithm 1 shows the steps of conducting forward simulation for obtaining an individual's lifetime trajectory. The input of policy parameter $\boldsymbol{\gamma}$ depends on both the structural parameter $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$. Since S^* is unknown, the first step is to sample S^*_{i1} using the initial distribution of S^* that depends on the initial distribution parameter. Combining observed state variables X_{i1} and sampled unobserved state variable S^*_{i1}, we can sample the choice J_{i1} using the policy function in Equation 3.8, conditional on the policy parameter $\boldsymbol{\gamma}$. After these steps, we obtained the state and choice variables in the first period. We can then move to the second period by using the transitional models of S^* and X, given the transitional model parameters $\boldsymbol{\xi}$ [3]. Similar to the first period, we can then use the policy function in Equation 3.8 to sample the choice in this period, given $(\widehat{X}_{i2}, \widehat{S}^*_{i2})$ and the policy parameters $\boldsymbol{\gamma}^*(\boldsymbol{\theta}, \boldsymbol{\xi})$. We can continue the same procedure and move forward to simulating the individual's lifetime trajectory.

Algorithm 2 discusses the procedure of using stochastic gradient descent for optimizing $\boldsymbol{\gamma}$. This part is similar to Algorithm A.2, except that the state space now contains unobserved variables S^*, and $\boldsymbol{\gamma}$ now depends on both $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$. When simulating the individual's lifetime trajectory, we follow Algorithm 1 to generate $\widehat{\boldsymbol{D}}_i = (\widehat{\boldsymbol{S}}_i, \widehat{\boldsymbol{X}}_i, \widehat{\boldsymbol{J}}_i)$ that contains both unobserved, observed, and choice variables. We can then calculate the lifetime value and the gradient of the policy function and use the policy gradient theorem to calculate the gradient of the value function to update the policy function parameters $\boldsymbol{\gamma}$.

Finally, Algorithm 3 presents the outer loop of conducting indirect inference for estimating structural parameters and transitional model parameters. Unlike Algorithm A.3 where we only need to estimate structural parameters $\boldsymbol{\theta}$, here we also need to estimate $\boldsymbol{\xi}$ because we cannot observe the transitional process of the unobserved state variable S^*. Therefore, embedding Algorithm A.3 and Algorithm 1 inside the loop, the algorithm minimizes the distance between the data and the simulated moments by choosing $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ at the same time. In order to separately identify these two sets of parameters, we need to adopt the identification results in Lemma 1, where we use the joint distribution of $(J_{t+1}, X_{t+1}, J_t, X_t, J_{t-1}, X_{t-1}, J_{t-2}, X_{t-2}, J_{t-3}, X_{t-3})$ to identify $f_{X_t, S^*_t | J_{t-1}, X_{t-1}, S^*_{t-1}}$ and the initial condition $f_{X_{t-3}, S^*_{t-3}, J_{t-3}}$. Therefore, the targeted moments should contain at least five consecutive periods of observed state variables and choice variables in order to separately identify $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$.

3.2.1 Estimation of the Rust Model Using RL + Indirect Inference Method

In this section, we discuss how to adopt our proposed method above to estimate this dynamic discrete choice model with continuous and time-varying unobserved state variables. The first step is to parametrize the choice variable as a function of observed and unobserved state variables using Equation 3.8. The second step is to parametrize the transitions of the observed and unobserved state variables and the initial distributions of the observed and unobserved state variables according to

[3] Notice that we assume S^* is realized before X does. Hence, X_t depends on S^*_t, not S^*_{t-1}.

Algorithm 1 Forward Simulation for Obtaining Individual's Lifetime Trajectory

```
Input: policy parameters $\boldsymbol{\gamma}(\boldsymbol{\theta}, \boldsymbol{\xi})$, utility parameters $\boldsymbol{\theta}$, transitional model parameter $\boldsymbol{\xi}$, initial state $X_{i1}$ for individual $i$
Sample $\widehat{S}^*_{i1}$ using the initial distribution of $S^*$ given $\boldsymbol{\xi}$
Sample $\widehat{J}_{i1}$ using Equation 3.8 given $(X_{i1}, \widehat{S}^*_{i1}; \boldsymbol{\gamma}(\boldsymbol{\theta}, \boldsymbol{\xi}))$
Sample $\widehat{S}^*_{i2}$ using the transitional model of $S^*$ given $(\widehat{S}^*_{i1}; \boldsymbol{\xi})$
Sample $\widehat{X}_{i2}$ using the transitional model of $X$ given $(\widehat{X}_{i1}, \widehat{J}_{i1}, \widehat{S}^*_{i2}; \boldsymbol{\xi})$
Sample $\widehat{J}_{i2}$ using Equation 3.8 given $(\widehat{X}_{i2}, \widehat{S}^*_{i2}; \boldsymbol{\gamma}(\boldsymbol{\theta}, \boldsymbol{\xi}))$
for $t = 3, \ldots, T$ do
    Sample $\widehat{S}^*_{it}$, $\widehat{X}_{it}$, and $\widehat{J}_{it}$ conditional on $\boldsymbol{\theta}$, $\boldsymbol{\xi}$, and $\boldsymbol{\gamma}(\boldsymbol{\theta}, \boldsymbol{\xi})$
end for
Obtain the final dataset $\widehat{\boldsymbol{D}}_i = (\widehat{\boldsymbol{S}}_i, \widehat{\boldsymbol{X}}_i, \widehat{\boldsymbol{J}}_i)$
Output: lifetime trajectory $\widehat{\boldsymbol{D}}_i$ for individual $i$
```

Algorithm 2 Stochastic Gradient Descent for optimizing $\boldsymbol{\gamma}$

```
Input: initial value $\boldsymbol{\gamma}_0$ for the policy parameters $\boldsymbol{\gamma}$, deep parameters $\boldsymbol{\theta}$, transitional model parameter $\boldsymbol{\xi}$, step size $s_q$, batch size I, initial state $\{X_{11}, X_{21}, ..., X_{I1}\}$ from the batch data $D_I$.
Initialize: $\boldsymbol{\gamma}_1 \leftarrow \boldsymbol{\gamma}_0$
for $q = 1, \ldots, Q$ do
    for $i = 1, 2, \ldots, I$ do
        Obtain $\widehat{\boldsymbol{D}}_i = (\widehat{\boldsymbol{S}}_i, \widehat{\boldsymbol{X}}_i, \widehat{\boldsymbol{J}}_i)$ following the procedure in Algorithm 1
        Calculate the lifetime value $V_i(\boldsymbol{\gamma}_q)$ and gradient $\nabla_{\boldsymbol{\gamma}}\log(\prod_{t>=1}\pi(\widehat{J}_{it}|\widehat{S}^*_{it}, \widehat{X}_{it}, t; \boldsymbol{\gamma}))$
    end for
    Average the lifetime value and gradient: $V(\boldsymbol{\gamma}_q) := \frac{1}{I}\sum_{i=1}^{I} V_i(\boldsymbol{\gamma}_q)$;
    $\nabla_{\boldsymbol{\gamma}}\log(\prod_{t>=1}\pi(J_t|\widehat{S}^*_t, \widehat{X}_t, t; \boldsymbol{\gamma})) := \frac{1}{I}\sum_{i=1}^{I} \nabla_{\boldsymbol{\gamma}}\log(\prod_{t>=1}\pi(\widehat{J}_{it}|\widehat{S}^*_{it}, \widehat{X}_{it}, t; \boldsymbol{\gamma}))$
    Update $\nabla_{\boldsymbol{\gamma}} V(\boldsymbol{\gamma}_q) \leftarrow V(\boldsymbol{\gamma}_q)\nabla_{\boldsymbol{\gamma}}\log(\prod_{t>=1}\pi(J_t|\widehat{S}^*_t, \widehat{X}_t, t; \boldsymbol{\gamma}))$
    Update $\boldsymbol{\gamma}_{q+1} \leftarrow \boldsymbol{\gamma}_q + s_q\nabla_{\boldsymbol{\gamma}} V(\boldsymbol{\gamma}_q)$
end for
Output: $\boldsymbol{\gamma}^* = \boldsymbol{\gamma}_{q^*}$
```

Algorithm 3 Indirect Inference for estimating $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$

```
Input: initial value $\boldsymbol{\theta}_0$, initial value $\boldsymbol{\xi}_0$; initial state $\{X_{11}, X_{21}, ..., X_{N1}\}$ from data D; data moments $\boldsymbol{\kappa}$.
Initialize: $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}_0$; $\boldsymbol{\xi}_1 \leftarrow \boldsymbol{\xi}_0$
for $k = 1, \ldots, K$ do
    Calculate $\boldsymbol{\gamma}^*(\boldsymbol{\theta}_k, \boldsymbol{\xi}_k)$ following Algorithm 2
    for $i = 1, 2, \ldots, N$ do
        Obtain $\widehat{\boldsymbol{D}}_i = (\widehat{\boldsymbol{S}}_i, \widehat{\boldsymbol{X}}_i, \widehat{\boldsymbol{J}}_i)$ using $\boldsymbol{\gamma}^*(\boldsymbol{\theta}_k, \boldsymbol{\xi}_k)$ following Algorithm 1
    end for
    Calculate simulated moments $\widehat{\boldsymbol{\kappa}}$ using $(\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{J}})$ according to targeted moments
    $(\boldsymbol{\theta}_{k+1}, \boldsymbol{\xi}_{k+1}) \leftarrow \min \text{Dis}(\boldsymbol{\kappa}, \widehat{\boldsymbol{\kappa}})$
end for
Output: $\boldsymbol{\theta}^* = \boldsymbol{\theta}_{k^*}$, $\boldsymbol{\xi}^* = \boldsymbol{\xi}_{k^*}$
```

Equation 3.2–Equation 3.5. Since we can observe X_1 and $X_{t+1}|J_t = 1$ from the data, we pre-estimate (β_1, β_2) directly from the data. We use $(\widehat{\beta}_1, \widehat{\beta}_2)$ to denote the estimated parameters, and for identification purposes we normalize $(\lambda_2 = 0.8, \lambda_3 = 0.2)$ in Equation 3.3. Combining these together, we obtain the initial distribution for X_1 and transitional process of the mileage as follows:

$$S_1^* \sim N(\alpha_1, \sigma_1^2); \quad X_1 \sim \exp(\widehat{\beta}_1), \tag{3.9}$$

The transitional process for mileage X_{it} is:

$$\begin{cases} X_{t+1} = X_t\big[1 + 0.2\exp(\eta_{t+1} + 0.8S_{t+1}^*)\big] & \text{if } J_t = 0; \quad f_{\eta_{t+1}}(\eta) = \exp(\eta - e^{\eta}) \\ p(X_{t+1}|X_t, J_t, \widehat{\beta}_2) = \beta_2\exp(-\widehat{\beta}_2 X_{t+1}) & \text{if } J_t = 1 \end{cases}. \tag{3.10}$$

With the pre-estimated and normalized parameters, we have seven remaining parameters to be estimated structurally. Out of the seven parameters, three of them are utility parameters, and four of them are transitional model parameters. We follow Algorithm 1–Algorithm 3 to estimate these parameters.

In Algorithm 1, the input of the algorithm includes the policy parameters $\boldsymbol{\gamma}^* = (\gamma_0, \gamma_1, \gamma_2, \gamma_3)$, the utility function parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$, and the transitional model parameters $\boldsymbol{\xi} = (\alpha_1, \sigma_1, \sigma_2, \lambda_1)$. We also need to obtain the initial mileage X_1 for the individual directly from the data since it is observed. Given all the structural parameters, transitional model parameters, and initial observed state variables, we can simulate the lifetime trajectory of the individual.

Firstly, we sample the initial unobserved state variable $\widehat{S}_{i1}^{*}$ from the initial distribution in Equation 3.9, given the parameter (α_1, σ_1^2). Secondly, we sample the choice variable $\widehat{J}_{i1}$ from the policy function in Equation 3.8, given the first period's state variables $(X_{i1}, \widehat{S}_{i1}^{*})$ and the policy function parameters $\boldsymbol{\gamma}^{*}(\boldsymbol{\theta}, \boldsymbol{\xi})$. Here, the dependence of $\boldsymbol{\gamma}^{*}$ on $(\boldsymbol{\theta}, \boldsymbol{\xi})$ means that the optimal policy function parameters depend on the structural parameters of the model. Up till now, we have obtained the state and choice variables in the first period: $(X_{i1}, \widehat{S}_{i1}^{*}, \widehat{J}_{i1})$. We move forward to the second period by using the transitional models for X and S^{*}. Specifically, we sample $\widehat{S}_{i2}^{*}$ using Equation 3.2 given last period's unobserved state variable $\widehat{S}_{i1}^{*}$ and transition parameters (λ_1, σ_2). At the same time, we sample $\widehat{X}_{i2}$ using Equation 3.10, given last period's state and choice variables $(\widehat{X}_{i1}, \widehat{S}_{i1}^{*}, \widehat{J}_{i1})$ and transition parameters $(\lambda_2, \widehat{\beta}_2)$. Finally, with the observed and unobserved state variables obtained in the second period, we use Equation 3.8 again to sample $\widehat{J}_{i2}$, conditional on $(\widehat{X}_{i2}, \widehat{S}_{i2}^{*})$ and policy function parameters $\boldsymbol{\gamma}^{*}(\boldsymbol{\theta}, \boldsymbol{\xi})$. Using the same strategy, we continue moving forward and sample for $(\widehat{S}_{it}^{*}, \widehat{X}_{it}, \widehat{J}_{it})$ for the remaining period until time T. This results in the final simulated dataset $\widehat{\boldsymbol{D}}_i = (\widehat{\boldsymbol{S}}_i, \widehat{\boldsymbol{X}}_i, \widehat{\boldsymbol{J}}_i)$, where $\widehat{\boldsymbol{S}}_i = \{\widehat{S}_{i1}^{*}, \widehat{S}_{i2}^{*}, ..., \widehat{S}_{iT}^{*}\}$, $\widehat{\boldsymbol{X}}_i = \{\widehat{X}_{i1}, \widehat{X}_{i2}, ..., \widehat{X}_{iT}\}$, and $\widehat{\boldsymbol{J}}_i = \{\widehat{J}_{i1}, \widehat{J}_{i2}, ..., \widehat{J}_{iT}\}$.

The inner loop of our algorithm contains the Stochastic Gradient Descent algorithm to optimize the policy function parameters $\boldsymbol{\gamma}$ as shown in Algorithm 2, and the outer loop of our algorithm adopts Simulated Method of Moments to estimate structural parameters $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ following Algorithm 3. Our goal is to find the set of parameters $(\boldsymbol{\theta}^{*}, \boldsymbol{\xi}^{*})$ to minimize the distance between the simulated and data moments. Therefore, our criterion function is a normalized metric of the distance between $\boldsymbol{\kappa}$ and $\widehat{\boldsymbol{\kappa}}$, where $\boldsymbol{\kappa}$ stands for the corresponding data moments. Notice that since we do not observe S in the real data, it is not included in the moments that we try to match. Only $(\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{J}})$ will be used in the SMM for estimating the structural parameters.

What remains to be discussed is the choice of moments to match in the SMM to estimate parameters in Algorithm 3. Using the result in Lemma 1, we target five consecutive periods of state and choice variables to non-parametrically identify the transitional model of $f(X_t, S_t^{*} | J_{t-1}, X_{t-1}, S_{t-1}^{*})$ and the initial condition $f(X_{t-3}, S_{t-3}^{*}, J_{t-3})$. We match the regression coefficients, mean, standard deviation, and correlation moments for the five periods of state and choice variables. Details of the targeted moments are provided in Section 1 of the Online Appendix.

3.3 Monte Carlo Evidence: Simulation Results

We simulate data for 1000 buses that live for 10 periods and make decisions in each period. The data is generated by deriving the value functions at each state using backward induction. Since both the unobserved state variable S_t^* and the observed state variable X_t are continuous, we need to discretize them before solving the full model. We discretize S_t^* into 5 grid points and discretize X_t into 20 grid points. We use S^* and X^* to denote the discretized version of S^* and X. Since neither of S_t^* and X_t has stationary distributions over time, the grid points for S_t^* and X_t are time-dependent, resulting in $5 \times 10 = 50$ grid points for S^* and $20 \times 10 = 200$ grid points for X. After getting the discretized S^* and X, we use sampling methods to calculate the transitional matrix of $Pr(S_{t+1}^*|S_t^*)$ and $Pr(X_{t+1}^*|X_t^*, S_t^*, J_t)$. The dimension of $Pr(S_{t+1}^*|S_t^*)$ is $9 \times 5 \times 5$ and the dimension of $Pr(X_{t+1}^*|X_t^*, S_t^*, J_t)$ is $9 \times 20 \times 20 \times 5 \times 2$. After the discretization of continuous state variables, we can solve the optimal policy for each state by starting from the last period and moving to the first period. With the optimal policy, we then start each bus engine in the first period by sampling from the initial distribution for X and S and forward-simulating the choices until period 10. When calculating the expected value in each period conditional on the last period's state and choice variables, we adopt the simulated transitional matrices $Pr(S_{t+1}^*|S_t^*)$ and $Pr(X_{t+1}^*|X_t^*, S_t^*, J_t)$. Summarizing the dimensions of the problem, we have 2 choices (changing the engine or not), 10 periods of data, 5 unobserved states, and 30 possible mileages in each period. The number of states is therefore $2 \times 10 \times 5 \times 20 = 2,000$.

With the generated data, we use the proposed algorithm in Section 3.2.1 to estimate the seven structural parameters. Following the proposed method, we pre-estimate β_1 and β_2 directly from the data and normalize ($\lambda_2 = 0.8, \lambda_3 = 0.2$). The remaining three utility parameters and four transitional model parameters are estimated using the reinforcement learning + indirect inference method. When running Algorithm 2, we set the step size $s_q = 10^{-4}$ and choose the batch size $I = 1$. When running Algorithm 3, we use the 111 moments specified in Section 1 of the Online Appendix in the procedure of SMM, where we use the standard errors of the moments as the weighting matrix when calculating the criterion function for minimization.

Table 1 presents the estimation results for the simulation study. There are seven parameters in total to estimate, with three of them being the utility function parameters and four of them being transitional-model parameters. The second, fourth, and fifth columns in the table present three sets of true parameters that generated the data. The third, fifth, and last columns show the corresponding estimation results for each set of true parameters, where we present the mean and standard deviations of the estimated parameters in 400 simulations. For all of the three estimations, we use 5000 iterations when calculating the optimal policy function in each outer loop of deep parameters.

For the three sets of estimations, we tried different combinations of utility parameters to check whether our algorithm can recover the true parameters under various true values while holding the transitional model parameters unchanged. From the table, it is obvious that in all three sets of simulations, our method produces estimates centered around the true values. All the 21 true parameters fall within the 95% confidence interval of the estimated values. This simulation result serves as evidence that our method can reasonably recover the underlying utility and transitional model parameters, even with the presence of time-varying continuous unobserved state variables.

4 DDCs with Discrete Unobserved Heterogeneity: A Special Case

After discussing the general case of dynamic discrete choice models with time-varying and continuous state variables, in this section, we look at a special case that is often seen in reality: a finite mixture model. We adopt a smilar model setup to the simulation study in Arcidiacono and Miller (2011) and compare our estimation results with the results in their paper to validate that our method works well for this special case.

We focus on the classic Rust bus engine problem (Rust, 1987) with an unobserved state variable $S_t^* \in \{1, 2\}$ that is assumed to be discrete and fixed over time. Other than S_t^*, the state space also contains an observed variable X_t, which is the accumulated mileage of the bus. The model has one binary choice variable, $J_t \in \{0, 1\}$, which denotes the engine replacement decision.

The flow utility in period t is presented in Equation 4.1. The payoff of keeping the current engine ($J_t = 0$), depends on both the unobserved state variable S_t^* and accumulated mileage X_t of the bus. S_t^* can be interpreted as the condition of the bus engine, where $S_t^* = 1$ denotes a bad condition and $S_t^* = 2$ denotes a good condition of the bus engine. Therefore, by directly letting S_t^* enter the utility function, we

Table 1: Estimation Results for Simulation Study: Continuous Unobserved State Variables

Parameters	**True (1)**	**Est. (1)**	**True (2)**	**Est. (2)**	**True (3)**	**Est. (3)**
			Utility parameters			
θ_1	0.5	0.494	1	1.072	1	1.067
		(0.057)		(0.078)		(0.080)
θ_2	5	4.857	5	4.869	5	4.996
		(0.502)		(0.531)		(0.563)
θ_3	2	2.011	2	1.833	3	2.816
		(0.458)		(0.398)		(0.383)
			Transitional process parameters			
α_1	1	0.976	1	0.985	1	0.978
		(0.093)		(0.096)		(0.092)
σ_1	1	0.980	1	0.976	1	0.962
		(0.090)		(0.101)		(0.100)
λ_1	0.5	0.432	0.5	0.486	0.5	0.488
		(0.063)		(0.067)		(0.067)
σ_2	0.2	0.184	0.2	0.218	0.2	0.211
		(0.038)		(0.039)		(0.039)

Note: This table presents the estimation results for the simulation study for the Rust bus engine model with a time-varying continuous unobserved state variable. We fix λ_2 to be 0.8. There are 7 remaining structural parameters to be estimated, where the top panel of the table presents the 3 utility parameters and the bottom panel presents the 4 parameters in the transitional model. The sample size of the data set is 1000 bus companies × 10 time periods. We test for three sets of parameters, where the second, fourth, and sixth columns show the true parameters, and the third, fifth, and last columns present the corresponding mean and standard deviations of the estimation results in 400 simulations. 5000 iterations are used to estimate the optimal policy function in each inner loop.

are assuming that $S_t^* = 2$ can bring more utility to the bus company than $S_t^* = 1$. In addition, we assume that maintenance costs increase linearly with accumulated mileage up to 25 and then flatten out. This is denoted by $\theta_1 \min\{X_t, 25\}$ in the utility function. For simplicity, we normalize the flow utility to zero when the engine is replaced ($J_t = 1$).

$$u(X_t, S^*) = \begin{cases} \theta_0 + \theta_1 \text{min}\{X_t, 25\} + \theta_2 S_t^* & \text{if } J_t = 0 \\ 0 & \text{if } J_t = 1 \end{cases} . \tag{4.1}$$

Mileage X_t accumulates in increments of 0.125. The accumulation of the mileage depends on both the decision to replace the engine J_t, and the previous mileage X_t. When $J_t = 0$, the probability of X_{t+1} conditional on X_t is denoted in Equation 4.2:

$$f(X_{t+1}|X_t) = \begin{cases} \exp(-(X_{t+1} - X_t)) - \exp(-(X_{t+1} + 0.125 - X_t)) \text{ if } X_{t+1} \geq X_t \\ 0, \text{ otherwise} \end{cases} . \tag{4.2}$$

This implies that the mileage transition follows a discrete analog of an exponential distribution. The reason for using a discrete version of an exponential distribution is to avoid discretizing any continuous state space variable. When engine replacement happens ($J_t = 1$), then $X_{t+1} = 0$ with probability $= 1$. This is to assume that the problem has limited dependence, with mileage going to zero with certainty if the engine is replaced.

In terms of S_t^*, we assume that the initial value of $S_t^* = 1$ with probability π_0:

$$S_1^* = \begin{cases} 1 & \text{with probability } \pi_0 \\ 2, & \text{with probability } 1 - \pi_0 \end{cases} , \tag{4.3}$$

and S_t^* stays constant over time.

Since this model is a simple special case of the model in Section 3 and it satisfies the assumptions in Hu and Shum (2012), we can follow similar arguments in Section 3 that a total of four consecutive periods of observed state variables and choice variables are needed to non-parametrically identify the conditional distribution $f_{X_t, S_t^* | J_{t-1}, X_{t-1}, S_{t-1}^*}$. At the same time, we can also identify the initial condition $f_{X_{t-2}, S_{t-2}^*, J_{t-2}}$ using the same set of observed state and choice variables.

To adopt our proposed method to estimate this model, the first step is again to parametrize the choice variable as a function of observed and unobserved state variables using Equation 3.8. We use the same functional form as in Section 3, since it makes no difference whether S^* is discrete or continuous, fixed or time-varying in our algorithm. Therefore, we use the same parametric function for the policy function in these two cases. We have three utility parameters $(\theta_0, \theta_1, \theta_2)$ to be estimated using the method. Again, we follow Algorithm 1–Algorithm 3 in Section 3 to estimate these parameters.

In Algorithm 1, the input includes the policy parameter $\boldsymbol{\gamma}^* = (\gamma_0, \gamma_1, \gamma_2, \gamma_3)$, the utility function parameters $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2)$, and the initial distribution parameter for S_t^*: $\boldsymbol{\xi} = \pi_0$. We obtain the initial mileage X_1 for the individual directly from the data. Given all the structural parameters and initial observed state variables, we simulate the lifetime trajectory for each individual. Firstly, we sample the initial unobserved state variable $\widehat{S}_{i1}^*$ from the initial distribution in Equation 4.3, given the parameter π_0. We then sample the choice variable $\widehat{J}_{i1}$ from the policy function in Equation 3.8, given the first period's state variables $(X_{i1}, \widehat{S}_{i1}^*)$ and the policy function parameters $\boldsymbol{\gamma}^*(\boldsymbol{\theta}, \boldsymbol{\xi})$. Hence, we have obtained the state and choice variables in the first period: $(X_{i1}, \widehat{S}_{i1}^*, \widehat{J}_{i1})$. We move forward to the second period by using the transitional models for X_t. Specifically, we can sample $\widehat{X}_{i2}$ using Equation 4.2, given last period's state and choice variables $(\widehat{X}_{i1}, \widehat{J}_{i1})$. Notice that the transitional model of X does not contain any unknown parameters and S^* stays constant over time. With the observed and unobserved state variables obtained in the second period, we use Equation 3.8 again to sample $\widehat{J}_{i2}$, conditional on $(\widehat{X}_{i2}, \widehat{S}_{i2}^*)$ and policy function parameters $\boldsymbol{\gamma}^*(\boldsymbol{\theta}, \boldsymbol{\xi})$. Using the same strategy, we continue moving forward and sample for $(\widehat{S}_{it}^*, \widehat{X}_{it}, \widehat{J}_{it})$ for the remaining periods until time T. This results in the simulated dataset $\widehat{\boldsymbol{D}}_i = (\widehat{\boldsymbol{S}}_i, \widehat{\boldsymbol{X}}_i, \widehat{\boldsymbol{J}}_i)$, where $\widehat{\boldsymbol{X}}_i = \{\widehat{X}_{i1}, \widehat{X}_{i2}, ..., \widehat{X}_{iT}\}$, and $\widehat{\boldsymbol{J}}_i = \{\widehat{J}_{i1}, \widehat{J}_{i2}, ..., \widehat{J}_{iT}\}$.

The inner loop of our algorithm contains the Stochastic Gradient Descent algorithm to optimize the policy function parameters $\boldsymbol{\gamma}$ as shown in Algorithm 2, and the outer loop of our algorithm adopts Simulated Method of Moments to estimate structural parameters $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ following Algorithm 3. Our goal is to find the set of parameters to minimize the distance between the simulated and data moments. Therefore, our criterion function is a normalized metric of the distance between $\boldsymbol{\kappa}$ and $\widehat{\boldsymbol{\kappa}}$, where $\boldsymbol{\kappa}$ stands for the corresponding data moments. Since we do not observe S^* in the real data, it is not included in the moments that we try to match. Only $(\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{J}})$ will be used in the SMM for estimating the structural parameters. What remains to be discussed is the choice of moments to match in the SMM to estimate parameters in Algorithm 3. We target the same moments as used in Section 3, including regression coefficients, mean, standard deviation, and the correlation moments.

We simulate data for 1000 buses that live for 30 periods and make decisions in each period. The data is generated by deriving the value functions at each state using backward induction, where we solve for the optimal policy for each state by starting from the last period and moving to the first period. With the optimal policy, we start each bus in the first period by assuming $X_{i1} = 0$ and sample S_i^* using Equation 4.3 and then forward-simulate the choices until period 30. After getting the lifetime path for each bus company, we keep the last 20 periods, which results in a final dataset of 1000×20 observations. Summarizing the dimensions of this problem, there are 2 choices, 20 periods of data, 2 unobserved states, and 201 possible mileages, resulting in $20 \times 2 \times 201 = 8,040$ number of states. With the generated data, we use the proposed algorithm in Section 3 to estimate the three structural parameters. When running Algorithm 2, we set the step size $s_q = 10^{-4}$ and choose the batch size $I = 1$. When running Algorithm 3, we target the moments specified in Section 3 in the procedure of SMM, where we use the standard errors of the moments as the weighting matrix when calculating the criterion function for minimization.

Table 2 presents the estimation results for the simulation study. There are three structural parameters in total to estimate. The second column presents the data-generating parameters for the simulation study. The third column shows estimation results using Arcidiacono and Miller (2011)'s two-step EM algorithm. The last two columns present results using our proposed reinforcement learning methods with different iteration numbers for solving the optimal policy function. To match the estimation setting used in Arcidiacono and Miller (2011), we calculate mean and standard deviations using 50 simulations. The first observation is that all of the three estimation results are performing well in terms of centering around the true values. The two estimation results from our proposed methods are both very close to the underlying data-generating parameters, as the two-step EM algorithm does. In terms of computational time, our proposed method with an iteration number equal to 5000 costs a relatively shorter time of computation than the two-step EM algorithm. The computational advantage can be enhanced with more complicated models having larger state space. At the same time, the loss in precision is limited compared to the traditional method. This result shows that even using iterations of 5000 can lead to robust estimates, of which the computational burden is reduced, without much sacrifice from the precision of estimation. Therefore, our method is validated in estimating DDCs with unobserved heterogeneity. In addition, our method has advantages over Arcidiacono and Miller (2011)'s two-step EM algorithm since the two-step EM algorithm is not suitable for estimating dynamic discrete choice models with continuous unobserved state variables. A procedure of discretizing the unobserved state variables is needed before running their algorithm. In comparison, our method is general enough to be applied to dynamic models with both discrete and continuous unobserved state variables, using the same algorithm for these different cases. Therefore, when the underlying true data-generating process features continuous unobserved state variables, our method is easy to implement in this scenario while maintaining the underlying continuous structure of the unobserved state variable.

Table 2: Estimation Results for Simulation Study: Discrete Unobserved State Variables

Parameters	**DGP**	**Est. (CCP)**	**Est. (RL+II)**	**Est. (RL+II)**
			5000	20000
		Utility parameters		
θ_0 (intercept)	2	2.0344	1.9606	1.9888
		(0.1394)	(0.1803)	(0.1755)
θ_1 (mileage)	-0.15	-0.1481	-0.1579	-0.1565
		(0.0057)	(0.0242)	(0.0204)
θ_2 (unobs. state)	1	1.0412	1.0055	0.9910
		(0.1129)	(0.0868)	(0.1031)
Time (minutes)		0.6553	0.3983	1.15

Note: Mean and standard deviations for 50 simulations. The observed data consists of 1000 buses for 20 periods. The Column CCP presents estimation results using Arcidiacono and Miller 2011's two-step EM algorithm. The rest columns show the estimation results using RL+Indirect Inference methods with different iteration numbers for solving the optimal policy function. The initial values for the estimated parameters are (1.8, -0.17, 0.9, 0.45).

5 Empirical Application: A Dynamic Fertility Model with Time-Varying Unobserved Pareto Weights

In this section, we run an empirical application with a dynamic model with continuous and time-varying unobserved state variables. Specifically, we focus on a dynamic collective fertility model with a partially observed state space, where the couple in a household j interact using a cooperative framework with limited commitment. In each period t, they choose private consumption, working hours, and a binary birth decision. We assume that before 2016, the couple was allowed to have at most one child. In 2016, the Two-Child Policy came as a shock for the agents, after which they could have two children if they wanted. The policy shock will have an impact on both the utility function through the penalty term for excess birth and the wage process. The Pareto weight updates in a reduced-form way depending on the relative wage and therefore is affected by the policy change. This empirical application is based on the model presented in the first chapter of Yang (2024), where the Pareto weight updating process differs. Below, we discuss the key aspects of the model.

Preferences Let $g \in \{f, m\}$ denote the wife and the husband in the household. The couple's individual flow utility depends on private consumption, leisure, and two fertility-related terms as shown in the following equation:

$$u(c_t^g, l_t^g, n_t, \widetilde{n}_t^g) = \alpha_1 \mathrm{ln} c_t^g + \alpha_2 \mathrm{ln} l_t^g - \alpha_3((n_t - \widetilde{n}_t^g))^2 - p_t \times \mathbb{I}(n_t > 1), \tag{5.1}$$

where p_t is the penalty for having excess birth that depends on the strictness of the policy, and the couple suffers a quadratic utility loss if the actual number of children is not equal to their ideal number of children.

The household flow utility is a sum of the individual utility weighted by the Pareto weight θ_t, plus a discrete-choice-specific random preference shock:

$$U(c_t^f, l_t^f, c_t^m, l_t^m, b_t, n_t, \widetilde{n}_t^f, \widetilde{n}_t^m, \theta_t) = \theta_t u(c_t^f, l_t^f, n_t, \widetilde{n}_t^f) + (1 - \theta_t) u(c_t^m, l_t^m, n_t, \widetilde{n}_t^m) + \epsilon_{bt}, \tag{5.2}$$

where the idiosyncratic shock ϵ_{bt} follows the type I extreme value distribution.

Wage Process and Childcare Costs we assume an exogenous wage process $\mathrm{log} w_{it}$ for men and women, and an exogenous childcare cost structure $\boldsymbol{X}_t$:

$$\begin{gathered} \mathrm{log} w_t = \boldsymbol{z}_t \boldsymbol{\beta} + u_t, \quad u_t = v_t + \epsilon_t, \quad v_t = \rho \cdot v_{t-1} + \xi_t \\ \epsilon_t \sim N(0, \sigma_\epsilon^2), \quad \xi_t \sim N(0, \sigma_\xi^2). \end{gathered} \tag{5.3}$$

$$\boldsymbol{X}_t = [\tau Q(n), (1 - \tau) Q(n)]. \tag{5.4}$$

Equation 5.3 shows the observed and unobserved parts of the wage process, where the unobserved part contains a persistent component that follows an AR(1) process and a measurement error. The total childcare cost $Q(n)$ is an increasing function in the number of children n, and the wife bears τ proportion of the total cost, as shown in Equation 5.4.

Pareto Weight Updating We assume the Pareto weight for women in the household is a continuous measure that is unobserved and varies over time. Specifically, we assume the transition process of the Pareto weights θ_t is:

$$\begin{aligned}
&\pi_1 = \alpha_4(\widetilde{\omega}_1^f - \widetilde{\omega}_1^m) + \nu_1, \quad \nu_1 \sim N(0, \sigma_1^2), \\
&\pi_t = \alpha_5\pi_{t-1} + \alpha_6\big[(w_t^f - \widetilde{\omega}_t^f) - (w_t^m - \widetilde{\omega}_t^m)\big] + \nu_2, \quad \nu_2 \sim N(0, \sigma_2^2), \\
&\theta_t = \exp(\pi_t)/(1 + \exp(\pi_t)),
\end{aligned} \tag{5.5}$$

where $(\widetilde{\omega}_t^f, \widetilde{\omega}_t^m)$ denotes the wife and the husband's expected wage at time t at the time of marriage, whereas (w_t^f, w_t^m) is the realized true wage for wife and husband at time t. The initial Pareto weight in the first period is decided by the difference in the expected wage at the time of marriage plus a normal error term. The higher the wife's expected wage at $t = 1$ is than the husband's expected wage, the larger the wife's Pareto weight will be in the first period. Therefore, we would expect α_4 to be positive. When $t \geq 2$, the Pareto weight follows an AR(1) process, equaling to $\alpha_5\pi_{t-1}$ plus two shock terms. The first shock term is related to the difference in the realized wage shocks for the wife and the husband. If the wife has a large positive shock in her realized wage compared to her husband, her Pareto weight will increase due to this change. On the other hand, if the husband enjoys a large positive wage shock compared to the wife, then the wife's Pareto weight should decrease. Therefore, we should expect α_6 to be positive as well. We use this specification to capture the impact of the labor market outcomes on the intra-household decision weights. The final step is to transfer π_t into a value between 0 and 1 using a logit form. The resulting θ_t is the time-varying Pareto weight for the wife in the dynamic household bargaining model. To summarize, the dynamic lifecycle model has limited commitment, featuring a changing Pareto weight that is unobserved in the data.

The Couple's Problem In each period, the couple chooses private consumption, working hours, and a binary choice of whether to give birth, subject to budget and time constraints. The state and choice variables of the couple are:

$$\begin{aligned}\boldsymbol{\Omega_t} &= \{A_t^f, E^f, E^m, t_M, t_P, S, w_t^f, w_t^m, n_{t-1}, \widetilde{n}_t^f, \widetilde{n}_t^m, \theta_t\},\\ q_t &= \{c_t^f, c_t^m, h_t^f, h_t^m, b_t\}.\end{aligned} \tag{5.6}$$

The state space consists of the age of the wife A_t^f, the couple's education levels $\{E^f, E^m\}$, the year they get married t_M, the age of the wife when the policy change happened t_P, the strictness of the One-Child policy for the province the couple is in S, wages of the couple $\{w_t^f, w_t^m\}$, the number of children from last period n_{t-1}, and the couple's ideal number of children θ_t respectively, and the Pareto weight of the wife within the household. The choice variables include the two private consumption levels and working hours, as well as the binary fertility choice.

Finally, the joint problem the couple solves subject to budget and time constraints is:

$$\begin{aligned}V_t(\boldsymbol{\Omega_t}) = \max_{q_t}\ & \theta_t u(c_t^f, l_t^f, n_t, \widetilde{n}_t^f) + (1-\theta_t)u(c_t^m, l_t^m, n_t, \widetilde{n}_t^m) + \epsilon_{bt} + \beta E_t[V_{t+1}(\boldsymbol{\Omega_{t+1}})]\\ & c_t^f + c_t^m = \Big(w_t^f h_t^f + w_t^m h_t^m - C_t(n_t)\Big)\cdot e(n)\\ & l_t^g + h_t^g = \bar{h}^g - x_t^g(n_t), \quad g \in \{f, m\}.\end{aligned} \tag{5.7}$$

Impact of the policy change The policy relaxation that happened during the lifetime of the couple has two effects on their dynamic optimization problem. Firstly, the penalty for having more than one child, denoted as p_t in Equation 5.1, is a positive number before the policy change but becomes zero after the relaxation. Secondly, as shown in the Pareto weight updating section, women's intra-household bargaining power decreases from 0.4 to 0.3 after the policy change. We assume that the policy comes as a shock to the couple, requiring them to resolve the whole dynamic problem after the policy relaxation.

We estimate the dataset with 1583 couples and 11 time periods from the age of 20 to the age of 40, where each period contains two years. We pre-estimate the wage process for men and women and the childcare time and monetary cost functions from the data. The remaining parameters are $(\alpha_1, \alpha_2, \alpha_3, p_1, p_2, \alpha_4, \alpha_5, \alpha_6, \sigma_1, \sigma_2)$, where the first five ones are utility function parameters, while the last five ones are transitional model parameters for the unobserved Pareto weight. We use the algorithm that is similar to Algorithm 1 – Algorithm 3 in Section 3. The algorithm has two main loops, where the outer loop searches over the parameter space for the 10 parameters to match the simulated moments with the data moments, and the inner loop iteratively updates the policy function parameters using the policy gradient method given the current structural parameters in the outer loop. Using the same identification argument as in Section 3, we identify the parameters in the equation for the latent Pareto weight using five periods' state and choice data. Following the same strategy in Equation **??**, we target the regression coefficients where we regress birth decisions on other variables, gathering five periods as a whole.

Table 3 presents the estimation results for the empirical study using our proposed reinforcement learning method. We present the initial values we use when estimating the parameters in the third column and show the resulting estimates we obtained by using the proposed reinforcement learning method for the parameters in the last column. The standard errors in the bracket are obtained by bootstrapping the whole data 400 times. When estimating the parameters, we set the iteration number to 5000 for calculating the optimal policy function. As shown in the table, the couple derives utility from consumption and leisure at the parameters of 8.2012 and 6.2856. Meanwhile, the couple will suffer from a utility loss if not reaching their ideal number of children (5.0005). As for the Pareto weight parameters, it is expected that the difference in the initial expected wage between the wife and the husband has a positive impact on women's weights (0.1660). The difference in the wage shocks between the wife and the husband has an even larger impact on the Pareto weights (0.1786) than the initial wage difference. For Pareto weights in later periods, the AR(1) parameter is estimated to be 0.9371, indicating a strong correlation over time. These results suggest that assuming the Pareto weight remains constant over time is inaccurate, as it is influenced by exogenous shocks, such as wage fluctuations in the labor market. As for the policy parameters, the estimated penalty for excess births is (1.9177, 0.6183), which aligns with our assumption that provinces with stricter fertility policies impose higher penalties on couples with excess births. The last row of the table shows that the model estimation takes 4.98 minutes, indicating that the proposed algorithm can efficiently handle complex models and provide estimates in a relatively short time.

6 Conclusion

The development of a unified estimation framework, integrating policy gradient methods from the reinforcement learning literature with the established indirect inference approach in economics, presents a significant advancement in the estimation of dynamic discrete choice models (DDCs). By addressing the computational challenges associated with high-dimensional state spaces and unobserved heterogeneity, this framework offers a useful solution applicable across various model specifications.

The proposed method effectively mitigates the computational burden inherent in traditional estimation approaches, particularly in cases where discretization of continuous state variables and looping over extensive grid points become impractical. Through the combination of the simulated method of moments (SMM) and policy gradient methods, the framework solves for the optimal policy by directly parametrizing the optimal policy as a function of the state variables and applying the policy gradient theorem to update the policy function parameters. By building a mapping from the structural parameters to the optimal policy function parameters, this algorithm allows an efficient search over the parameter space in the outer loop by matching the simulated and true data moments, avoiding the calculation of the value function over massive state space. Our Monte Carlo study shows that the proposed method is able to reduce the computational time by a large scale compared to the traditional dynamic programming backward induction method, especially for dynamic models with high-dimensional state space.

A noteworthy aspect of this approach is its easy adaptability to models with different kinds of partially observed state variables. Leveraging non-parametric identification results in Hu and Shum (2012), our proposed framework demonstrates efficacy in estimating models with continuous, time-varying unobserved state variables, under conditions outlined in Hu and Shum (2012). This capability underscores its applicability to a wide range of dynamic decision-making scenarios. In addition, our approach offers distinct advantages over the existing EM algorithm since ours seamlessly handles both discrete and continuous unobserved variables. This simplifies implementation while maintaining the integrity of the underlying continuous structure, making our approach highly effective across various modeling scenarios.

The empirical validation of the proposed method across various model specifications further validates its usefulness. From simplified toy models to more complex dynamic discrete choice models, the framework consistently yields estimates that closely approximate true parameter values. Notably, the computational efficiency of the method is evident, with significantly reduced processing time compared to traditional dynamic programming methods.

In summary, this paper introduces a robust and novel estimation framework for dynamic discrete choice models, effectively combining insights from reinforcement learning with established econometric techniques. By streamlining the estimation process and enhancing computational efficiency, this framework provides a way for a more comprehensive and simple way of estimating dynamic discrete choice models.

Table 3: Estimation Results for Empirical Study: Continuous Unobserved State Variables

Parameters	**Symbol**	**Initial Value**	**Est. (RL)**
			Iter. = 5000
Utility parameters			
Utility Function parameters			
Utility from ln(c)	α_1	7.6	8.2012 (0.4699)
Utility from ln(l)	α_2	5.4	6.2856 (0.9487)
Dis-utility from not ideal num. of child	α_3	4.5	5.0005 (1.0421)
Pareto weight parameters			
First period in initial wage diff.	α_4	0.1355	0.1660 (0.0348)
First period standard deviation	σ_1	0.5	0.7022 (0.1560)
Later periods AR(1) parameter	α_5	0.9	0.9371 (0.1329)
Later periods wage shock diff.	α_6	0.1770	0.1786 (0.0140)
Later periods standard deviation	σ_2	0.5	0.7341 (0.1464)
Policy parameters			
Penalty on excess birth in strict provinces	p_1	1.2	1.9177 (0.2376)
Penalty on excess birth in loose provinces	p_2	0.5	0.6183 (0.1798)
Time (minutes)			4.98

Note: Standard errors obtained using Bootstrap for 400 times. The observed data consists of 1583 couples for 11 periods. The column Initial Value presents initial values for each parameter estimation. The rest of the columns show the estimation results using RL + Indirect Inference methods with 5000 iterations. The final row shows the time needed for estimating the model. A total of 100 moments are used for estimation.

References

Aguirregabiria, V. and Mira, P. (2007). Sequential estimation of dynamic discrete games. *Econometrica*, 75(1):1–53.

Arcidiacono, P. and Jones, J. B. (2003). Finite mixture distributions, sequential likelihood and the em algorithm. *Econometrica*, 71(3):933–946.

Arcidiacono, P. and Miller, R. A. (2011). Conditional choice probability estimation of dynamic discrete choice models with unobserved heterogeneity. *Econometrica*, 79(6):1823–1867.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., Pineau, J., et al. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354.

Gallant, A. R., Hong, H., and Khwaja, A. (2018). A bayesian approach to estimation of dynamic models with small and large number of heterogeneous players and latent serially correlated states. *Journal of econometrics*, 203(1):19–32.

Gourieroux, C., Monfort, A., and Renault, E. (1993). Indirect inference. *Journal of applied econometrics*, 8(S1):S85–S118.

Hong, M., Qi, Z., and Xu, Y. (2023). A policy gradient method for confounded pomdps. *arXiv preprint arXiv:2305.17083*.

Hotz, V. J. and Miller, R. A. (1993). Conditional choice probabilities and the estimation of dynamic models. *The Review of Economic Studies*, 60(3):497–529.

Hotz, V. J., Miller, R. A., Sanders, S., and Smith, J. (1994). A simulation estimator for dynamic models of discrete choice. *The Review of Economic Studies*, 61(2):265–289.

Hu, Y. and Shum, M. (2012). Nonparametric identification of dynamic models with unobserved state variables. *Journal of Econometrics*, 171(1):32–44.

Hwang, Y. (2024). Identification and estimation of a dynamic discrete choice model with time-varying unobserved heterogeneity using proxies. *Available at SSRN 3535098*.

Jin, W., Ni, Y., O'halloran, J., Spence, A. B., Rubin, L. H., and Xu, Y. (2023). A bayesian decision framework for optimizing sequential combination antiretroviral therapy in people with hiv. *Annals of Applied Statistics*, 17(4):3035–3055.

Kakade, S. M. (2001). A natural policy gradient. *Advances in neural information processing systems*, 14.

Kasahara, H. and Shimotsu, K. (2009). Nonparametric identification of finite mixture models of dynamic discrete choices. *Econometrica*, 77(1):135–175.

Lange, S., Gabel, T., and Riedmiller, M. (2012). Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pages 45–73. Springer.

Li, G., Li, S., Li, S., and Qu, X. (2022). Continuous decision-making for autonomous driving at intersections using deep deterministic policy gradient. *IET Intelligent Transport Systems*, 16(12):1669–1681.

Peters, J. and Schaal, S. (2006). Policy gradient methods for robotics. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225. IEEE.

Rust, J. (1987). Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica: Journal of the Econometric Society*, pages 999–1033.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Yang, F. (2024). *Essays on China's Population Policies: Impacts and Methodological Innovations*. PhD thesis, Johns Hopkins University.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

A RL + Indirect Inference Algorithm for Estimating DDCs with Fully Observed State Space

We discuss our proposed method combining reinforcement learning and indirect inference method for estimating DDCs with fully observed state space through Algorithm A.1, A.2, and A.3 below.

Algorithm A.1 discusses the procedure of forward simulation to obtain an agent's lifetime trajectory, given the optimal policy function parameters and her initial state variables. For the initial time period, we can direct sample the agent's action in this period using the policy function in Equation 2.1, since we already know the policy parameters $\boldsymbol{\gamma}^*(\boldsymbol{\theta})$. After getting $\widehat{J}_{i1}$, we can move forward to the second period by sampling $\widehat{\boldsymbol{X}}_{it}$ using the transitional model of the state variables, conditional on $\boldsymbol{X}_{it-1}$ and $\widehat{J}_{it-1}$. Similar to the first period, we can then simulate the choice variable $\widehat{J}_{it}$ by using Equation 2.1 again. Looping over the total T periods results in a complete lifetime trajectory $\widehat{\boldsymbol{D}}_i$ for individual i, where $\widehat{\boldsymbol{D}}_i$ contains the state and the choice variables.

Algorithm A.2 describes the procedure for conducting stochastic gradient descent for optimizing the policy function parameter $\boldsymbol{\gamma}$. This serves as the inner loop of our whole algorithm, where the structural parameter $\boldsymbol{\theta}$ is given in the outer loop, and the inner loop is updating $\boldsymbol{\gamma}$ to find the optimal policy function conditional on the given structural parameters. We start with some initial values $\boldsymbol{\gamma}_0$ for the policy parameters, together with structural parameters $\boldsymbol{\theta}$ and the initial state variable $\{X_{11}, X_{21}, ..., X_{I1}\}$ from the batch data D_I as input. Meanwhile, we choose step size s_q for updating the policy parameters and the batch size I, using which we randomly sample from the whole dataset to get the batch data D_I. After initializing $\boldsymbol{\gamma}$, we start the two loops in the algorithm, where the outer loop contains the iteration of the Stochastic Gradient Descent algorithm until the convergence of the policy function parameters, and in the inner loop, we sample the lifetime trajectory $\widehat{\boldsymbol{D}}_i$ for each couple i following the steps in Algorithm A.1. Using the simulated lifetime path, we can then calculate the lifetime value $\{V_i(\boldsymbol{\gamma}_q)\}$ and the policy gradient $\nabla_{\boldsymbol{\gamma}} \log(\prod_{t>=1} \pi(\widehat{J}_{it}|\widehat{X}_{it}, t; \boldsymbol{\gamma}_q))$. Notice that these objects depend on $\boldsymbol{\gamma}_q$, which is the policy parameter in the current outer loop iteration. After obtaining the lifetime values and policy gradients for each individual in the batch data, the inner loop ends and we begin to calculate the average of the lifetime values and policy gradients across all individuals in order to calculate the gradient of the value function.

Using the average lifetime value and the average policy gradient, we can proceed to get $\nabla_{\boldsymbol{\gamma}} V(\boldsymbol{\gamma}_q)$, adopting the policy gradient theorem. Finally, in each of the outer loop iterations, we update the policy function parameter $\boldsymbol{\gamma}_{q+1}$ by adding the policy gradient normalized by the step size onto the previous $\boldsymbol{\gamma}_q$. We repeat this process until $\boldsymbol{\gamma}$ converges. The final output of the algorithm contains the parameters we obtained in the final iteration of the outer loop, which is defined as the optimal policy function parameter, conditional on the structural parameters $\boldsymbol{\theta}$.

Finally, Algorithm A.3 shows the outer loop of the whole algorithm that adopts indirect inference (Simulated Method of Moments) to estimate structural parameters $\boldsymbol{\theta}$. The input of the algorithm includes initial values $\boldsymbol{\theta}_0$ for the parameters to estimate, the initial state variable X_{i1} from the data for each couple i, and the data moments $\boldsymbol{\kappa}$ we plan to target for estimating the deep parameters. The loop $k = 1, ..., K$ stands for the iterations of updating the structural parameters by minimizing the criterion function value. In each iteration, we calculate the optimal policy function parameters $\boldsymbol{\gamma}(\boldsymbol{\theta}_k)$ following the procedure in Algorithm A.2, given the parameter $\boldsymbol{\theta}_k$ in the current iteration. Using the policy function parameters, we then simulate the lifetime trajectory $\widehat{\boldsymbol{D}}_i$ for each couple i in the data, following Algorithm A.1. Having obtained the simulated data path for all individuals, we are able to calculate the simulated aggregate moments $\widehat{\boldsymbol{\kappa}}$ for these targeted moments. Our goal is to find the set of parameters to minimize the distance between the simulated moments and real data moments. Hence, our criterion function is a normalized metric of the distance between $\boldsymbol{\kappa}$ and $\widehat{\boldsymbol{\kappa}}$, where $\boldsymbol{\kappa}$ stands for the corresponding data moments.

Combining Algorithm A.1–Algorithm A.3, our proposed method is able to update the inner and outer loops together, which results in both a mapping from the structural parameter to the optimal policy function parameter and the optimal structural parameter to minimize the distance between the simulated and true data moments. Therefore, it allows a convenient and time-saving estimation of the dynamic discrete choice models.

Algorithm A.1 Forward Simulation for Obtaining Individual i's Lifetime Trajectory

```
Input: policy parameters $\boldsymbol{\gamma}(\boldsymbol{\theta})$, utility parameters $\boldsymbol{\theta}$, initial state $\boldsymbol{X}_{i1}$ for individual $i$
Sample $\widehat{J}_{i1}$ using Equation 2.1 , given $\boldsymbol{\gamma}(\boldsymbol{\theta})$ and $\boldsymbol{X}_{i1}$
for $t = 2, \ldots, T$ do
    Sample $\widehat{\boldsymbol{X}}_{it}$ using $f_{\boldsymbol{X}_t|\boldsymbol{X}_{t-1},J_{t-1}}$
    Sample $\widehat{J}_{it}$ Equation 2.1 , conditional on $\boldsymbol{\gamma}(\boldsymbol{\theta})$ and $\widehat{\boldsymbol{X}}_{it}$
end for
Obtain the final dataset $\widehat{\boldsymbol{D}}_i = (\widehat{\boldsymbol{X}}_i, \widehat{\boldsymbol{J}}_i)$
Output: lifetime trajectory $\widehat{\boldsymbol{D}}_i$ for individual $i$
```

Algorithm A.2 Stochastic Gradient Descent for optimizing $\boldsymbol{\gamma}$

```
Input: initial value $\boldsymbol{\gamma}_0$ for the policy parameters $\boldsymbol{\gamma}$, deep parameters $\boldsymbol{\theta}$, step size $s_q$, batch size I, initial state $\{X_{11}, X_{21}, ..., X_{I1}\}$ from the batch data $D_I$.
Initialize: $\boldsymbol{\gamma}_1 \leftarrow \boldsymbol{\gamma}_0$
for $q = 1, \ldots, Q$ do
    for $i = 1, 2, \ldots, I$ do
        Obtain $\widehat{\boldsymbol{D}}_i = (\widehat{\boldsymbol{X}}_i, \widehat{\boldsymbol{J}}_i)$ following the procedure in Algorithm A.1
        Calculate the lifetime value $V_i(\boldsymbol{\gamma}_q)$ and gradient $\nabla_{\boldsymbol{\gamma}}\text{log}(\prod_{t>=1} \pi(\widehat{J}_{it}|\widehat{X}_{it}, t; \boldsymbol{\gamma}))$
    end for
    Average the lifetime value and gradient across individuals:
    $V(\boldsymbol{\gamma}_q) := \frac{1}{I}\sum_{i=1}^{I} V_i(\boldsymbol{\gamma}_q)$; $\nabla_{\boldsymbol{\gamma}}\text{log}(\prod_{t>=1} \pi(\widehat{J}_t|\widehat{X}_t, t; \boldsymbol{\gamma})) := \frac{1}{I}\sum_{i=1}^{I} \nabla_{\boldsymbol{\gamma}}\text{log}(\prod_{t>=1} \pi(\widehat{J}_{it}|\widehat{X}_{it}, t; \boldsymbol{\gamma}))$
    $\nabla_{\boldsymbol{\gamma}} V(\boldsymbol{\gamma}_q) \leftarrow V(\boldsymbol{\gamma}_q)\nabla_{\boldsymbol{\gamma}}\text{log}(\prod_{t>=1} \pi(\widehat{J}_t|\widehat{X}_t, t; \boldsymbol{\gamma}))$
    $\boldsymbol{\gamma}_{q+1} \leftarrow \boldsymbol{\gamma}_q + s_q \nabla_{\boldsymbol{\gamma}} V(\boldsymbol{\gamma}_q)$
end for
Output: $\boldsymbol{\gamma}^* = \boldsymbol{\gamma}_{q^*}$
```

Algorithm A.3 Indirect Inference for estimating $\boldsymbol{\theta}$

```
Input: initial value $\boldsymbol{\theta}_0$, initial state $\{X_{11}, X_{21}, ..., X_{N1}\}$ from data D; data moments $\boldsymbol{\kappa}$.
Initialize: $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}_0$
for $k = 1, \ldots, K$ do
    Calculate $\boldsymbol{\gamma}^*(\boldsymbol{\theta}_k)$ following Algorithm A.2
    for $i = 1, 2, \ldots, N$ do
     Obtain $\widehat{\boldsymbol{D}}_i = (\widehat{\boldsymbol{X}}_i, \widehat{\boldsymbol{J}}_i)$ using $\boldsymbol{\gamma}^*(\boldsymbol{\theta}_k)$ following Algorithm A.1
    end for
    Calculate simulated moments $\widehat{\boldsymbol{\kappa}}$ using $(\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{J}})$ according to targeted moments
    $\boldsymbol{\theta}_{k+1} \leftarrow \min \text{Dis}(\boldsymbol{\kappa}, \widehat{\boldsymbol{\kappa}})$
end for
Output: $\boldsymbol{\theta}^* = \boldsymbol{\theta}_{k^*}$
```

B Proof for the Model Satisfying the Identification Assumptions

In this section, we show that the model in the Monte Carlo simulation in Section 3.1.1 satisfies the four assumptions required by Hu and Shum (2012) to obtain the identification result. Assumption 2 is satisfied for this model because the law of motions has a Markov structure, where both X_t and S_t^* do not depend on historical values if already conditional on X_{t-1} and S_{t-1}^*. Limited feedback is also satisfied since we assume in our model that S_t^* realizes before X_t, and X_t depends on S_t^* in period t. Therefore, conditional on S_t^*, there is no additional effect from S_{t-1}^* on X_t. In summary, the two conditions in this assumption are satisfied by our model.

Since we are focusing on the stationary case in the model, we label the four observed periods of data as $t = 1, 2, 3, 4$ without loss of generosity. As long as we can establish the injectivity of the operators L_{X_1,w_2,w_3,X_4}, $L_{X_4|w_3,S_3^*}$, and L_{X_1,w_2,X_3}, it is sufficient to prove that Assumption 2 holds in our model. We only need to have injectivity for $L_{X_4|w_3,S_4^*}$, $D_{w_3|w_2,S_3^*}$, $L_{S_3^*|w_2,S_2^*}$ and $L_{S_2^*,w_2,X_1}$.

The diagonal operator $D_{w_3|w_2,S_3^*}$ has the kernel function:

$$\begin{aligned} f_{w_3|w_2,S_3^*} &= f_{x_3,j_3|x_2,j_2,S_3^*} \\ &= f_{j_3|x_2,j_2,x_3,S_3^*} f_{x_3|x_2,j_2,S_3^*} \\ &= f_{j_3|x_3,S_3^*} f_{x_3|x_2,j_2,S_3^*}. \end{aligned} \tag{B.1}$$

It is obvious that $f_{j_3|x_3,S_3^*}$ is nonzero along its support and $f_{x_3|x_2,j_2,S_3^*}$ is nonzero along its support as well. Therefore, we have that $D_{w_3|w_2,S_3^*}$ is injective.

From Equation 3.3, we know that X_4 is a convolution of S_4^*, for every x_3 when $j_3 = 0$. This is because from Equation 3.3, we have $\log\big[X_4 - X_3\big] - \log(\lambda_3 X_3) = \lambda_2 S_4^* + \eta_4$. Therefore, applying the result in the convolution literature, we have that $L_{X_4|w_3,S_4^*}$ is injective. Similarly, Equation 3.2 implies that S_3^* is a convolution of S_2^* for fixed w_2, since $S_3^* = \lambda_1 S_2^* + \nu_3$. Therefore, we can also reach the conclusion that $L_{S_3^*|w_2,S_2^*}$ is injective according to the findings in the convolution literature.

Our model has the special case that the condition of the bus S_t^* evolves exogenously and does not involve the choice variable J_t. Together with the assumption that the initial values of the state variables (S_1^*, X_1) are independently distributed, these two conditions guarantee the injectivity of $L_{S_2^*,w_2,X_1}$.

Having proved that the four linear operators are injective, we have shown that

Assumption 2 is satisfied.

As shown in Equation B.1, the density $f_{W_3|W_2,X_3^*}$ factors as follows:

$$f_{W_3|W_2,X_3^*} = f_{J_3|X_3,S_3^*} f_{X_3|X_2,J_2,S_3^*}. \tag{B.2}$$

Plug Equation B.2 into Equation 3.1, we have

$$\begin{aligned}
&k(w_3, \overline{w}_3, w_2, \overline{w}_2, s_3^*) \\
&= \frac{f_{J_3|X_3,S_3^*}(j_3|x_3, s_3^*) f_{X_3|X_2,J_2,S_3^*}(x_3|x_2, j_2, s_3^*) \cdot f_{J_3|X_3,S_3^*}(j_3|\overline{x}_3, s_3^*) f_{X_3|X_2,J_2,S_3^*}(\overline{x}_3|\overline{x}_2, j_2, s_3^*)}{f_{J_3|X_3,S_3^*}(j_3|\overline{x}_3, s_3^*) f_{X_3|X_2,J_2,S_3^*}(\overline{x}_3|x_2, j_2, s_3^*) \cdot f_{J_3|X_3,S_3^*}(j_3|x_3, s_3^*) f_{X_3|X_2,J_2,S_3^*}(x_3|\overline{x}_2, j_2, s_3^*)} \\
&= \frac{f_{X_3|X_2,J_2,S_3^*}(x_3|x_2, j_2, s_3^*) \cdot f_{X_3|X_2,J_2,S_3^*}(\overline{x}_3|\overline{x}_2, j_2, s_3^*)}{f_{X_3|X_2,J_2,S_3^*}(\overline{x}_3|x_2, j_2, s_3^*) \cdot f_{X_3|X_2,J_2,S_3^*}(x_3|\overline{x}_2, j_2, s_3^*)}.
\end{aligned} \tag{B.3}$$

Therefore, to have unique eigenvalues, we need to have $f_{J_3|X_3,S_3^*} > 0$ for all X_3 since this term gets canceled out from the numerator and denominator of the eigenvalues. We have that $f_{J_3|X_3,S_3^*} > 0$ is true due to the assumption that the discrete-choice-specific error term ϵ_{jt} in the utility function (Equation 3.6) follows the type I extreme value distribution.

From Equation 3.3, we have

$$\begin{aligned}
f_{X_3|X_2,J_2,S_3^*}(x_3|x_2, 0, s_3^*) &= \frac{1}{x_3 - x_2}\exp\Big[\log\frac{x_3 - x_2}{\lambda_3 x_2} - \lambda_2 s_3^* - e^{\log\frac{x_3 - x_2}{\lambda_3 x_2} - \lambda_2 s_3^*}\Big] \\
&= \frac{\exp[-\lambda_2 s_3^*]}{\lambda_3 x_2}\exp[-e^{-\lambda_2 s_3^*}\frac{x_3 - x_2}{\lambda_3 x_2}]
\end{aligned} \tag{B.4}$$

Plugging Equation B.4 into Equation B.3, we have

$$k(w_3, \overline{w}_3, w_2, \overline{w}_2, s_3^*) = \exp\Big(- e^{-\lambda_2 s_3^*} \cdot \frac{-(\overline{x}_3 - x_3)(\overline{x}_2 - x_2)}{\lambda_3 x_2 \overline{x}_2}\Big), \quad \text{when } j_3 = 0.$$

We set $\lambda_2 = 0.8, \lambda_3 = 0.2$. Given these parameters and the functional form of the eigenvalue, it is easy to show that $0 < k(w_3, \overline{w}_3, w_2, \overline{w}_2, s_3^*) < C$ for some finite C, and $k(w_3, \overline{w}_3, w_2, \overline{w}_2, \overline{s}_3^*) \neq k(w_3, \overline{w}_3, w_2, \overline{w}_2, \widetilde{s}_3^*)$ since $k(w_3, \overline{w}_3, w_2, \overline{w}_2, s_3^*)$ is monotone in s_3^*. Therefore, we have shown that Assumption 3 is satisfied.

We set $V_t = X_t$ for all t. From Equation 3.3, we have

$$\log\Big[\frac{X_4 - x_3}{\lambda_3 x_3}\Big] = \eta_4 + \lambda_2 s_4^*.$$

Therefore, we have

$$E\Big[\log\frac{X_4 - x_3}{\lambda_3 x_3}|x_3, j_3, s_3^*\Big] = E(\eta_4) + \lambda_2 E[S_4^*|s_3^*] \tag{B.5}$$

Plugging Equation 3.2 into Equation B.5, we get

$$E\Big[\log\frac{X_4 - x_3}{\lambda_3 x_3}|x_3, j_3, s_3^*\Big] = E(\eta_4) + \lambda_2 \lambda_1 s_3^*, \tag{B.6}$$

which is monotonic in s_3^*. Therefore, we can set G to be

$$G(x_3, j_3, s_3^*) = E\Big[\log\frac{X_4 - x_3}{\lambda_3 x_3}|x_3, j_3, s_3^*\Big],$$

and we normalize $s_3^* = E\Big[\log\frac{X_4 - x_3}{\lambda_3 x_3}|x_3, j_3, s_3^*\Big]$. Therefore, we have shown that Assumption 4 is satisfied by our model. □