Identifying Latent State-Transition Processes for Individualized Reinforcement Learning

Yuewen Sun[1,2], **Biwei Huang**[3], **Yu Yao**[4], **Donghuo Zeng**[5], **Xinshuai Dong**[2], **Songyao Jin**[3], **Boyang Sun**[1], **Roberto Legaspi**[5], **Kazushi Ikeda**[5], **Peter Spirtes**[2], **Kun Zhang**[1,2]
[1]Mohamed bin Zayed University of Artificial Intelligence, [2]Carnegie Mellon University,
[3]University of California San Diego, [4]The University of Sydney, [5]KDDI Research

Abstract

The application of reinforcement learning (RL) involving interactions with individuals has grown significantly in recent years. These interactions, influenced by factors such as personal preferences and physiological differences, causally influence state transitions, ranging from health conditions in healthcare to learning progress in education. As a result, different individuals may exhibit different state-transition processes. Understanding individualized state-transition processes is essential for optimizing individualized policies. In practice, however, identifying these state-transition processes is challenging, as individual-specific factors often remain latent. In this paper, we establish the identifiability of these latent factors and introduce a practical method that effectively learns these processes from observed state-action trajectories. Experiments on various datasets show that the proposed method can effectively identify latent state-transition processes and facilitate the learning of individualized RL policies.

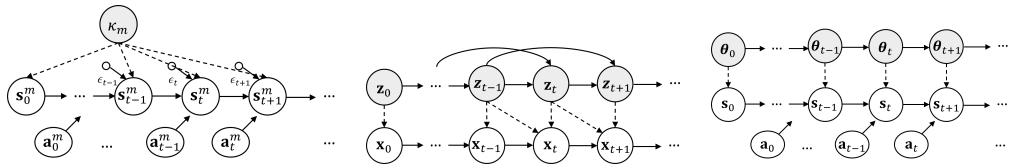
1 Introduction

Reinforcement Learning (RL) [46] involves training agents to make decisions by interacting with the environment. The agent observes its current state, takes an action, and transitions to a new state with a reward. Such a sequence of moving from one state to another is known as a *state-transition process*.

Individualized RL focuses on adapting the policy for each individual. It has recently seen increasing application in various sectors, including healthcare [89, 57, 21], education [68, 2, 14], and e-commerce [53, 87, 1]. The *individual-specific factors* [60], which capture the unique characteristics of each individual, play an important role in causally influencing the transitions between states. These factors cover a range of aspects, including individual preferences, past experiences, and physiological differences. For example, in education, different learning styles can influence how students with the same prior knowledge benefit from a tutorial. In healthcare, genetic differences can affect how patients with hypertension respond to the same treatments. Understanding individual-specific factors is essential for designing better RL systems that provide more individualized and effective decisions [41, 28, 4, 65]. By understanding learning styles in education, RL agents can recommend personalized tutorials, such as animated content for visual learners or hands-on exercises for kinesthetic learners. Similarly, in healthcare, knowledge of genetic makeup can help agents suggest treatment plans tailored to individual needs, leading to improved health outcomes.

However, these individual-specific factors are not always observable, making it challenging to understand the Latent Individualized State-Transition (LIST) processes, as illustrated in Figure 1(a). These latent factors, such as learning styles and genetic makeup, are unique to each individual and have a time-invariant influence on the state-transition process. This raises the question: can the identifiability of these latent factors be guaranteed?

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) **iMDP for individual m.** (b) Latent causal processes. (c) Factored nonstationary MDP.

Figure 1: Comparisons of different state-transition processes. Latent variables are colored in grey.

Such identifiability is easier to achieve when the observations are either i.i.d., or i.i.d. given side information (e.g., domain index, time index, etc.), by exploiting sparsity [97], variability [47], or functional complexity [43]. To the best of our knowledge, only a few studies have explored the identifiability of latent factors from temporal observations. These methods primarily focus on the time-varying latent factors, which differs from our work on time-invariant latent factors. Specifically, existing works [86, 85, 7] assume time-varying latent variables without considering the influence of actions in the generative process (see Figure 1(b)). Factored MDPs [15] incorporate actions into the process but still assume that the latent factors change over time (see Figure 1(c)). Thus, the findings from these studies cannot be applied to our setting. Intuitively, this is because time-invariant latent factors cannot provide the variability that many current methods rely on to achieve identifiability. Therefore, it remains unclear how to derive the identifiability of the latent individual-specific factors and the corresponding latent state-transition processes from observed states and actions.

Recent advances in finite mixture models [77, 67] have demonstrated strong identifiability results by exploiting group information in nonparametric settings. By assuming that observations within the same group are known to come from the same component, the mixture of probability measures can be uniquely identified under appropriate assumptions. Inspired by these works, we establish the identifiability of latent factors by leveraging group information from the data, making it easier to distinguish different underlying components. We propose both a finite latent, nonparametric setting and an infinite latent, parametric setting, and develop a theoretically grounded framework that effectively learns these processes from observed state-action trajectories. Our contributions are summarized as follows:

- We introduce Individualized Markov Decision Processes (iMDPs), a novel framework that incorporates latent individual-specific factors κ into state-transition processes. We treat such latent factors that influence each state in the decision process and vary across individuals.
- Our work provides theoretical guarantees and new insights into learning state-transition processes with latent factors. For finite κ, we consider two scenarios to establish identifiability in nonparametric settings. For infinite κ, we demonstrate identifiability in the post-nonlinear case. To the best of our knowledge, this is the first work to provide a theoretical guarantee for identifying latent individual-specific factors from observed transitions.
- We propose a practical generative method that can effectively estimate the latent individual-specific factors. Empirical results on various datasets demonstrate the effectiveness of the method both in inferring latent factors and in learning individualized policies.

2 Related Work

Individualized Machine-Learning Applications Recently, machine learning has created highly individualized solutions across various domains. In healthcare, algorithms support individualized treatment for physical activity, weight loss, and diabetes management [89, 57, 18, 17]. In finance, it provides accurate stock predictions for stock market activities [56]. Education is benefiting from individualized ICT systems that address the individual learning needs of students [16, 40]. Furthermore, transportation has seen the development of individualized car-following strategies [69] that improve driving safety and efficiency. Meanwhile, entertainment platforms such as YouTube and TikTok are using it to provide individualized video recommendations [6, 33].

Reinforcement Learning with Latent State-Transition Processes In the field of RL, various models explore the state transition dynamics with latent variables. One such approach is Partially Observable Markov Decision Processes (POMDPs) [64], where the full information about the state is

unknown. In POMDPs, observations are generated from the latent states, which do not match our individual latent setting. For example, block MDPs [92, 96] assume that there is a fixed and unknown mapping from observations to the latent states. Factored MDPs [35, 15], which provide the partial identifiability of latent factors, assume that the latent factors evolve over time following a Markov process. On the other hand, there exists a piece of work focusing on estimating state transitions with time-invariant latent factors. Models such as contextual MDPs [29, 60, 65], latent MDPs [51, 50] and multitask RL [73, 24] consider similar scenarios with our latent individual-specific factors. However, these works lack theoretical guarantees on the identifiability of the latent factors thus it is hard for them to guarantee individualized decision-making.

3 Problem Formulation

Consider a population with M individuals that can be divided into G groups, where the exact group memberships are unknown. We introduce iMDPs to model individualized decision-making processes, where *observed* individual uniqueness is captured by u, and *latent* group-level properties are encoded by κ. Specifically, each individual is assigned a unique value of u, with the cardinality of u being M. Meanwhile, individuals within the same group share the same value of κ, which differs across groups, and the cardinality of κ is G. For each individual, the value of κ is predetermined and κ has a time-invariant influence on the state-transition process. Furthermore, all individuals are assumed to share the same state and action spaces. The iMDP is defined as follows.

Definition 3.1 (iMDP)**.** *An iMDP consists of a tuple $\langle \mathcal{S}, \mathcal{A}, R, \{s_0^m\}_{m=1}^M, \{u_m\}_{m=1}^M, \{\mathbb{T}_m\}_{m=1}^M \rangle$, where M is the number of individuals; $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively; $R \in \mathbb{R}$ is the immediate reward received after transitioning from current state s to new state s' via action a, i.e., $r = R_a(s, s')$ for $s \in \mathcal{S}$, $s' \in \mathcal{S}$ and $a \in \mathcal{A}$. To model individualized decision-making, each individual is associated with a unique index and an individualized state-transition process. In the mth MDP, u_m is the unique index identifying each individual, and s_0^m is the individualized initial state. The individualized state-transition probability is denoted by $\mathbb{T}_m := \mathbb{P}_m(s'|s, a, \kappa_m) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|\times|\mathcal{S}|}$, where κ_m is the latent individual-specific factor with a cardinality of G. Thus, the joint distribution of any adjacent state-action pairs (s, a, s') can be fully characterized by u and κ as:*

$$\mathbb{P}(s, a, s'|u) = \mathbb{P}(s'|s, a, \kappa)\mathbb{P}(s, a|u). \tag{1}$$

Data Generation Process Here we introduce the LIST processes based on iMDP framework. For individual m, the observed states $\mathbf{s}_t^m$ are generated according to the following generation process:

$$\mathbf{s}_t^m = f(\mathbf{s}_{t-1}^m, \mathbf{a}_{t-1}^m, \kappa_m, \epsilon_t^m), \tag{2}$$

where $\mathbf{s}_t^m = (s_{0,t}^m, \ldots, s_{d_s,t}^m)^\top \in \mathbb{R}^{d_s}$ denotes the d_s-dimensional state at time t, and $\mathbf{a}_t^m = (a_{0,t}^m, \ldots, a_{d_a,t}^m)^\top \in \mathbb{R}^{d_a}$ denotes the d_a-dimensional action at time t. The term ϵ_t^m represents independent noise, while κ characterizes the group-level properties within the population. The transition function f is identical across individuals, governs the dynamics and is consistent with Eq. (1). During interaction with the environment, the trajectory $\tau_m = \{\mathbf{s}_0^m, \mathbf{a}_0^m, \mathbf{s}_1^m, \ldots, \mathbf{s}_T^m\}$ is recorded as a sequence of observed state-action tuples, where T denotes the length of the trajectory.

Objectives In this work, we investigate RL with a focus on capturing individualized latent state-transition dynamics. Our objectives are twofold: 1) to identify the latent individual-specific factors κ from observed trajectories, and 2) to learn individualized policies for each agent, facilitating policy adaptation for newcomers. Consider the example of hypertension diagnosis in healthcare. Treating all patients identically may lead to different outcomes due to the dynamics of state transitions, which are influenced by latent κ. Therefore, accurate identification of κ from the population provides crucial dynamic insights. Once κ is uncovered, we can categorize patients into different groups and tailor individualized treatments for each, which motivates our second goal.

4 Identifiability Analysis

We establish the identifiability of latent individualized state-transition processes under two conditions: (1) finite latent condition under group determinacy assumption (see Theorem 4.1 and Theorem 4.2) and (2) infinite latent condition under functional constraint (see Theorem 4.3). The corresponding identifiability results are presented below.

Finite Latent Condition Suppose the value of κ is finite; we first provide the definition of group-wise identifiability. For the detailed assumptions discussion and proofs, please see Appendix B.

Definition 4.1 (Group-wise Identifiability)**.** *Let $\{\tau_m\}_{m=1}^M$ be sequences of observed states and actions collected from G groups under a fixed policy, following the true latent individualized state-transition processes described in Eq.* (2)*. A learned generative model $(\hat{f}, \hat{\kappa}, \hat{\epsilon})$ is observational equivalent to (f, κ, ϵ) if the joint distribution $\mathbb{P}_{\hat{f},\hat{\kappa},\hat{\epsilon}}(s, a, s')$ matches $\mathbb{P}_{f,\kappa,\epsilon}(s, a, s')$ everywhere. We say that the latent individualized state-transition processes are group-wise identifiable if observational equivalence can always lead to the identifiability of latent individual-specific factors across the population up to the invertible transformation g:*

$$\mathbb{P}_{\hat{f},\hat{\kappa},\hat{\epsilon}}(s, a, s') = \mathbb{P}_{f,\kappa,\epsilon}(s, a, s') \iff \hat{\kappa} = g(\kappa). \tag{3}$$

Assumption 4.1 (Group Determinacy)**.** *Consider a finite mixture model $\sum_{g=1}^G \pi_g \delta_{\kappa_g}(\kappa)$, where π_g represents mixing proportions with $\sum_{g=1}^G \pi_g = 1$, and δ_{κ_g} is the Dirac function centered at κ_g. Each unique value of κ corresponds to a specific group in the population, with $\delta_{\kappa_g}(\kappa) = 1$ if $\kappa = \kappa_g$ and 0 otherwise, and the number of individuals per group is greater than $2G - 1$.*

Assumption 4.1 guarantees that identifiability can be derived from the finite mixture model perspective using group information. We consider two scenarios to establish identifiability under finite latent conditions. Theorem 4.1 considers finite samples, establishing identifiability under specific assumptions about the initial states $\{\mathbf{s}_0^m\}_{m=1}^M$. Theorem 4.2 guarantees asymptotic identifiability for sufficiently long trajectories without imposing constraints on the initial states.

Theorem 4.1. *Assume the LIST processes described in Eq.* (2)*. Suppose the distributions of initial states are the same for all individuals within the same groups, and the trajectory length is finite. Under Assumption 4.1, the identifiability of the individual-specific factor κ is guaranteed.*

Theorem 4.2. *Assume the LIST processes described in Eq.* (2)*. Suppose the distribution of the initial state varies across individuals, and the trajectory length is sufficiently long, i.e., there exist two different individuals within the same group have overlap condition $\mathbb{P}(s, a|u = u_i) = \mathbb{P}(s, a|u = u_j)$, where $i \neq j$. Under Assumption 4.1, the identifiability of κ is asymptotically guaranteed.*

Infinite Latent Condition Theorem 4.3 demonstrates that under specific functional constraints, the identifiability of latent individual-specific variables can be extended to multiple and infinite latent factors. Specifically, we consider the post-nonlinear temporal model [93] and allow multiple instances of κ to influence the state-transition dynamics. The identifiability and cardinality of latent factors are decided by the rank conditions of specific covariance submatrices derived from the observed data. Furthermore, empirical results in Section 6 indicate that even when multiple latent factors with infinite cardinality are present, our estimation framework (see Section 5) still encourages the identification of latent factors. For a detailed proof, please see Appendix C.

Theorem 4.3. *Consider a trajectory collected from the post-nonlinear temporal model (Definition C.1) with d_s-dimensional observed states over time $t = 1, \ldots, T$. Let m latent factors κ_j, $j = 1, \ldots, m$, have direct causal influence on all states, and $\mathcal{S}_t = \{s_{1,t}, s_{2,t}, \ldots, s_{d_s,t}\}$ represent the set of all state variables at time t. These latent factors, as well as the state-transition process, can be identified if and only if for every $i = m + 2, \ldots, T - (m + 1)$, there exist pairs of minimal rank sets (Definition C.2) $(\mathbf{A_i}, \mathbf{B_i})$, denoted as $\mathbf{A_i} = \mathcal{R}_{i,i^-}$ and $\mathbf{B_i} = \mathcal{R}_{i,i^+}$, where $i^- < i < i^+$, that satisfy:*

- *(Rank Deficiency for Identification) In addition to $\mathcal{S}_i$ shared by $\mathbf{A_i}$ and $\mathbf{B_i}$, each subset should include a randomly selected set of $m + 1$ additional state variables. If the covariance matrices $\Sigma_{\mathbf{A_i},\mathbf{B_i}}$ exhibit a consistent rank r (where $r > d_s$) across all distinct indices i, this consistency implies the existence of latent factors within the system.*
- *(Quantification of Latent Factors) Once identification is established, the number of latent factors m can be inferred from the rank deficiency of $\Sigma_{\mathbf{A_i},\mathbf{B_i}}$ relative to the dimensionality of the observed variables, specifically given by $m = \mathrm{rank}(\Sigma_{\mathbf{A_i},\mathbf{B_i}}) - d_s$.*

5 Estimation and Policy Learning Framework

Overview We propose a two-stage approach that addresses two key objectives: (1) developing an estimation framework to recover the latent factors κ from individual trajectories, and (2) implementing

individualized policy learning to facilitate policy adaptation for new individuals. The estimation framework is designed to meet the conditions outlined in the identifiability theorems. According to Definition 4.1, identifiability is achieved if and only if observational equivalence implies latent factor equivalence. This motivates the use of a generative model to estimate latent factors and ensure that the reconstructed distribution aligns closely with the true observed distribution.

To fulfill this requirement, we make a modification to the variational autoencoder (VAE) architecture [45], which enables the unsupervised estimation of latent factors. Theorems 4.1 and 4.2 guarantee the asymptotic accuracy of this alignment. As illustrated in Figure 2, individual trajectories are encoded in a discrete embedding space, which is consistent with the assumptions in the theorems. Detailed pseudocode for the proposed approach is provided in Appendix I, and a comprehensive description of each component is provided in Appendix H.

5.1 Latent Estimation Framework

Temporal Encoding and Latent Factor Quantization The group determinacy assumption suggests the identifiability of the latent individual-specific factor κ. Given that κ is time-invariant and influences each state in the transition process, we begin by employing a sequential encoder to capture the high-level representation, denoted as z_m, based on the input from all states across each trajectory. We then utilize a vector quantization layer [76] to discretize the latent space and estimate the latent factor $\hat{\kappa}_m$. This quantization step ensures that the learned representation aligns with the group-level characteristics of the latent factors, as assumed in our framework, thereby supporting our objectives.

Specifically, to capture temporal dependency from sequential observations, we use sequential neural networks such as Conv1D [52] or Long Short-Term Memory (LSTM) [31] as encoders. The encoder function, denoted as g, maps the input trajectory $\{\mathbf{s}_0^m, \ldots, \mathbf{s}_T^m\}$ to a continuous high-level representation $z_m = g(\mathbf{s}_0^m, \ldots, \mathbf{s}_T^m)$, where Conv1D extracts local temporal patterns from subsequences, and LSTM aggregates information over time by sequentially updating its hidden and cell states. After processing the whole trajectory sequentially, the final hidden state of the LSTM and the final output of the Conv1D layer serve as the representation z_m.

However, continuous representations z_m are incompatible with our framework's requirements. To address this, we apply a vector quantization layer to discretize the latent space and approximate the latent factor. This layer maps z_m to the nearest vector in a predefined embedding dictionary $E = \{e_1, e_2, \ldots, e_G\}$, where each vector e_i represents a distinct group in the discrete embedding space. The assignment is realized by finding the nearest neighbor in the dictionary as $\hat{\kappa}_m = \arg\min_{e_i} \|z_m - e_i\|_2$, where $\hat{\kappa}_m$ represents the quantized vector e_i closest to the continuous representation z_m.

Latent Factor Estimation via Conditional Reconstruction Reconstruction plays a critical role in unsupervised latent factor estimation by ensuring that the reconstructed distribution closely matches the true observed distribution. As stated in Definition 4.1, this alignment allows the estimated latent factors to approximate the true latent factors. Given the nature of the transition processes, we design a conditional decoder that uses the state-action pairs $(\mathbf{s}_{t-1}^m, \mathbf{a}_{t-1}^m)$ as conditions to guide the reconstruction of $\hat{\mathbf{s}}_t^m$. These conditions, together with the estimated latent factors $\hat{\kappa}_m$, serve as inputs to the decoder. The reconstruction accuracy is evaluated through the reconstruction likelihood $p_{\text{Recon}}(\hat{\mathbf{s}}_t^m | \mathbf{s}_{t-1}^m, \mathbf{a}_{t-1}^m, \hat{\kappa}_m)$, where p_{Recon} denotes the reconstruction distribution. It provides a probabilistic measure of how well $\hat{\mathbf{s}}_t^m$ approximates $\mathbf{s}_t^m$. This likelihood quantitatively evaluates the performance of the decoder and the ability of the model to accurately reconstruct observed states.

Training Objectives The parameters are optimized according to the following ELBO objective:

$$\mathcal{L}_{\text{ELBO}} = \mathcal{L}_{\text{Recon}} + \alpha \mathcal{L}_{\text{Quant}} + \beta \mathcal{L}_{\text{Commit}} \tag{4}$$

where α and β are weights for the corresponding loss components. Specifically, (1) *Reconstruction loss* $\mathcal{L}_{\text{Recon}} = \sum_t \|\mathbf{s}_t^m - \text{De}(\text{En}(\mathbf{s}_{0:T}^m), \mathbf{s}_{t-1}^m, \mathbf{a}_{t-1}^m)\|^2$. This measures the discrepancy between the reconstructed state $\hat{\mathbf{s}}_t^m$ and the original state $\mathbf{s}_t^m$, where En and De are the encoder and decoder, respectively. (2) *Quantization loss* $\mathcal{L}_{\text{Quant}} = \sum_i \|\text{sg}[z_{m,i}] - e_{m,i}\|^2$. This evaluates the discrepancy between the encoder output z_m and its discretized representation e_m. Since the quantization step is undifferentiable, we use the stop-gradient operation $\text{sg}[\cdot]$ to update the dictionary vectors without affecting the encoder parameters. (3) *Commitment loss* $\mathcal{L}_{\text{Commit}} = \sum_i \|z_{m,i} - \text{sg}[e_{m,i}]\|^2$. This term minimizes the discrepancy between z_m and e_m, encouraging z_m to align closely with the embedding

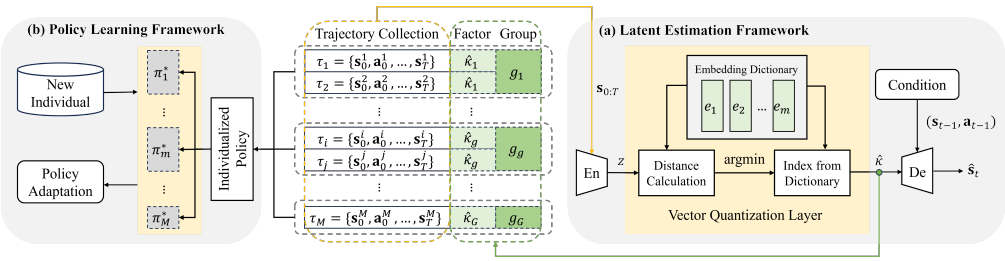


Figure 2: (a) Latent estimation framework takes each trajectory $\mathbf{s}_{0:T}$ as input and processes it through a quantized encoder to estimate the latent factor $\hat{\kappa}$. A conditional decoder then reconstructs $\hat{\mathbf{s}}_t$, using $(\mathbf{s}_{t-1}, \mathbf{a}_{t-1})$ as condition and $\hat{\kappa}$ as input. (b) Once the estimated latent factors are assigned to each trajectory, the policy learning framework integrates these latents as augmented labels to optimize the RL policy. For new individuals, the initial policy is adapted based on their group affiliation, enabling individualized policy adaptation for newcomers.

space. By applying the stop gradient to $e_{m,i}$, gradients from this loss do not change the dictionary vectors, but instead optimize the encoder parameters.

5.2 Policy Learning Framework

The estimation network is pre-trained offline. When a new individual arrives, we estimate κ and adapt the policy simultaneously through new interactions. Specifically,

Latent-based Policy Individualization The estimated latent individual-specific factors $\hat{\kappa}$, together with the offline trajectories from all individuals, are used to learn the individualized policy π^*_κ. We view the estimated factors as an augmented component of the policy input, and the policy training objective is adjusted to match the unique characteristics of each individual.

Take Q-learning [58] as an example. In the individualized process, the latent factor is augmented as a policy input as $\mu_\pi(\mathbf{s}_t; \theta^\mu) \rightarrow \mu^m_\pi(\mathbf{s}^m_t, \hat{\kappa}_m; \theta^\mu)$, where θ^μ represents the parameters of the policy network. This incorporation allows the policy to adapt effectively to individual-specific dynamics. The training objective is updated accordingly as $\mathcal{J}(\theta^\mu) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t Q\left(\mathbf{s}_t, \mu^m_\pi(\mathbf{s}^m_t, \hat{\kappa}_m; \theta^\mu); \theta^Q\right)\right]$, where γ is the discount factor, and Q represents the Q value. Such individualization improves policy adaptability across varying environments, and our framework is general enough to be seamlessly integrated with various RL algorithms.

Policy Adaptation for New Individual Policy adaptation for a new individual involves two steps: initializing the policy from the individualized policy π^*_κ and fine-tuning it through new interactions. For a new individual from group g_n, the group factor $\hat{\kappa}_n$ is first estimated. The policy network π_{new} is then initialized by directly transferring parameters from $\pi^*_{\kappa=\hat{\kappa}_n}$. Specifically, $\pi_{\text{new}} = \arg\max_\pi \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \in D_n}(R_{\mathbf{a}_t}(\mathbf{s}_t))$ is fine-tuned based on the new individual's trajectory D_n by maximizing the expected reward. The dataset D_n is incrementally augmented with new observations $(\mathbf{s}_t, \mathbf{a}_t, R_t, \mathbf{s}_{t+1})$ collected under π_{new}. This process refines the policy to better fit the specific characteristics of the new individual. Additionally, the updated D_n helps to accurately estimate the latent factor of the new individual, further improving the adaptation process.

6 Experiment

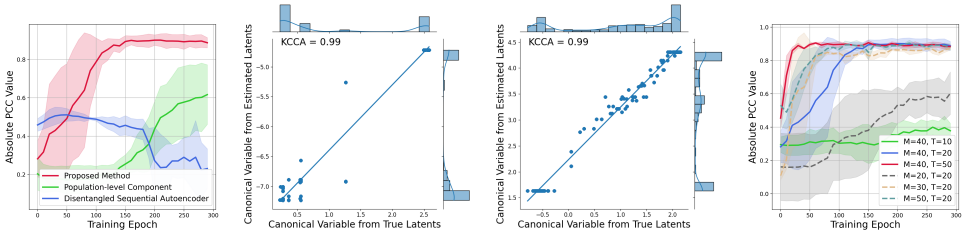
Evaluation Metrics To measure the latent identification, we quantify the correlation between the estimated and true latent factors by: (1) Pearson Correlation Coefficient (PCC) for single latent, which quantifies the linear correlation between individual estimated and true factors, and (2) Kernel Canonical Correlation Analysis (KCCA) for multiple latents, which evaluates the correlation between sets of estimated and true factors. An absolute value close to 1 indicates a strong correlation and better latent recovery. To evaluate the control performance, we measure the adaptation performance using: (3) jumpstart, which records the improvement in initial performance when a learning agent leverages knowledge from source tasks, and (4) accumulative reward, which indicates the learning

quality over the learning process. (5) initial and final reward, which measures the initial performance benefited from policy adaptation and the performance after the full training process.

Baselines For estimation evaluation, our baselines include: (1) disentangled sequential autoencoder [88], which disentangles the latent representations into static and dynamic parts instead of considering the global influence of κ. (2) Population-level component, which embeds the latent factors using population data rather than on an individual basis. For policy evaluation, our baselines include: (3) aligned latent models [22], which jointly optimizes a latent-space model and a policy to achieve high returns. (4) Soft Actor-Critic (SAC) [26], which uses entropy as part of the objective function to encourage exploration and improve robustness. (5) Deep Deterministic Policy Gradient (DDPG) [58], which combines deterministic policy gradients with deep neural networks to effectively handle continuous action spaces. (6) Dueling Double Deep Q-Network (D3QN) [81], which introduces a dueling architecture for value function estimation and improves value estimation. (7) Rainbow DQN [30], which combines prioritized experience replay and dueling network architectures to improve performance and learning stability.

6.1 Evaluation on Latent Estimation Framework

Synthetic Experiments We first run experiments on the synthetic dataset to demonstrate the effectiveness of the estimation framework, which is manually generated following the post-nonlinear model. We design three types of latent factors κ, each either satisfying or violating the required assumptions. **Case 1**: κ is a finite latent factor following the categorical distribution $\text{Cat}(0.1, 0.2, 0.3, 0.4)$ with cardinality equal to 4. **Case 2**: κs are three-dimensional finite latent factors, and each factor follows the categorical distributions $\text{Cat}(0.2, 0.8)$, $\text{Cat}(0.2, 0.3, 0.5)$, $\text{Cat}(0.1, 0.2, 0.3, 0.4)$, with cardinality equal to 2, 3, 4, respectively. **Case 3**: κs are three-dimensional infinite latent factors, and each factor follows the Gaussian distribution $\mathcal{N}(0, 1)$, uniform distribution $\text{Uniform}(0, 1)$, and exponential distribution $\text{Exp}(1)$, respectively. We synthetically generate 40 unique trajectories, each representing an individual, with a maximum length of 20 per trajectory.



(a) (b) (c) (d)

Figure 3: Synthetic results. (a) Comparisons of PCC trajectories in Case 1. (b-c) Scatterplot of the canonical variables in Case 2 and 3. (d) Identifiability performance responses of the sample size.

For case 1, we use PCC to measure the estimation performance and report the training curve in Figure 3(a), where the shaded regions represent the standard deviation. The comparative results show that our method can recover the true latent factors, which outperforms other baselines. In particular, the population-level component overlooks the underlying components between different groups, thus failing to identify the individual-specific factor. Furthermore, although the disentangled sequential autoencoder achieves compromised identifiability in the early training stage by considering the static part in the latent space, it fails to reach full identifiability because it overlooks the individualized transition processes, leading to worse recovery performance over time. For cases 2 and 3, we use KCCA to quality the correlation and visualize the scatter plots in Figure 3(b) and Figure 3(c). These results highlight the strong identifiability in the infinite cases and verify the claim in Theorem 4.3.

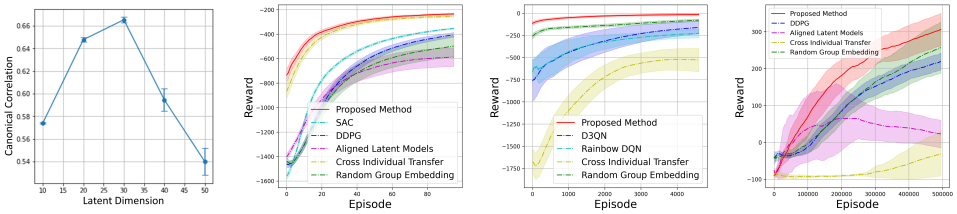
Moreover, we slightly violate the required sample size assumption in Theorem 4.1 and report the change in the training curve under different population sizes in Figure 3(d), represented by the dashed lines. The result shows that satisfying the sample sufficiency assumption is necessary to recover the latent factor. In addition, we evaluate the effect of the trajectory length as described in the Theorem 4.2. The findings, shown as solid lines, indicate that increasing the sample size would apparently improve the identifiability performance, which is consistent with the proposed theorem.

Ablation Study The contributions of the different components in the latent estimation framework are reported in Table 1. We build on the autoencoder framework with a quantization layer and add each component sequentially to the previous module. Incorporating a sequential encoder significantly improves the identifiability, which is important for the accurate recovery of latent factors. In the implementation, we use a noise estimator during optimization to minimize bias and improve identifiability. The results suggest that the noise estimator contributes to fine-tuning the overall performance of the model, allowing for more accurate and reliable recovery of latent individual-specific factors.

Table 1: Contribution of each module.

Module	**PCC** ($\mu \pm \sigma$)	**Bias** ($\mu \pm \sigma$)
Quantized Encoding	0.646 $\pm$ 3.1e-04	0.099 $\pm$ 2.3e-04
+ Sequential Encoder	0.910 $\pm$ 1.3e-04	0.077 $\pm$ 5.7e-06
+ Noise Estimator	0.942 $\pm$ 4.0e-05	0.072 $\pm$ 3.0e-07

PersuasionForGood Corpus We further evaluate our framework on the real-world dataset, PersuasionForGood corpus [80], which is widely used for analyzing persuasion strategies [66, 8, 90]. It consists of 1017 person-to-person dialogues and 32 personality traits of each participant. In each dialogue, the persuader tries to convince the persuadee to donate to a charity. In the iMDP context, the state refers to the persuadee's response, the action refers to the persuader's utterance, and the reward is the final donation. Since this offline dataset lacks real-time interactions necessary for assessing control performance, we instead use it to identify the latent personality of each individual. We use BERT [11] as the backbone to embed each utterance into a 768-dimensional feature representation and then use an LSTM encoder followed by the quantization layer to recover the latent personalities. The CCA results under different latent dimensions are shown in Figure 4(a), demonstrating that our method can achieve remarkable performance on the real dataset by appropriately fine-tuning the latent dimensions.



(a) (b) (c) (d)

Figure 4: (a) Canonical correlation with respect to the latent dimensions in the PersuasionForGood corpus. (b-d) Accumulative reward curves in Pendulum, HeartPole, and Half Cheetah, respectively.

6.2 Evaluation on Policy Learning Framework

Pendulum Pendulum [5] is a continuous control task for RL study with the goal of swinging up and stabilizing in an upright position. The states are the x-y coordinates and angular velocity, and the action is the torque applied to the pendulum. For simplicity, we choose DDPG as the foundational optimization algorithm and manually create 20 individualized environments. In these environments, the gravity g is randomly drawn from a categorical distribution over the set $\{3, \ldots, 12\}$. The performance of the policy adaptation is evaluated on a new individual with $g = 10$.

We compare our method against several baselines: (1) SAC; (2) DDPG without prior knowledge; (3) aligned latent models; (4) pre-trained DDPG incorporating knowledge from given individuals, termed cross-individual transfer; (5) individualized policy incorporating randomly defined group embedding, termed random group embedding. The training curves over accumulative reward are reported in Figure 4(b), showing that the proposed method outperforms other baselines in both jumpstart and accumulative reward. Specifically, methods that benefit from population knowledge (our method and cross-individual transfer) outperform non-transfer methods, indicating that the pre-trained policy would accelerate the learning process. However, since cross-individual transfer ignores the individual-specific information, such mixed policy knowledge yields worse initial performance compared to the individualized policies derived from our method.

HeartPole HeartPole [59] is a discrete healthcare environment that explores the long-term health outcomes of short-term decisions. The six-dimensional states represent different health conditions, including alertness, hypertension, intoxication, time since sleep, time elapsed, and work done. Actions can be chosen from work, coffee, alcohol, and sleep. We create 100 individualized scenarios and assign each patient with individual characteristics, such as coffee tolerance, hypertension risk, and alcohol tolerance, according to a categorical distribution over the set $\{0.6, 0.8, 1.2\}$. The adaptation performance is evaluated on a new individual with all indices set to 1.

We compare our method against the following baselines: (1) D3QN without prior knowledge, (2) Rainbow DQN, (3) cross-individual transfer with D3QN, and (4) random group embedding. The training curves over accumulative reward are shown in Figure 4(c), and our method outperforms other baselines in both jumpstart and accumulative reward. Interestingly, although inappropriate source domain knowledge may harm the control performance (see cross-individual transfer), the result of random group embedding indicates that group embedding knowledge can encourage the performance of generalization. The group structure, together with properly estimated group information, jointly allows our method to converge better and faster than other baselines.

Half Cheetah Half Cheetah [74] is a Mujoco-based task aiming to control a 2D bipedal robot. The agent consists of 9 links and 8 joints, and the goal is to apply torque to the joints to make the cheetah run forward as fast as possible. We introduced 50 individualized settings with the gravity g following a categorical distribution with probabilities $p = 0.2$ and corresponding g values from $\{8, 8.5, \ldots, 10\}$. The adaptation performance is evaluated on a new individual with $g = 9.8$. We compare our method with (1) DDPG without prior knowledge, (2) aligned latent models, (3) cross-individual transfer with DDPG, and (4) random group embedding. The comparative results are shown in Figure 4(d). We found that inappropriate source domain data can degrade the control performance (see cross-individual transfer), but the integration of group embedding facilitates generalization, allowing our method to outperform baselines in terms of convergence speed and efficiency.

7 Conclusion and Limitations

Our work focuses on learning latent state-transition processes from observed state-action trajectories, guaranteeing identifiability in the presence of latent individual-specific factors. To the best of our knowledge, this study provides novel identifiability guarantees for several settings that have not been previously addressed. Despite these contributions, our approach has three major limitations. (1) It currently does not account for instantaneous causal dependencies within $\mathbf{s}_t$. This limitation could be addressed by adjusting the temporal resolution of the data and explicitly modeling these dependencies. Integrating causal graphical models or advanced inference techniques could further enhance the framework's ability to handle instantaneous causal relationships. (2) While empirical results suggest the framework may adapt to scenarios with continuous latent factors, a formal nonparametric proof for these settings remains absent. Developing such a proof is an important avenue for future work. (3) The model assumes latent factors are time-invariant. Establishing such theoretical identifiability for time-varying latent factors is highly non-trivial and would require additional theoretical constraints to establish identifiability. These limitations highlight key directions for future research.

Furthermore, practical concerns such as privacy, robustness, and reliability are essential for real-world applications. To address privacy risks, techniques like de-identification (e.g., removing direct identifiers, data perturbation, pseudonymization) and differential privacy approaches should be explored. Ensuring privacy and security will enhance the framework's applicability in practice.

Acknowledgement

This material is based upon work supported by NSF Award No. 2229881, AI Institute for Societal Decision Making (AI-SDM), the National Institutes of Health (NIH) under Contract R01HL159805, and grants from Salesforce, Apple Inc., Quris AI, and Florin Court Capital. BH is supported by NSF DMS-2428058.

References

[1] M Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7):1–38, 2022.

[2] Jonathan Bassen, Bharathan Balaji, Michael Schaarschmidt, Candace Thille, Jay Painter, Dawn Zimmaro, Alex Games, Ethan Fast, and John C Mitchell. Reinforcement learning for the adaptive scheduling of educational activities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[3] Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023.

[4] Andrew Bennett, Nathan Kallus, Lihong Li, and Ali Mousavi. Off-policy evaluation in infinite-horizon reinforcement learning with latent confounders. In *International Conference on Artificial Intelligence and Statistics*, pages 1999–2007. PMLR, 2021.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[6] Qingpeng Cai, Ruohan Zhan, Chi Zhang, Jie Zheng, Guangwei Ding, Pinghua Gong, Dong Zheng, and Peng Jiang. Constrained reinforcement learning for short video recommendation. *arXiv preprint arXiv:2205.13248*, 2022.

[7] Guangyi Chen, Yifan Shen, Zhenhao Chen, Xiangchen Song, Yuewen Sun, Weiran Yao, Xiao Liu, and Kun Zhang. Caring: Learning temporal causal representation under non-invertible generation process. *arXiv preprint arXiv:2401.14535*, 2024.

[8] Maximillian Chen, Weiyan Shi, Feifan Yan, Ryan Hou, Jingwen Zhang, Saurav Sahay, and Zhou Yu. Seamlessly integrating factual information and social content with persuasive dialogue. *arXiv preprint arXiv:2203.07657*, 2022.

[9] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009.

[10] Florent Delgrange, Ann Nowe, and Guillermo A Pérez. Wasserstein auto-encoded mdps: Formal verification of efficiently distilled rl policies with many-sided guarantees. *arXiv preprint arXiv:2303.12558*, 2023.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Xinshuai Dong, Biwei Huang, Ignavier Ng, Xiangchen Song, Yujia Zheng, Songyao Jin, Roberto Legaspi, Peter Spirtes, and Kun Zhang. A versatile causal discovery framework to allow causally-related hidden variables. *arXiv preprint arXiv:2312.11001*, 2023.

[13] F. Ebert, C. Finn, A. X. Lee, and S. Levine. Self-supervised visual planning with temporal skip connections. *ArXiv Preprint ArXiv:1710.05268*, 2017.

[14] Bisni Fahad Mon, Asma Wasfi, Mohammad Hayajneh, Ahmad Slim, and Najah Abu Ali. Reinforcement learning in education: A literature review. In *Informatics*, volume 10, page 74. MDPI, 2023.

[15] Fan Feng, Biwei Huang, Kun Zhang, and Sara Magliacane. Factored adaptation for non-stationary reinforcement learning. *Advances in Neural Information Processing Systems*, 35:31957–31971, 2022.

[16] Apple WP Fok, Hau-San Wong, and YS Chen. Hidden markov model based characterization of content access patterns in an e-learning environment. In *2005 ieee international conference on multimedia and expo*, pages 201–204. IEEE, 2005.

[17] Evan M Forman, Michael P Berry, Meghan L Butryn, Charlotte J Hagerman, Zhuoran Huang, Adrienne S Juarascio, Erica M LaFata, Santiago Ontañón, J Mick Tilford, and Fengqing Zhang. Using artificial intelligence to optimize delivery of weight loss treatment: Protocol for an efficacy and cost-effectiveness trial. *Contemporary Clinical Trials*, 124:107029, 2023.

[18] Evan M Forman, Stephanie G Kerrigan, Meghan L Butryn, Adrienne S Juarascio, Stephanie M Manasse, Santiago Ontañón, Diane H Dallal, Rebecca J Crochiere, and Danielle Moskow. Can the artificial intelligence technique of reinforcement learning use continuously-monitored digital data to optimize treatment for weight loss? *Journal of behavioral medicine*, 42:276–290, 2019.

[19] C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning (ICML)*, 2019.

[20] D. Ghosh, A. Gupta, and S. Levine. Learning actionable representations with goal conditioned policies. *ICLR*, 2019.

[21] Susobhan Ghosh, Raphael Kim, Prasidh Chhabria, Raaz Dwivedi, Predrag Klasjna, Peng Liao, Kelly Zhang, and Susan Murphy. Did we personalize? assessing personalization by an online reinforcement learning algorithm using resampling. *arXiv preprint arXiv:2304.05365*, 2023.

[22] Raj Ghugare, Homanga Bharadhwaj, Benjamin Eysenbach, Sergey Levine, and Ruslan Salakhutdinov. Simplifying model-based rl: learning representations, latent-space models, and policies with one objective. *arXiv preprint arXiv:2209.08466*, 2022.

[23] K. Gregor, G. Papamakarios, F. Besse, L. Buesing, and T. Weber. Temporal difference variational auto-encoder. *arXiv preprint arXiv:1806.03107*, 2018.

[24] Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *International Conference on Machine Learning*, pages 3875–3886. PMLR, 2020.

[25] D. Ha and J. Schmidhuber. World models. In *Advances in Neural Information Processing Systems*, 2018.

[26] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[27] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

[28] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[29] Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

[30] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[31] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[32] Jesse Hoey, Pascal Poupart, Craig Boutilier, and Alex Mihailidis. Pomdp models for assistive technology. In *Assistive Technologies: Concepts, Methodologies, Tools, and Applications*, pages 120–140. IGI Global, 2014.

[33] William Hoiles, Vikram Krishnamurthy, and Kunal Pattanayak. Rationally inattentive inverse reinforcement learning explains youtube commenting behavior. *The Journal of Machine Learning Research*, 21(1):6879–6917, 2020.

[34] Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics: methodology and distribution*, pages 162–190. Springer, 1992.

[35] Biwei Huang, Fan Feng, Chaochao Lu, Sara Magliacane, and Kun Zhang. Adarl: What, where, and how to adapt in transfer reinforcement learning. *arXiv preprint arXiv:2107.02729*, 2021.

[36] Biwei Huang, Charles Jia Han Low, Feng Xie, Clark Glymour, and Kun Zhang. Latent hierarchical causal structure discovery with rank constraints. *Advances in Neural Information Processing Systems*, 35:5549–5561, 2022.

[37] Biwei Huang, Kun Zhang, Pengtao Xie, Mingming Gong, Eric P Xing, and Clark Glymour. Specific and shared causal relation modeling and mechanism-based clustering. *Advances in Neural Information Processing Systems*, 32, 2019.

[38] Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in neural information processing systems*, 29, 2016.

[39] Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019.

[40] Xiaoyuan Ji, Hu Ye, Jianxin Zhou, Yajun Yin, and Xu Shen. An improved teaching-learning-based optimization algorithm and its application to a combinatorial optimization problem in foundry industry. *Applied Soft Computing*, 57:504–516, 2017.

[41] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, $\cdots$, and H. Michalewski. Model-based reinforcement learning for Atari. *arXiv preprint arXiv:1903.00374*, 2019.

[42] M. Karl, M. Soelch, J. Bayer, and P. van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016.

[43] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

[44] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[45] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[46] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

[47] Lingjing Kong, Shaoan Xie, Weiran Yao, Yujia Zheng, Guangyi Chen, Petar Stojanov, Victor Akinwande, and Kun Zhang. Partial identifiability for domain adaptation. *arXiv preprint arXiv:2306.06510*, 2023.

[48] R.G. Krishnan, U. Shalit, and D. Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[49] T. D. Kulkarni, A. Saeedi, S. Gautam, and S. J. Gershman. Deep successor reinforcement learning. *arXiv preprint arXiv:1606.02396*, 2016.

[50] Jeongyeol Kwon, Yonathan Efroni, Constantine Caramanis, and Shie Mannor. Rl for latent mdps: Regret guarantees and a lower bound. *Advances in Neural Information Processing Systems*, 34:24523–24534, 2021.

[51] Jeongyeol Kwon, Yonathan Efroni, Shie Mannor, and Constantine Caramanis. Prospective side information for latent mdps. *arXiv preprint arXiv:2310.07596*, 2023.

[52] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[53] Yu Lei and Wenjie Li. Interactive recommendation with user-specific deep reinforcement learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(6):1–15, 2019.

[54] T. Lesort, N. Díaz-Rodríguez, J. F. Goudou, and D. Filliat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, 2018.

[55] Minne Li, Mengyue Yang, Furui Liu, Xu Chen, Zhitang Chen, and Jun Wang. Causal world models by unsupervised deconfounding of physical dynamics. *arXiv preprint arXiv:2012.14228*, 2020.

[56] Zhige Li, Derek Yang, Li Zhao, Jiang Bian, Tao Qin, and Tie-Yan Liu. Individualized indicator for all: Stock-wise technical indicator optimization with stock embedding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 894–902, 2019.

[57] Peng Liao, Kristjan Greenewald, Predrag Klasnja, and Susan Murphy. Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–22, 2020.

[58] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[59] Vadim Liventsev, Aki Härmä, and Milan Petković. Towards effective patient simulators. *Frontiers in artificial intelligence*, 4:798659, 2021.

[60] Chaochao Lu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Deconfounding reinforcement learning in observational settings. *arXiv preprint arXiv:1812.10576*, 2018.

[61] Zhiyao Luo, Mingcheng Zhu, Fenglin Liu, Jiali Li, Yangchen Pan, Jiandong Zhou, and Tingting Zhu. Dtr-bench: An in silico environment and benchmark platform for reinforcement learning based dynamic treatment regime. *arXiv preprint arXiv:2405.18610*, 2024.

[62] S. Mahadevan and M.. Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research (JMLR)*, 8:2169–2231, 2007.

[63] Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6:355–378, 2019.

[64] Kevin P Murphy. A survey of pomdp solution techniques. *environment*, 2(10), 2000.

[65] Alizée Pace, Hugo Yèche, Bernhard Schölkopf, Gunnar Rätsch, and Guy Tennenholtz. Delphic offline reinforcement learning under nonidentifiable hidden confounding. *arXiv preprint arXiv:2306.01157*, 2023.

[66] Wei Peng, Yue Hu, Luxi Xing, Yuqiang Xie, and Yajing Sun. Do you know my emotion? emotion-aware strategy recognition towards a persuasive dialogue system. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 724–739. Springer, 2022.

[67] Alexander Ritchie, Robert A Vandermeulen, and Clayton Scott. Consistent estimation of identifiable nonparametric mixture models from grouped observations. *Advances in Neural Information Processing Systems*, 33:11676–11686, 2020.

[68] Doaa Shawky and Ashraf Badawi. Towards a personalized learning experience using reinforcement learning. *Machine learning paradigms: Theory and application*, pages 169–187, 2019.

[69] Dongjian Song, Bing Zhu, Jian Zhao, Jiayi Han, and Zhicheng Chen. Personalized car-following control based on a hybrid of reinforcement learning and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[70] Seth Sullivant, Kelli Talaska, and Jan Draisma. Trek separation for gaussian graphical models. 2010.

[71] Yuewen Sun, Erli Wang, Biwei Huang, Chaochao Lu, Lu Feng, Changyin Sun, and Kun Zhang. Acamda: Improving data efficiency in reinforcement learning through guided counterfactual data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15193–15201, 2024.

[72] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[73] Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[74] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[75] Momchil S Tomov, Eric Schulz, and Samuel J Gershman. Multi-task reinforcement learning in humans. *Nature Human Behaviour*, 5(6):764–773, 2021.

[76] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[77] Robert A Vandermeulen and Clayton D Scott. On the identifiability of mixture models from grouped samples. *arXiv preprint arXiv:1502.06644*, 2015.

[78] Thanh Vinh Vo, Pengfei Wei, Wicher Bergsma, and Tze Yun Leong. Causal modeling with stochastic confounders. In *International Conference on Artificial Intelligence and Statistics*, pages 3025–3033. PMLR, 2021.

[79] Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Provably efficient causal reinforcement learning with confounded observational data. *Advances in Neural Information Processing Systems*, 34:21164–21175, 2021.

[80] Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, Florence, Italy, July 2019. Association for Computational Linguistics.

[81] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016.

[82] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *NeurIPS*, 2015.

[83] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

[84] Xinghao Yang, Weifeng Liu, Wei Liu, and Dacheng Tao. A survey on canonical correlation analysis. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2349–2368, 2019.

[85] Weiran Yao, Guangyi Chen, and Kun Zhang. Temporally disentangled representation learning. *Advances in Neural Information Processing Systems*, 35:26492–26503, 2022.

[86] Weiran Yao, Yuewen Sun, Alex Ho, Changyin Sun, and Kun Zhang. Learning temporally causal latent processes from general temporal data. *arXiv preprint arXiv:2110.05428*, 2021.

[87] Chunli Yin and Jinglong Han. Dynamic pricing model of e-commerce platforms based on deep reinforcement learning. *CMES-Computer Modeling in Engineering & Sciences*, 127(1), 2021.

[88] Li Yingzhen and Stephan Mandt. Disentangled sequential autoencoder. In *International Conference on Machine Learning*, pages 5670–5679. PMLR, 2018.

[89] Elad Yom-Tov, Guy Feraru, Mark Kozdoba, Shie Mannor, Moshe Tennenholtz, and Irit Hochberg. Encouraging physical activity in patients with diabetes: Intervention using a reinforcement learning system. *Journal of medical Internet research*, 19(10):e338, 2017.

[90] Donghuo Zeng, Roberto S Legaspi, Yuewen Sun, Xinshuai Dong, Kazushi Ikeda, Peter Spirtes, and Kun Zhang. Counterfactual reasoning using predicted latent personality dimensions for optimizing persuasion outcome. In *International Conference on Persuasive Technology*, pages 287–300. Springer, 2024.

[91] A. Zhang, R. McAllister, R. Calandra, Y. Gal, and S. Levine. Learning invariant representations for reinforcement learning without reconstruction. *ICLR*, 2021.

[92] Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*, pages 11214–11224. PMLR, 2020.

[93] Kun Zhang and Aapo Hyvarinen. On the identifiability of the post-nonlinear causal model. *arXiv preprint arXiv:1205.2599*, 2012.

[94] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. Johnson, and S. Levine. Self-supervised visual planning with temporal skip connections. *ICML*, 2019.

[95] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. J. Johnson, and S. Levine. Solar: deep structured representations for model-based reinforcement learning. *arXiv preprint arXiv:1808.09105*, 2018.

[96] Xuezhou Zhang, Yuda Song, Masatoshi Uehara, Mengdi Wang, Alekh Agarwal, and Wen Sun. Efficient reinforcement learning in block mdps: A model-free representation learning approach. In *International Conference on Machine Learning*, pages 26517–26547. PMLR, 2022.

[97] Yujia Zheng, Ignavier Ng, and Kun Zhang. On the identifiability of nonlinear ica: Sparsity and beyond. *Advances in Neural Information Processing Systems*, 35:16411–16422, 2022.

Supplementary Materials for "Identifying Latent State-Transition Processes for Individualized Reinforcement Learning"

A Notation and Terminology

We summarize the notations used throughout the paper in the following table.

Index	
τ	Trajectory
t	Time index
T	Total length of time series
G	Number of groups
M	Number of individuals
m	Index for a specific individual
i, j	Variable element index
α, β	Weights of ELBO objective
$[G] = \{1, 2, \ldots, G\}$	Sequence of integers from 1 to G inclusive
Variable	
ϵ_{it}	i.i.d. noise term for s_i at time t
f_i	Nonparametric state transition function
$\mathbf{s}_t$, $\hat{\mathbf{s}}_t$	Observed & reconstructed states at time t
$\mathbf{s}^m, \mathbf{a}^m, \kappa^m$	State, action, and latent factor from individual m
$\mathbf{s} = [s_1, s_2, \ldots, s_{d_s}]^\top$	d_s-dimensional observed states
$\mathbf{a} = [a_1, a_2, \ldots, a_{d_a}]^\top$	d_a-dimensional observed actions
$\kappa = [\kappa_1, \kappa_2, \ldots, \kappa_{d_\kappa}]^\top$	d_κ-dimensional latent individual-specific factors

B Identifiability Theory

Given the identifiability theorems, we first provide intuitive explanations for each assumption and discuss their relevance to real-world applications. Then, we provide the proof. Finally, we introduce some preliminaries related to our theorems, which are essential for the proof.

B.1 Preliminaries for Theorem 4.1 and 4.2

B.1.1 Markov Property

The first-order Markov property implies that the transition probability to the next state depends only on the current state, uninfluenced by the sequence of previous states. Specifically,

Definition B.1 (First-order Markov Property [72]). *A stochastic process $\{X_t : t \in \mathcal{N}\}$ has the first-order Markov property if, for each set of times $t, t-1, \ldots, 0$ and corresponding state $x_t, x_{t-1}, \ldots, x_0$ in the state space, the following conditional independence property holds:*

$$\mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \ldots, X_0 = x_0) = \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}) \tag{5}$$

The first-order Markov property implies that the transition probability to the next state depends only on the current state, uninfluenced by the previous states. In the context of the state transition process, it possesses the first-order Markov property. Mathematically, it can be represented as:

$$\mathbb{P}(\mathbf{s}_t | \mathbf{s}_{t-1}, \mathbf{a}_{t-1}, \mathbf{s}_{t-2}, \mathbf{a}_{t-2}, \ldots, \mathbf{s}_0, \mathbf{a}_0) = \mathbb{P}(\mathbf{s}_t | \mathbf{s}_{t-1}, \mathbf{a}_{t-1}), \tag{6}$$

where $\mathbb{P}(\mathbf{s}_t | \mathbf{s}_{t-1}, \mathbf{a}_{t-1})$ is the transition probability from $(\mathbf{s}_{t-1}, \mathbf{a}_{t-1})$ to the state $\mathbf{s}_t$.

B.1.2 Finite Mixture Model

A finite mixture model is used for modeling a total population that comprises unobserved or hidden groups. Each of these groups is assumed to follow its own distinct probability distribution. In this context, the overall population model is expressed as a weighted sum of these individual distributions [63]. Specifically,

Definition B.2 (Finite Mixture Models [77])**.** *A finite mixture model is a probability law based on a finite number of probability measures, $\mu_1, \ldots, \mu_m$, and a discrete distribution $\omega_1, \ldots, \omega_m$. A realization of a mixture model is generated by generating a component at random k, $1 \leq k \leq m$, and then drawing from $\mu_k \sim \mathcal{P}$. Then, the mixture measure $\mathcal{P}$ is defined as a weighted sum of probability measures μ_i with weights w_i. Specifically,*

$$\mathcal{P} = \sum_{i=1}^{m} w_i \delta_{\mu_i}. \tag{7}$$

B.2 Discussions on Assumptions

Group Determinacy The distinct values of the latent factor κ categorize the population into separate groups, with each group characterized by its unique probability distribution and denoted as $\sum_{g=1}^{G} \pi_g \delta_{\kappa_g}(\kappa)$. The mixture formulation implies that the latent factor κ serves as a categorical variable, with each unique value explicitly specifying a distinct group within the population. Such a formulation facilitates the identification and analysis of heterogeneous subpopulations within the finite mixture model.

The idea of group determinacy is important in real-world applications. Take personalized education as an example. For each student m, $\mathbf{s}^m$ represents their current knowledge state, $\mathbf{a}^m$ denotes their personalized learning action, and the function f determines the unique educational trajectory for each student. The latent factor κ^m influences how a student's learning progresses over time. It can be based on factors such as learning style preferences that help to logically group students. Specifically, one group might consist of visual learners who excel in interactive, graphically-oriented subjects, while another group might include students who prefer textual information and excel in reading and writing-intensive subjects. Each group exhibits its own set of learning outcomes and patterns, allowing educators to personalize teaching methods and materials to effectively meet the different needs of each group.

Sample Sufficiency In a finite mixture model with G groups, each group requires sufficient observations to identify the latent group factor κ. This assumption provides a minimum number of observation samples, which is $2G - 1$ observations in each group. Such a threshold ensures that we have enough information and variability in the observed data to distinguish the characteristics of each group. This assumption helps to identify the unique characteristics of each individual, which is critical for identifiability.

Sample sufficiency indicates that sufficient data are needed to achieve identifiability, which is a fundamental assumption in many analytical models. For example, in the context of nonlinear ICA using auxiliary variables [39], it is necessary to have at least $2n + 1$ values for the auxiliary variables to ensure sufficient variability and guarantee identifiability. Similarly, for successful disentanglement with minimal change [47], at least $2n + 1$ domain embeddings are required to ensure identifiability. Intuitively, without sufficient data to provide us with relevant information about the parameters, it is impossible to determine the values of these parameters.

Infinite Samples and Overlapping Conditions Asymptotic identifiability refers to the property that a model becomes identifiable as the sample size goes to infinity. In practical terms, this means that given an infinite amount of data, one would be able to consistently estimate the parameters of the model. In the context of finite mixture models, if there are enough samples for each individual, then the corresponding components can be identified directly from each individual [37].

The overlapping condition requires the existence of at least two different individuals within the same group of the population, who have identical conditional probabilities. This assumption is crucial as it ensures that the model accounts for overlapping behavioral responses between different individuals, which is a common phenomenon in heterogeneous populations. Consider personalized education as an example, where students come from different academic backgrounds and have different levels of prior knowledge. Despite this initial heterogeneity, it is possible for two students in the same learning group to have the same probability of successfully completing a task.

B.3 Proof of Theorem 4.1 and Theorem 4.2

We first show that the individualized transition processes can be viewed as a finite mixture model with grouped samples and then derive the identifiability under two scenarios.

B.3.1 Necessary Lemmas

Lemma B.1 addresses the identifiability of mixture models from grouped samples. It implies that with sufficient data per group, each component of the mixture model can be determined without ambiguity from the observed data. Specifically, suppose we have a mixture model consisting of G different probability distributions, that is, G components $c_1, c_2, \ldots, c_G$. Each component c_i corresponds to a unique probability density function $\mathbb{P}_i(\cdot)$. These components are mixed together, with each component having a mixing weight π_i, satisfying $\pi_i \geq 0$ and $\sum_{i=1}^{G} \pi_i = 1$. Now suppose we have G observation groups $g_1, g_2, \ldots, g_G$, with each group g_i containing observations that are independent and identically distributed drawn from the same component c_i.

Lemma B.1 (Identifiability of Mixture Models from Grouped Samples [77])**.** *Suppose we have observations from a mixture model and that they are grouped such that observations in the same group are known to be drawn from the same component. Denote by G the number of groups. If there are at least* $2G - 1$ *observations per group, any mixture of G probability measures can be uniquely identified.*

B.3.2 Proof of Identifiability

Proof. The observation model for each group g_i with a unique latent factor κ_i can be defined as $\mathbb{P}(s, a, s'|\kappa_i) = \int \mathbb{P}(s, a, s', u|\kappa_i)du$, where u denotes individual-specific factors within the group g_i. Since u comprises both κ and the initial state z, we have $\mathbb{P}(u) = \mathbb{P}(u)\mathbb{P}(\kappa|u) = \mathbb{P}(u, \kappa) = \mathbb{P}(u|\kappa)\mathbb{P}(\kappa)$. In this work, we consider two scenarios to provide identifiability from the observed temporal data collected from the population.

Theorem 4.1 Assumptions in theorem 4.1 and Eq. 2 ensure that individuals within the same group share identical joint distributions. Suppose the observations can be grouped into G finite components, then the joint distribution can be factorized as:

$$\mathbb{P}(s, a, s') = \int \mathbb{P}(s, a, s'|u)\mathbb{P}(u)du \tag{8}$$

$$= \int \mathbb{P}(u)\mathbb{P}(s'|s, a, \kappa)\mathbb{P}(s, a|u)du \tag{9}$$

$$= \int\int \mathbb{P}(u|\kappa)\mathbb{P}(\kappa)\mathbb{P}(s'|s, a, \kappa)\mathbb{P}(s, a|u)dud\kappa \tag{10}$$

$$= \int\int \mathbb{P}(\kappa)\mathbb{P}(s'|s, a, \kappa)\mathbb{P}(s, a|u)\mathbb{P}(u|\kappa)dud\kappa \tag{11}$$

$$= \int\int \mathbb{P}(\kappa)\mathbb{P}(s'|s, a, \kappa)\mathbb{P}(s, a, u|\kappa)dud\kappa \tag{12}$$

$$= \int \mathbb{P}(\kappa)\mathbb{P}(s'|s, a, \kappa)\mathbb{P}(s, a|\kappa)d\kappa \tag{13}$$

$$= \int \mathbb{P}(\kappa)\mathbb{P}(s', s, a|\kappa)d\kappa \tag{14}$$

This formulation asserts that the joint distribution of $\mathbb{P}(s, a, s')$ for the entire population can be modeled as a mixture model governed by the respective κ_i values. Since sample sufficiency ensures that the sample size of observations within each group is greater than $2G - 1$, then the identifiability of κ is guaranteed by Lemma B.1.

Theorem 4.2 According to assumptions in theorem 4.2, if each individual's trajectory is sufficiently long, the conditional model $\mathbb{P}(s'|s, a)$, as well as $\mathbb{P}(s, a)$ for the individual, would become asymptotically identifiable. Consider an extreme scenario where each individual is treated as a distinct group. Asymptotically, it is possible to identify the mixture distribution of the population. However, some

individuals may share the same κ and can be grouped together. Then, we need to find the similarity between different individuals and merge them into the same group.

An intuitive merging criterion is as follows. Asymptotically, it can be inferred that for some particular value of $(\mathbf{s}, \mathbf{a}) = (\mathbf{s}^*, \mathbf{a}^*)$, the probability of $\mathbf{s}'$ given $(\mathbf{s}^*, \mathbf{a}^*)$ will be the same across individuals in the same group. Define t^j as the time of the j-th occurrence of $(\mathbf{s}^*, \mathbf{a}^*)$. We can then define the collection of variables $\mathbf{X}^j$ as $\mathbf{X}^j = \{\mathbf{s}_{t^j+1}, t^j = 1, \ldots\}$, representing the state at time $t+1$ given the fixed state and action $(\mathbf{s}^*, \mathbf{a}^*)$ at time t. In this way, for any j, $\mathbf{X}^j$ sampled from a particular group j. Then, the identifiability is ensured by Lemma B.1.

□

Remark B.1. *Prior work [37] used a Gaussian mixture model as a prior on the coefficients, while the latent confounder variable, denoted as Z, was constrained to a binary state, thereby indicating group membership for a given individual. We extend this foundation by generalizing the latent confounder to a set of discrete values and considering a nonparametric model for broader applications.*

C Further Discussion on Identifiability Theorem

Recent work [12, 36] provides the necessary and sufficient conditions for the identifiability of certain latent structural patterns, but it rules out the case of triangle structure involving latent variables. In this paper, we extend their work to the temporal case and provide the identifiability of the latent group factor κ, where κ can be either continuous or discrete. Furthermore, we allow multiple instances of κ to influence the state transition dynamics.

C.1 Problem Setting

In Theorem 4.3, we aim to identify the latent group factors based on a post-nonlinear temporal causal model, as shown in Figure 5. We assume the existence of a learnable and invertible embedding mapping f, which is able to preserve the causal structure intrinsic to the state s. Specifically,

Definition C.1 (Post-nonlinear Temporal Causal Models)**.** *Consider a scenario where d_s observed states from the k-th individual are denoted as $\mathbf{s}_t^k = (s_{1,t}^k, \ldots, s_{d_s,t}^k)^{\mathrm{T}}$, which is a direct observation of an embedded representation $f(s)$, alongside m unobserved group factors $\kappa = (\kappa_1, \ldots, \kappa_m)^{\mathrm{T}}$. The state transition dynamics satisfy*

$$s_{i,t+1}^k = f^{-1}\left(\sum_{j \in \mathcal{P}_i} \alpha_{ij} f(s_{j,t}^k) + \sum_{j \in \mathcal{L}_i} \beta_{ij} a_{j,t}^k + \sum_{j=1}^{m} \lambda_j \kappa_j + \epsilon_{i,t+1}^k\right), \tag{15}$$

for $i = 1, \ldots, n$. Here, α_{ij} and β_{ij} represent causal coefficients that quantify the influence of the state $f(s_{j,t})$ and the action $a_{j,t}$ on $f(s_{i,t+1})$, respectively. $\mathcal{P}_i$ and $\mathcal{L}_i$ denote the sets of direct state and action that influences $s_{i,t+1}^k$ (or $f(s_{i,t+1}^k)$). Actions are considered to be stochastic. The coefficients λ_j are individual-specific and show variation across individuals. The random noise term $\epsilon_{i,t+1}^k$ is independent of $s_{j,t}$ and $a_{j,t}$ for all $j \in \mathcal{N}^+$ to account for unmeasured influences.

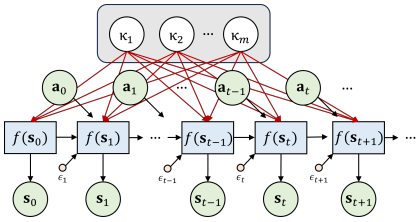


Figure 5: Post-nonlinear temporal causal model.

Definition C.2 (Minimal Rank Set)**.** *Let $\mathcal{S}_t = \{s_{1,t}, s_{2,t}, \ldots, s_{d_s,t}\}$ represent the set of all state variables in the system at any time $t = 1, \ldots, T$, and let $\mathcal{L} = \{\kappa_1, \kappa_2, \ldots, \kappa_m\}$ represent the set of latent variables, where m and d_s are the numbers of latent and state dimensions, respectively. A subset $\mathcal{R}_{t,t^-} \subseteq \mathcal{S}_t \cup \mathcal{S}_{<t}$ (or $\mathcal{R}_{t,t^+} \subseteq \mathcal{S}_t \cup \mathcal{S}_{>t}$) with cardinality r, is called a minimal rank set if it satisfies the following conditions:*

(i) The bottleneck set, defined as $\mathcal{B} = \mathcal{L} \cup \mathcal{S}_t$ for any given time t, can t-separate (see Definition C.5) any pair of minimal rank sets $(\mathcal{R}_{t,t^-}, \mathcal{R}_{t,t^+})$, where $t^- < t < t^+$.

(ii) There does not exist a subset $\mathcal{R}'_{t,t^\pm} \subset \mathcal{R}_{t,t^\pm}$ with $|\mathcal{R}'_{t,t^\pm}| < |\mathcal{R}_{t,t^\pm}|$ that can satisfy condition (i).

The set $\mathcal{R}_{t,t^\pm}$ is considered *minimal* in the sense that it is the smallest cardinality subset of observed state variables that includes a bottleneck set and disjoint state variables, capable of representing the essential separation status within the system. An illustrative example of a minimal rank set (see Figure 6) is shown in the yellow area, and a bottleneck set is depicted in the green area.

C.2 Proof of Theorem 4.3

The underlying intuition of Theorem 4.3 is that, in the absence of latent variables, rank information should align with what Conditional Independence (CI) skeleton (see Definition C.6) provides; if not, then there must exist at least one latent variable.

C.2.1 Necessary Lemmas

The following lemma indicates that the rank of the covariance matrix (see Definition C.3) $\Sigma_{\mathbf{A},\mathbf{B}}$ between any two sets of variables $\mathbf{A}$ and $\mathbf{B}$ is less than or equal to the sum of cardinalities of any trek-separating (see Definition C.5) sets $\mathbf{C_A}$ and $\mathbf{C_B}$. The equality holds for generic covariance matrices consistent with the graph $\mathcal{G}$.

Lemma C.1 (Trek Separation for Directed Graphical Models [70])**.** *The submatrix $\Sigma_{\mathbf{A},\mathbf{B}}$ has rank less than or equal to r for all covariance matrices consistent with the graph $\mathcal{G}$ if and only if there exist subsets $\mathbf{C_A}, \mathbf{C_B} \subset V(\mathcal{G})$ with $|\mathbf{C_A}| + |\mathbf{C_B}| \leq r$ such that $\mathbf{C_A}, \mathbf{C_B}$ t-separates $\mathbf{A}$ from $\mathbf{B}$. Consequently,*

$$\operatorname{rank}(\Sigma_{\mathbf{A},\mathbf{B}}) \leq \min\{|\mathbf{C_A}| + |\mathbf{C_B}| : (\mathbf{C_A}, \mathbf{C_B}) \text{ t-separates } \mathbf{A} \text{ from } \mathbf{B}\} \tag{16}$$

and equality holds for generic covariance matrices consistent with $\mathcal{G}$.

Lemma C.2 (Identifiability of Linear Regression Models)**.** *Consider a linear regression model with a response variable Y and p predictors $X_1, X_2, \ldots, X_p$. The linear relationship is defined as:*

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon \tag{17}$$

where $\beta_0, \beta_1, \ldots, \beta_p$ are the regression coefficients and ε is the error term. The matrix representation can be expressed as:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{18}$$

where $\mathbf{Y}$ is the response vector, $\mathbf{X}$ is the design matrix including predictors, $\boldsymbol{\beta}$ is the vector of regression coefficients, and $\boldsymbol{\varepsilon}$ is the vector of error terms. For the regression coefficients $\boldsymbol{\beta}$ to be identifiable, the design matrix $\mathbf{X}$ must have full column rank, meaning no predictor is a perfect linear combination of the others. This ensures that the matrix $\mathbf{X}^T\mathbf{X}$ is invertible, allowing for the unique estimation of $\boldsymbol{\beta}$ through:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{19}$$

Lemma C.3 (Identifiability of Factor Analysis)**.** *Consider a factor analysis model with p observations for each of n individuals and k common factors ($k < p$). The relationship is defined by the factor loading matrix $L \in \mathbb{R}^{p\times k}$ and the factor matrix $F \in \mathbb{R}^{k\times n}$. Specifically,*

$$X = LF + \varepsilon \tag{20}$$

where $X \in \mathbb{R}^{p\times n}$ is the observation matrix and $\varepsilon \in \mathbb{R}^{p\times n}$ is the error term matrix. The factor loading matrix L and the factor matrix F are unique up to an orthogonal transformation. Specifically, for any orthogonal matrix Q, if we set $L' = LQ$ and $F' = Q^T F$, the transformed matrices L' and F' also satisfy the model criteria.

C.2.2 Proof of Structure Identifiability

Proof. Suppose latent factors exist and influence the embedding of the observed states $f(s)$, which preserve the causal structure intrinsic to the states s. According to Lemma C.1, the rank of $\Sigma_{\mathbf{A_i},\mathbf{B_i}}$ should be less than or equal to $\min\{|\mathbf{C_{A_i}}| + |\mathbf{C_{B_i}}|\}$. In the absence of latent factors, according to the CI skeleton, the minimal configuration to t-separate $\mathbf{A_i}$ from $\mathbf{B_i}$ is by

$$(\{f(s_{1,i}), \ldots, f(s_{d_s,i})\}, \emptyset) \quad \text{or} \quad (\emptyset, \{f(s_{1,i}), \ldots, f(s_{d_s,i})\}). \tag{21}$$

Consequently, the rank of the covariance matrix is $\text{rank}(\Sigma_{\mathbf{A_i},\mathbf{B_i}}) = |\{f(s_{1,i}), \ldots, f(s_{d_s,i})\}| + |\emptyset| = d_s$. If the calculated rank is greater than d_s, it implies the presence of latent variables accounting for the unexplained variance since the observed variables alone would not result in such rank deficiency.

In scenarios with latent factors, the maximum rank deficiency observed across covariance submatrices, representing the discrepancy between the expected and actual ranks, establishes a lower bound for the number of latent variables. Considering the minimal t-separation of $\mathbf{A_i}$ and $\mathbf{B_i}$ occurs via

$$(\{f(s_{1,i}), \ldots, f(s_{d_s,i}), \kappa_1, \ldots, \kappa_m\}, \emptyset) \quad \text{or} \quad (\emptyset, \{f(s_{1,i}), \ldots, f(s_{d_s,i}), \kappa_1, \ldots, \kappa_m\}). \tag{22}$$

Then the rank of the covariance matrix is $\text{rank}(\Sigma_{\mathbf{A_i},\mathbf{B_i}}) = |\{f(s_{1,i}), \ldots, f(s_{d_s,i}), \kappa_1, \ldots, \kappa_m\}| + |\emptyset| = m + d_s$. By iteratively computing the rank of $\text{rank}(\Sigma_{\mathbf{A_i},\mathbf{B_i}})$, a consistent value corroborates the existence of latent factors influencing all observed states. Furthermore, the count of latent factors can be deduced by $m = \text{rank}(\Sigma_{\mathbf{A_i},\mathbf{B_i}}) - d_s$.

In conclusion, under the conditions of the theorem, if the observed rank deficiency in the covariance matrix of observed variables cannot be explained by the observed variables alone, it implies the existence of latent variables. Furthermore, the number of such latent variables can be inferred from the extent of the rank deficiency. □

C.2.3 Proof of Parameter Identifiability

Proof. For each individual k, consider the proposed model at any time t and $t+1$:

$$f(s_{i,t}^k) = \sum_{j \in \mathcal{P}_i} \alpha_{ij} f(s_{j,t-1}^k) + \sum_{j \in \mathcal{L}_i} \beta_{ij} a_{j,t-1}^k + \sum_{j=1}^{m} \lambda_j \kappa_j + \epsilon_{i,t}^k,$$

$$f(s_{i,t+1}^k) = \sum_{j \in \mathcal{P}_i} \alpha_{ij} f(s_{j,t}^k) + \sum_{j \in \mathcal{L}_i} \beta_{ij} a_{j,t}^k + \sum_{j=1}^{m} \lambda_j \kappa_j + \epsilon_{i,t+1}^k.$$

Subtracting these two equations, we obtain:

$$f(s_{i,t+1}^k) - f(s_{i,t}^k) = \sum_{j \in \mathcal{P}_i} \alpha_{ij} (f(s_{j,t}^k) - f(s_{j,t-1}^k)) + \sum_{j \in \mathcal{L}_i} \beta_{ij} (a_{j,t}^k - a_{j,t-1}^k) + (\epsilon_{i,t+1}^k - \epsilon_{i,t}^k).$$

Define $x_{i,t+1}^k = f(s_{i,t+1}^k) - f(s_{i,t}^k)$, $y_{i,t+1}^k = a_{i,t+1}^k - a_{i,t}^k$, and $\eta_{i,t+1}^k = \epsilon_{i,t+1}^k - \epsilon_{i,t}^k$. Substituting these, the model transforms to:

$$x_{i,t+1}^k = \sum_{j \in \mathcal{P}_i} \alpha_{ij} x_{j,t}^k + \sum_{j \in \mathcal{L}_i} \beta_{ij} y_{j,t}^k + \eta_{i,t+1}^k.$$

In that case, the identifiability of α and β can be directly derived by Lemma C.2. We further assume that the m latent factors follow the Normal distribution. Drawing on methodologies used in factor analysis C.3, then λ is orthogonal-wise identifiable. □

C.3 Examples For Theorem 4.3

In this section we present four illustrative examples to describe cases where identifiability is achieved. For the sake of simplicity, we define X as $X = f(s)$ and omit the terms a and ϵ from our illustration for simplicity, under the assumption that they are random and independent variables.

Example 1 In Example 1, as shown in Figure 6, there is only one latent factor and one-dimensional states. The bottleneck set is (X_3, κ), and the pairs of minimal rank sets are $(\mathbf{A}, \mathbf{B}) = ((X_1, X_2, X_3), (X_3, X_4, X_5))$. According to the causal graph, the minimal way to t-separate $\mathbf{A}$ from $\mathbf{B}$ is either $(\{\kappa, X_3\}, \emptyset)$ or $(\emptyset, \{\kappa, X_3\})$. Consequently, the rank of the covariance matrix is $\text{rank}(\Sigma_{\mathbf{A},\mathbf{B}}) = |\{\kappa, X_3\}| + |\emptyset| = 2$. According to Theorem 2, the fact that $\text{rank}(\Sigma_{\mathbf{A},\mathbf{B}}) = 2 > 1$ indicates the presence of the latent factor. Consequently, the number of latent variables can be deduced as $m = \text{rank}(\Sigma_{\mathbf{A},\mathbf{B}}) - 1 = 2 - 1 = 1$.

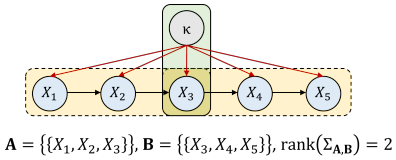


Figure 6: Single latent variable and one-dimension state.

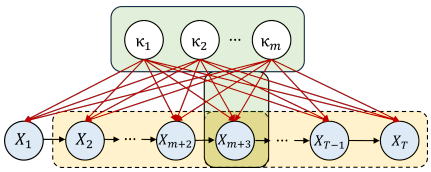


Figure 7: Multiple latent variables and one-dimension state.

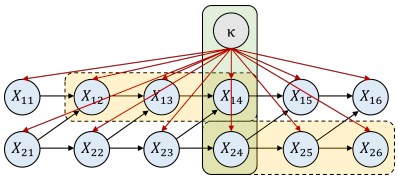


Figure 8: Single latent variable and multi-dimension states.

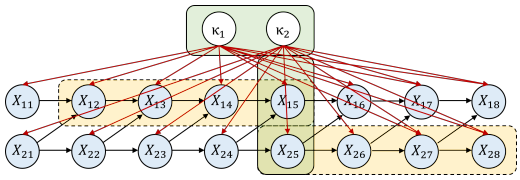


Figure 9: Multiple latent variables and multi-dimension states.

Figure 10: Examples that illustrate different identifiable cases.

Example 2 As shown in Figure 7, there are m latent factors and one-dimensional states. The bottleneck sets are $((X_{m+2}, \kappa_1, \ldots, \kappa_m), \ldots, (X_{T-m-1}, \kappa_1, \ldots, \kappa_m))$. Suppose $T = 2m + 4$, then the pairs of minimal rank sets are $(\mathbf{A_1}, \mathbf{B_1}) = ((X_1, \ldots, X_{m+2}), (X_{m+2}, \ldots, X_{2m+3}))$ and $(\mathbf{A_2}, \mathbf{B_2}) = ((X_2, \ldots, X_{m+3}), (X_{m+3}, \ldots, X_{2m+4}))$. According to the graph, the minimal configuration to t-separate $\mathbf{A_1}$ from $\mathbf{B_1}$ is either $(\{\kappa_1, \ldots, \kappa_m, X_{m+2}\}, \emptyset)$ or $(\emptyset, \{\kappa_1, \ldots, \kappa_m, X_{m+2}\})$ with $\mathrm{rank}(\Sigma_{\mathbf{A_1},\mathbf{B_1}}) = |\{\kappa_1, \ldots, \kappa_m, X_{m+2}\}| + |\emptyset| = m + 1$. The minimal configuration to t-separate $\mathbf{A_2}$ from $\mathbf{B_2}$ is either $(\{\kappa_1, \ldots, \kappa_m, X_{m+3}\}, \emptyset)$ or $(\emptyset, \{\kappa_1, \ldots, \kappa_m, X_{m+3}\})$, resulting in $\mathrm{rank}(\Sigma_{\mathbf{A_2},\mathbf{B_2}}) = |\{\kappa_1, \ldots, \kappa_m, X_{m+3}\}| + |\emptyset| = m + 1$. According to Theorem 2, the fact that $\mathrm{rank}(\Sigma_{\mathbf{A_1},\mathbf{B_1}}) = \mathrm{rank}(\Sigma_{\mathbf{A_2},\mathbf{B_2}}) = m + 1 > 1$ indicates the presence of the latent factor. Consequently, the number of latent variables can be deduced as $m = \mathrm{rank}(\Sigma_{\mathbf{A_i},\mathbf{B_i}}) - 1 = m + 1 - 1 = m$.

Example 3 As shown in Figure 8, there is one latent factor and two-dimensional states. The bottleneck sets are $((X_{13}, X_{23}, \kappa), (X_{14}, X_{24}, \kappa))$, and one possible pairs of minimal rank sets are $(\mathbf{A_1}, \mathbf{B_1}) = ((X_{11}, X_{12}, X_{13}, X_{23}), (X_{13}, X_{23}, X_{24}, X_{25}))$ and $(\mathbf{A_2}, \mathbf{B_2}) = ((X_{12}, X_{13}, X_{14}, X_{24}), (X_{14}, X_{24}, X_{25}, X_{26}))$. According to the causal graph, the minimal configuration to t-separate $\mathbf{A_1}$ from $\mathbf{B_1}$ is either $(\{\kappa, X_{13}, X_{23}\}, \emptyset)$ or $(\emptyset, \{\kappa, X_{13}, X_{23}\})$. Consequently, the rank of the covariance matrix is $\mathrm{rank}(\Sigma_{\mathbf{A_1},\mathbf{B_1}}) = |\{\kappa, X_{13}, X_{23}\}| + |\emptyset| = 3$. Similarly, the minimal configuration to t-separate $\mathbf{A_2}$ from $\mathbf{B_2}$ is either $(\{\kappa, X_{14}, X_{24}\}, \emptyset)$ or $(\emptyset, \{\kappa, X_{14}, X_{24}\})$, resulting in $\mathrm{rank}(\Sigma_{\mathbf{A_2},\mathbf{B_2}}) = |\{\kappa, X_{14}, X_{24}\}| + |\emptyset| = 3$. According to Theorem 2, the fact that $\mathrm{rank}(\Sigma_{\mathbf{A_1},\mathbf{B_1}}) = \mathrm{rank}(\Sigma_{\mathbf{A_2},\mathbf{B_2}}) = 3 > 2$ indicates the presence of the latent factor. Consequently, the number of latent variables can be deduced as $m = \mathrm{rank}(\Sigma_{\mathbf{A_i},\mathbf{B_i}}) - 2 = 3 - 2 = 1$.

Example 4 As shown in Figure 9, there are two latent factors and two-dimensional states. The bottleneck sets are $((X_{14}, X_{24}, \kappa_1, \kappa_2), (X_{15}, X_{25}, \kappa_1, \kappa_2)$, and one possible pairs of minimal rank sets are $(\mathbf{A_1}, \mathbf{B_1}) = ((X_{11}, X_{12}, X_{13}, X_{14}, X_{24}), (X_{14}, X_{24}, X_{25}, X_{26}, X_{27}))$ and $(\mathbf{A_2}, \mathbf{B_2}) = ((X_{12}, X_{13}, X_{14}, X_{15}, X_{25}), (X_{15}, X_{25}, X_{26}, X_{27}, X_{28}))$. According to the graph, the minimal configuration to t-separate $\mathbf{A_1}$ from $\mathbf{B_1}$ is either $(\{\kappa_1, \kappa_2, X_{14}, X_{24}\}, \emptyset)$ or $(\emptyset, \{\kappa_1, \kappa_2, X_{14}, X_{24}\})$. Consequently, the rank of the covariance matrix is $\mathrm{rank}(\Sigma_{\mathbf{A_1},\mathbf{B_1}}) = |\{\kappa_1, \kappa_2, X_{14}, X_{24}\}| + |\emptyset| = 4$. Similarly, the minimal configuration to t-separate $\mathbf{A_2}$ from $\mathbf{B_2}$ is either $(\{\kappa_1, \kappa_2, X_{15}, X_{25}\}, \emptyset)$ or $(\emptyset, \{\kappa_1, \kappa_2, X_{15}, X_{25}\})$ with $\mathrm{rank}(\Sigma_{\mathbf{A_2},\mathbf{B_2}}) = |\{\kappa_1, \kappa_2, X_{15}, X_{25}\}| + |\emptyset| = 4$. According to Theorem 2, the fact that $\mathrm{rank}(\Sigma_{\mathbf{A_1},\mathbf{B_1}}) = \mathrm{rank}(\Sigma_{\mathbf{A_2},\mathbf{B_2}}) = 4 > 2$ indicates the presence of the latent factor. Consequently, the number of latent variables can be deduced as $m = \mathrm{rank}(\Sigma_{\mathbf{A_i},\mathbf{B_i}}) - 2 = 4 - 2 = 2$.

C.4 Related Definitions of Theorem 4.3

C.4.1 Covariance Matrix of Random Vector

In this discussion, we introduce the concept of the covariance matrix within the framework of latent variable models. By examining the properties of the covariance matrix, such as its rank, we are able to identify signs of latent variables—rank deficiencies, which serve as a measure of the cardinality of the minimal set of latent variables required to explain the observed dependencies. Such rank deficiencies indicate the presence of latent variables that extend beyond the observable scope.

Consider a directed acyclic graph (DAG), denoted as $\mathcal{G}$, whose vertices $V(\mathcal{G})$ form the set $[m] := \{1, 2, \ldots, m\}$. Each node i in $\mathcal{G}$ is associated with a random variable X_i and an independent error term $\epsilon_i \sim \mathcal{N}(0, \phi_i)$ with $\phi_i > 0$. The DAG structure imposes a recursive relationship among the variables, where the value of X_j can be expressed as a linear combination of the variables X_i of its parent vertices $\text{pa}(j)$, alongside the error term ϵ_j and regression coefficients λ_{ij} that correspond to the edges $i \to j$ in $\mathcal{G}$:

$$X_j = \sum_{i \in \text{pa}(j)} \lambda_{ij} X_i + \epsilon_j. \tag{23}$$

where $\text{pa}(j)$ denotes the set of parent nodes of vertex j, where a parent node i is one that has an edge leading to j in $\mathcal{G}$. From this recursive sequence of regressions, one can solve for the covariance matrix Σ of the jointly normal random vector $\mathbf{X}$, which is defined as follows.

Definition C.3 (Covariance Matrix of Random Vector [70])**.** *The covariance matrix of the random vector is given by the matrix factorization*

$$\Sigma = \Lambda^{-\top} \Phi \Lambda^{-1}. \tag{24}$$

where matrix Φ is defined as a diagonal matrix with the variances of the error terms as its diagonal elements: $\Phi = \text{diag}(\phi_1, \ldots, \phi_m)$. The matrix M is an $m \times m$ upper triangular matrix where $M_{ij} = \lambda_{ij}$ if $i \to j$ is an edge in $\mathcal{G}$, and $M_{ij} = 0$ otherwise. Thus the matrix Λ is defined as $\Lambda = I - M$, where I is the $m \times m$ identity matrix.

Specifically, given two subsets $\mathbf{A}, \mathbf{B} \subset [m]$, $\Sigma_{\mathbf{A},\mathbf{B}} = (\sigma_{ab})_{a \in \mathbf{A}, b \in \mathbf{B}}$ is defined as the submatrix of covariance with row index set $\mathbf{A}$ and column index set $\mathbf{B}$.

C.4.2 Trek and Trek Separation

The concepts of Trek and Trek Separation precede a crucial need to address the presence of latent variables and intricate dependency structures that are not directly observable. The Trek represents a particular path that interconnects variables within a graph, even if they are not directly linked, while Trek Separation delineates a criterion to ascertain whether two sets of variables are independent, conditional on a set of other variables. Below, we give the formation definitions of these two concepts.

Definition C.4 (Trek [70])**.** *A trek in $\mathcal{G}$ from i to j is an ordered pair of directed paths (P_1, P_2) where P_1 has sink i, P_2 has sink j, and both P_1 and P_2 have the same source k. The common source k is called the top of the trek, denoted top(P_1, P_2). Note that one or both of P_1 and P_2 may consist of a single vertex, that is, a path with no edges. A trek (P_1, P_2) is simple if the only common vertex among P_1 and P_2 is the common source top(P_1, P_2). We let $\mathcal{T}(i, j)$ and $\mathcal{S}(i, j)$ denote the sets of all treks and all simple treks from i to j, respectively.*

Definition C.5 (Trek Separation [70])**.** *Let $\mathbf{A}, \mathbf{B}, \mathbf{C_A}$ and $\mathbf{C_B}$ be four subsets of $V(\mathcal{G})$ which need not be disjoint. We say that the pair $(\mathbf{C_A}, \mathbf{C_B})$ trek separates (or t-separates) $\mathbf{A}$ from $\mathbf{B}$ if for every trek (P_1, P_2) from a vertex in $\mathbf{A}$ to a vertex in $\mathbf{B}$, either P_1 contains a vertex in $\mathbf{C_A}$ or P_2 contains a vertex in $\mathbf{C_B}$.*

C.4.3 Conditional Independence Skeleton

The Conditional Independence (CI) skeleton in graphical models refers to a structure that represents the conditional independence among observed variables. The CI skeleton can be used to infer the existence of latent variables. If the observed data suggests dependencies not represented in the CI skeleton, it may indicate hidden factors at play. The formal definition is given as follows.

Definition C.6 (Conditional Independence Skeleton [12])**.** *A CI skeleton of $\mathbf{X}$ is an undirected graph where the edge between X_1 and X_2 exists if and only if there does not exist a set of observed variables $\mathbf{C}$ such that $X_1, X_2 \notin \mathbf{C}$ and $X_1 \perp\!\!\!\perp X_2 | \mathbf{C}$.*

D Background

Reinforcement Learning In RL, an agent learns to make decisions by interacting with the environment. The agent receives rewards for taking actions in the environment and uses this feedback to learn optimal behavior. It is often modeled as a Markov Decision Process (MDP) represented by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathbb{P}, R, \gamma \rangle$, where $\mathcal{S}$ denotes a finite set of states representing different situations an agent might encounter, $\mathcal{A}$ a finite set of actions representing different decisions an agent can make, $\mathbb{P}$ a state transition function defining the probability of transitioning to a new state s' given a current state s and action a, denoted as $\mathbb{P}(s'|s,a)$, R a reward function assigning a scalar value to each state-action pair (s,a), representing the immediate reward received after performing action a in state s. $\gamma \in [0,1]$ is the discount factor, representing the agent's consideration for future rewards. The agent's goal is to learn an optimal policy π^*, which defines the optimal set of actions in different states to maximize the expected cumulative discounted reward over the long run. Developing this optimal policy involves estimating value functions such as the action-value function, defined as $Q^\pi(s,a) = \mathbb{E}_\pi\big[\sum_{t=0}^{\infty} \gamma^t R_t | S_0 = s, A_0 = a\big]$, which represents the expected reward of taking action a in state s following policy π. The pursuit of optimal policy π^* involves maximizing the value functions over all possible state-action pairs: $\pi^* = \arg\max_\pi Q^\pi(s,a)$.

Variational Autoencoder Variational Autoencoders (VAEs) [44] are a class of generative models in deep learning, adept at unsupervised learning of complex data distributions. Rooted in the framework of Bayesian inference, VAEs are designed to approximate probability density functions of input data. The architecture of a VAE consists of two primary components: an encoder $q_\phi(z|x)$ and a decoder $p_\theta(x|z)$. The encoder maps input data x to a latent space, represented by a probability distribution, typically Gaussian, with parameters μ and σ signifying the mean and standard deviation, respectively. The decoder reconstructs the input data from a sampled latent representation z.

The distinct feature of VAEs lies in their probabilistic approach. The encoder outputs parameters of a latent distribution, from which a sample z is drawn:

$$z \sim q_\phi(z|x) = \mathcal{N}(z; \mu, \sigma^2 I) \tag{25}$$

The decoder then attempts to reconstruct the input from this latent sample. VAEs optimize the Evidence Lower Bound (ELBO) objective, which balances two aspects: the reconstruction quality and the regularization of the latent space. The traditional ELBO is given by:

$$\text{ELBO} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \text{KL}[q_\phi(z|x)||p(z)] \tag{26}$$

Here, the first term measures the reconstruction quality, while the second term, the Kullback-Leibler (KL) divergence, imposes a regularization by encouraging the latent distribution $q_\phi(z|x)$ to be close to a prior $p(z)$, typically assumed to be a standard normal distribution $\mathcal{N}(0, I)$. VAEs, through this optimization, are capable of generating new data points that are similar to the input data, making them highly valuable in applications like image generation, denoising, and anomaly detection within the domain of unsupervised learning.

E Detailed Related Work

Individualized Machine-Learning Applications In the modern era, the power of machine learning has been harnessed to create highly individualized solutions across a myriad of domains. In the realm of health and wellness, machine learning aids in tailoring interventions for increasing physical activity [89, 57], promoting weight loss [18, 17], improving adherence for diabetes [89]. For the elderly, personalized algorithms assist in both technology adaptation and specialized care for conditions [32]. The financial sector benefits from machine learning's prowess in optimizing technical indicators, making stock market predictions more precise and individualized [56]. In the educational landscape, Information and Communication Technology (ICT) leverages machine learning to offer personalized education systems such as adaptive e-learning interfaces [16] and individualized tutorial planning [40]. Furthermore, the transportation sector sees advancements with car-following control strategies tailored for individual drivers [69]. Multimedia platforms, such as YouTube and TikTok, are enhancing user experiences by offering video content recommendations fine-tuned to individual preferences using reinforcement learning [6, 33]. These examples merely scratch the surface, emphasizing the vast and diverse applications of individualized machine learning in today's world.

Reinforcement Learning for Latent State-Transition Processes RL has witnessed significant advancements in recent years, particularly with the integration of latent variable models to capture the underlying dynamics of environments. A primary focus in this domain is learning low-dimensional, latent Markovian representations from observed data [54, 48, 42, 25, 82, 95, 49, 62, 19, 23, 20, 91]. Common strategies for state representation learning include reconstructing the observation, learning a forward model, or learning an inverse model. Additionally, prior knowledge, such as temporal continuity [83], can be leveraged to constrain the state space. Numerous studies have proposed methods to estimate the underlying state-transition process from high-dimensional input sequences [82, 13, 25, 27, 94, 19, 41, 28]. Using the learned world model, agents can engage in model-based RL or planning. Furthermore, these methods encode structural constraints, ensuring the sufficiency and minimality of the estimated state representations from both generative and selection processes. Recently, several studies [60, 55, 78, 79, 4, 65] have aimed to estimate the state-transition process in the presence of latent confounders. A handful of work [60, 65] can be viewed as addressing similar settings involving individual-specific factors. However, to the best of our knowledge, we have yet to identify a systemic approach that offers a clear identifiability result for the state-transition process when individual-specific factors are present.

Comparisons with Related Works Contextual MDPs [29] consider the general contextual influence on transition probabilities and rewards. However, the context variables are assumed to be partially observable and do not guarantee the identifiability of the context variables. In our case, when the latent factors are finite, our method guarantees group-wise identifiability even when the transition processes are nonparametric. In the cases of infinite latent factors, identification could be achieved under proper assumptions.

Multi-task RL [75] involves learning policies for a variety of tasks simultaneously. The goal of the agent is to perform well on all these tasks, which may have similar or different objectives. often involves sharing information between tasks to improve learning efficiency and policy performance. Instead of focusing on policy optimization for all tasks, our work identifies latent individual-specific factors that implicitly influence the decision-making process. These factors indicate the unique properties of each individual, providing explanatory clues for policy adaptation.

Meta-RL [3] trains a learning model on a variety of tasks so that it can efficiently apply what it has learned to new tasks. Unlike our method, it does not assume a time-invariant latent factor, has no guarantee of identifiability, and does not provide a clear clue of adaptation. While iMDP captures how an individual's belonging to a certain group affects their interactions within an environment, allowing for individualized policy adaptation. Moreover, iMDP provides a guarantee of identifiability and develops a corresponding estimation framework that potentially offers better interpretability.

Factored MDP [92] and Factored Non-stationary MDP [24] assume there are no unobserved confounders in the state transitions. Block MDP [10], POMDP [38], and Latent MDP [86] consider latent states/spaces, but such latent factors do not influence each state in the transition process. Specifically, existing works [92, 10, 24] usually focus on latent variables that are time-varying. When they are time-varying, they can benefit from many recent advances in nonlinear ICA to achieve strong identifiability results [38, 86]. However, the identifiability of time-invariant latent confounders, though not well-studied, has numerous applications. The aforementioned settings differ significantly from our work since we consider a latent group factor that influences each state in the state transition process, and the proposed individualized transition process is motivated by numerous applications. We provide theoretical results that when individual-specific factors are finite, our method ensures the identifiability of the entire latent state-transition process, even in the case of nonparametric transitions. This establishes novel theoretical insights for learning state-transition processes with latent factors.

F Experiment Details

F.1 Evaluation Metrics

Pearson Correlation Coefficient Pearson Correlation Coefficient (PCC) [9] is a statistical measure that quantifies the degree of linear relationship between two variables. It provides a value between -1 and 1, where 1 implies a perfect positive linear relationship, -1 implies a perfect negative linear relationship, and 0 implies no linear relationship between the variables. The equation for calculating

the Pearson Correlation Coefficient r between two variables X and Y is as follows:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \tag{27}$$

where n is the number of paired samples, $\sum xy$ is the sum of the product of paired scores, $\sum x$ and $\sum y$ are the sums of the x scores and y scores respectively, $\sum x^2$ and $\sum y^2$ are the sums of the squared x scores and y scores respectively.

Canonical Correlation Analysis Canonical Correlation Analysis (CCA) [34] is designed to identify bases for two sets of variables in order to maximize the mutual correlations between the projections onto these bases. In our work, CCA is used as an evaluation metric to validate that the recovered latent variable is meaningfully related to the ground truth latent variable, thus proving the relevance of the estimated representations. Let X and Y be the two sets of observed variables. This algorithm starts by centering the columns of X and Y so that they have zero mean. Then the covariance matrices $C_{XX} = X^\top X, C_{YY} = Y^\top Y$, and $C_{XY} = X^\top Y$ are calculated. After that, the canonical correlations are obtained by solving the following generalized eigenvalue problem: $C_{XX}^{-1} C_{XY} C_{YY}^{-1} C_{YX} \nu = \lambda \nu$. The square roots of the eigenvalues λ indicate the canonical correlations between the linear combinations of X and Y. The corresponding eigenvectors ν and $u = C_{XY}\nu$ are the canonical weights used to construct the canonical variables. Finally, the canonical variables of X and Y are $U = X\nu$ and $V = Yu$, respectively, representing the linear combinations of the original variables that are maximally correlated. The correlation of the primary pair of canonical variables is the highest, followed by the secondary pair, and so on. When employing CCA as an evaluation metric, a higher canonical correlation indicates a stronger and more relevant relationship between the recovered latent variable and the ground truth latent variable.

To extend the capability of CCA for analyzing nonlinear relationships, Kernel Canonical Correlation Analysis (KCCA) [84] is employed in the experiment which uses kernel functions to map the original variables into a higher-dimensional feature space. This allows for the capture of more complex, nonlinear correlations between the variables, thus potentially increasing the robustness and relevance of the relationships discovered in scenarios where linear methods fall short.

F.2 Dataset Descriptions

Synthetic Data Generation Processes In this paper, we created three synthetic datasets: Case 1 corresponds to a finite latent factor that satisfies our assumptions, and Case 2 and Case 3 allow for multiple finite and infinite latent variables. The dimensions of states and actions are set to 3 and 2, respectively. The actions taken are generated randomly, following a uniform distribution $\mathrm{Uniform}(0, 1)$. The noise term follows a mean-zero Gaussian distribution. The mixing function f corresponds to the post-nonlinear model [93], where f_1 represents the nonlinear effect, and f_2 denotes the invertible post-nonlinear distortion on $\mathbf{s}_t$, embodied by a randomly initialized three-layer MLP with Tanh activation function. The data generation process follows:

$$\mathbf{s}_t = f_2(f_1(\mathbf{s}_{t-1}, \mathbf{a}_{t-1}, \kappa), \epsilon_t)). \tag{28}$$

PersuasionForGood The PersuasionForGood dataset reveals the mechanics of persuasion in the context of charitable giving. It contains 1017 dialogues from 1285 participants in which one participant, called the persuader (ER), tries to convince the other participant, called the persuadee (EE), to donate to a charity. An example dialog is shown in Figure 11. All participants underwent personality assessments, which included detailed participant-level information such as demographics, Big Five personality traits, moral foundations, and so on, allowing for a multifaceted analysis of persuasion strategies and allowing us to use the labeled 32-dimensional personalities of each persuader as the ground-truth latent factor in our experiments.

We use this dataset to evaluate the performance of our estimation framework. We compare the estimated factors to the documented personality traits of persuaders. By examining the interactions between participants with different backgrounds and personalities, we aim to identify underlying patterns that could create effective persuasive agents. Specifically, we use BERT embeddings to generate a 768-dimensional feature vector for each dialog utterance. This process starts with tokenization, segmenting words into smaller units. BERT then processes these tokens to produce contextual embeddings.

Speaker	Utterance	Extrovert	Agreeable	Conscientious	Neurotic	Open
ER	Hello. How are you?	3.6	4.4	4.4	3	4
EE	I'm good, how are you doing?	4	5	4.2	3.6	4.8
ER	Very well. I'm just up organizing info for my charity. Are you involved with charities?	3.6	4.4	4.4	3	4
EE	Yes! I work with children who have terminal illnesses. What charity are you involved in?	4	5	4.2	3.6	4.8
ER	That's great! I help with Save The Children.	3.6	4.4	4.4	3	4
EE	Amazing! Working with kids is the best. What do you do for Save the Children?	4	5	4.2	3.6	4.8
ER	I help raise donations and volunteer time.	3.6	4.4	4.4	3	4
EE	That's so important. How do you raise donations?	4	5	4.2	3.6	4.8
ER	By directly asking for aid. Do you currently donate to your charity?	3.6	4.4	4.4	3	4
EE	Yes I do, but I'm happy to donate to yours as well!	4	5	4.2	3.6	4.8
ER	Wonderful! Would you be will to donate $1.00 of your task money to help Save the Children? Save The Children is an international non-governmental organization that promotes children's rights, provides relief and helps support children in developing countries.	3.6	4.4	4.4	3	4
EE	Yes, I would be happy to!	4	5	4.2	3.6	4.8
ER	Would $2.00 be too much to ask?	3.6	4.4	4.4	3	4
EE	No, I can do it.	4	5	4.2	3.6	4.8
ER	Thank you. Can we make it $1.50? These children really need the assistance.	3.6	4.4	4.4	3	4
EE	$1.50 sounds good then.	4	5	4.2	3.6	4.8
ER	Why not $1.75 then? :-)	3.6	4.4	4.4	3	4
EE	I can do $2.00! Happy to help.	4	5	4.2	3.6	4.8
ER	Thank you so much! Do you have any more questions for me?	3.6	4.4	4.4	3	4
EE	Nope. Thank you!	4	5	4.2	3.6	4.8

Figure 11: A sample persuasive dialog between persuader (ER) and persuadee (EE) from the PersuasionForGood corpus, along with the Big Five personality test scores, including Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism.

Pendulum The pendulum environment, provided by OpenAI Gym, is a classic control task used for the evaluation RL models. This environment presents a continuous control task where the agent must learn to control a frictionless pendulum with the goal of swinging it to the highest point and keeping it in the inverted position. The pendulum starts at a random position, and the goal is to bring it to a standstill at the inverted position with the least amount of effort. The system is characterized by a continuous action space, representing the torque applied to the pendulum's fulcrum. For a pendulum of length l and mass m, subject to gravity g and a control input u, the equations of motion can be described by the following second-order nonlinear ordinary differential equations $\dot{\theta} = \omega$, $\dot{\omega} = -\frac{g}{l}\sin(\theta) + \frac{u}{ml^2}$, where θ is the angle of the pendulum from the vertical upright position, and ω is the angular velocity of the pendulum. The state of the pendulum at any time t can be represented as $\vec{s}_t = [\cos\theta_t, \sin\theta_t, \omega_t]$, action represents the torque applied to the free end of the pendulum in the range $a_t \in [-2, 2]$, and the reward function is defined as: $r_t = -(\theta_t^2 + 0.1 * \omega_t^2 + 0.001 * a_t^2)$.

The goal of RL algorithms is to determine an optimal control policy π^* that minimizes the effort to swing and balance the pendulum upright, typically by minimizing a cost function defined over states and actions. Each episode provides a continuous stream of observations, actions, and rewards, allowing the development and evaluation of algorithms capable of learning effective control policies in continuous action spaces. In academic studies, the Pendulum environment serves as a benchmark to investigate the effectiveness of RL algorithms in handling continuous control tasks.

HeartPole HeartPole provides a straightforward scenario for assessing healthcare treatment, highlighting the complex interplay between productivity, health, and decision-making strategies. It simulates a professional's quest for increased productivity and examines the long-term health impacts of short-term choices, such as insufficient sleep, and intake of coffee and alcohol. The states include alertness, hypertension, intoxication, time since last sleep, total elapsed time, and total work done.

A productivity function and a heart attack risk function are defined over these variables, rewarding incremental productivity while imposing a significant penalty for heart attacks. Every thirty minutes, the agent evaluates the current state and chooses from a set of actions: work, drink coffee (which increases alertness and hypertension), drink alcohol (which decreases alertness while increasing hypertension and intoxication), or sleep (time-consuming but essential to reduce hypertension and intoxication to maintain alertness).

Half Cheetah Half Cheetah is an integral part of the Mujoco physics engine, designed to simulate the agility and mechanics of a cheetah through a 2D robotic model. This model consists of 9 body parts, including a torso, two front and two back thighs, shins and feet, connected by 8 joints to provide fluid motion reminiscent of a cheetah's natural gait. The primary goal for this robot is to achieve maximum forward speed while maintaining stability, mirroring the efficiency and speed of its biological counterpart.

The observational data in this environment includes both the position and velocity of each segment of the half cheetah, methodically ordered with all position data provided before velocity information. This systematic ordering allows for a detailed understanding of the dynamics of the robot at any given time. Actions within this simulation are defined by the torque applied to the joints, which directly affects its acceleration and motion patterns. The reward function for Half Cheetah is designed to encourage rapid forward motion and operational efficiency and consists of two main components: a forward motion reward proportional to the increase in the robot's horizontal displacement over time and a control cost penalty for unnecessary control effort and external force application. This reward structure is carefully designed to encourage the optimization of forward motion, with an emphasis on reducing control effort and mitigating forces that may interfere with the robot's streamlined motion.

F.3 Additional Experiment Results

Ablation Study: Variability in number of samples Here, we vary the number of samples to further verify this effectiveness. The data generation process is the same as Case 1, except that we change the number of samples to $\{100, 150, 200, 300, 500, 800, 1000\}$. The comparison results shown in Figure 14 indicate that our method can achieve consistently good recovery performance under different numbers of individuals. This further confirms that the identifiability of our framework is guaranteed by the mathematical relationship between the trajectory length and the number of groups, which is constrained by the sample sufficiency assumption under the conditions given in Theorem 4.1. Moreover, Figures 14(b) and 14(c) show the successful recovery of the latent group factor, validated by high-frequency similarity and a remarkable PCC value, confirming the ability of our method to skillfully recover latent variables in practical pendulum tasks.

Ablation Study: Variability in initial states We conduct additional experiments to verify how the variability in initial states across individuals affects performance. The initial state distributions are defined with two types: normal and uniform. For the normal distribution, the means are set to [0, 1, 1] and the standard deviations are set to [1, 2, 1], respectively. For the uniform distribution, the range for each dimension is defined with lower bounds [0, -1, 1] and upper bounds [1, 1, 1.5]. The experiment results in Figure 15 show that although the initial states have high variability, the estimated values of the latent factors corresponding to the 200 individuals are ultimately highly classified and can be divided into 4 groups.

Ablation Study: Consideration of transformer as encoder Our framework is flexible enough to integrate various encoders and decoders, depending on the application tasks. To demonstrate this flexibility, we incorporated Transformers into our framework and conducted a comparative analysis against the existing models. The result, shown in Figure 15(c), indicates that while both frameworks achieve identifiability, the Transformer-based encoder demonstrates faster convergence compared to our previous approach.

Added Experiment: Inventory Inventory management [71] is an important real-world problem that aims to keep inventories of goods at optimal levels to minimize inventory costs while maximizing revenue from demand fulfillment. We tested the performance of our algorithm on the inventory with state dimensions of 50, 100, and 200 and added additional baselines (8) Meta gradient RL, (9) Multitask RL, (10) Policy distillation, and (11) Non-policy adaptation to verify the model. The experimental results in Figure 12 show that our framework outperforms other algorithms in terms of initial reward and final reward.

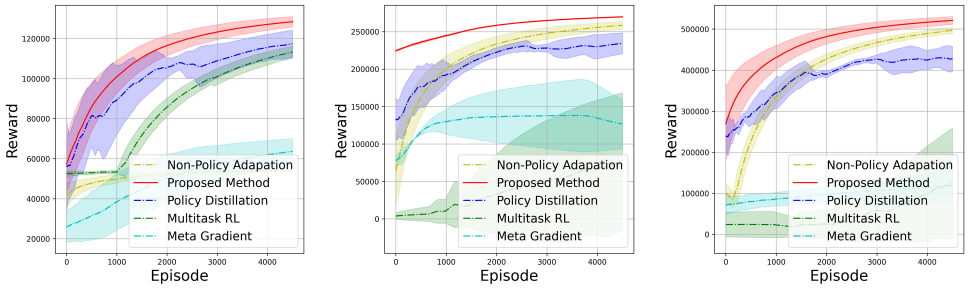
Added Experiment: AhnChemo AhnChemoEnv [61] is designed to simulate cancer treatment through chemotherapy, allowing realistic modeling of tumor growth and response to treatment. We create different groups with PK/PD variation. The experimental results in Figure 13 show that our framework outperforms other algorithms in terms of initial and final reward. Our method achieves

the highest initial and final rewards compared to the baselines. Specifically, it shows a significant jump-start compared to non-policy adaptation, validating the effectiveness of our adaptation approach.

The meta-gradient method optimizes the hyperparameters of the learning algorithm by calculating the gradient of the learning process, allowing rapid adaptation to new tasks as they change. However, due to the continuous adjustment of learning strategies during training, it converges more slowly and the adaptation effect is less significant compared to our algorithm. Multitask RL improves learning efficiency by sharing model strategies across different tasks. This requires first training policies on multiple tasks, which can be time-consuming (and even risky) during exploration. Moreover, identifying which new task corresponds to a previously trained task can be challenging. Our algorithm addresses this by estimating directly without requiring prior knowledge. Policy distillation transfers the knowledge of already trained teacher models to a student model, allowing the student to perform well across multiple tasks. However, this approach highly relies on the performance of the teacher models; insufficiently trained teacher models can negatively impact the final performance. Our algorithm does not depend on the source policy performance; subsequent policy optimization is based on the new environment, leading to better final performance.

F.4 Training Details

The estimation framework is trained using AdamW optimizer for a maximum of 200 epochs and early stops if the validation ELBO loss does not decrease for ten epochs. A learning rate of 0.001 and a mini-batch size of 32 are used. We used three random seeds in each experiment and reported the mean performance with standard deviation averaged across random seeds. We used a machine with the following CPU specifications: 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz with 16 logical processors. The machine has one GeForce RTX 3080 GPU with 32GB GPU memory.

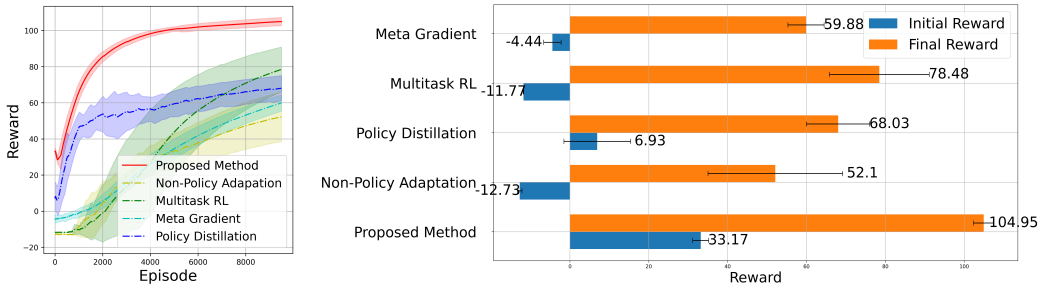


(a) Reward curve with $d_s = 50$. (b) Reward curve with $d_s = 100$. (c) Reward curve with $d_s = 200$.

Figure 12: **Results in Inventory**. We evaluated the performance of different methods under different state dimensions. Our algorithm scales well to high-dimensional cases and outperforms other baselines in terms of initial reward and final reward.

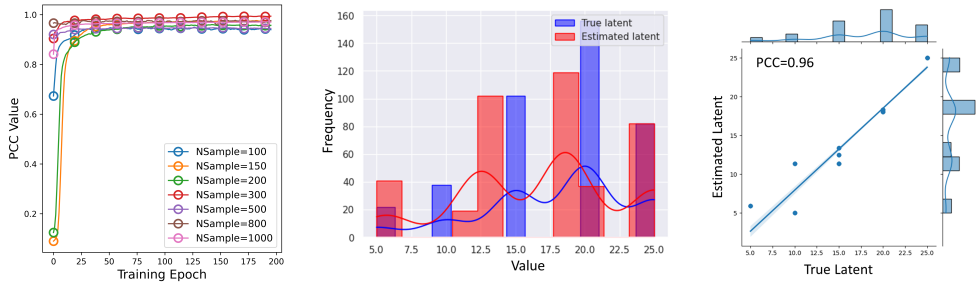
G Impact Statement

Our work has a significant impact on ethics, society, and future applications. We emphasize the importance of individualized policies in systems and advocate for a deeper understanding and respect for individual differences. Tailoring interventions to different individuals has the potential to improve user experience and outcomes in healthcare, education, and other areas. This approach avoids a one-size-fits-all policies. Our method can greatly improve individualized services, transforming the delivery of educational content, the management of healthcare, and the recommendation of products. This makes these services more effective and aligned with individual needs. However, the implementation of this method requires careful consideration of privacy and data security, as personalized systems require the collection and analysis of personal data. Maintaining user trust, preventing misuse, and ensuring ethical use of such data are of utmost importance.



(a) Accumulative reward curves. (b) Initial and final reward under different benchmarks.

Figure 13: **Results in AhnChemoEnv**. We evaluated the performance of our method against several baselines, including (1) meta gradient RL, (2) multitask RL, (3) policy distillation, and (4) non-policy adaptation. Our method outperforms these benchmarks and achieves superior performance in terms of initial reward and final reward.



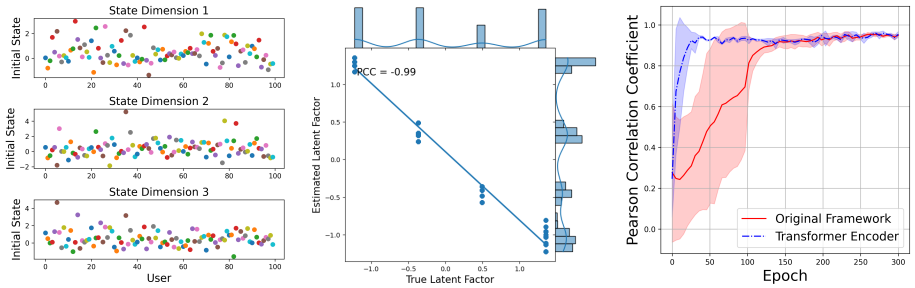
(a) (b) (c)

Figure 14: (a) PCC trajectory comparisons under different numbers of individuals. (b-c) Successful recovery of the latent group factor in the Pendulum.

H Estimation Framework Details

The proposed framework is customized based on the requirements of the identifiability theorems given in Section 4. We would like to emphasize that our proposed framework differs from the traditional VAE and model-based RL in three main aspects: (1) Our framework uses a quantization layer to discretize the continuous latent representations. This mapping of continuous latent representations to an embedding dictionary is well suited to the group determinacy requirement. (2) Our decoder reconstructs individualized state transition processes to simulate the data generation process, incorporating additional conditions as well as the estimated latent factor. (3) We further extract latent factors for each individual as additional information to facilitate individual policy learning. The detailed implementations of each component are summarized below.

Encoder For any individual m, the Conv1D layer transforms an input sequence $\mathbf{s}_t^m$, using learned kernel filters. These filters slide over the sequence to produce a feature map, denoting the response of the filter at each position. Mathematically, the transformation by a single filter in the Conv1D layer at time t is described as $\mathbf{o}_t = \sigma\left(W * \mathbf{s}_{t:H+t}^m + b\right)$, where $\mathbf{o}_t$ is the feature map, W the kernel to be learned during training, $*$ the convolution operation, $\mathbf{s}_{t:H+t}^m$ the input sub-sequence from time t to $t+H$, where H is the size of the kernel. σ is the activation function, and b is the bias term to be learned during training. The layer may contain multiple such filters, each learning different features of the input sequence. The resulting feature maps serve as a transformed representation z_m, which embeds the information about the latent group factor κ.



(a) Initial states with high variability.

(b) Estimated latent factor.

(c) PCC trajectories comparison.

Figure 15: (a-b) **Evaluation on variability in initial states**. The estimated values of κ are highly clustered into four classes. (c) **Incorporating Transformers**. The transformer encoder achieves faster convergence compared to the original framework.

As for the LSTM, let the hidden states and cell states of the LSTM at time t denote as h_t and c_t, respectively. Then, the LSTM updates are given by:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [\mathbf{s}_t^m, h_{t-1}] + b_f), \\
i_t &= \sigma(W_i \cdot [\mathbf{s}_t^m, h_{t-1}] + b_i), \\
\tilde{c}_t &= \tanh(W_c \cdot [\mathbf{s}_t^m, h_{t-1}] + b_c), \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\
o_t &= \sigma(W_o \cdot [\mathbf{s}_t^m, h_{t-1}] + b_o), \\
h_t &= o_t \odot \tanh(c_t),
\end{aligned}
$$

where σ is the sigmoid activation function, $\odot$ element-wise multiplication. W_f, W_i, W_c, W_o and b_f, b_i, b_c, b_o are the weight matrices and bias terms to be learned during training. $f_t, i_t, \tilde{c}_t, c_t, o_t$ and h_t are the forget gate, input gate, candidate cell state, cell state, output gate, and hidden state at time t, respectively. The final hidden state of LSTM h_T, after the sequential processing of the entire trajectory, serves as the representative z_m that embeds the information about the latent group factor κ.

Quantization Layer Let the output of the encoder be a continuous latent representation denoted as $z_m \in \mathbb{R}$, and define an embedding dictionary E consisting of G vectors, where each vector represents a unique discrete category: $E = \{e_1, e_2, \ldots, e_G\}$, where $e_i \in \mathbb{R}$. The quantized vector $\hat{\kappa}_m$ is obtained by mapping z_m to the nearest dictionary vector. The mapping can be expressed mathematically as $\hat{\kappa}_m = \arg\min_{e_i \in E} \|z_m - e_i\|_2$. Subsequently, the quantized output is the vector from the dictionary that is closest to the encoder output. Thus, the continuous representation z_m is effectively mapped to a discrete $\hat{\kappa}_m$ by finding the nearest neighbor in the dictionary, aligning the representation learning with the discrete nature of the latent variable.

Decoder Suppose $\mathbf{s}_{t-1}^m$ and $\mathbf{a}_{t-1}^m$ as the true previous state and action, respectively. Let $\hat{\kappa}_m$ be the approximated latent group factor for the m-th individual. The inputs to the conditional decoder are a combination of the aforementioned variables: $\text{Input}_t = (\mathbf{s}_{t-1}, \mathbf{a}_{t-1}, \hat{\kappa}_m)$. The output of the decoder is the reconstructed next state, $\hat{\mathbf{s}}_t$, which is a function of the decoder input: $\hat{\mathbf{s}}_t = \text{De}(\text{Input}_t)$. The reconstruction likelihood measures how closely the reconstructed state matches the true subsequent state, which is defined as $\mathcal{L}_{\text{Recon}} = p_{\text{Recon}}(\mathbf{s}_t^m | \mathbf{s}_{t-1}^m, \mathbf{a}_{t-1}^m, \hat{\kappa}_m)$. The objective in this process is to optimize the decoder parameters to maximize the reconstruction likelihood $\max \mathcal{L}_{\text{Recon}}$ so that the reconstructed state $\hat{\mathbf{s}}_t$ is as close as possible to the true next state $\mathbf{s}_t$.

I Algorithm

The pseudocode for the proposed algorithm is presented in Algorithm 1 and Algorithm 2.

Algorithm 1 Algorithm of Individualized Policy based on Latent Factor Analysis.

Input: $\{f_{\text{Env}}^m\}_{m=1}^M$: individualized environments; Encoder: encoder; Quantization: embedding dictionary; Decoder: decoder; π: policy network
Output: $\{\hat{\kappa}_g\}_{g=1}^G$: estimated group factor; $\{\pi_m^*\}_{m=1}^M$: optimized individualized policy

Main loop
Main(f_{Env}, Encoder, Quantization, Decoder, π)
$\text{Encoder}, \text{Quantization}, \text{Decoder}, \pi \sim \text{N}(0, \text{I})$ *# Randomly initialize the network*
$\mathcal{H} \leftarrow \{\tau_m\}_{m=1}^M$ *# Collect individual trajectories by interaction with* $\{f_{\text{Env}}^m\}_{m=1}^M$
for each individual m **do**
  $z_m = \text{Encoder}(\mathbf{s}_{0:T}^m)$ *# Capture the high-level representations*
  $\hat{\kappa}_m = \text{Quantization}(z_m)$ *# Vector quantization*
  for each state $\mathbf{s}_t^m$ in the trajectory **do**
    $\hat{\mathbf{s}}_t^m = \text{Decoder}(\mathbf{s}_{t-1}^m, \mathbf{a}_{t-1}^m, \hat{\kappa}_m)$ *# Reconstruct the next state*
  end for
end for
return $\{\pi_g^*\}_{g=1}^G = \text{PolicyLearning}(\mathcal{H}, \{\hat{\kappa}_g\}_{g=1}^G)$ *# Optimize the individualized policies*

EncoderFunction($\mathbf{s}_{0:T}^m$)
if dataset is synthetic **then**
  for each t in $\mathbf{s}_{0:T}^m$ **do**
    $o_t^m \leftarrow \text{Conv1D}(\mathbf{s}_{t:t+H}^m)$
  end for
else if dataset is corpus **then**
  Initialize h_0^m, c_0^m
  for each t in $\mathbf{s}_{0:T}^m$ **do**
    $h_t^m, c_t^m \leftarrow \text{LSTM}(h_{t-1}^m, c_{t-1}^m, \mathbf{s}_t^m; \theta)$
  end for
end if
return $z_m \leftarrow$ Final output of Conv1D or final hidden state of LSTM

QuantizationFunction(z_m)
Initialize $E = \{e_1, e_2, \ldots, e_G\}$, $d_{\text{min}} = \infty$
for each e_i in E **do**
  if $\|z_m - e_i\|_2 < d_{\text{min}}$ **then**
    Update d_{min} and $\hat{\kappa}_m \leftarrow e_i$
  end if
end for
return $\hat{\kappa}_m$

DecoderFunction($\mathbf{s}_{t-1}^m, \mathbf{a}_{t-1}^m, \hat{\kappa}_m$)
Combine inputs to reconstruct $\hat{\mathbf{s}}_t^m \leftarrow \text{Decoder}(\mathbf{s}_{t-1}^m, \mathbf{a}_{t-1}^m, \hat{\kappa}_m)$
return Reconstructed state $\hat{\mathbf{s}}_t^m$

PolicyLearningFunction($\mathcal{H}, \{\hat{\kappa}_g\}_{g=1}^G$)
for each individual m **do**
  Update policy input to $\mu_\pi(\mathbf{s}_t; \theta^\mu) \rightarrow \mu_\pi^m(\mathbf{s}_t^m, \hat{\kappa}^m; \theta^\mu)$
  Update training objective:
    $\mathcal{J}(\theta^\mu) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t Q\left(\mathbf{s}_t, \mu_\pi^m(\mathbf{s}_t^m, \hat{\kappa}_m; \theta^\mu); \theta^Q\right)\right]$
  Optimize μ_π^m for individual m
end for
return Optimized individual policy μ_π^*

Algorithm 2 Training Process with Extended ELBO Objective.

Initialize parameters of the encoder $\mathrm{Encoder}$ and decoder $\mathrm{Decoder}$
Initialize weights α and β
repeat
  for each individual m **do**
    Compute encoded representation: $z_m \leftarrow \mathrm{Encoder}(\mathbf{s}^m_{0:T})$
    Estimate individual-specific factor: $\hat{\kappa}_m \leftarrow \mathrm{Quantization}(z_m)$
    Compute reconstructed state: $\hat{\mathbf{s}}^m_t \leftarrow \mathrm{Decoder}(\mathbf{s}^m_{t-1}, \mathbf{a}^m_{t-1}, \hat{\kappa}_m)$
    Calculate $\mathcal{L}_{\mathrm{Recon}} = \sum_t \|\mathbf{s}^m_t - \hat{\mathbf{s}}^m_t\|^2$
    Calculate $\mathcal{L}_{\text{Quant}} = \sum_i \|\mathrm{sg}[z_{m,i}] - e_{m,i}\|^2$, $\mathcal{L}_{\text{Commit}} = \sum_i \|e_{m,i} - \mathrm{sg}[z_{m,i}]\|^2$
    Compute extended ELBO objective: $\mathcal{L}_{\mathrm{ELBO}} = \mathcal{L}_{\mathrm{Recon}} + \alpha\mathcal{L}_{\text{Quant}} + \beta\mathcal{L}_{\text{Commit}}$
    Update parameters to minimize $\mathcal{L}_{\mathrm{ELBO}}$
  end for
until convergence

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We list the contributions and scope in both abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: See Section 4 and Appendix B and C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide details in both the main content and appendix, with code link.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The access to the data is referred and code is linked.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Details are in appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars are provided in the figures.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Provided in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Checked.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Not related to our work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All are referred.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: Not related to our work.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Not related to our work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not related to our work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.