Q-LDA: Uncovering Latent Patterns in Text-based Sequential Decision Processes

Jianshu Chen*, Chong Wang†, Lin Xiao*, Ji He‡, Lihong Li† and Li Deng‡

*Microsoft Research, Redmond, WA, USA
{jianshuc,lin.xiao}@microsoft.com
†Google Inc., Kirkland, WA, USA*
{chongw,lihong}@google.com
‡Citadel LLC, Seattle/Chicago, USA
{Ji.He,Li.Deng}@citadel.com

Abstract

In sequential decision making, it is often important and useful for end users to understand the underlying patterns or causes that lead to the corresponding decisions. However, typical deep reinforcement learning algorithms seldom provide such information due to their black-box nature. In this paper, we present a probabilistic model, Q-LDA, to uncover latent patterns in text-based sequential decision processes. The model can be understood as a variant of latent topic models that are tailored to maximize total rewards; we further draw an interesting connection between an approximate maximum-likelihood estimation of Q-LDA and the celebrated Q-learning algorithm. We demonstrate in the text-game domain that our proposed method not only provides a viable mechanism to uncover latent patterns in decision processes, but also obtains state-of-the-art rewards in these games.

1 Introduction

Reinforcement learning [21] plays an important role in solving sequential decision making problems, and has seen considerable successes in many applications [16, 18, 20]. With these methods, however, it is often difficult to understand or examine the underlying patterns or causes that lead to the sequence of decisions. Being more interpretable to end users can provide more insights to the problem itself and be potentially useful for downstream applications based on these results [5].

To investigate new approaches to uncovering underlying patterns of a text-based sequential decision process, we use text games (also known as interactive fictions) [11, 19] as the experimental domain. Specifically, we focus on choice-based and hypertext-based games studied in the literature [11], where both the action space and the state space are characterized in natural languages. At each time step, the decision maker (i.e., *agent*) observes one text document (i.e., *observation text*) that describes the current observation of the game environment, and several text documents (i.e., *action texts*) that characterize different possible actions that can be taken. Based on the history of these observations, the agent selects one of the provided actions and the game transits to a new state with an *immediate reward*. This game continues until the agent reaches a final state and receives a *terminal reward*.

In this paper, we present a probabilistic model called *Q-LDA* that is tailored to maximize total rewards in a decision process. Specially, observation texts and action texts are characterized by two separate topic models, which are variants of latent Dirichlet allocation (LDA) [4]. In each topic model, topic proportions are chained over time to model the dependencies for actions or states. And

*The work was done while Chong Wang, Ji He, Lihong Li and Li Deng were at Microsoft Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

these proportions are partially responsible for generating the immediate/terminal rewards. We also show an interesting connection between the maximum-likelihood parameter estimation of the model and the Q-learning algorithm [22, 18]. We empirically demonstrate that our proposed method not only provides a viable mechanism to uncover latent patterns in decision processes, but also obtains state-of-the-art performance in these text games.

Contribution. The main contribution of this paper is to seamlessly integrate topic modeling with Q-learning to uncover the latent patterns and interpretable causes in text-based sequential decision-making processes. Contemporary deep reinforcement learning models and algorithms can seldom provide such information due to their black-box nature. To the best of our knowledge, there is no prior work that can achieve this and learn the topic model in an end-to-end fashion to maximize the long-term reward.

Related work. Q-LDA uses variants of LDA to capture observation and action texts in text-based decision processes. In this model, the dependence of immediate reward on the topic proportions is similar to supervised topic models [3], and the chaining of topic proportions over time to model long-term dependencies on previous actions and observations is similar to dynamic topic models [6]. The novelty in our approach is that the model is estimated in a way that aims to maximize long-term reward, thus producing near-optimal policies; hence it can also be viewed as a topic-model-based reinforcement-learning algorithm. Furthermore, we show an interesting connection to the DQN variant of Q-learning [18]. The text-game setup used in our experiment is most similar to previous work [11] in that both observations and actions are described by natural languages, leading to challenges in both representation and learning. The main difference from that previous work is that those authors treat observation-texts as Markovian states. In contrast, our model is more general, capturing both partial observability and long-term dependence on observations that are common in many text-based decision processes such as dialogues. Finally, the choice of reward function in Q-LDA share similarity with that in Gaussian process temporal difference methods [9].

Organization. Section 2 describes the details of our probabilistic model, and draws a connection to the Q-learning algorithm. Section 3 presents an end-to-end learning algorithm that is based on mirror descent back-propagation. Section 4 demonstrates the empirical performance of our model, and we conclude with discussions and future work in Section 5.

2 A Probabilistic Model for Text-based Sequential Decision Processes

In this section, we first describe text games as an example of sequential decision processes. Then, we describe our probabilistic model, and relate it to a variant of Q-learning.

2.1 Sequential decision making in text games

Text games are an episodic task that proceeds in discrete time steps $t \in \{1, \ldots, T\}$, where the length T may vary across different *episodes*. At time step t, the agent receives a text document of N words describing the current observation of the environment: $w_t^S \triangleq \{w_{t,n}^S\}_{n=1}^N$.[2] We call these words *observation text*. The agent also receives A_t text documents, each of which describes a possible action that the agent can take. We denote them by $w_t^a \triangleq \{w_{t,n}^a\}_{n=1}^N$ with $a \in \{1, \ldots, A_t\}$, where A_t is the number of feasible actions and it could vary over time. We call these texts *action texts*. After the agent takes one of the provided actions, the environment transits to time $t+1$ with a new state and an immediate reward r_t; both dynamics and reward generation may be stochastic and unknown. The new state then reveals a new observation text w_{t+1}^S and several action texts w_{t+1}^a for $a \in \{1, \ldots, A_{t+1}\}$. The transition continues until the end of the game at step T when the agent receives a *terminal reward* r_T. The reward r_T depends on the ending of the story in the text game: a good ending leads to a large positive reward, while bad endings negative rewards.

The goal of the agent is to maximize its cumulative reward by acting optimally in the environment. At step t, given all observation texts $w_{1:t}^S$, all action texts $w_{1:t}^A \triangleq \{w_{1:t}^a : \forall a\}$, previous actions $a_{1:t-1}$ and rewards $r_{1:t-1}$, the agent is to find a *policy*, $\pi(a_t | w_{1:t}^S, w_{1:t}^A, a_{1:t-1}, r_{1:t-1})$, a conditional

[2] For notation simplicity, we assume all texts have the same length N.

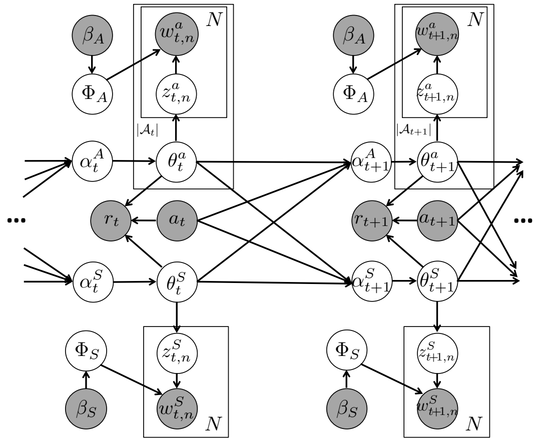


Figure 1: Graphical model representation for the studied sequential decision process. The bottom section shows the observation topic models, which share the same topics in Φ_S, but the topic distributions θ^S_t changes with time t. The top section shows the action topic models, sharing the same action topics in Φ_A, but with time varying topic distribution θ^a_t for each $a \in A_t$. The middle section shows the dependence of variables between consecutive time steps. There are no plates for the observation text (bottom part of the figure) because there is only one observation text document at each time step. We follow the standard notation for graphical models by using shaded circles as observables. Since the topic distributions θ^S_t and θ^a_t and the Dirichlet parameters α^S_t and α^A_t (except α^S_1 and α^A_1) are not observable, we need to use their MAP estimate to make end-to-end learning feasible; see Section 3 for details. The figure characterizes the general case where rewards appear at each time step, while in our experiments the (non-zero) rewards only appear at the end of the games.

probability of selecting action a_t, that maximizes the expected long-term reward $\mathbb{E}\{\sum_{\tau=t}^{T} \gamma^{\tau-t} r_\tau\}$, where $\gamma \in (0, 1)$ is a discount factor. In this paper, for simplicity of exposition, we focus on problems where the reward is nonzero only in the final step T. While our algorithm can be generalized to the general case (with greater complexity), this special case is an important case of RL (e.g., [20]). As a result, the policy is independent of $r_{1:t-1}$ and its form is simplified to $\pi(a_t|w^S_{1:t}, w^A_{1:t}, a_{1:t-1})$.

The problem setup is similar to previous work [11] in that both observations and actions are described by natural languages. For actions described by natural languages, the action space is inherently discrete and large due to the exponential complexity with respect to sentence length. This is different from most reinforcement learning problems where the action spaces are either small or continuous. Here, we take a probabilistic modeling approach to this challenge: the observed variables—observation texts, action texts, selected actions, and rewards—are assumed to be generated from a probabilistic latent variable model. By examining these latent variables, we aim to uncover the underlying patterns that lead to the sequence of the decisions. We then show how the model is related to Q-learning, so that estimation of the model leads to reward maximization.

2.2 The Q-LDA model

The graphical representation of our model, Q-LDA, is depicted in Figure 1. It has two instances of topic models, one for observation texts and the other for action texts. The basic idea is to chain the topic proportions (θs in the figure) in a way such that they can influence the topic proportions in the future, thus capturing long-term effects of actions. Details of the generative models are as follows.

For the observation topic model, we use the columns of $\Phi_S \sim \text{Dir}(\beta_S)$[3] to denote the topics for the observation texts. For the action topic model, we use the columns of $\Phi_A \sim \text{Dir}(\beta_A)$ to denote the topics for the action texts. We assume these topics do not change over time. Given the initial topic proportion Dirichlet parameters—α^S_1 and α^A_1 for observation and action texts respectively—the Q-LDA proceeds sequentially from $t = 1$ to T as follows (see Figure 1 for all latent variables).

[3] Φ_S is a word-by-topic matrix. Each column is drawn from a Dirichlet distribution with hyper-parameter β_S, representing the word-emission probabilities of the corresponding topic. Φ_A is similarly defined.

1. Draw observation text w_t^S as follows,
   (a) Draw observation topic proportions $\theta_t^S \sim \mathrm{Dir}(\alpha_t^S)$.
   (b) Draw all words for the observation text $w_t^S \sim \mathrm{LDA}(w_t^S|\theta_t^S, \Phi_S)$, where $\mathrm{LDA}(\cdot)$ denotes the standard LDA generative process given its topic proportion θ_t^S and topics Φ_S [4]. The latent variable $z_{t,n}^S$ indicates the topic for the word $w_{t,n}^S$.
2. For $a = 1, ..., A_t$, draw action text w_t^a as follows,
   (a) Draw action topic proportions $\theta_t^a \sim \mathrm{Dir}(\alpha_t^A)$.
   (b) Draw all words for the a-th action text using $w_t^a \sim \mathrm{LDA}(w_t^a|\theta_t^a, \Phi_A)$, where the latent variable $z_{t,n}^a$ indicates the topic for the word $w_{t,n}^a$.
3. Draw the action: $a_t \sim \pi_b(a_t|w_{1:t}^S, w_{1:t}^A, a_{1:t-1})$, where π_b is an *exploration policy* for data collection. It could be chosen in different ways, as discussed in the experiment Section 4. After model learning is finished, a greedy policy may be used instead (c.f., Section 3).
4. The immediate reward r_t is generated according to a Gaussian distribution with mean function $\mu_r(\theta_t^S, \theta_t^{a_t}, U)$ and variance σ_r^2:
$$r_t \sim \mathcal{N}\left(\mu_r(\theta_t^S, \theta_t^{a_t}, U), \sigma_r^2\right) . \tag{1}$$
   Here, we defer the definitions of $\mu_r(\theta_t^S, \theta_t^{a_t}, U)$ and its parameter U to the next section, where we draw a connection between likelihood-based learning and Q-learning.
5. Compute the topic proportions Dirichlet parameters for the next time step $t + 1$ as
$$\alpha_{t+1}^S = \sigma\left(W_{SS}\theta_t^S + W_{SA}\theta_t^{a_t} + \alpha_1^S\right), \quad \alpha_{t+1}^A = \sigma\left(W_{AS}\theta_t^S + W_{AA}\theta_t^{a_t} + \alpha_1^A\right) , \tag{2}$$
   where $\sigma(x) \triangleq \max\{x, \epsilon\}$ with ϵ being a small positive number (e.g., 10^{-6}), a_t is the action selected by the agent at time t, and $\{W_{SS}, W_{SA}, W_{AS}, W_{AA}\}$ are the model parameters to be learned. Note that, besides θ_t^S, the only topic proportions from $\{\theta_t^a\}_{a=1}^{A_t}$ that will influence α_{t+1}^S and α_{t+1}^A is $\theta_t^{a_t}$, i.e., the one corresponding to the chosen action a_t. Furthermore, since θ_t^S and $\theta_t^{a_t}$ are generated according to $\mathrm{Dir}(\alpha_t^S)$ and $\mathrm{Dir}(\alpha_t^A)$, respectively, α_{t+1}^S and α_{t+1}^A are (implicitly) chained over time via θ_t^S and $\theta_t^{a_t}$ (c.f. Figure 1).

This generative process defines a joint distribution $p(\cdot)$ among all random variables depicted in Figure 1. Running this generative process—step 1 to 5 above for T steps until the game ends—produces one episode of the game. Now suppose we already have M episodes. In this paper, we choose to directly learn the conditional distribution of the rewards given other observations. By learning the model in a *discriminative* manner [2, 7, 12, 15, 23], we hope to make better predictions of the rewards for different actions, from which the agent could obtain the best policy for taking actions. This can be obtained by applying Bayes rule to the joint distribution defined by the generative process. Let Θ denote all model parameters: $\Theta = \{\Phi_S, \Phi_A, U, W_{SS}, W_{SA}, W_{AS}, W_{AA}\}$. We have the following loss function

$$\min_{\Theta}\left\{-\ln p(\Theta) - \sum_{i=1}^{M} \ln p\left(r_{1:T_i}|w_{1:T_i}^S, w_{1:T_i}^A, a_{1:T_i}, \Theta\right)\right\} , \tag{3}$$

where $p(\Theta)$ denotes a prior distribution of the model parameters (e.g., Dirichlet parameters over Φ_S and Φ_A), and T_i denotes the length of the i-th episode. Let K_S and K_A denote the number of topics for the observation texts and action texts, and let V_S and V_A denote the vocabulary sizes for the observation texts and action texts, respectively. Then, the total number of learnable parameters for Q-LDA is: $V_S \times K_S + V_A \times K_A + K_A \times K_S + (K_S + K_A)^2$.

We note that a good model learned through Eq. (3) may predict the values of rewards well, but might not imply the best policy for the game. Next, we show by defining the appropriate mean function for the rewards, $\mu_r(\theta_t^S, \theta_t^{a_t}, U)$, we can achieve both. This closely resembles Q-learning [21, 22], allowing us to effectively learn the policy in an iterative fashion.

2.3 From Q-LDA to Q-learning

Before relating Q-LDA to Q-learning, we first give a brief introduction to the latter. Q-learning [22, 18] is a reinforcement learning algorithm for finding an optimal policy in a Markov decision process (MDP) described by $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, and $\gamma \in (0, 1)$ is a discount factor. Furthermore, $\mathcal{P}$ defines a transition probability $p(s'|s, a)$ for going to the next state $s' \in \mathcal{S}$ from the current state $s \in \mathcal{S}$ after taking action $a \in \mathcal{A}$, and $r(s,a)$ is the immediate reward corresponding to this transition. A policy $\pi(a|s)$ in an MDP is defined to be the probability of taking action a at state s. Let s_t and a_t be the state and action at time t, and let $r_t = r(s_t, a_t)$ be the immediate reward at time t. An optimal policy is the one that maximizes the expected long-term reward $\mathbb{E}\{\sum_{t=1}^{+\infty} \gamma^{t-1} r_t\}$. Q-learning seeks to find the optimal policy by estimating the Q-function, $Q(s,a)$, defined as the expected long-term discounted reward for taking action a at state s and then following an optimal policy thereafter. It satisfies the Bellman equation [21]

$$Q(s,a) = \mathbb{E}\{r(s,a) + \gamma \cdot \max_b Q(s', b)|s,a\}, \tag{4}$$

and directly gives the optimal action for any state s: $\arg\max_a Q(s,a)$.

Q-learning solves for $Q(s,a)$ iteratively based on observed state transitions. The basic Q-learning [22] requires storing and updating the values of $Q(s,a)$ for all state–action pairs in $\mathcal{S} \times \mathcal{A}$, which is not practical when $\mathcal{S}$ and $\mathcal{A}$ are large. This is especially true in our text games, where they can be exponentially large. Hence, $Q(s,a)$ is usually approximated by a parametric function $Q_\theta(s,a)$ (e.g., neural networks [18]), in which case the model parameter θ is updated by:

$$\theta \leftarrow \theta + \eta \cdot \nabla_\theta Q_\theta \cdot (d_t - Q_\theta(s_t, a_t)), \tag{5}$$

where $d_t \triangleq r_t + \gamma \cdot \max_{a'} Q_{\theta'}(s_{t+1}, a')$ if s_t nonterminal and $d_t \triangleq r_t$ otherwise, and θ' denotes a delayed version of the model parameter updated periodically [18]. The update rule (5) may be understood as applying stochastic gradient descent (SGD) to a regression loss function $J(\theta) \triangleq \mathbb{E}[d_t - Q_\theta(s,a)]^2$. Thus, d_t is the *target*, computed from r_t and $Q_{\theta'}$, for the prediction $Q_\theta(s_t, a_t)$.

We are now ready to define the mean reward function μ_r in Q-LDA. First, we model the Q-function by $Q(\theta_t^S, \theta_t^a) = (\theta_t^a)^T U \theta_t^S$, where U is the same parameter as the one in (1).[4] This is different from typical deep RL approaches, where black-box models like neural networks are used. In order to connect our probabilistic model to Q-learning, we define the mean reward function as follows,

$$\mu_r(\theta_t^S, \theta_t^{a_t}, U) = Q(\theta_t^S, \theta_t^{a_t}) - \gamma \cdot \mathbb{E}\big[\max_b Q(\theta_{t+1}^S, \theta_{t+1}^b)|\theta_t^S, \theta_t^{a_t}\big] \tag{6}$$

Note that μ_r remains as a function of θ_t^S and $\theta_t^{a_t}$ since the second term in the above expression is a conditional expectation given θ_t^S and $\theta_t^{a_t}$. The definition of the mean reward function in Eq. (6) has a strong relationship with the Bellman equation (4) in Q-learning; it relates the long-term reward $Q(\theta_t^S, \theta_t^{a_t})$ to the mean immediate reward μ_r in the same manner as the Bellman equation (4). To see this, we move the second term on the right-hand side of (6) to the left, and make the identification that μ_r corresponds to $\mathbb{E}\{r(s,a)\}$ since both of them represent the mean immediate reward. The resulting equation share a same form as the Bellman equation (4). With the mean function μ_r defined above, we show in Appendix B that the loss function (3) can be approximated by the one below using the maximum a posteriori (MAP) estimate of θ_t^S and $\theta_t^{a_t}$ (denoted as $\hat{\theta}_t^S$ and $\hat{\theta}_t^{a_t}$, respectively):

$$\min_\Theta \Big\{ -\ln p(\Phi_S|\beta_S) - \ln p(\Phi_A|\beta_A) + \sum_{i=1}^{M} \sum_{t=1}^{T_i} \frac{1}{2\sigma_r^2} \Big[d_t - Q(\hat{\theta}_t^S, \hat{\theta}_t^{a_t})\Big]^2 \Big\} \tag{7}$$

where $d_t = r_t + \gamma \max_b Q(\hat{\theta}_{t+1}^S, \hat{\theta}_{t+1}^b)$ for $t < T_i$ and $d_t = r_t$ for $t = T_i$. Observe that the first two terms in (7) are regularization terms coming from the Dirichlet prior over Φ_S and Φ_A, and the third term shares a similar form as the cost $J(\theta)$ in Q-learning; it can also be interpreted as a regression problem for estimating the Q-function, where the target d_t is constructed in a similar manner as Q-learning. Therefore, optimizing the discriminative objective (3) leads to a variant of Q-learning. After learning is finished, we can obtain the greedy policy by taking the action that maximizes the Q-function estimate in any given state.

We also note that we have used the MAP estimates of θ_t^S and $\theta_t^{a_t}$ due to the intractable marginalization of the latent variables [14]. Other more advanced approximation techniques, such as Markov Chain Monte Carlo (MCMC) [1] and variational inference [13] can also be used, and we leave these explorations as future work.

3 End-to-end Learning by Mirror Descent Back Propagation

[4] The intuition of choosing $Q(\cdot,\cdot)$ to be this form is that we want θ_t^S to be aligned with θ_t^a of the correct action (large Q-value), and to be misaligned with the θ_t^a of the wrong actions (small Q-value). The introduction of U allows the number and the meaning of topics for the observations and actions to be different.

Algorithm 1 The training algorithm by mirror descent back propagation

```
Input: $D$ (number of experience replays), $J$ (number of SGD updates), and learning rate.
Randomly initialize the model parameters.
for $m = 1, \dots, D$ do
    Interact with the environment using a behavior policy $\pi_b^m(a_t|x_{1:t}^S, x_{1:t}^A, a_{1:t-1})$ to collect $M$ episodes of data $\{w_{1:T_i}^S, w_{1:T_i}^A, a_{1:T_i}, r_{1:T_i}\}_{i=1}^M$ and add them to $\mathcal{D}$.
    for $j = 1, \dots, J$ do
        Randomly sample an episode from $\mathcal{D}$.
        For the sampled episode, compute $\hat{\theta}_t^S$, $\hat{\theta}_t^a$ and $Q(\hat{\theta}_t^S, \hat{\theta}_t^a)$ with $a = 1, \dots, A_t$ and $t = 1, \dots, T_i$ according to Algorithm 2.
        For the sampled episode, compute the stochastic gradients of (7) with respect to $\Theta$ using back propagation through the computational graph defined in Algorithm 2.
        Update $\{U, W_{SS}, W_{SA}, W_{AS}, W_{AA}\}$ by stochastic gradient descent and update $\{\Phi_S, \Phi_A\}$ using stochastic mirror descent.
    end for
end for
```

Algorithm 2 The recursive MAP inference for one episode

```
Input: $\alpha_1^S$, $\alpha_1^A$, $L$, $\delta$, $x_t^S$, $\{x_t^a : a = 1, \dots, A_t\}$ and $a_t$, for all $t = 1, \dots, T_i$.
Initialization: $\hat{\alpha}_1^S = \alpha_1^S$ and $\hat{\alpha}_1^A = \alpha_1^A$
for $t = 1, \dots, T_i$ do
    Compute $\hat{\theta}_t^S$ by repeating $\hat{\theta}_t^S \leftarrow \frac{1}{C}\hat{\theta}_t^S \odot \exp\left(\delta\left[\Phi_S^T \frac{x_t^S}{\Phi_S \hat{\theta}_t^S} + \frac{\hat{\alpha}_t^S - \mathbb{1}}{\hat{\theta}_t^S}\right]\right)$ for $L$ times with initialization $\hat{\theta}_t^S \propto \mathbb{1}$, where $C$ is a normalization factor.
    Compute $\hat{\theta}_t^a$ for each $a = 1, \dots, A_t$ by repeating $\hat{\theta}_t^a \leftarrow \frac{1}{C}\hat{\theta}_t^a \odot \exp\left(\delta\left[\Phi_A^T \frac{x_t^a}{\Phi_A \hat{\theta}_t^a} + \frac{\hat{\alpha}_t^A - \mathbb{1}}{\hat{\theta}_t^a}\right]\right)$ for $L$ times with initialization $\hat{\theta}_t^a \propto \mathbb{1}$, where $C$ is a normalization factor.
    Compute $\hat{\alpha}_{t+1}^S$ and $\hat{\alpha}_{t+1}^A$ from $\hat{\theta}_t^S$ and $\hat{\theta}_t^{a_t}$ according to (11).
    Compute the Q-values: $Q(\hat{\theta}_t^S, \hat{\theta}_t^a) = (\hat{\theta}_t^a)^T U \hat{\theta}_t^S$ for $a = 1, \dots, A_t$.
end for
```

In this section, we develop an end-to-end learning algorithm for Q-LDA, by minimizing the loss function given in (7). As shown in the previous section, solving (7) leads to a variant of Q-learning, thus our algorithm could be viewed as a reinforcement-learning algorithm for the proposed model.

We consider learning our model with experience replay [17], a widely used technique in recent state-of-the-art systems [18]. Specifically, the learning process consists of multiple stages, and at each stage, the agent interacts with the environment using a fixed exploration policy $\pi_b(a_t|x_{1:t}^S, x_{1:t}^A, a_{1:t-1})$ to collect M episodes of data $\{w_{1:T_i}^S, w_{1:T_i}^A, a_{1:T_i}, r_{1:T_i}\}_{i=1}^M$ and saves them into a *replay memory* $\mathcal{D}$. (We will discuss the choice of π_b in section 4.) Under the assumption of the generative model Q-LDA, our objective is to update our estimates of the model parameters in Θ using $\mathcal{D}$; the updating process may take several randomized passes over the data in $\mathcal{D}$. A stage of such learning process is called one *replay*. Once a replay is done, we let the agent use a new behavior policy π_b' to collect more episodes, add them to $\mathcal{D}$, and continue to update Θ from the augmented $\mathcal{D}$. This process repeats for multiple stages, and the model parameters learned from the previous stage will be used as the initialization for the next stage. Therefore, we can focus on learning at a single stage, which was formulated in Section 2 as one of solving the optimization problem (7). Note that the objective (7) is a function of the MAP estimates of θ_t^S and $\theta_t^{a_t}$. Therefore, we start with a recursion for computing $\hat{\theta}_t^S$ and $\hat{\theta}_t^{a_t}$ and then introduce our learning algorithm for Θ.

3.1 Recursive MAP inference by mirror descent

The MAP estimates, $\hat{\theta}_t^S$ and $\hat{\theta}_t^a$, for the topic proportions θ_t^S and θ_t^a are defined as

$$(\hat{\theta}_t^S, \hat{\theta}_t^a) = \arg\max_{\theta_t^S, \theta_t^a} p(\theta_t^S, \theta_t^a | w_{1:t}^S, w_{1:t}^A, a_{1:t-1}) \tag{8}$$

Solving for the exact solution is, however, intractable. We instead develop an approximate algorithm that recursively estimate $\hat{\theta}_t^S$ and $\hat{\theta}_t^a$. To develop the algorithm, we rely on the following result, whose proof is deferred to Appendix A.

Proposition 1. *The MAP estimates in* (8) *could be approximated by recursively solving the problems:*

$$\hat{\theta}_t^S = \arg\max_{\theta_t^S} \left[\ln p(x_t^S|\theta_t^S, \Phi_S) + \ln p\big(\theta_t^S|\hat{\alpha}_t^S\big)\right] \tag{9}$$

$$\hat{\theta}_t^a = \arg\max_{\theta_t^a} \left[\ln p(x_t^a|\theta_t^a, \Phi_A) + \ln p\big(\theta_t^a|\hat{\alpha}_t^A\big)\right], \quad a \in \{1, \ldots, A_t\}, \tag{10}$$

where x_t^S and x_t^a are the bag-of-words vectors for the observation text w_t^S and the a-th action text w_t^a, respectively. To compute $\hat{\alpha}_t^S$ and $\hat{\alpha}_t^A$, we begin with $\hat{\alpha}_1^S = \alpha_1^S$ and $\hat{\alpha}_1^A = \alpha_1^A$ and update their values for the next $t+1$ time step according to

$$\hat{\alpha}_{t+1}^S = \sigma\left(W_{SS}\hat{\theta}_t^S + W_{SA}\hat{\theta}_t^{a_t} + \alpha_1^S\right), \quad \hat{\alpha}_{t+1}^A = \sigma\left(W_{AS}\hat{\theta}_t^S + W_{AA}\hat{\theta}_t^{a_t} + \alpha_1^A\right) \tag{11}$$

Note from (9)–(10) that, for given $\hat{\theta}_t^S$ and $\hat{\theta}_t^a$, the solution of θ_t^S and θ_t^a now becomes A_t+1 decoupled sub-problems, each of which has the same form as the MAP inference problem of Chen et al. [8]. Therefore, we solve each sub-problem in (9)–(10) using their mirror descent inference algorithm, and then use (11) to compute the Dirichlet parameters at the next time step. The overall MAP inference procedure is summarized in Algorithm 2. We further remark that, after obtaining $\hat{\theta}_t^S$ and $\hat{\theta}_t^a$, the Q-value for the t step is readily estimated by:

$$\mathbb{E}\big[Q(\theta_t^S, \theta_t^a)|w_{1:t}^S, w_{1:t}^A, a_{1:t-1}\big] \approx Q(\hat{\theta}_t^S, \hat{\theta}_t^a), \quad a \in \{1, \ldots, A_t\}, \tag{12}$$

where we approximate the conditional expectation using the MAP estimates. After learning is finished, the agent may extract a greedy policy for any state s by taking the action $\arg\max_a Q(\hat{\theta}^S, \hat{\theta}^a)$. It is known that if the learned Q-function is closed to the true Q-function, such a greedy policy is near-optimal [21].

3.2 End-to-end learning by backpropagation

The training loss (7) for each learning stage has the form of a finite sum over M episodes. Each term inside the summation depends on $\hat{\theta}_t^S$ and $\hat{\theta}_t^{a_t}$, which in turn depend on all the model parameters in Θ via the computational graph defined by Algorithm 2 (see Appendix E for a diagram of the graph). Therefore, we can learn the model parameters in Θ by sampling an episode in the data, computing the corresponding stochastic gradient in (7) by back-propagation on the computational graph given in Algorithm 2, and updating Θ by stochastic gradient/mirror descent. More details are found in Algorithm 1, and Appendix E.4 gives the gradient formulas.

4 Experiments

In this section, we use two text games from [11] to evaluate our proposed model and demonstrate the idea of interpreting the decision making processes: (i) "Saving John" and (ii) "Machine of Death" (see Appendix C for a brief introduction of the two games).[5] The action spaces of both games are defined by natural languages and the feasible actions change over time, which is a setting that Q-LDA is designed for. We choose to use the same experiment setup as [11] in order to have a fair comparison with their results. For example, at each m-th experience-replay learning (see Algorithm 1), we use the softmax action selection rule [21, pp.30–31] as the exploration policy to collect data (see Appendix E.3 for more details). We collect $M = 200$ episodes of data (about 3K time steps in "Saving John" and 16K in "Machine of Death") at each of $D = 20$ experience replays, which amounts to a total of $4,000$ episodes. At each experience replay, we update the model with 10 epochs before the next replay. Appendix E provides additional experimental details.

We first evaluate the performance of the proposed Q-LDA model by the long-term rewards it receives when applied to the two text games. Similar to [11], we repeat our experiments for five times with different random initializations. Table 1 summarize the means and standard deviations of the rewards

[5] The simulators are obtained from `https://github.com/jvking/text-games`

Table 1: The average rewards (higher is better) and standard deviations of different models on the two tasks. For DRRN and MA-DQN, the number of topics becomes the number of hidden units per layer.

Tasks	# topics	Q-LDA	DRRN (1-layer)	DRRN (2-layer)	MA-DQN (2-layer)
Saving John	20	**18.8** (0.3)	17.1 (0.6)	18.4 (0.1)	4.9 (3.2)
	50	**18.6** (0.6)	18.3 (0.2)	18.5 (0.3)	9.0 (3.2)
	100	**19.1** (0.6)	18.2 (0.2)	18.7 (0.4)	7.1 (3.1)
Machine of Death	20	**19.9** (0.8)	7.2 (1.5)	9.2 (2.1)	2.8 (0.9)
	50	**18.7** (2.1)	8.4 (1.3)	10.7 (2.7)	4.3 (0.9)
	100	**17.5** (2.4)	8.7 (0.9)	11.2 (0.6)	5.2 (1.2)

on the two games. We include the results of Deep Reinforcement Relevance Network (DRRN) proposed in [11] with different hidden layers. In [11], there are several variants of DQN (deep Q-networks) baselines, among which MA-DQN (max-action DQN) is the best performing one. We therefore only include the results of MA-DQN. Table 1 shows that Q-LDA outperforms all other approaches on both tasks, especially "Machine of Death", where Q-LDA even beats the DRRN models by a large margin. The gain of Q-LDA on "Saving John" is smaller, as both Q-LDA and DRRN are approaching the upper bound of the reward, which is 20. "Machine of Death" was believed to be a more difficult task due to its stochastic nature and larger state and action spaces [11], where the upper bound on the reward is 30. (See Tables 4–5 for the definition of the rewards for different story endings.) Therefore, Q-LDA gets much closer to the upper bound than any other method, although there may still be room for improvement. Finally, our experiments follow the standard online RL setup: after a model is updated based on the data observed so far, it is tested on newly generated episodes. Therefore, the numbers reported in Table 1 are *not* evaluated on the training dataset, so they truthfully reflect the actual average reward of the learned models.

We now proceed to demonstrate the analysis of the latent pattern of the decision making process using one example episode of "Machine of Death". In this episode, the game starts with the player wandering in a shopping mall, after the peak hour ended. The player approaches a machine that prints a death card after inserting a coin. The death card hints on how the player will die in future. In one of the story development, the player's death is related to a man called Bon Jovi. The player is so scared that he tries to combat with a cardboard standee of Bon Jovi. He reveals his concern to a friend named Rachel, and with her help he finally overcomes his fear and maintains his friendship. This episode reaches a good ending and receives the highest possible reward of 30 in this game.

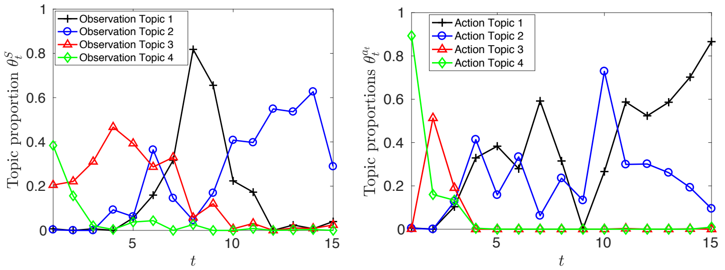
In Figure 2, we show the evolution of the topic proportions for the four most active topics (shown in Table 2)[6] for both the observation texts and the selected actions' texts. We note from Figure 2 that the most dominant observation topic and action topic at beginning of the episode are "wander at mall" and "action at mall", respectively, which is not surprising since the episode starts at a mall scenario. The topics related to "mall" quickly dies off after the player starts the death machine. Afterwards, the most salient observation topic becomes "meet Bon Jovi" and then "combat" ($t = 8$). This is because after the activation of death machine, the story enters a scenario where the player tries to combat with a cardboard standee. Towards the end of the episode, the observation topic "converse w/rachel" and the topic "kitchen & chat" corresponding to the selected action reach their peaks and then decay right before the end of the story, where the action topic "relieve" climbs up to its peak. This is consistent with the story ending, where the player chooses to overcome his fear after chatting with Rachel. In Appendix D, we show the observation and the action texts in the above stages of the story.

Finally, another interesting observation is about the matrix U. Since the Q-function value is computed from $[\hat{\theta}_t^a]^T U \hat{\theta}_t^S$, the (i, j)-th element of the matrix U measures the positive/negative correlation between the i-th action topic and the j-th observation topic. In Figure 2(c), we show the value of the learned matrix U for the four observation topics and the four action topics in Table 2. Interestingly, the largest value (39.5) of U is the $(1, 2)$-th element, meaning that the action topic "relieve" and the state topic "converse w/rachel" has strong positive contribution to a high long-term reward, which is what happens at the end of the story.

[6] In practice, we observe that some topics are never or rarely activated during the learning process. This is especially true when the number of topics becomes large (e.g., 100). Therefore, we only show the most active topics. This might also explain why the performance improvement is marginal when the number of topics grows.

Table 2: The four most active topics for the observation texts and the action texts, respectively.

Observation Topics	
1: combat	minutes, lights, firearm, shoulders, whiff, red, suddenly, huge, rendition
2: converse w/ rachel	rachel, tonight, grabs, bar, towards, happy, believing, said, moonlight
3: meet Bon Jovi	small, jovi, bon, door, next, dog, insists, room, wrapped, standees
4: wander at mall	ended, catcher, shopping, peak, wrapped, hanging, attention, door
Action Topics	
1: relieve	leave, get, gotta, go, hands, away, maybe, stay, ability, turn, easy, rachel
2: kitchen & chat	wait, tea, look, brisk, classics, oysters, kitchen, turn, chair, moment
3: operate the machine	coin, insert, west, cloth, desk, apply, dollars, saying, hands, touch, tell
4: action at mall	alarm, machine, east, ignore, take, shot, oysters, win, gaze, bestowed



(a) Observation topic θ_t^S

(b) Selected action topic $\theta_t^{a_t}$

$$\begin{bmatrix} 1.2 & \mathbf{39.5} & 20.7 & 12.2 \\ 22.1 & 12.4 & 1.4 & -0.2 \\ 2.5 & 4.8 & 4.1 & 1.9 \\ 5.3 & 8.4 & 13.3 & 4.1 \end{bmatrix}$$

(c) Learned values of matrix U

Figure 2: The evolution of the most active topics in "Machine of Death."

5 Conclusion

We proposed a probabilistic model, Q-LDA, to uncover latent patterns in text-based sequential decision processes. The model can be viewed as a latent topic model, which chains the topic proportions over time. Interestingly, by modeling the mean function of the immediate reward in a special way, we showed that discriminative learning of Q-LDA using its likelihood is closely related to Q-learning. Thus, our approach could also be viewed as a Q-learning variant for sequential topic models. We evaluate Q-LDA on two text-game tasks, demonstrating state-of-the-art rewards in these games. Furthermore, we showed our method provides a viable approach to finding interesting latent patterns in such decision processes.

Acknowledgments

The authors would like to thank all the anonymous reviewers for their constructive feedback.

References

[1] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1):5–43, 2003.

[2] C. M. Bishop and J. Lasserre. Generative or discriminative? getting the best of both worlds. *Bayesian Statistics*, 8:3–24, 2007.

[3] D. M. Blei and J. D. Mcauliffe. Supervised topic models. In *Proc. NIPS*, pages 121–128, 2007.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[5] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[6] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[7] G. Bouchard and B. Triggs. The tradeoff between generative and discriminative classifiers. In *Proc. COMPSTAT*, pages 721–728, 2004.

[8] Jianshu Chen, Ji He, Yelong Shen, Lin Xiao, Xiaodong He, Jianfeng Gao, Xinying Song, and Li Deng. End-to-end learning of lda by mirror-descent back propagation over a deep architecture. In *Proc. NIPS*, pages 1765–1773, 2015.

[9] Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with Gaussian processes. In *Proceedings of the Twenty-Second International Conference on Machine Learning (ICML-05)*, pages 201–208, 2005.

[10] Matthew Hausknecht and Peter Stone. Deep recurrent Q-learning for partially observable MDPs. In *Proc. AAAI-SDMIA*, November 2015.

[11] Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. Deep reinforcement learning with a natural language action space. In *Proc. ACL*, 2016.

[12] A. Holub and P. Perona. A discriminative framework for modelling object classes. In *Proc. IEEE CVPR*, volume 1, pages 664–671, 2005.

[13] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. In *Learning in graphical models*, pages 105–161. Springer, 1998.

[14] Michael Irwin Jordan. *Learning in graphical models*, volume 89. Springer Science & Business Media, 1998.

[15] S. Kapadia. *Discriminative Training of Hidden Markov Models*. PhD thesis, University of Cambridge, 1998.

[16] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[17] Long-Ji Lin. Reinforcement learning for robots using neural networks. Technical report, Technical report, DTIC Document, 1993.

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[19] Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. In *Proc. EMNLP*, 2015.

[20] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

[21] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[22] Christopher Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[23] Oksana Yakhnenko, Adrian Silvescu, and Vasant Honavar. Discriminatively trained Markov model for sequence classification. In *Proc. IEEE ICDM*, 2005.

Supplementary Material for "Q-LDA: Uncovering Latent Patterns in Text-based Sequential Decision Processes"

A Proof of Proposition 1

We first write out the joint probability of the Q-LDA model as

$$\prod_{i=1}^{M}\prod_{t=0}^{T_i}\Big\{\pi_b(a_t|w^S_{1:t}, w^A_{1:t}, a_{1:t-1}) \times p(w^S_t, z^S_t, \theta^S_t|\alpha^S_t, \Phi_S)p(\alpha^S_t|\theta^S_{t-1}, \theta^{a_{t-1}}_{t-1}, W_{SS}, W_{SA})p(\Phi_S|\beta_S)$$
$$\times\, p(w^A_t, z^A_t, \theta^A_t|\alpha^A_t, \Phi_A)p(\alpha^A_t|\theta^S_{t-1}, \theta^{a_{t-1}}_{t-1}, W_{AS}, W_{AA})p(\Phi_A|\beta_A) \times p(r_t|\theta^S_t, \theta^{a_t}_t, U)\Big\} \quad (13)$$

where $\theta^A_t \triangleq \{\theta^a_t\}$, $z^S_t \triangleq \{z^S_{t,n}\}$ and $z^A_t \triangleq \{z^a_{t,n}\}$. Following the same line of argument as in [8], we marginalize the variables z^S_t and z^A_t in joint probability of the Q-LDA model and obtain

$$\prod_{i=1}^{M}\prod_{t=0}^{T_i}\Big\{\pi_b(a_t|w^S_{1:t}, w^A_{1:t}, a_{1:t-1}) \times p(x^S_t, \theta^S_t|\alpha^S_t, \Phi_S)p(\alpha^S_t|\theta^S_{t-1}, \theta^{a_{t-1}}_{t-1}, W_{SS}, W_{SA})p(\Phi_S|\beta_S)$$
$$\times\, p(x^A_t, \theta^A_t|\alpha^A_t, \Phi_A)p(\alpha^A_t|\theta^S_{t-1}, \theta^{a_{t-1}}_{t-1}, W_{AS}, W_{AA})p(\Phi_A|\beta_A) \times p(r_t|\theta^S_t, \theta^{a_t}_t, U)\Big\} \quad (14)$$

where x^S_t is the bag-of-words (BOW) vector for the observation text at the t-th time step. Note that the probability depends on $w^S_{1:t}$ and $w^A_{1:t}$ via $x^S_{1:t}$ and $x^A_{1:t}$. Therefore, we can also write the policy as $\pi_b(a_t|x^S_{1:t}, x^A_{1:t}, a_{1:t-1})$ so that

$$\prod_{i=1}^{M}\prod_{t=0}^{T_i}\Big\{\pi_b(a_t|x^S_{1:t}, x^A_{1:t}, a_{1:t-1}) \times p(x^S_t, \theta^S_t|\alpha^S_t, \Phi_S)p(\alpha^S_t|\theta^S_{t-1}, \theta^{a_{t-1}}_{t-1}, W_{SS}, W_{SA})p(\Phi_S|\beta_S)$$
$$\times\, p(x^A_t, \theta^A_t|\alpha^A_t, \Phi_A)p(\alpha^A_t|\theta^S_{t-1}, \theta^{a_{t-1}}_{t-1}, W_{AS}, W_{AA})p(\Phi_A|\beta_A) \times p(r_t|\theta^S_t, \theta^{a_t}_t, U)\Big\} \quad (15)$$

First, by Bayes rule, we have

$$p(\theta^S_t, \theta^A_t|x^S_{1:t}, x^A_{1:t}, a_{1:t-1}) = \frac{p(\theta^S_t, \theta^A_t, x^S_t, x^A_t|x^S_{1:t-1}, x^A_{1:t-1}, a_{1:t-1})}{p(x^S_t, x^A_t|x^S_{1:t-1}, x^A_{1:t-1}, a_{1:t-1})} \quad (16)$$

where for simplicity of notation we dropped the dependency on the model parameters $\Theta \triangleq (\Phi_S, \Phi_A, W_{SS}, W_{SA}, W_{AS}, W_{AA}, U)$. Note that the denominator is independent of (θ^S_t, θ^A_t). Therefore, the MAP estimate of (θ^S_t, θ^A_t) is the same as maximizing the numerator:

$$(\hat{\theta}^S_t, \hat{\theta}^A_t) \triangleq \arg\max_{\theta_t} p(\theta^S_t, \theta^A_t, x^S_t, x^A_t|x^S_{1:t-1}, x^A_{1:t-1}, a_{1:t-1}) \quad (17)$$

We now proceed to compute the probability $p(\theta^S_t, \theta^A_t, x^S_t, x^A_t|x^S_{1:t-1}, x^A_{1:t-1}, a_{1:t-1})$. Note that

$$\begin{aligned}&p(\theta^S_t, \theta^A_t, x^S_t, x^A_t|x^S_{1:t-1}, x^A_{1:t-1}, a_{1:t-1})\\ &\quad= \int p(x^S_t|\theta^S_t, \Phi_S)p(\theta^S_t|\alpha^S_t)p(\alpha^S_t|\theta^S_{t-1}, \theta^{a_{t-1}}_{t-1}, W_{SS}, W_{SA})\\ &\qquad\times p(x^A_t|\theta^A_t, \Phi_A)p(\theta^A_t|\alpha^A_t)p(\alpha^A_t|\theta^S_{t-1}, \theta^{a_{t-1}}_{t-1}, W_{AA}, W_{AS})\\ &\qquad\times p(\theta^S_{t-1}, \theta^A_{t-1}|x^S_{1:t-1}, x^A_{1:t-1}, a_{1:t-1})d\alpha^S_t d\alpha^A_t d\theta^S_{t-1}d\theta^A_{t-1}\end{aligned} \quad (18)$$

Note that the random variable a_{t-1} is generated according to $\pi_b(a_{t-1}|x^S_{1:t-1}, x^A_{1:t-1}, a_{1:t-2})$, which is conditioned on $x^S_{1:t-1}$, $x^A_{1:t-1}$ and $a_{1:t-2}$. Therefore, knowing a_{t-1} does not provide additional information regarding θ^S_{t-1} and θ^A_{t-1} once $x^S_{1:t-1}$, $x^A_{1:t-1}$ and $a_{1:t-2}$ are known, which leads to the following relation:

$$p(\theta^S_{t-1}, \theta^A_{t-1}|x^S_{1:t-1}, x^A_{1:t-1}, a_{1:t-1}) = p(\theta^S_{t-1}, \theta^A_{t-1}|x^S_{1:t-1}, x^A_{1:t-1}, a_{1:t-2}) \quad (19)$$

Substituting the above expression into (18), we obtain

$$\begin{aligned}
p(\theta_t^S,&\theta_t^A, x_t^S, x_t^A | x_{1:t-1}^S, x_{1:t-1}^A, a_{1:t-1}) \\
&= \int p(x_t^S|\theta_t^S, \Phi_S) p(\theta_t^S|\alpha_t^S) p(\alpha_t^S|\theta_{t-1}^S, \theta_{t-1}^{a_{t-1}}, W_{SS}, W_{SA}) \\
&\quad \times p(x_t^A|\theta_t^A, \Phi_A) p(\theta_t^A|\alpha_t^A) p(\alpha_t^A|\theta_{t-1}^S, \theta_{t-1}^{a_{t-1}}, W_{AS}, W_{AA}) \\
&\quad \times p(\theta_{t-1}^S, \theta_{t-1}^A | x_{1:t-1}^S, x_{1:t-1}^A, a_{1:t-1}) d\alpha_t^S d\alpha_t^A d\theta_{t-1}^S d\theta_{t-1}^A \\
&= \int p(x_t^S|\theta_t^S, \Phi_S) p(\theta_t^S|\alpha_t^S) p(\alpha_t^S|\theta_{t-1}^S, \theta_{t-1}^{a_{t-1}}, W_{SS}, W_{SA}) \\
&\quad \times p(x_t^A|\theta_t^A, \Phi_A) p(\theta_t^A|\alpha_t^A) p(\alpha_t^A|\theta_{t-1}^S, \theta_{t-1}^{a_{t-1}}, W_{AS}, W_{AA}) \\
&\quad \times p(\theta_{t-1}^S, \theta_{t-1}^A | x_{1:t-1}^S, x_{1:t-1}^A, a_{1:t-2}) d\alpha_t^S d\alpha_t^A d\theta_{t-1}^S d\theta_{t-1}^A \\
&\overset{(a)}{=} \int p(x_t^S|\theta_t^S, \Phi_S) p(\theta_t^S|\alpha^S(\theta_{t-1}^S, \theta_{t-1}^{a_{t-1}}, W_{SS}, W_{SA})) \\
&\quad \times p(x_t^A|\theta_t^A, \Phi_A) p(\theta_t^A|\alpha^A(\theta_{t-1}^S, \theta_{t-1}^{a_{t-1}}, W_{AS}, W_{AA})) \\
&\quad \times p(\theta_{t-1}^S, \theta_{t-1}^A | x_{1:t-1}^S, x_{1:t-1}^A, a_{1:t-2}) d\theta_{t-1}^S d\theta_{t-1}^A \\
&\overset{(b)}{\approx} p(x_t^S|\theta_t^S, \Phi_S) p\big(\theta_t^S|\alpha^S(\hat{\theta}_{t-1}^S, \hat{\theta}_{t-1}^{a_{t-1}}, W_{SS}, W_{SA})\big) \\
&\quad \times p(x_t^A|\theta_t^A, \Phi_A) p\big(\theta_t^A|\alpha^A(\hat{\theta}_{t-1}^S, \hat{\theta}_{t-1}^{a_{t-1}}, W_{AA}, W_{AS})\big)
\end{aligned} \tag{20}$$

where step (a) uses the fact that the probability distribution of α_t^S and α_t^A are Dirac delta functions and step (b) samples the integral with MAP estimates of θ_t^S and θ_t^A. Therefore, substituting (20) into (17), we get

$$\begin{aligned}
(\hat{\theta}_t^S, \hat{\theta}_t^A) &\approx \arg\max_{(\theta_t^S, \theta_t^A)} \Big\{ p(x_t^S|\theta_t^S, \Phi_S) p\big(\theta_t^S|\alpha^S(\hat{\theta}_{t-1}^S, \hat{\theta}_{t-1}^{a_{t-1}}, W_{SS}, W_{SA})\big) \\
&\qquad\qquad \times p(x_t^A|\theta_t^A, \Phi_A) p\big(\theta_t^A|\alpha^A(\hat{\theta}_{t-1}^S, \hat{\theta}_{t-1}^{a_{t-1}}, W_{AS}, W_{AA})\big) \Big\} \\
&= \arg\max_{(\theta_t^S, \theta_t^A)} \Big\{ \ln p(x_t^S|\theta_t^S, \Phi_S) + \ln p\big(\theta_t^S|\alpha^S(\hat{\theta}_{t-1}^S, \hat{\theta}_{t-1}^{a_{t-1}}, W_{SS}, W_{SA})\big) \\
&\qquad\qquad + \ln p(x_t^A|\theta_t^A, \Phi_A) + \ln p\big(\theta_t^A|\alpha^A(\hat{\theta}_{t-1}^S, \hat{\theta}_{t-1}^{a_{t-1}}, W_{AS}, W_{AA})\big) \Big\}
\end{aligned} \tag{21}$$

Using the definition of these probability distributions, we can show that the above MAP estimation problem can be decomposed into

$$\hat{\theta}_t^S = \arg\max_{\theta_t^S} \Big[\ln p(x_t^S|\theta_t^S, \Phi_S) + \ln p\big(\theta_t^S|\alpha^S(\hat{\theta}_{t-1}^S, \hat{\theta}_{t-1}^{a_{t-1}}, W_{SS}, W_{SA})\big) \Big] \tag{22}$$

$$\begin{aligned}
\hat{\theta}_t^a &= \arg\max_{\theta_t^a} \Big[\ln p(x_t^a|\theta_t^a, \Phi_A) + \ln p\big(\theta_t^a|\alpha^A(\hat{\theta}_{t-1}^S, \hat{\theta}_{t-1}^{a_{t-1}}, W_{AS}, W_{AA})\big) \Big] \\
&\qquad a = 1, \ldots, A_t
\end{aligned} \tag{23}$$

Note that the approximate MAP inference of θ_t^S and θ_t^a ($a = 1, \ldots, A_t$) is completely decoupled into independent optimization problems, which could be solved by mirror descent separately. Therefore, we complete our proof of Proposition 1.

B Approximation of the learning objective function

In this appendix, we show that the learning objective function (3) can be approximated by the cost function (7). For convenience, we repeat (3) below:

$$\max_{\Theta} \Big\{ \ln p(\Theta) + \sum_{i=1}^{M} \ln p(r_{1:T_i}|x_{1:T_i}^S, x_{1:T_i}^A, a_{1:T_i}, \Theta) \Big\} \tag{24}$$

An important step of our derivation is to write $p(r_{1:T_i}|x_{1:T_i}^S, x_{1:T_i}^A, a_{1:T_i}, \Theta)$ as an expression of probabilities for each time step t. We begin by examining the joint probability $p(x_{1:T_i}^S, x_{1:T_i}^A a_{1:T_i}, r_{1:T_i}|\Theta)$:

$$p(x_{1:T_i}^S, x_{1:T_i}^A, a_{1:T_i}, r_{1:T_i}|\Theta)$$

$$
\begin{aligned}
&= \prod_{t=1}^{T_i} p(x_t^S, x_t^A, a_t, r_t | x_{1:t-1}^S, x_{1:t-1}^A, a_{1:t-1}, r_{1:t-1}, \Theta) \\
&= \prod_{t=1}^{T_i} p(x_t^S, x_t^A | x_{1:t-1}^S, x_{1:t-1}^A, a_{1:t-1}, r_{1:t-1}, \Theta) \times \pi(a_t | x_{1:t}^S, x_{1:t}^A, a_{1:t-1}, r_{1:t-1}) \\
&\quad \times p(r_t | x_{1:t}^S, x_{1:t}^A, a_{1:t}, r_{1:t-1}, \Theta) \\
&= \prod_{t=1}^{T_i} p(x_t^S, x_t^A | x_{1:t-1}^S, x_{1:t-1}^A, a_{1:t-1}, \Theta) \pi(a_t | x_{1:t}^S, x_{1:t}^A, a_{1:t-1}) p(r_t | x_{1:t}^S, x_{1:t}^A, a_{1:t}, \Theta) \qquad (25)
\end{aligned}
$$

where the last step uses the fact that the behavior policy for exploring the environment does not depend on the current model parameter to be optimized and the fact that the intermediate rewards are known deterministic quantities except the terminal reward. Likewise, we can also get

$$
\begin{aligned}
p(x_{1:T_i}^S, x_{1:T_i}^A, a_{1:T_i} | \Theta) &= \prod_{t=1}^{T_i} p(x_t^S, x_t^A, a_t | x_{1:t-1}^S, x_{1:t-1}^A, a_{1:t-1}, \Theta) \\
&= \prod_{t=1}^{T_i} p(x_t^S, x_t^A | x_{1:t-1}^S, x_{1:t-1}^A, a_{1:t-1}, \Theta) \pi(a_t | x_{1:t}^S, x_{1:t}^A, a_{1:t-1}) \qquad (26)
\end{aligned}
$$

Dividing (25) by the above expression leads to

$$
p(r_{1:T_i} | x_{1:T_i}^S, x_{1:T_i}^A, a_{1:T_i}, \Theta) = \frac{p(x_{1:T_i}^S, x_{1:T_i}^A, a_{1:T_i}, r_{1:T_i} | \Theta)}{p(x_{1:T_i}^S, x_{1:T_i}^A, a_{1:T_i} | \Theta)} = \prod_{t=1}^{T_i} p(r_t | x_{1:t}^S, x_{1:t}^A, a_{1:t}, \Theta) \qquad (27)
$$

We now examine the term inside the product of (27). Unfortunately, the exact expression is not tractable as it requires to marginalize out all the latent variables, which cannot be done in closed-form. Instead, we develop approximate expressions for it. Note that

$$
\begin{aligned}
p(r_t | x_{1:t}^S, x_{1:t}^A, a_{1:t}, \Theta) &= \int p(r_t | \theta_t^S, \theta_t^{a_t}, U) p(\theta_t^S, \theta_t^A | x_{1:t}^S, x_{1:t}^A, a_{1:t}, \Theta) d\theta_t^S d\theta_t^A \\
&\overset{(a)}{=} \int p(r_t | \theta_t^S, \theta_t^{a_t}, U) p(\theta_t^S, \theta_t^A | x_{1:t}^S, x_{1:t}^A, a_{1:t-1}, \Theta) d\theta_t^S d\theta_t^A \\
&= \mathbb{E}_{\theta_t^S, \theta_t^{a_t} | x_{1:t}^S, x_{1:t}^A, a_{1:t-1}} \big[p(r_t | \theta_t^S, \theta_t^{a_t}, U) \big] \\
&\overset{(b)}{\approx} p(r_t | \hat{\theta}_t^S, \hat{\theta}_t^{a_t}, U) \qquad (28)
\end{aligned}
$$

where step (a) uses the fact that the action a_t is generated only by $x_{1:t}^S$, $x_{1:t}^A$ and $a_{1:t-1}$, and step (b) approximate the expectation by sampling it with the MAP estimate. Substituting (28) into (27), we get

$$
p(r_{1:T_i} | x_{1:T_i}^S, x_{1:T_i}^A, a_{1:T_i}, \Theta) = \prod_{t=1}^{T_i} \mathbb{E}_{\theta_t^S, \theta_t^{a_t} | x_{1:t}^S, x_{1:t}^A, a_{1:t-1}} \big[p(r_t | \theta_t^S, \theta_t^{a_t}, U) \big] \approx \prod_{t=1}^{T_i} p(r_t | \hat{\theta}_t^S, \hat{\theta}_t^{a_t}, U) \qquad (29)
$$

Substituting (29) into (24), we obtain

$$
\max_{\Theta} \Big\{ \ln p(\Theta) + \sum_{i=1}^{M} \sum_{t=1}^{T_i} \ln p(r_t | \hat{\theta}_t^S, \hat{\theta}_t^{a_t}, U) \Big\} \qquad (30)
$$

Recalling from (1) that, conditioned on $\hat{\theta}_t^S$ and $\hat{\theta}_t^{a_t}$, r_t is a Gaussian random variable with mean $\mu_r(\hat{\theta}_t^S, \hat{\theta}_t^{a_t}, U)$ and variance σ_r^2, we can express $p(r_t | \hat{\theta}_t^S, \hat{\theta}_t^{a_t}, U)$ as:

$$
\begin{aligned}
p(r_t | \hat{\theta}_t^S, \hat{\theta}_t^{a_t}, U) &= \frac{1}{\sqrt{2\pi\sigma_r^2}} \exp\left(-\frac{1}{2\sigma_r^2} (r_t - \mu_r(\hat{\theta}_t^S, \hat{\theta}_t^{a_t}, U))^2 \right) \\
&\overset{(a)}{=} \frac{1}{\sqrt{2\pi\sigma_r^2}} \exp\left(-\frac{1}{2\sigma_r^2} (r_t - Q(\hat{\theta}_t^S, \hat{\theta}_t^{a_t}) + \gamma \cdot \mathbb{E}\big[\max_{a_{t+1}} Q(\theta_{t+1}^S, \theta_{t+1}^{a_{t+1}}) | \hat{\theta}_t^S, \hat{\theta}_t^{a_t} \big])^2 \right) \\
&\qquad (31)
\end{aligned}
$$

where step (a) substituted (6). Substituting (31) into (30), we obtain

$$\min_{\Theta}\left\{-\ln p(\Theta)+\sum_{i=1}^{M}\sum_{t=1}^{T_i}\frac{1}{2\sigma_r^2}\left\|r_t-Q(\hat{\theta}_t^S,\hat{\theta}_t^{a_t})+\gamma\cdot\mathbb{E}\left[\max_{a_{t+1}}Q(\theta_{t+1}^S,\theta_{t+1}^{a_{t+1}})|\hat{\theta}_t^S,\hat{\theta}_t^{a_t}\right]\right\|^2\right\} \tag{32}$$

where we have dropped some constant terms. Introduce

$$d_t=\begin{cases}r_t+\gamma\cdot\mathbb{E}_{\theta_{t+1}^S,\theta_{t+1}^b|\hat{\theta}_t^S,\hat{\theta}_t^{a_t}}[\max_b Q(\theta_{t+1}^S,\theta_{t+1}^b)] & t<T_i\\ r_{T_i} & t=T_i\end{cases} \tag{33}$$

Then we can write (32) as

$$\min_{\Theta}\left\{-\ln p(\Theta)+\sum_{i=1}^{M}\sum_{t=1}^{T_i}\frac{1}{2\sigma_r^2}\left[d_t-Q(\hat{\theta}_t^S,\hat{\theta}_t^{a_t})\right]^2\right\} \tag{34}$$

A remaining problem is that d_t has a conditional expectation with respect to θ_{t+1}^S and $\theta_{t+1}^{a_{t+1}}$. First, note that we can have the following approximation:

$$\mathbb{E}_{\theta_t^S,\theta_t^{a_t}|x_{1:t}^S,x_{1:t}^A,a_{1:t-1}}\left\{\mathbb{E}_{\theta_{t+1}^S,\theta_{t+1}^b|\theta_t^S,\theta_t^{a_t}}[\max_b Q(\theta_{t+1}^S,\theta_{t+1}^b)]\right\}\approx\mathbb{E}_{\theta_{t+1}^S,\theta_{t+1}^b|\hat{\theta}_t^S,\hat{\theta}_t^{a_t}}[\max_b Q(\theta_{t+1}^S,\theta_{t+1}^b)]$$

where we sample the outer conditional expectation by the MAP samples $\hat{\theta}_t^S$ and $\hat{\theta}_t^{a_t}$. Then, we have

$$\begin{aligned}&\mathbb{E}_{\theta_{t+1}^S,\theta_{t+1}^b|\hat{\theta}_t^S,\hat{\theta}_t^{a_t}}[\max_b Q(\theta_{t+1}^S,\theta_{t+1}^b)]\\&\qquad\approx\mathbb{E}_{\theta_t^S,\theta_t^{a_t}|x_{1:t}^S,x_{1:t}^A,a_{1:t-1}}\left\{\mathbb{E}_{\theta_{t+1}^S,\theta_{t+1}^b|\theta_t^S,\theta_t^{a_t}}[\max_b Q(\theta_{t+1}^S,\theta_{t+1}^b)]\right\}\\&\qquad=\mathbb{E}_{\theta_{t+1}^S,\theta_{t+1}^b|x_{1:t}^S,x_{1:t}^A,a_{1:t-1}}\left\{\max_b Q(\theta_{t+1}^S,\theta_{t+1}^b)\right\}\\&\qquad\approx\max_b Q(\hat{\theta}_{t+1}^S,\hat{\theta}_{t+1}^b)\end{aligned} \tag{35}$$

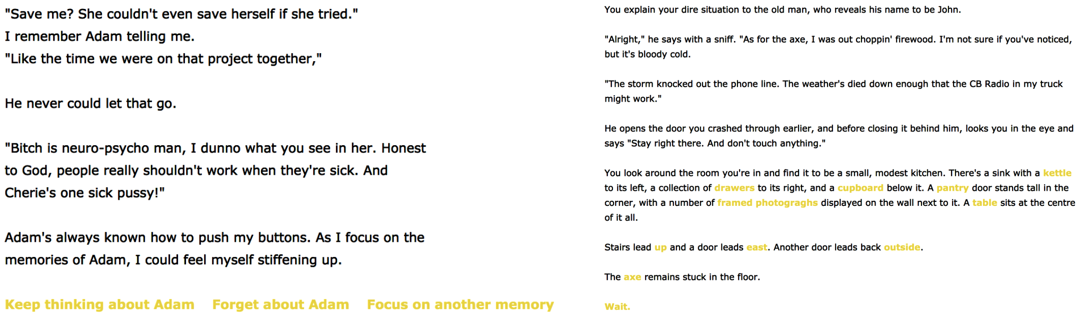
In summary, we have the approximation:

$$d_t=\begin{cases}r_t+\gamma\max_b Q(\hat{\theta}_{t+1}^S,\hat{\theta}_{t+1}^b) & t<T_i\\ r_t & t=T_i\end{cases} \tag{36}$$

which completes our proof.

C Introduction of the two text games

In Figure 3, we show two screenshots of the two text games used in this paper. The first game belongs to choice-based game, where the feasible actions at each time are listed separately as candidate choices. And the second game is a mix between choice-based and hypertext-based game (where the actions are embedded in the observation text as substrings with hyperlinks). The action spaces of both games are defined by natural languages and the feasible actions change over time, which is a setting that Q-LDA is designed for. This setting was believed to be more challenging than the parser-based text games in [19], which accepts a (small) fixed set of pre-defined typed-in commands (e.g., "eat apple", "get key"). Therefore, we do not consider parser-based game and will focus on the choice-based and hyperlink-based games. To be self-contained, we include the description of the two text games ("Saving John" and "Machine of Death") from [11] (Tables 3, 4, and 5). Table 3 gives the basic statistics of the two text games, Tables 4-5 give the rewards for different endings of the two games. In Table 6, we give an example text flow when playing "Machine of Death". In addition, the number of conversation turns (number of steps) per episode is 10-30 for "SavingJohn" and is 10-200 for "Machine of Death". When the training converges, the length is around 7 for "SavingJohn" and is between 10-20 for "Machine of Death". For more details, the readers are referred to [11] and its supplementary material.



"Save me? She couldn't even save herself if she tried."
I remember Adam telling me.
"Like the time we were on that project together,"

He never could let that go.

"Bitch is neuro-psycho man, I dunno what you see in her. Honest to God, people really shouldn't work when they're sick. And Cherie's one sick pussy!"

Adam's always known how to push my buttons. As I focus on the memories of Adam, I could feel myself stiffening up.

Keep thinking about Adam **Forget about Adam** **Focus on another memory**

(a) "Saving John"

You explain your dire situation to the old man, who reveals his name to be John.

"Alright," he says with a sniff. "As for the axe, I was out choppin' firewood. I'm not sure if you've noticed, but it's bloody cold.

"The storm knocked out the phone line. The weather's died down enough that the CB Radio in my truck might work."

He opens the door you crashed through earlier, and before closing it behind him, looks you in the eye and says "Stay right there. And don't touch anything."

You look around the room you're in and find it to be a small, modest kitchen. There's a sink with a **kettle** to its left, a collection of **drawers** to its right, and a **cupboard** below it. A **pantry** door stands tall in the corner, with a number of **framed photograghs** displayed on the wall next to it. A **table** sits at the centre of it all.

Stairs lead **up** and a door leads **east**. Another door leads back **outside**.

The **axe** remains stuck in the floor.

Wait.

(b) "Machine of Death"

Figure 3: The user interface of the two text games used for evaluation.

Table 3: Statistics for the games "Saving John" and and "Machine of Death".

Game	Saving John	Machine of Death
Text game type	Choice	Choice & Hypertext
Vocab size	1762	2258
Action vocab size	171	419
Avg. words/description	76.67	67.80
State transitions	Deterministic	Stochastic
# of states (underlying)	≥ 70	≥ 200

D Additional experiment results

In Table 6, we show the snapshots of the text game "Machine of Death" at three different time steps: beginning ($t = 2$), in the middle ($t = 8$), and approaching the end ($t = 15$). In the table, we show the observation texts and the action texts for all the actions. The action texts highlighted in boldface correspond to the selected action. Below, we show the value of the matrix U in the learned model parameter on "Machine of Death" task:

$$U = \begin{bmatrix} 1.2014 & \mathbf{39.5233} & 20.7054 & 12.2296 \\ 22.1366 & 12.4041 & 1.3726 & -0.1604 \\ 2.5195 & 4.8452 & 4.1210 & 1.9419 \\ 5.3332 & 8.3989 & 13.3208 & 4.1159 \end{bmatrix} \tag{37}$$

E Implementation details

E.1 Details of the inference algorithm

As we discussed in the paper, we use mirror descent algorithm to perform MAP inference. In Algorithm 2, the MAP inference is implemented with constant step-size δ. In practice, we found that it converges faster if we use adaptive step-size determined by line search. In Algorithm 3, we include the mirror descent inference algorithm with line search.

E.2 Details of the learning algorithm

In Figure 4, we visualize the computation graph of the inference step time t, which illustrates the recursive inference steps in Algorithm 2 (or Algorithm 3). We observe that the recursive inference process could be interpreted as a recurrent neural network (RNN) with the following special structures. The topic distributions θ_t^S and $\{\theta_t^a\}$ can be viewed as $A_t + 1$ (time-varying) sets of hidden units that satisfy probabilistic simplex constraints, which are computed by $A_t + 1$ feedforward mirror descent networks (Figure 4(b)) from the input vectors x_t^S and $\{x_t^a\}$ and the Dirichlet parameters $\hat{\alpha}_t^S$ and $\hat{\alpha}_t^A$. The recurrent links from the current hidden units (θ_t^S and $\{\theta_t^A\}$) to the next ones are through

Table 4: Final rewards defined for the text game “Saving John"

Reward	Endings (partially shown)
-20	Suspicion fills my heart and I scream. Is she trying to kill me? I don’t trust her one bit...
-10	Submerged under water once more, I lose all focus...
0	Even now, she’s there for me. And I have done nothing for her...
10	Honest to God, I don’t know what I see in her. Looking around, the situation’s not so bad...
20	Suddenly I can see the sky... I focus on the most important thing - that I’m happy to be alive.

Table 5: Final rewards for the text game “Machine of Death.” Scores are assigned according to whether the character survives, how the friendship develops, and whether he overcomes his fear.

Reward	Endings (partially shown)
-20	You spend your last few moments on Earth lying there, shot through the heart, by the image of Jon Bon Jovi.
-20	you hear Bon Jovi say as the world fades around you.
-20	As the screams you hear around you slowly fade and your vision begins to blur, you look at the words which ended your life.
-10	You may be locked away for some time.
-10	Eventually you’re escorted into the back of a police car as Rachel looks on in horror.
-10	Fate can wait.
-10	Sadly, you’re so distracted with looking up the number that you don’t notice the large truck speeding down the street.
-10	All these hiccups lead to one grand disaster.
10	Stay the hell away from me! She blurts as she disappears into the crowd emerging from the bar.
20	You can’t help but smile.
20	Hope you have a good life.
20	Congratulations!
20	Rachel waves goodbye as you begin the long drive home. After a few minutes, you turn the radio on to break the silence.
30	After all, it’s your life. It’s now or never. You ain’t gonna live forever. You just want to live while you’re alive.

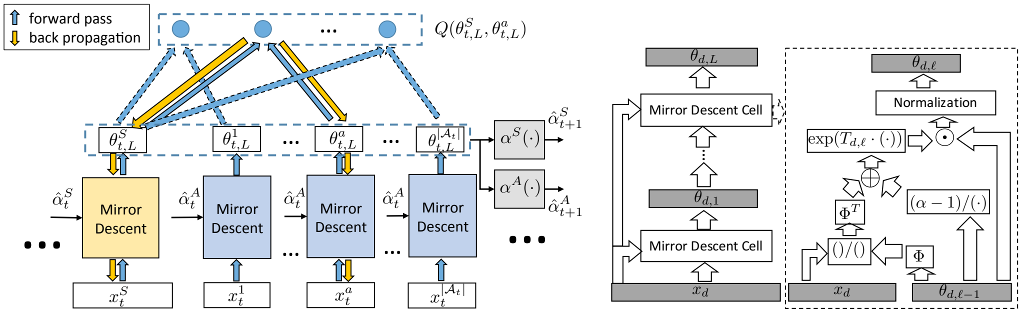
the Dirichlet parameters computed via (11). Furthermore, there are $A_t + 1$ output units, which are pairwise bilinear functions of θ_t^S and θ_t^a for each $a = 1, \dots, A_t$. Therefore, the entire inference process could be interpreted as using a special structured RNN to approximate the Q-function in reinforcement learning. From this perspective, our work is related to DRRN [11] in that both of them use separate embedding vectors for the state and action texts and that they both use bilinear functions to map the embeddings into a Q-value. However, our work uses a special structured RNN to embed the input texts into their respective representation vectors while DRRN uses standard feedforward DNN. Our work is also related to the deep recurrent Q-network (DRQN) [10], which uses standard RNN (rather than the special structured RNN in our case) to approximate the Q-function to address the partial observability problem in reinforcement learning. Different from our model, the DRQN only works in the case with a fixed action space and could not handle the situation where the actions are described by natural languages. Finally, the above special RNN structures are designed from the generative model of Q-LDA, while both DRRN and DRQN are constructed as a black-box model for function approximation in Q-learning. This enables Q-LDAto be more interpretable during the decision making process.

E.3 Details of the experiments

The softmax action selection rule for behavior policy can be written as $\pi_b^m(a_t|x_{1:t}^S, x_{1:t}^A, a_{1:t-1}) \propto \exp[\frac{1}{\tau}Q(\hat{\theta}_t^S, \hat{\theta}_t^a)]$ for all $a_t = 1, \dots, A_t$, where τ is a temperature parameter that controls the sharpness of the softmax. $Q(\hat{\theta}_t^S, \hat{\theta}_t^a)$ is computed according to Algorithm 2 using the model parameter Θ_{m-1} from the previous experience replay. That is, at the exploration stage of each m-th experience replay, the behavior policy $\pi_b^m(\cdot)$ is parameterized by Θ_{m-1}, which will be fixed during the exploration stage. With this, the behavior policy π_b^m can be viewed as independent of the model parameter Θ to be optimized in the m-th replay. During the exploration stage, we will terminate the episode

Table 6: Snapshots of game observation and actions at different times for “Machine of Death”

Time step	$t = 2$	$t = 8$	$t = 15$
Observation text (partially shown)	You approach The Machine, which has the very charming street name of The Machine of Death. The device has only been around for a few years, but it’s already hard to imagine a world without it, as it completely reshaped it, creating a culture of death. ... You never did get yourself tested. Maybe today is the day.	You decided that you don’t need a firearm. You already have a set of guns sitting below your shoulders, after all. ... You take a moment to relish the drunken merriment. Then, in a corner, you see rock idol Jon Bon Jovi.	‘It makes me feel normal,’ she admits. ... she says with a laugh. ’I’m going to go let Bonny have a run. You better be careful around him,’ she adds with a mischievous grin. ...... People call Rachel the crazy one, but you’re the one carrying a gun around in case you bump into members of Bon Jovi!
Action texts (selected action in **bold**)	[1] Return your eyes to the mall. [2] A slip of paper is stuck to the side of the Machine. Examine it. [3] Stand back and watch people use the Machine. **[4] Insert a coin.**	**[1] Duck! DUCK!** [2] Tackle him to the ground! [3] Ignore him.	**[1] It’s time to let it go. Dismantle the gun to the best of your ability and get rid of it.** [2] Things could have gone a lot worse tonight. Who knows when I’ll need that gun to survive!



(a) The computational graph at each time step t

(b) The computation graph of mirror descent

Figure 4: (a) Feedforward computation graph for the model in Figure 1. We use the same blue color for the mirror descent graphs on the action texts to represent that they share the same model parameter Φ_A. The mirror descent graph for the state text uses a different yellow color to imply that it uses a different model parameter Φ_S. (b) The mirror descent graph in (a), where Φ is either Φ_S or Φ_A.

when its length exceeds 100 in “Saving John” and will terminate the episode when the length exceeds 500 in “Machine of Death”.

For learning algorithm, we use RMSProp to adaptively adjust the learning rate for each model parameter, with exponential decaying parameter 0.999. The overall learning rates are chosen to be:

- $\mu_U = 1.0$ for both games
- In “Saving John”, $\mu_{\Phi_S} = \mu_{\Phi_A} = 10^{-4}$ when the number of topics is 20 and 50, and 10^{-5} when the number of topics is 100. In “Machine of Death”, $\mu_{\Phi_S} = \mu_{\Phi_A} = 10^{-4}$ when the number of topics is 20 and 50, and 10^{-6} when the number of topics is 100.
- In “Saving John”, the learning rates for all $W_{SS}, W_{SA}, W_{AS}, W_{AA}$ are chosen to be 10^{-2}. In “Machine of Death”, they are chosen to be 10^{-2} when the number of topics is 20 and 50, and 10^{-3} when the number of topics is 100.

Algorithm 3 Inference with Mirror-Descent over one episode (with line search)

```
for $t = 0, \dots, T_i$ do
  if $t = 0$ then
    $\hat{\alpha}^S_0 = \alpha^S_0$ and $\hat{\alpha}^A_0 = \alpha^A_0$.
  else
    $\hat{\alpha}^S_t = \alpha^S(\theta^S_{t-1,L}, \theta^{a_{t-1}}_{t-1,L}, W_S)$ and $\hat{\alpha}^A_t = \alpha^A(\theta^S_{t-1,L}, \theta^{a_{t-1}}_{t-1,L}, W_A)$
  end if
  Initialization: $\theta^S_{t,0} = \frac{1}{K}\mathbb{1}$ and $T_{t,0}$.
  for $\ell = 1, \dots, L$ do
    $T_{t,\ell} = T_{t,\ell-1}/\eta$, where $0 < \eta < 1$ (e.g., $\eta = 0.5$).
    while 1 do
      $\theta^S_{t,\ell} = \frac{1}{C_\theta} \cdot \theta^S_{t,\ell-1} \odot \exp\left(T_{t,\ell}\left[\Phi^T \frac{x_t}{\Phi\theta^S_{t,\ell-1}} + \frac{\alpha - \mathbb{1}}{\theta^S_{t,\ell-1}}\right]\right)$
      if $f^S(\theta^S_{t,\ell}) > f^S(\theta^S_{t,\ell-1}) + [\nabla_{\theta^S_t} f^S(\theta^S_{t,\ell-1})]^T(\theta^S_{t,\ell} - \theta^S_{t,\ell-1}) + \frac{1}{T_{t,\ell}}\Psi(\theta^S_{t,\ell}, \theta^S_{t,\ell-1})$ then
        $T_{t,\ell} \leftarrow \eta \cdot T_{t,\ell}$
      else
        break
      end if
    end while
  end for
  for $a = 1, \dots, |\mathcal{A}_t|$ do
    Initialization: $\theta^a_{t,0} = \frac{1}{K}\mathbb{1}$ and $T_{t,0}$.
    for $\ell = 1, \dots, L$ do
      $T_{t,\ell} = T_{t,\ell-1}/\eta$, where $0 < \eta < 1$ (e.g., $\eta = 0.5$).
      while 1 do
        $\theta^a_{t,\ell} = \frac{1}{C_\theta} \cdot \theta^a_{t,\ell-1} \odot \exp\left(T_{t,\ell}\left[\Phi^T \frac{x_t}{\Phi\theta^a_{t,\ell-1}} + \frac{\alpha - \mathbb{1}}{\theta^a_{t,\ell-1}}\right]\right)$
        if $f^a(\theta^a_{t,\ell}) > f^a(\theta^a_{t,\ell-1}) + [\nabla_{\theta^a_t} f^a(\theta^a_{t,\ell-1})]^T(\theta^a_{t,\ell} - \theta^a_{t,\ell-1}) + \frac{1}{T_{t,\ell}}\Psi(\theta^a_{t,\ell}, \theta^a_{t,\ell-1})$ then
          $T_{t,\ell} \leftarrow \eta \cdot T_{t,\ell}$
        else
          break
        end if
      end while
    end for
  end for
  Output: $\hat{\theta}^S_t = \theta^S_{t,L}$, $\hat{\theta}^a_t = \theta^a_{t,L}$, and

      $\mathbb{E}\left\{Q(\theta^S_t, \theta^a_t) | x^S_{1:t}, x^A_{1:t}, a_{1:t-1}\right\} \approx (\theta^a_{t,L})^T U \theta^S_{t,L}, \quad a = 1, \dots, |\mathcal{A}_t|$      (38)

end for
```

- The initial Dirichlet parameters $\alpha^S_1 = \alpha^A_1 = 1.001$. The rest of the α^S_t and α^A_t is dynamically determined by the model itself and could be less than or greater than one. $\beta_S = \beta_A = 1.001$.
- The discount factor $\gamma = 0.9$, same as the choice in [11].
- $\sigma^2_r = 3.2$.
- We clip the gradient of Φ_S and Φ_A with threshold 10^4, and we clip the gradients of $W_{SS}, W_{SA}, W_{AS}, W_{AA}$ with threshold 100.

E.4 Derivation of the Back Propagation Formula

In this appendix, we derive the back propagation formula for learning the LDA model from feedbacks. The cost function of the problem can be expressed as

$$L(\Theta) = \sum_{i=1}^{N} l_i(\Theta) - \ln p(\Theta) = N\left(\frac{1}{N}\sum_{i=1}^{N} l_i(\Theta) - \frac{1}{N}\ln p(\Theta)\right) \tag{39}$$

where

$$\begin{aligned} l_i(\Theta) &\triangleq \sum_{t=1}^{T_i} \frac{1}{2\sigma_r^2} \|d_t - q_t\|^2 \\ q_t &\triangleq (\theta_{t,L}^{a_t})^T U \theta_{t,L}^S \\ d_t &\triangleq r_t + \gamma \cdot \max_b Q(\theta_{t+1,L}^S, \theta_{t+1,L}^b) \end{aligned}$$

The gradients of $\ln p(\Theta)$ with respect to model parameters are relatively easy. Below, we mainly focus on deriving the gradient of $l_i(\Theta)$. We summarize the result before the derivation. Then, we have

$$\Delta q_t = -\frac{1}{\sigma_r^2}(d_t - q_t) \tag{40}$$

$$\frac{\partial l_i}{\partial U} = \sum_{t=1}^{T_i} \Delta q_t \cdot \theta_{t,L}^{a_t} [\theta_{t,L}^S]^T \tag{41}$$

$$\frac{\partial l_i}{\partial \Phi_S} = \sum_{t=1}^{T_i} \sum_{\ell=1}^{L} T_{t,\ell}^S \left\{ \frac{x_t^S}{\Phi_S \theta_{t,\ell-1}^S} (\theta_{t,\ell}^S \odot \xi_{t,\ell}^S)^T - \left[\Phi_S (\theta_{t,\ell}^S \odot \xi_{t,\ell}^S) \odot \frac{x_t^S}{(\Phi_S \theta_{t,\ell-1}^S)^2} \right] [\theta_{t,\ell-1}^S]^T \right\} \tag{42}$$

$$\frac{\partial l_i}{\partial \Phi_A} = \sum_{t=1}^{T_i} \sum_{\ell=1}^{L} T_{t,\ell}^{a_t} \left\{ \frac{x_t^{a_t}}{\Phi_A \theta_{t,\ell-1}^{a_t}} (\theta_{t,\ell}^{a_t} \odot \xi_{t,\ell}^{a_t})^T - \left[\Phi_A (\theta_{t,\ell}^{a_t} \odot \xi_{t,\ell}^{a_t}) \odot \frac{x_t^{a_t}}{(\Phi_A \theta_{t,\ell-1}^{a_t})^2} \right] [\theta_{t,\ell-1}^{a_t}]^T \right\} \tag{43}$$

$$\frac{\partial l_i}{\partial W_{SS}} = \sum_{t=2}^{T_i} (D_t^S \Delta \alpha_t^S)(\theta_{t-1,L}^S)^T \tag{44}$$

$$\frac{\partial l_i}{\partial W_{SA}} = \sum_{t=2}^{T_i} (D_t^S \Delta \alpha_t^S)(\theta_{t-1,L}^{a_{t-1}})^T \tag{45}$$

$$\frac{\partial l_i}{\partial W_{AS}} = \sum_{t=2}^{T_i} (D_t^A \Delta \alpha_t^A)(\theta_{t-1,L}^S)^T \tag{46}$$

$$\frac{\partial l_i}{\partial W_{AA}} = \sum_{t=2}^{T_i} (D_t^A \Delta \alpha_t^A)(\theta_{t-1,L}^{a_{t-1}})^T \tag{47}$$

$$\begin{aligned} \xi_{t,\ell-1}^S = (I - \mathbb{1}[\theta_{t,\ell-1}^S]^T) \Bigg\{ & \frac{\theta_{t,\ell}^S \odot \xi_{t,\ell}^S}{\theta_{t,\ell-1}^S} \\ & - T_{t,\ell}^S \left[\Phi_S^T \mathrm{diag}\left(\frac{x_t^S}{(\Phi_S \theta_{t,\ell-1}^S)^2} \right) \Phi_S + \mathrm{diag}\left(\frac{\alpha_t^S - \mathbb{1}}{(\theta_{t,\ell-1}^S)^2} \right) \right] (\theta_{t,\ell}^S \odot \xi_{t,\ell}^S) \Bigg\} \end{aligned} \tag{48}$$

$$\begin{aligned} \xi_{t,\ell-1}^{a_t} = (I - \mathbb{1}[\theta_{t,\ell-1}^{a_t}]^T) \Bigg\{ & \frac{\theta_{t,\ell}^{a_t} \odot \xi_{t,\ell}^{a_t}}{\theta_{t,\ell-1}^{a_t}} \\ & - T_{t,\ell}^{a_t} \left[\Phi_A^T \mathrm{diag}\left(\frac{x_t^{a_t}}{(\Phi_A \theta_{t,\ell-1}^{a_t})^2} \right) \Phi_A + \mathrm{diag}\left(\frac{\alpha_t^A - \mathbb{1}}{(\theta_{t,\ell-1}^{a_t})^2} \right) \right] (\theta_{t,\ell}^{a_t} \odot \xi_{t,\ell}^{a_t}) \Bigg\} \end{aligned} \tag{49}$$

$$\xi_{t,L}^S = [I - \mathbb{1}(\theta_{t,L}^S)^T] \Big(U^T \theta_{t,L}^{a_t} \Delta q_t + W_{SS}^T D_{t+1}^S \Delta \alpha_{t+1}^S + W_{AS}^T D_{t+1}^A \Delta \alpha_{t+1}^A \Big) \tag{50}$$

$$\xi_{t,L}^{a_t} = [I - \mathbb{1}(\theta_{t,L}^{a_t})^T] \Big(U \theta_{t,L}^S \Delta q_t + W_{SA}^T D_{t+1}^S \Delta \alpha_{t+1}^S + W_{AA}^T D_{t+1}^A \Delta \alpha_{t+1}^A \Big) \tag{51}$$

$$\Delta \alpha_t^S = \sum_{\ell=1}^{L} T_{t,\ell}^S \frac{\theta_{t,\ell}^S}{\theta_{t,\ell-1}^S} \odot \xi_{t,\ell}^S, \qquad \Delta \alpha_{T+1}^S = 0 \tag{52}$$

$$\Delta\alpha_t^A = \sum_{\ell=1}^{L} T_{d,\ell}^{a_t} \frac{\theta_{t,\ell}^{a_t}}{\theta_{t,\ell-1}^{a_t}} \odot \xi_{t,\ell}^{a_t}, \qquad \Delta\alpha_{T+1}^A = 0 \tag{53}$$

E.5 Δq_t

By the definition of l_i we have

$$\Delta q_t = \frac{\partial l_i}{\partial q_t} = -\frac{1}{\sigma_r^2}(d_t - q_t) \tag{54}$$

E.6 $\frac{\partial l_i}{\partial U}$

By chain rule, we have

$$\begin{aligned} \frac{\partial l_i}{\partial U} &= \sum_{t=1}^{T_i} \frac{\partial q_t}{\partial U} \cdot \frac{\partial l_i}{\partial q_t} \\ &= \sum_{t=1}^{T_i} \frac{\partial q_t}{\partial U} \cdot \Delta q_t \end{aligned} \tag{55}$$

By definition, $q_t = (\theta_{t,L}^{a_t})^T U \theta_{t,L}^S$, we have

$$\frac{\partial q_t}{\partial U} = \theta_{t,L}^{a_t} (\theta_{t,L}^S)^T \tag{56}$$

Substituting the above expression, we arrive at the expression of $\frac{\partial l_i}{\partial U}$.

E.7 $\frac{\partial l_i}{\partial \Phi_S}$

The expression of $\frac{\partial l_i}{\partial \Phi_S}$ and the recursion of $\xi_{t,\ell}^S$ can be derived in the same manner as that in BP-sLDA. We only derive the expression of $\xi_{t,L}^S$ here. First, note that $\xi_{t,\ell}^S = \mathbb{1}^T p_{t,\ell}^S \cdot \delta_{t,\ell}^S$, where $\delta_{t,\ell}^S \triangleq \frac{\partial l_i}{\partial p_{t,\ell}^S}$. We start by deriving the expression of $\delta_{t,L}^S$. Introduce the notation

$$\Delta\alpha_{t+1}^S = \frac{\partial l_i}{\partial \alpha_{t+1}^S}, \quad \Delta\alpha_{t+1}^A = \frac{\partial l_i}{\partial \alpha_{t+1}^A} \tag{57}$$

Then, we have

$$\begin{aligned} \delta_{t,L}^S &= \frac{\partial l_i}{\partial p_{t,L}^S} \\ &= \frac{\partial q_t}{\partial p_{t,L}^S} \cdot \frac{\partial l_i}{\partial q_t} + \frac{\partial \alpha_{t+1}^S}{\partial p_{t,L}^S} \cdot \frac{\partial l_i}{\partial \alpha_{t+1}^S} + \frac{\partial \alpha_{t+1}^A}{\partial p_{t,L}^S} \cdot \frac{\partial l_i}{\partial \alpha_{t+1}^A} \\ &= \frac{\partial q_t}{\partial p_{t,L}^S} \cdot \Delta q_t + \frac{\partial \alpha_{t+1}^S}{\partial p_{t,L}^S} \cdot \Delta\alpha_{t+1}^S + \frac{\partial \alpha_{t+1}^A}{\partial p_{t,L}^S} \cdot \Delta\alpha_{t+1}^A \end{aligned} \tag{58}$$

By the expression of $q_t = (\theta_{t,L}^{a_t})^T U \theta_{t,L}^S$ and $\theta_{t,L}^S = \frac{p_{t,L}^S}{\mathbb{1}^T p_{t,L}^S}$, we have

$$\frac{\partial q_t}{\partial p_{t,L}^S} = \frac{1}{\mathbb{1}^T p_{t,L}^S} (I - \mathbb{1}(\theta_{t,L}^S)^T) U^T \theta_{t,L}^{a_t} \tag{59}$$

Noting that

$$\begin{aligned} \alpha_{t+1}^S &= \sigma\left(W_{SS}\theta_{t,L}^S + W_{SA}\theta_{t,L}^{a_t} + \alpha_0^S\right) \\ &= \sigma(p_{\alpha_{t+1}^S}) \end{aligned}$$

where

$$p_{\alpha^S_{t+1}} \triangleq W_{SS}\theta^S_{t,L} + W_{SA}\theta^{a_t}_{t,L} + \alpha^S_0$$

we have

$$\begin{aligned}
\frac{\partial(\alpha^S_{t+1})^T}{\partial p^S_{t,L}} &= \frac{\partial(\theta^S_{t,L})^T}{\partial p^S_{t,L}} \cdot \frac{\partial p^T_{\alpha^S_{t+1}}}{\partial \theta^S_{t,L}} \cdot \frac{\partial(\alpha^S_t)^T)}{\partial p_{\alpha^S_{t+1}}} \\
&= \frac{1}{\mathbb{1}^T p^S_{t,L}}[I - \mathbb{1}(\theta^S_{t,L})^T]W^T_{SS}\mathrm{diag}\big(\sigma'(p_{\alpha^S_{t+1}}\big) \\
&= \frac{1}{\mathbb{1}^T p^S_{t,L}}[I - \mathbb{1}(\theta^S_{t,L})^T]W^T_{SS}D^S_{t+1} \qquad (60)
\end{aligned}$$

where

$$D^S_{t+1} \triangleq \mathrm{diag}\big(\sigma'(p_{\alpha^S_{t+1}}\big)$$

Likewise, we can get

$$\frac{\partial(\alpha^A_{t+1})^T}{\partial p^S_{t,L}} = \frac{1}{\mathbb{1}^T p^S_{t,L}}[I - \mathbb{1}(\theta^S_{t,L})^T]W^T_{AS}D_{\alpha^A_{t+1}} \qquad (61)$$

where

$$\begin{aligned}
&D^A_{t+1} \triangleq \mathrm{diag}\big(\sigma'(p_{\alpha^A_{t+1}}\big) \\
&p_{\alpha^A_{t+1}} \triangleq W_{AS}\theta^S_{t,L} + W_{AA}\theta^{a_t}_{t,L} + \alpha^A_0 \qquad (62)
\end{aligned}$$

Substituting (59), (60) and (61) into (58), we obtain

$$\begin{aligned}
\delta^S_{t,L} = &\frac{1}{\mathbb{1}^T p^S_{t,L}}(I - \mathbb{1}(\theta^S_{t,L})^T)U^T\theta^{a_t}_{t,L}\Delta q_t \\
&+ \frac{1}{\mathbb{1}^T p^S_{t,L}}[I - \mathbb{1}(\theta^S_{t,L})^T]W^T_{SS}D^S_{t+1}\Delta\alpha^S_{t+1} \\
&+ \frac{1}{\mathbb{1}^T p^S_{t,L}}[I - \mathbb{1}(\theta^S_{t,L})^T]W^T_{AS}D_{\alpha^A_{t+1}}\Delta\alpha^A_{t+1}
\end{aligned}$$

Multiplying both sides by $\mathbb{1}^T p^S_{t,L}$, we obtain the desired result.

E.8 $\frac{\partial l_i}{\partial \Phi_A}$

The related expression for $\frac{\partial l_i}{\partial \Phi_A}$ can be derived in a similar manner as that of $\frac{\partial l_i}{\partial \Phi_S}$. Therefore, we omit the derivation for brevity.

E.9 $\Delta\alpha^S_t$ and $\Delta\alpha^A_t$

By chain rule, it holds that

$$\begin{aligned}
\Delta\alpha^S_t &= \frac{\partial l_i}{\partial \alpha_S} \\
&= \sum_{\ell=1}^{L} \frac{\partial z^S_{t,\ell}}{\partial \alpha^S_t} \cdot \frac{\partial p^S_{t,\ell}}{\partial z^S_{t,\ell}} \cdot \frac{\partial l_i}{\partial p^S_{t,\ell}} \\
&= \sum_{\ell=1}^{L} \frac{\partial z^S_{t,\ell}}{\partial \alpha^S_t} \cdot \frac{\partial p^S_{t,\ell}}{\partial z^S_{t,\ell}} \cdot \delta^S_{t,\ell} \qquad (63)
\end{aligned}$$

Noting that

$$p^S_{t,\ell} = \theta^S_{t,\ell-1} \odot \exp(z^S_{t,\ell})$$

$$z_{t,\ell}^S = T_{t,\ell}^S \cdot \left[(\Phi_\ell^S)^T \frac{x_t^S}{\Phi_\ell^S \theta_{t,\ell-1}^S} + \frac{\alpha_t^S - \mathbb{1}}{\theta_{t,\ell-1}^S} \right] \tag{64}$$

we have

$$\begin{aligned} \frac{\partial (p_{t,\ell}^S)^T}{\partial z_{t,\ell}^S} &= \operatorname{diag}\big(\theta_{t,\ell-1}^S \odot \exp(z_{t,\ell}^S)\big) = \operatorname{diag}(p_{t,\ell}^S) \\ \frac{\partial (z_{t,\ell}^S)^T}{\partial \alpha_t^S} &= T_{t,\ell}^S \cdot \operatorname{diag}\left(\frac{1}{\theta_{t,\ell-1}^S}\right) \end{aligned} \tag{65}$$

Substituting the above expressions, we have

$$\begin{aligned} \Delta\alpha_t^S &= \sum_{\ell=1}^L T_{t,\ell}^S \cdot \frac{p_{t,\ell}^S}{\theta_{t,\ell-1}^S} \odot \delta_{t,\ell}^S \\ &= \sum_{\ell=1}^L T_{t,\ell}^S \cdot \frac{\theta_{t,\ell}^S}{\theta_{t,\ell-1}^S} \odot \xi_{t,\ell}^S \end{aligned} \tag{66}$$

where in the last step, we used the fact that $\xi_{t,\ell}^S = \mathbb{1}^T p_{t,\ell}^S \cdot \delta_{t,\ell}^S$ and $\theta_{t,\ell}^S = \frac{p_{t,\ell}^S}{\mathbb{1}^T p_{t,\ell}^S}$. In a similar manner, we can derive the expression for $\Delta\alpha_t^A$.

E.10 $\frac{\partial \ell_i}{\partial W_{SS}}$, $\frac{\partial l_i}{\partial W_{SA}}$, $\frac{\partial l_i}{\partial W_{AS}}$ and $\frac{\partial l_i}{\partial W_{AA}}$

We will only derive the expression for $\frac{\partial \ell_i}{\partial W_{SS}}$ and the derivation of the others is similar. Let $[W_{SS}]_{ij}$ denote the (i,j)-th component of the matrix W_{SS}. Then, by chain rule, we have

$$\frac{\partial l_i}{\partial [W_{SS}]_{ij}} = \frac{\partial \alpha_t^S}{\partial [W_{SS}]_{ij}} \cdot \frac{\partial l_i}{\partial \alpha_t^S} = \frac{\partial \alpha_t^S}{\partial [W_{SS}]_{ij}} \cdot \Delta\alpha_t^S \tag{67}$$

By the fact that

$$p_{\alpha_t^S} = W_{SS}\theta_{t-1,L}^S + W_{SA}\theta_{t-1,L}^{a_{t-1}} + \alpha_0^S$$

we have

$$\begin{aligned} \frac{\partial (\alpha_t^S)^T}{\partial [W_{SS}]_{ij}} &= \frac{\partial p_{\alpha_t^S}^T}{\partial [W_{SS}]_{ij}} \cdot \frac{\partial (\alpha_t^S)^T}{\partial p_{\alpha_t^S}} \\ &= \frac{\partial p_{\alpha_t^S}^T}{[W_{SS}]_{ij}} D_t^S \\ &= [\theta_{t-1,L}^S]_j \cdot e_i^T D_t^S \end{aligned} \tag{68}$$

where e_i is a vector with i-th element being one and zero otherwise. Then, it holds that

$$\frac{\partial (\alpha_t^S)^T}{\partial [W_{SS}]_{ij}} \cdot \Delta\alpha_t^S = [\theta_{t-1,L}^S]_j \cdot [D_t^S]_{ii} \cdot [\Delta\alpha_t^S]_i \tag{69}$$

so that putting in matrix form:

$$\frac{\partial l_i}{\partial W_{SS}} = (D_t^S \Delta\alpha_t^S)(\theta_{t-1,L}^S)^T$$