Abstract then Play: A Skill-centric Reinforcement Learning Framework for Text-based Games

Anjie Zhu[1], Peng-Fei Zhang[2], Yi Zhang[2], Zi Huang[2] and Jie Shao[1]
[1]University of Electronic Science and Technology of China, Chengdu, China
[2]The University of Queensland, Brisbane, Australia
anjiezhu@std.uestc.edu.cn mima.zpf@gmail.com y.zhang30@uqconnect.edu.au
huang@itee.uq.edu.au shaojie@uestc.edu.cn

Abstract

Text-based games present an exciting test-bed for reinforcement learning algorithms in the natural language environment. In these adventure games, an agent must learn to interact with the environment through text in order to accomplish tasks, facing large and combinational action space as well as partial observability issues. However, existing solutions fail to decompose the task and abstract the action autonomously, which either pre-specify the subtasks or pre-train on the human gameplay dataset. In this work, we introduce a novel skill-centric reinforcement learning framework, which is capable of abstracting the action in an end-to-end manner. To learn a more disentangled skill, we focus on the informativeness and distinguishability of the skill in accordance with the information bottleneck principle. Specifically, we introduce a discriminator to enable the skill to reflect the trajectory and push their representations onto the unit hypersphere to distribute uniformly. Moreover, a self-predictive mechanism is employed to learn inverse and forward dynamics, and a self-recovery mechanism is leveraged to refine the action representation, thus resulting in a more comprehensive perception of dynamics and more effective representations of textual state and action. Empirical experiments are carried out on the Jericho environment and the results validate the superiority against state-of-the-art baselines.

1 Introduction

Mastering the ability of understanding and responding using natural language is essential for a wide range of technologies and applications (e.g., in customer consultation and service systems). The interactive adventure games (Hausknecht et al., 2020), such as zork1 (can be seen in Table 1), provide a good test-bed for reinforcement learning (RL) agents (Osborne et al., 2022) in the pursuit of intelligence, which can be regarded as long-horizon puzzles or quests through navigating and interacting with multiple objects and locations. The game player not only needs to accurately understand the information of the environment, but also needs to make an effective and well-performed reaction, both in the form of natural language. This structure comes with two critical issues: (1) ***Large and combinational action space***. The agent faces about 1.64×10^{14} possible actions at every step in the game zork1 (Ammanabrolu and Hausknecht, 2020), which is a dauntingly large combinatorially-sized action space and thus brings significant difficulties in making decisions. Some works investigate filtering out the irrelevant ones (Zahavy et al., 2018) or generating contextually-relevant action candidates (Yao et al., 2020). Another line of research is dedicated to hierarchical reinforcement learning (Adolphs and Hofmann, 2020; Xu et al., 2022, 2021), which hierarchically decomposes tasks into simpler ones and learns by hierarchical policies. (2) ***Partial observability***. Due to the limited information provided from the environment, incomplete textual description brings great obstacles to the exploration and reasoning of the agent. There

Zork1
Observation: South of House You are facing the south side of a white house. There is no door here, and all the windows are boarded. **Action**: *Go east*
Observation: Behind House You are behind the white house. A path leads into the forest to the east. In one corner of the house there is a small window which is slightly ajar. **Action**: *Enter house*
Observation: Kitchen You are in the kitchen of the white house. A table seems to have been used recently for the preparation of food. A passage leads to the west and a dark staircase can be seen leading upward. **Action**: *Go west*

Table 1: The transcript of zork1 in textworld games. The agent receives observation from the game environment. According to the textual information, the agent executes its action, and then the environment transits to the next observation.

Findings of the Association for Computational Linguistics: ACL 2023, pages 13225–13236

are some attempts to define and discover under-explored states (Madotto et al., 2020), or bottleneck states (Ammanabrolu et al., 2020), hoping to grasp and explore the environment more comprehensively. Other works resort to building a knowledge graph from textual description (Ammanabrolu and Hausknecht, 2020), hoping to capture more structural information about the environment. Unfortunately, these previous works heavily rely on hard-crafted design. They pre-specify subtasks or pre-train on the human gameplay dataset in advance, which all fail to abstract the tasks autonomously.

Action abstraction, a.k.a. "skill", refers to a set of pertinent long-horizon behaviors that an agent is capable of engaging in, which is a temporally extended macro-action in hierarchical reinforcement learning. For instance, there are several steps involved in entering a house, such as finding the door, approaching it and opening the door. Leveraging such action abstraction can assist the intelligent agent in autonomously breaking down the challenging task into a hierarchy, where the action space issue becomes trivial. Motivated by this, to cope with the large action space challenge, we choose to abstract the action from a novel perspective, without requiring the assistance of any pre-trained models or prior knowledge. Additionally, observing the textual state and action in the environment, the traditional representation extractor designed for vector-based state and action is insufficient, necessitating the mining and exploitation of comprehensive information. Thus, it is desirable to employ a better representation learner for textual state and action. With valuable semantic representation acquired from those textual observations, intelligent agents deployed on top of them will bring more effective use of them naturally.

In this work, we introduce a skill-centric action abstraction framework in an end-to-end manner for text-based games, in the hopes of alleviating the combinatorially-size action space and partial observability issues. A novel skill learning strategy is designed according to the information bottleneck principle, with the goal of a disentangled skill by enhancing the informativeness and distinguishability of the skill. In particular, in pursuing informativeness, we pull the representations of skill and the current trajectory closely by maximizing the mutual information between them, and push the representations of skill and the other unrelated trajectory away by minimizing their mutual information. By this means, the skill is enforced to cover sufficient information about the current trajectory while being parsimonious to exclude unrelated noisy information. In pursuing distinguishability, the representations of skill and trajectory are separated onto the unit hypersphere, thus preserving maximal information. The discovered skill can guide the agent to make optimal decisions. Besides, within the skill-centric framework, the self-predictive mechanisms for inverse and forward dynamics and the self-recovery dynamics learning for the action itself are presented. The core idea lies in easing the partial information issue and reinforcing the representations of textual state and action effectively. Overall, our contributions can be summarized as follows:

- A novel skill-centric reinforcement learning framework in an end-to-end manner for text-based games is presented, where the information bottleneck-based action abstraction is performed to improve the exploration of the agent and lessen the burden of large action space.
- Two simple yet effective representation learning strategies for the textual state and action are developed, i.e., a self-predictive mechanism and a self-recovery mechanism, investigating the partial and textual information and reinforcing effective representations.
- Promising results through extensive experiments on text-based games demonstrate the superiority of the proposed framework.

2 Related Work

2.1 RL agent for text-based games

Thus far, a considerable amount of literature has investigated text-based games. Among them, reinforcement learning dominates interactive fiction text-based games consisting of human-written text. In Jericho (Hausknecht et al., 2020), there are three categories of restrictions on the action space, parser-based, template-based and choice-based. LSTM-DQN (Narasimhan et al., 2015) is the first work on combining reinforcement learning and natural language understanding, which selects the verbs and objects independently according to deep Q-network. Deep Reinforcement Relevance Network (DRRN) (He et al., 2016; Yao et al., 2021) and Template-DQN (TDQN) (Hausknecht et al., 2020) extend LSTM-DQN on template-based and choice-

based action space, respectively. MPRC-DQN (Guo et al., 2020) reformulates text-based games as a multi-paragraph reading comprehension task which utilizes context-query attention mechanism and object-centric history retrieval strategy. Another representative series of research incorporates knowledge graph to enhance state representation (Ammanabrolu and Hausknecht, 2020). Additionally, Adhikari et al. (2020) build a Graph Aided Transformer Agent (GATA) to learn belief graph for action selection during planning and generalizing. Xu et al. (2020) conduct explicit reasoning with knowledge graph with relational and temporal awareness and design a stacked hierarchical attention mechanism to build state representation from multi-model inputs. Assisted by a fine-tuned GPT-2, CALM (Yao et al., 2020) leverages the pre-trained language model to generate candidate actions, which needs human gameplay and pre-training in advance. CBR (Atzeni et al., 2022) employs case-based reasoning which is built on a collection of past experience and reuses the relevant one. All of these earlier approaches, however, make an effort to create action-reduction strategies, or they enlist the assistance of knowledge graphs or pre-trained models. Our mission is to free our hands and decompose the tasks by abstracting actions in an end-to-end manner.

2.2 Hierarchical RL agent

Hierarchical reinforcement learning is a promising strategy for figuring out tough puzzles, which decomposes a challenging long-horizon task into simpler subtasks. Similar to the idea of feudal learning, LeDeepChef (Adolphs and Hofmann, 2020) combines multiple actions into a single "high-level" command that designs a recipe manager to identify which recipe action still needs to be performed, which is not flexible because it is customized for ingredients and inventory in cooking tasks. L-GAT (Kohita et al., 2021) builds a hierarchical action generation algorithm where the agent first defines abstractive action template consisting of frame, role and lexicon, and then generates a concrete action commanding with hierarchical predicating, word masking and template masking. Xu et al. (2022) present a world-perceiving module to autonomously decompose tasks and prune actions by answering questions, although built on a question-answering dataset and still require supervised pre-training. H-KGA (Xu et al., 2021) applies hierarchical reinforcement learning where high-level policy is for goal generation and selection, and low-level policy is for goal-conditioned reinforcement learning. The process of goal-set generation is non-learning, which is based on a fixed rule and cannot be adapted to different types of games. It is worth noting that all these works either presume the accessibility of a set of subtasks, or decompose a task through pre-defined rules or a pre-trained module. In this work, we aim to introduce a novel skill-centric framework that aims at abstracting action, without any extra involvement.

3 Preliminaries

3.1 Text-based games as POMDP

The agent in text-based games never has access to global environment state, but only local textual information. Therefore, the text-based games can be formally defined as a discrete-time Partially Observable Markov Decision Process (POMDP). It can be represented as a 7-tuple $\langle S, T, A, \Psi, P, R, \gamma \rangle$: the state set S, the state transition function T, the action set A, the observation set Ψ, the reward function R, the conditional observation probability P, and the discount factor $\gamma \in [0, 1]$. Specifically, the agent receives an observation obs_t based on the current state s_t and last action a_{t-1} through $P(obs_t|s_t, a_{t-1})$ at timestep t. Then, the agent acts its corresponding action a_t based on its policy π, and receives the next state and reward feedback based on $T(s_{t+1}|s_t, a_t)$ and $R(s_t, a_t)$ separately from the environment. The objective of reinforcement learning agent is to optimize the policy that maximizes its cumulative discounted reward $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$.

3.2 Skill learning in reinforcement learning

Skill is defined as a set of empowered actions and we parameterize skill as a latent z. The skill-centric policy is represented as $\pi(a|s, z)$, where the skill latent z is combined with the state to produce the optimal action. It is assumed that skills are sampled from a prior distribution $p(z)$. Recent lines of research show the interest in discovering skills without the assumption of any other extrinsic rewards, i.e., in an unsupervised manner and without any other human-designed rewards. It complies with the demand that we do not have to impose an artificially complex abstraction structure. Leveraging information theory such as measuring the mutual information between state, action and skill

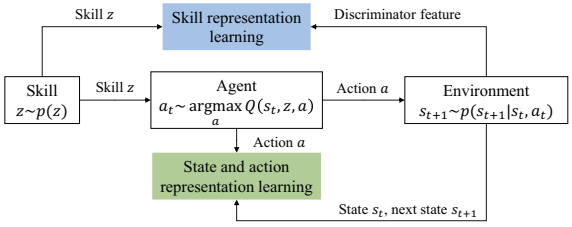

Figure 1: The overall process of our proposed framework.

(e.g., DIAYN (Eysenbach et al., 2019) and HIDIO (Zhang et al., 2021)), RL agents are able to autonomously discover such skills. To the best of our knowledge, we are the first to consider skill learning in text-based games with reinforcement learning.

4 Methodology

Figure 1 depicts the whole process of our proposed framework. Besides the general reinforcement learning framework with agent and environment, we introduce the skill and textual state and action representation learning in text-based games, which focus on large action space and partial observability issues, and aid the agent in making the optimal decision.

4.1 Informative and distinguishable skill learning

Pursuing effective skills without any extrinsic supervision, we simultaneously investigate extracting the most shared information between the representations of skill and the current trajectory and remaining invariant to irrelevant ones, as shown in the top half of Figure 2. Concretely, in light of the informativeness, we propose an information bottleneck-based way to learn a more disentangled and interpretable skill. Then, in light of the distinguishability of skill, we propose to push their representations away on the unit hypersphere, which ensures preserving maximal information.

Informativeness. In accordance with the information bottleneck principle (Alemi et al., 2017), our objective of learning skill is to make skill provide sufficient information coverage of trajectory while being parsimonious to leave out unrelated noisy information. Formally, our aim is to maximize

$$J = I(\Omega; Z) - \alpha I(Z; \Omega'), \tag{1}$$

where I denotes mutual information, Z denotes the skill latent variable, Ω is the trajectory and Ω' is the irrelevant trajectory for which we intend to keep minimal information. We can rewrite the objective of Eq. (1) in the following:

$$J = H(Z) - H(Z|\Omega) - \alpha\big(H(Z) - H(Z|\Omega')\big), \tag{2}$$

where $H(\cdot)$ denotes entropy. The conditional entropy $H(Z|\Omega)$ and $H(Z|\Omega')$ are the entropy of skill representation conditioned on the state and unrelated noisy information. Our objective then becomes to maximize:

$$\begin{aligned} J &= (1-\alpha)H(Z) - H(Z|\Omega) + \alpha H(Z|\Omega') \\ &= -(1-\alpha)\mathbb{E}_z[\log p(z)] + \mathbb{E}_{z,\omega}[\log p(z|\omega)] \\ &\quad - \alpha\mathbb{E}_{z,\omega'}[\log p(z|\omega')] \\ &\geq \mathbb{E}_{z,\omega}\big[\log q_\phi(z|\omega) - (1-\alpha)p(z)\big] \\ &\quad - \alpha\mathbb{E}_{z,\omega'}[\log q_\phi(z|\omega')], \end{aligned} \tag{3}$$

which gives us a variational lower bound. Approximating the posterior $p(z|\omega)$ is intractable, so instead, we estimate it with a discriminator $q_\phi(z|\omega)$ to obtain the lower bound, which can be known from Jensen's inequality. The trajectory comprises a sequence of states and/or actions, forming a cohesive representation of interactions with the environment over a specific period. About the selection of trajectory we displayed here, we will conduct more experiments to illustrate. Rethinking the first part of Eq. (3), an idea of a contrastive way comes to us, specifically, maximizing $\log q_\phi(z|\omega)$, i.e., maximizing the alignment between skill and trajectory. For our practical implementation, we parameterize this function through calculating the cross-correlation matrix between skill and trajectory. We convert to minimize the following loss function which seeks to decorrelate the feature dimension and maintain non-redundant information:

$$\sum_i (1 - C_{ii})^2 + \lambda \sum_i \sum_{j \neq i} C_{ij}^2, \tag{4}$$

where C denotes the cross-correlation matrix calculated between skill and trajectory along the batch dimension. Naturally, the above is dedicated to identifying and isolating the underlying factors of variation in the hopes of improving the representation of skill.

Distinguishability. The learned representations can easily degenerate to one dominant dimension

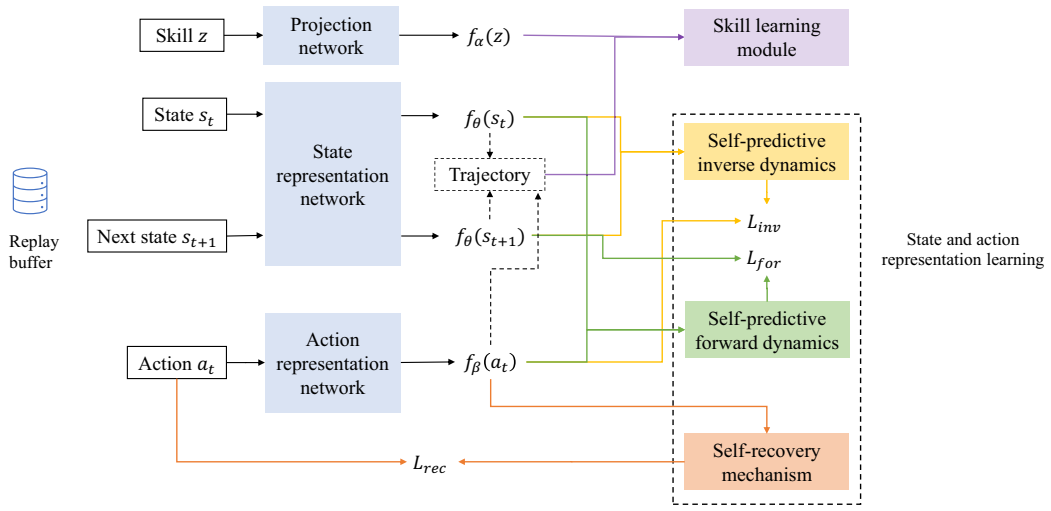

Figure 2: The illustration of our framework in detail.

while others are less important. In order to circumvent this common issue in representation learning, inspired by forcing representations to distribute more uniformly and separably, we consider pushing the representations to distribute uniformly on the unit hypersphere through minimizing

$$\log \mathbb{E}_{z,\omega} \left[\exp(-2||f_\alpha(z) - f(\omega)||^2) \right], \quad (5)$$

where $f_\theta(\cdot)$ and $f(\cdot)$ are the encoder networks that transform the skill and trajectory into representations, for subsequent RL policy optimization. The reason for choosing the logarithm of average pairwise Gaussian potential as shown in Eq. (5) is that it has a tight relationship to the universal optimal point configuration (Cohn and Kumar, 2006).

Discussion. Recall that the objective of general contrastive objective can be decomposed into two components, each of which correlates with our two objectives, informativeness and distinguishability, respectively. The expansion is in the following:

$$\begin{aligned} L &= -\log \frac{\exp(\text{sim}(z,\omega))}{\sum_{\omega' \in \Omega \setminus \omega} \exp(\text{sim}(z,\omega'))}, \\ &= \underbrace{-\text{sim}(z,\omega)}_{\text{informativeness}} + \underbrace{\log \sum_{\omega' \in \Omega \setminus \omega} \exp(\text{sim}(z,\omega'))}_{\text{distinguishability}}, \end{aligned} \quad (6)$$

where $\text{sim}(\cdot)$ is the similarity function. Two terms of the above equation indicate bringing the representation of skill and trajectory as close as possible, namely, aligning them closer, and making the representation of skill and unrelated trajectory as further as possible, namely, spreading them out. In this work, we have our own corresponding manifestations and realization respectively. Forcing each dimension to carry out its functions and fulfill its related obligations, we measure the cross-correlation matrix beginning from a feature-wise point of view and concentrate more on the disentanglement of representation, which ensures informativeness. At the same time, the representation of skill and trajectory will also be projected into a unit hypersphere to strive for uniform distribution, which ensures distinguishability. They work in harmony to achieve a greater representation of the skill.

4.2 Self-predictive state and action representation learning

When compared with vector-based ones in general RL, the state and action in the Jericho environment are text, so it is crucial to extract effective information from textual data, which is the only feedback we got from the partial observation. In this section, we introduce the self-predictive mechanism for learning inverse and forward dynamics, and the self-recovery mechanism for improving the action representation, as shown in the bottom half of Figure 2. They intend to acquire effective representations of state and action, so as to investigate the world model in greater detail, without the assistance of any additional auxiliary tasks.

Our framework is based on the DQN algorithm (Mnih et al., 2015), consisting of experience (s_t, a_t, s_{t+1}, r_t), which we simply utilize for self-supervised dynamics learning. First, rather than manually building representations (feature extrac-

tion), the self-predictive inverse dynamics mechanism predicts the action based on the state and next state. The objective of the inverse dynamics model is described as:

$$
\begin{aligned}
&L_{inv}(\theta, \beta, t) \\
&= \ell\Big(F_{inv}\big(f_\theta(s_t), f_\theta(s_{t+1})\big), f_\beta(a_t)\Big) \\
&= -\log p_d\Big(f_\beta(a_t) | F_{inv}\big(f_\theta(s_t), f_\theta(s_{t+1})\big)\Big), \\
&L_{inv}(\theta, \beta) = \sum_{t=0}^{T} L_{inv}(\theta, t),
\end{aligned}
\tag{7}
$$

where T is the path length. The inverse dynamics model is denoted as F_{inv}, which we implement as a multi-layer perceptron. ℓ calculates the discrepancy between the outputs of the inverse dynamics model and the real one. Directly calculating the Euclidean distance for the text embedding could cause the representations to collapse. Here, we thus employ GRU decoder d which is widely used in language processing, and p_d is the probability of decoding the output of inverse dynamics to action sequence. Next, for the self-recovery mechanism for action, we propose to recover the output of f_β back into real action, which is represented as:

$$
\begin{aligned}
L_{rec}(\beta, t) &= \ell\big(f_\beta(a_t), a_t\big) \\
&= -\log p_d\big(a_t | f_\beta(a_t)\big), \\
L_{rec}(\beta) &= \sum_{t=0}^{T} L_{rec}(\beta, t).
\end{aligned}
\tag{8}
$$

Similar to the inverse dynamics, self-predictive forward dynamics is represented as follows, which predicts the next state based on the current state and action:

$$
\begin{aligned}
&L_{for}(\theta, \beta, t) \\
&= \ell\Big(F_{for}\big(f_\theta(s_t), f_\beta(a_t)\big), f_\theta(s_{t+1})\Big) \\
&= -\log p_d\Big(f_\theta(s_{t+1}) | F_{for}\big(f_\theta(s_t), f_\beta(a_t)\big)\Big), \\
&L_{for}(\theta, \beta) = \sum_{t=0}^{T} L_{for}(\theta, \beta, t),
\end{aligned}
\tag{9}
$$

where F_{for} is the forward dynamics model. Having the above mechanisms, we will embrace a better perception of state and action, resulting in better representations of state and action while alleviating the partial observability issue.

Algorithm 1: Our framework.

```
Initialize replay buffer B;
Initialize state-action value function Q with random weights;
s ← s_0, t ← 0;
Sample a skill z ~ p(z);
for timestep t = 1 to T do
    With probability ε select the random action a_t, otherwise select a_t = max_a Q(s_t, z, a);
    Execute action a_t and receive s_{t+1}, r_t from environment;
    Store (s_t, a_t, s_{t+1}, r_t, z_t) in B;
    Sample random minibatch of transitions from B;
    if t%UPDATE SKILL ==0 then
        Sample a skill z ~ p(z);
    end
    Perform the update of skill;
    Perform the update of representations of state and action;
    Perform the update of Q-value;
end
```

4.3 Overall objective

Our framework is based on deep Q-value, and the optimal action is according to the maximal of state action Q-value: $a_t = \text{argmax}_a Q(s_t, z, a)$. Following classic reinforcement learning, we deploy ϵ-greedy policy to enhance exploration through selecting the random action with probability ϵ. Thus, the objective function of Q-value is:

$$
\begin{aligned}
J_Q =& \mathbb{E}_{s,a,s',z}\big[Q(s, z, a) - (r(s, a) \\
&+ \gamma \max_{a'} Q(s', z, a'))\big]^2.
\end{aligned}
\tag{10}
$$

The overall learning algorithm is summarized as Algorithm 1.

5 Experiments and Analysis

5.1 Experimental setup

Game environment. Jericho games (Hausknecht et al., 2020) provide human-made interactive fiction to verify the performance of intelligent agents. For example, the most famous game zork1, is a treasure collecting game where the dungeon crawler needs to explore a vast labyrinth and solve puzzles. Furthermore, other significant obstacles that intelligent agent faces are sparse rewards and unpredictable enemy attacks. In terms of action space which can be categorized into parser-based, template-based and choice-based, we choose choice-based here. We conduct experiments on different games of Jericho suite.

Implementation details. The size of the replay buffer is 10000. The discount factor γ is 0.9.

Game	$\|T\|$	$\|V\|$	DRRN	TDQN	KG-A2C	CALM	SHA-KG	MPRC-DQN	L-GAT	CBR	Ours	MaxR
balances	156	452	10	4.8	10	9.1	10	10	8.8	11.9	**12.5**	51
deephome	173	760	1	1	1	1	-	1	14.9	1	**29.1**	300
inhumane	141	409	0	0.7	3	25.7	5.4	0	0	24.2	**28.7**	90
jewel	161	657	1.6	0	1.8	0.1	1.8	4.5	0	6.4	**6.5**	90
karn	178	615	2.1	0.7	0	2.3	-	10	-	0	**10**	170
library	173	510	17	6.3	14.3	9.0	15.8	17.7	7.6	22.3	19.75	30
ludicorp	187	503	13.8	6	17.8	10.1	17.8	19.7	6.1	23.8	21	150
pentari	155	472	27.2	17.4	50.7	-	51.3	44.4		52.1	**53.7**	70
reverb	183	526	8.2	0.3	7.4	-	10.6	2	1.0	6.5	**11.3**	50
spellbrkr	333	844	37.8	18.7	21.3	40	40	25	39.4	41.2	**49.4**	600
temple	175	622	7.4	7.9	7.6	0	7.9	8	5.0	7.8	**8**	35
tryst205	197	871	9.6	0	6.7	-	6.9	10	-	13.4	11.9	350
yomomma	141	619	0.4	0	-	-	-	1	-	1	**1**	35
zork1	237	697	32.6	9.9	34	30.4	34.5	38.3	17.1	44.3	**51**	350
zork3	214	564	0.5	0	0.1	0.5	0.7	3.6	0.4	3.2	1	7

Table 2: The result scores of our proposed method and other baselines on Jericho games. $|T|$ and $|V|$ are the size of template set and vocabulary set, respectively. MaxR is the maximum possible score for the game.

The maximum timestep is 100000. We use the Adam optimizer (Kingma and Ba, 2015) to update the weights under the learning rate 1e-4. For every 1000 training episodes, we validate the model and report the testing performance. Our code is available at `https://github.com/AnneZhu1020/Abstract-then-play/`.

Baselines. We compare with the following baselines:

- DRRN (He et al., 2016): The Q-value function is approximated using interaction function on state and action embeddings.
- TDQN (Hausknecht et al., 2020): For template-based action space, there are three Q-value approximations, one template and two objects.
- KG-A2C (Ammanabrolu and Hausknecht, 2020): The Advantage Actor Critic is to estimate the value of templates and objects with knowledge graph enhanced state while constraining the type of actions.
- CALM (Yao et al., 2020): It generates action candidates through training a GPT-2 based language model with a large number of human gameplay.
- SHA-KG (Xu et al., 2020): It considers a stacked hierarchical attention for multi-modal inputs and subgraphs of knowledge graph with different semantic meanings.
- MPRC-DQN (Guo et al., 2020): For template-based action space, it generates action by finding supportive evidence from the observation and augments the current observation with relevant history.
- L-GAT (Kohita et al., 2021): Facilitated by general semantic schemes (FrameNet, VerbNet, WordNet), it designs the general action template based on prior knowledge.
- CBR (Atzeni et al., 2022): It is based on case-based reasoning whose process consists of retrieving, reusing, revising and retaining.

5.2 Comparison with state-of-the-arts

We compare our framework with recent reinforcement learning models designed for text-based games, and the results are reported in Table 2. Our framework outperforms the existing methods in most games, especially games zork1, spellbrkr and inhumane with a large improvement which are representative games for possible, difficult and extreme (difficulty level). On the game karn and yomomma, we show a competitive performance, reaching the same maximum reward as the previous state-of-the-art baselines. The promising results demonstrate the effectiveness and practicability of our proposed framework.

5.3 Detailed analysis

Effect of skill learning. This experiment is conducted with other representative skill learning strategies, DIAYN (Eysenbach et al., 2019)

Strategy	zork1	spellbrkr	inhumane	Objective formula
Ours	51	49.395	28.75	maximize $I(S;Z) - \alpha I(Z;S')$
w/o skill learning	40	41.5	23.8	-
w/. DIAYN	42	42.3	24.2	maximize $I(S;Z) + H(A\|S) - I(A;Z\|S)$
w/. contrastive representation (with InfoNCE loss)	44	41.1	25.1	maximize $I(S;Z)$

Table 3: Ablation study of our proposed framework with different skill learning strategies.

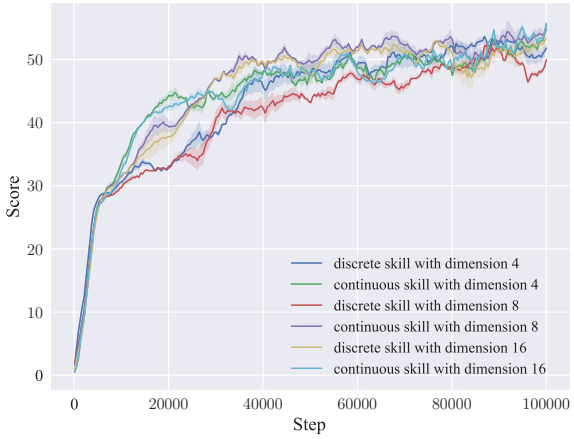

Figure 3: Ablation study of skill on zork1. Shaded regions indicate the standard deviations.

trajectory selection	zork1	spellbrkr	inhumane
$[s_t; a_t]$	43	41	27.2
$[s_t; a_t; s_{t+1}]$	44	43.65	27.3
$[s_{t+1} - s_t; a_t]$	47	46.5	28.35
$s_{t+1} - s_t$	51	49.395	28.75

Table 4: Ablation study of our proposed framework with different trajectory selections across three games.

Strategies	zork1	spellbrkr	inhumane
Ours	51	49.395	28.75
w/o self-predictive forward dynamics	43	43.15	28.0
w/o self-predictive inverse dynamics	46	43.65	27.8
w/o self-recovery for action	47	44.5	28.35

Table 5: Ablation study of our proposed framework with the self-predictive and self-recovery mechanisms.

and contrastive learning with InfoNCE loss (van den Oord et al., 2018). Specific objectives of the above strategies are illustrated in the last column of Table 3. Concretely, for the last one, we optimize: $\frac{f(\omega)^T f(z)}{\|f(\omega)\|\|f(z)\|T} - \log \frac{1}{N} \sum_{j=1}^{N} \exp \frac{f(\omega_j)^T f(z)}{\|f(\omega_j)\|\|f(z)\|T}$. As we can see, ours shows the best performance in representative games in Jericho, compared with previous skill learning strategies and the one without skill learning. This indicates the necessity of our proposed skill learning.

Analysis on different settings of skill. Here are two main questions about skill, discrete or continuous skill, and the dimension embedding of skill. We conduct experiments on discrete and continuous with dimensions in the range of {4, 8, 16}, separately. It can be seen from Figure 3 that the discrete embedding may be greatly affected by the dimension of embedding. In contrast, the continuous embedding can present a good performance more stably. We guess this is because the discrete skill embedding does not provide information about the proximity of skills, thus learning a continuous embedding is more effective. Regarding the dimension of continuous skill, there is only a slight difference among them where the continuous skill with dimension 8 narrowly beats the others.

Analysis on trajectory selection. In this experiment, aiming at answering which feature is the best for extracting skill based on states and actions, we perform four candidates across the above games. From Table 4, we can conclude that selecting the difference between two adjacent states as the trajectory performs almost the best. Moreover, observing the last two rows, it is surprised that the addition of action does not bring about an improvement in the effect. This finding is in line with the intention of DIAYN (Eysenbach et al., 2019) which is to have the state distinguish skill rather than action.

Ablation study of self-predictive and self-recovery mechanisms. To analyze the proposed self-predictive mechanism for forward and inverse dynamics, and our self-recovery for action representation itself, we conduct experiments across three games. As shown in Table 5, we can find that without the above mechanisms, the performance will decrease to a certain extent in different games. The effect of action reconstruction is the most significant, followed by the learning of inverse dynamics.

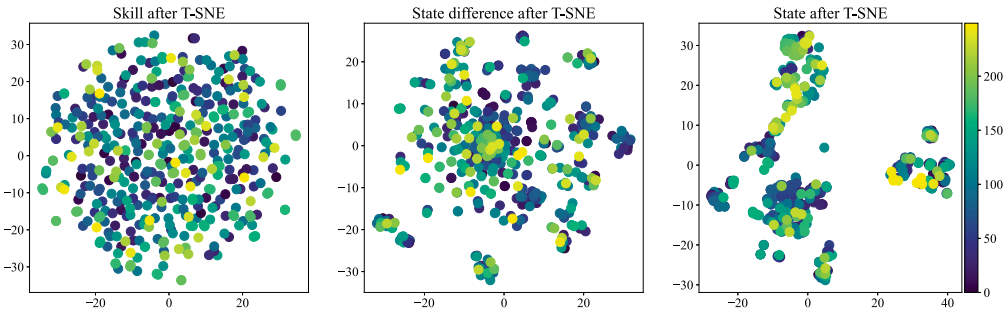

Figure 4: T-SNE visualization of the representations of skill, state difference and state.

5.4 Skill visualization

In order to analyze the representations of state and skill after learning, we perform t-SNE (der Maaten and Hinton, 2008) to project them into a vector of 2 dimensions. As shown in Figure 4, the state representation visualization depicted in the right figure differs significantly from the skill representation in the left figure. The result demonstrates the consistency between the representation of skill and trajectory selected (i.e., state difference), thereby highlighting the effectiveness of our skill and state representation learning through discriminator optimization.

6 Conclusion

In this work, we present a novel skill-centric reinforcement learning framework for text-based games. Our framework aims at abstracting the action in an end-to-end manner, in line with the information bottleneck principle which pursues the informativeness and distinguishability of skill. For better perceiving textual state and action, we employ self-predictive mechanisms for forward and inverse dynamics and a self-recovery mechanism for the action itself. To the best of our knowledge, this is the first work to consider skill discovery and representation learning in text-based games aiming at eliminating partial observability and large action space issues, without the need of any pre-trained models or prior knowledge.

Limitations

For now, the latent of our skill is sampled from a distribution, whose flexibility is not fully investigated. We intend to exploit more flexible skills and goal discovery, or direct generation via state or action. Additionally, the sparse reward in text-based games is also a burning challenge, which hinders the efficient exploration of agents. Our abstracted action, skill, to some extent eases off this issue, but is not enough. We will dive into this more and design a fancy solution later.

Acknowledgements

This work was partially supported by National Natural Science Foundation of China (No. 62276047) and Australian Research Council (No. DP190102353 and No. CE200100025).

References

Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláŝ Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and William L. Hamilton. 2020. Learning dynamic belief graphs to generalize on text-based games. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*.

Leonard Adolphs and Thomas Hofmann. 2020. Ledeepchef deep reinforcement learning agent for families of text-based games. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, pages 7342–7349.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR*.

Prithviraj Ammanabrolu and Matthew J. Hausknecht. 2020. Graph constrained reinforcement learning for natural language action spaces. In *8th International Conference on Learning Representations, ICLR*.

Prithviraj Ammanabrolu, Ethan Tien, Matthew J. Hausknecht, and Mark O. Riedl. 2020. How to avoid being eaten by a grue: Structured exploration strategies for textual worlds. *CoRR*, abs/2006.07409.

Mattia Atzeni, Shehzaad Zuzar Dhuliawala, Keerthiram Murugesan, and Mrinmaya Sachan. 2022. Case-based reasoning for better generalization in textual reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR*.

Henry Cohn and Abhinav Kumar. 2006. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, 20(1):98–148.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Value function based reinforcement learning in changing markovian environments. *J. Mach. Learn. Res.*, 9(86):2579–2605.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. 2019. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR*.

Xiaoxiao Guo, Mo Yu, Yupeng Gao, Chuang Gan, Murray Campbell, and Shiyu Chang. 2020. Interactive fiction game playing as multi-paragraph reading comprehension with reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 7755–7765.

Matthew J. Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. 2020. Interactive fiction games: A colossal adventure. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, pages 7903–7910.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, Volume 1: Long Papers*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*.

Ryosuke Kohita, Akifumi Wachi, Daiki Kimura, Subhajit Chaudhury, Michiaki Tatsubori, and Asim Munawar. 2021. Language-based general action template for reinforcement learning agents. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*, pages 2125–2139.

Andrea Madotto, Mahdi Namazifar, Joost Huizinga, Piero Molino, Adrien Ecoffet, Huaixiu Zheng, Alexandros Papangelis, Dian Yu, Chandra Khatri, and Gökhan Tür. 2020. Exploration based language learning for text-based games. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*, pages 1488–1494.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533.

Karthik Narasimhan, Tejas D. Kulkarni, and Regina Barzilay. 2015. Language understanding for text-based games using deep reinforcement learning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 1–11.

Philip Osborne, Heido Nõmm, and André Freitas. 2022. A survey of text games for reinforcement learning informed by natural language. *Trans. Assoc. Comput. Linguistics*, 10:873–887.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Yunqiu Xu, Meng Fang, Ling Chen, Yali Du, and Chengqi Zhang. 2021. Generalization in text-based games via hierarchical reinforcement learning. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 1343–1353.

Yunqiu Xu, Meng Fang, Ling Chen, Yali Du, Joey Zhou, and Chengqi Zhang. 2022. Perceiving the world: Question-guided reinforcement learning for text-based games. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL*, pages 538–560.

Yunqiu Xu, Meng Fang, Ling Chen, Yali Du, Joey Tianyi Zhou, and Chengqi Zhang. 2020. Deep reinforcement learning with stacked hierarchical attention for text-based games. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*.

Shunyu Yao, Karthik Narasimhan, and Matthew J. Hausknecht. 2021. Reading and acting while blindfolded: The need for semantics in text game agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 3097–3102.

Shunyu Yao, Rohan Rao, Matthew J. Hausknecht, and Karthik Narasimhan. 2020. Keep CALM and explore: Language models for action generation in text-based games. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 8736–8754.

Tom Zahavy, Matan Haroush, Nadav Merlis, Daniel J. Mankowitz, and Shie Mannor. 2018. Learn what not to learn: Action elimination with deep reinforcement learning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 3566–3577.

Jesse Zhang, Haonan Yu, and Wei Xu. 2021. Hierarchical reinforcement learning by discovering intrinsic options. In *9th International Conference on Learning Representations, ICLR*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section limitations

☑ A2. Did you discuss any potential risks of your work?
Section 5 and limitations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.