Self-imitation Learning for Action Generation in Text-based Games

Zijing Shi[1], Yunqiu Xu[1], Meng Fang[2], Ling Chen[1]
[1] University of Technology Sydney, NSW, Australia
zijing.shi@student.uts.edu.au, {yunqiu.xu,ling.chen}@uts.edu.au
[2] University of Liverpool, Liverpool, UK
Meng.Fang@liverpool.ac.uk

Abstract

In this work, we study reinforcement learning (RL) in solving text-based games. We address the challenge of combinatorial action space, by proposing a confidence-based self-imitation model to generate action candidates for the RL agent. Firstly, we leverage the self-imitation learning to rank and exploit past valuable trajectories to adapt a pre-trained language model (LM) towards a target game. Then, we devise a confidence-based strategy to measure the LM's confidence with respect to a state, thus adaptively pruning the generated actions to yield a more compact set of action candidates. In multiple challenging games, our model demonstrates promising performance in comparison to the baselines.

1 Introduction

Text-based games are situated systems where the game agents observe textual descriptions, and generate textual commands to interact with the environment. These games have proven to be suitable test-beds for studying various natural language processing (NLP) tasks, such as question answering (Yuan et al., 2019; Xu et al., 2022), dialogue systems (Ammanabrolu et al., 2022), situated language learning (Shridhar et al., 2020) and commonsense reasoning (Murugesan et al., 2021; Ryu et al., 2022). Recent years have witnessed the thriveness of designing reinforcement learning (RL) agents in solving these games (Narasimhan et al., 2015; Hausknecht et al., 2020; Ammanabrolu and Riedl, 2019; Xu et al., 2020b), while the combinatorial action space remains as a challenging issue, preventing RL agents from being deployed in real world applications.

In general, text-based games accept free-form actions, resulting in a large combinatorial action space. Fig. 1 shows a raw excerpt from the classic game "Zork1". A 4-word action has to be selected from $|\mathcal{V}|^4$ candidates, where $\mathcal{V}$ denotes the vocabulary set (Xu et al., 2020a). Given that only 130

Observation: The door reluctantly opens to reveal a rickety staircase descending into darkness.

Action: go down

Observation: The trap door crashes shut, and you hear someone barring it.

Action: go north

Observation: This is a small room with passages to the east and south and a forbidding hole leading west. A nasty-looking troll, brandishing a bloody axe, blocks all passages out of the room. Your sword has begun to glow very brightly.

Action: kill troll with sword

Observation: Almost as soon as the troll breathes his last breath, a cloud of sinister black fog envelops him, and when the fog lifts, the carcass has disappeared. Your sword is no longer glowing.

Figure 1: Excerpt from the game "Zork1". With a vocabulary size of 697, there are around $697^4 \approx 200$ billion potential 4-word actions in the game.

actions are required to solve this game, the agent wastes both training data and time in attempting irrelevant actions (Dulac-Arnold et al., 2015). To handle the combinatorial action space, early efforts either heavily rely on hand-crafted rules, or simply assume the availability of the action candidate set. For example, some works consider a set of currently admissible actions (He et al., 2016), or a template-based action space (Hausknecht et al., 2020). Alternatively, some other works alleviated this challenge by filtering inadmissible actions through methods such as action affordance (Jain et al., 2020), bandit-based elimination (Zahavy et al., 2018) and rule-based scoring (Ammanabrolu and Riedl, 2019).

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 703–726
May 2-6, 2023

In handling the combinatorial action space for text-based games, recent pre-trained language models (PLMs) (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020; Andreas and Klein, 2016) can help generate actions. However, the potential of LM is still less effectively explored. As one of the pioneer works, Yao et al. (2020) proposed the CALM, which is a GPT-2 model pre-trained on human gameplay trajectories, to generate the action candidate set for the RL agent. However, when solving a previously unseen game, CALM tends to generate actions with less satisfying qualities, leading to two consequences that may affect RL training: 1) the action set may contain a large proportion of inadmissible actions, and 2) the useful actions may not be generated. As a mitigation, the CALM model is set to generate a relatively huge action candidate set, followed by ad-hoc operations to filter out the inadmissible actions, which requires prior knowledge. Micheli and Fleuret (2021) extended the LM-based agent to goal-conditioned tasks to follow instructions. Besides the offline pre-training data, the LM is further improved with the successful trajectories collected during online interaction. However, text-based games do not have well-defined goals. Furthermore, some games are so challenging that it is impossible to collect successful trajectories (Tuyls et al., 2022).

In this work, we address the crux of combinatorial action space in solving text-based games. We propose the Confidence-based Self-imitation Model (CSM) to generate the action candidates for the RL agent.* Firstly, we leverage the self-imitation learning method (Oh et al., 2018) to rank and exploit past trajectories of high values to adapt a pre-trained LM towards the target game. Then, we propose a confidence-based strategy to measure the LM's confidence (Gandrabur et al., 2006) with respect to a state, thus adaptively pruning the action candidates based on the confidence value. Our model achieves promising performance in six challenging man-made games. Apart from significantly outperforming an action generation-based baseline, our strategy helps the RL agent to even achieve comparable performance to a baseline armed with the oracle action candidate set.

Our main contributions are summarized as follows: Firstly, we develop a LM-based framework to handle the issue of combinatorial action space in solving text-based games. Secondly, we propose a strategy to further improve the LM via self-imitation learning during the RL training. Thirdly, our experiments demonstrate that, the proposed method significantly improve the performance on multiple games compared with the strong contemporary method.

2 Related Work

2.1 RL Agents for Text-based Games

Inspired by the success of RL in playing games (Silver et al., 2016) and various NLP tasks (Fang et al., 2017; Yuan et al., 2019; Ammanabrolu et al., 2022), Narasimhan et al. (2015) and He et al. (2016) introduce RL to solve text-based games. Compared with non-learning-based agents (Hausknecht et al., 2019; Atkinson et al., 2019), the RL-based agents reduce the demand for extensive expert knowledge to develop gameplay strategies, and become the predominant modelling paradigm for solving text-based games. Subsequently, many variants of RL-based agents with different architectures and learning schemes have been proposed (Yuan et al., 2018; Jain et al., 2020; Guo et al., 2020; Xu et al., 2021; Tuyls et al., 2022; Shi et al., 2023). Innovations include modeling state space utilising knowledge graphs (Adhikari et al., 2020; Xu et al., 2020b), integrating question-answering and reading comprehension modules (Ammanabrolu et al., 2020; Xu et al., 2022). While these approaches focus on the problems of partial observability and language semantics, they still face the challenge of the combinatorial action space.

2.2 Combinatorial Action Space in TBGs

The combinatorial language-based action space is one primary challenge in solving text-based games. Early efforts mainly utilise hand-crafted rules or assume the agent has a predefined set of actions to choose from. For instance, the Jericho benchmark provides a valid action handicap that filters out inadmissible actions (i.e. actions that are either unrecognized by the game engine or do not change the underlying game state) at each game state (Hausknecht et al., 2020). This handicap has been widely used as the reduced action space by approaches like DRRN (He et al., 2016). In addition, the template-based action space is introduced where the agent selects first a template, and then a verb-object pair either individually (Hausknecht et al., 2020) or conditioned on the selected template (Ammanabrolu and Hausknecht, 2020). Even

*Code is available at https://github.com/winni18/CSM.

using the reduced action space, approaches filtering unnecessary actions can further improve the computational tractability and speed up the learning convergence (Zahavy et al., 2018; Jain et al., 2020).

2.3 Pre-training Methods for TBGs

Recent studies focus on enhancing the language understanding capability of agents by introducing pre-trained language processing modules. For instance, Singh et al. (2021) utilise the DistilBERT (Sanh et al., 2019) fine-tuned on human gameplay trajectories to represent game states. Ammanabrolu et al. (2020) employ the pre-trained ALBERT (Lan et al., 2019) to extract information from the textual observation by answering questions, and then update the knowledge graph during training. Adolphs and Hofmann (2020) use a pre-trained task-specific module to predict what is left to complete the tasks. In general, RL-based agents are initialised with knowledge using pre-trained modules before exploring game environments.

Some studies leverage pre-trained language models for action generation (Hausknecht et al., 2020) or word embeddings for affordance detection (Fulda et al., 2017). The approach closest to our work is Yao et al. (2020), which is state-of-the-art without requiring access to admissible actions. In their study, a GPT-2 language model trained on human gameplay trajectories is used to generate action candidates for the RL agent to select. To ensure that the correct actions are provided, the GPT-2 model is set to generate a relatively huge action candidate set, followed by ad-hoc operations to predict the admissibility of an action based on environmental feedback. In contrast, our work intends to narrow down the action space via self-imitation learning and make learning tractable.

3 Preliminaries

Text-based Games as POMDPs The text-based game can be formally formulated as a partially observable Markov Decision Process (POMDP) $(\mathcal{S}, T, \mathcal{A}, \mathcal{O}, R, \gamma)$. At each step t, the agent receives a textual observation $o_t \in \mathcal{O}$ from the game environment, while the latent state $s_t \in S$, which contains the complete internal information of the environment, could not be observed. By executing an action $a_t \in \mathcal{A}$, the environment will transit to the next state according to the latent transition function T, and the agent will receive the reward signal $r_t = R(s_t, a_t)$ and the next observation o_{t+1}. The objective of the agent is to take actions to maximize the expected cumulative discounted rewards $R_t = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$, where $\gamma \in [0, 1]$ is the discount factor.

Trajectory and Episode We define the trajectory τ as the sequence of observation-action pairs collected in an RL episode, i.e., $\tau = (o_1, a_1, o_2, a_2, \ldots, o_l, a_l)$, where l_τ is the length of τ. An RL episode is the process of an agent interacting with the environment from the beginning of a game to a termination state (e.g., the agent dies) or the step exceeding the pre-defined limit.

DRRN Existing RL methods for solving text-based games use game rewards to learn a value function. For instance, the Deep Reinforcement Relevance Network (DRRN) (He et al., 2016) is a choice-based game agent, where each action candidate a is paired with the state o to check its relevance. The agent then passes each pair through a deep neural network with parameters ϕ to estimate the Q-values $Q_\phi(o, a)$. The parameters ϕ of DRRN are trained using tuples (o, a, r, o') sampled from a prioritized experience replay buffer with the temporal difference (TD) loss:

$$\mathcal{L}_{\text{TD}}(\phi) = \left(r + \gamma \max_{a' \in A} Q_\phi \left(o', a' \right) - Q_\phi(o, a) \right)^2 \tag{1}$$

where r is the game reward and γ is the discount factor. The next action is then selected by softmax sampling the predicted Q-values:

$$\pi_\phi(a|o) = \frac{\exp\left(Q_\phi(o, a)\right)}{\sum_{a' \in A} \exp\left(Q_\phi\left(o, a'\right)\right)} \tag{2}$$

To circumvent the challenge of combinatorial action space, DRRN assumes access to the valid action handicap provided by the environment at each game state.

4 Methodology

4.1 Overview

To address the combinatorial action space, we propose the Confidence-based Self-imitation Model (CSM), which leverages the advantages of pre-trained LM and Self-imitation Learning (SiL) for adaptive action generation. Fig.2 shows an overview of CSM. At time step t, the LM is provided with the context $c_t = (o_{t-1}, a_{t-1}, o_t)$ as the input, and generates a set of action candidates $\mathcal{A}_t$

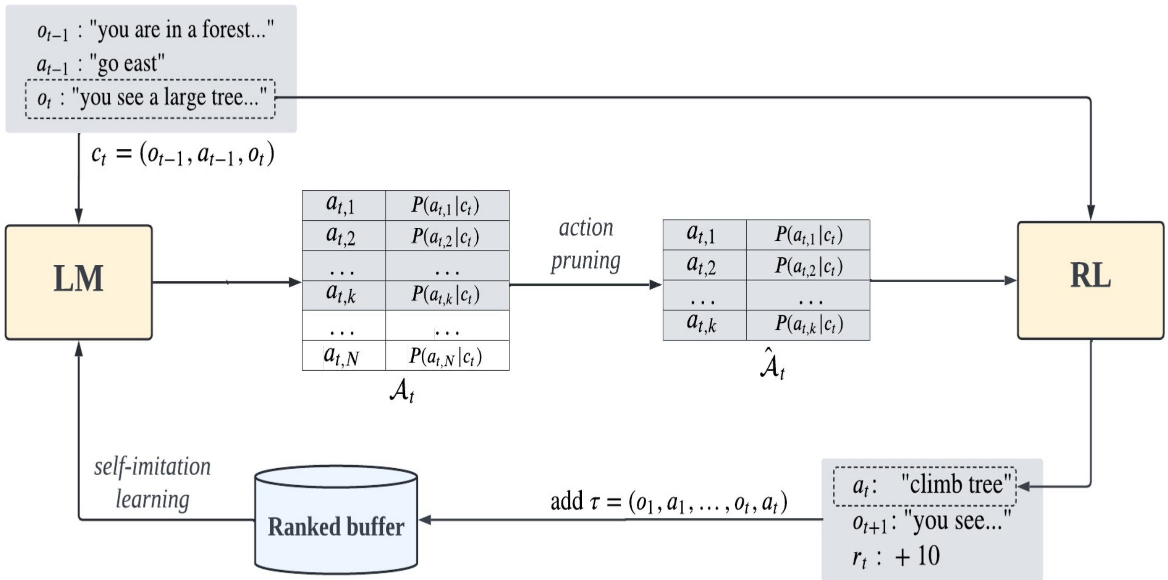

Figure 2: An overview of CSM. The LM takes the context c_t and generates action candidates $\mathcal{A}_t$, and conducts action pruning to further reduce the action space. The RL agent takes the observation o_t, and selects an action $a_t \in \hat{\mathcal{A}}_t$. The valuable trajectories τ are collected to further improve the LM through self-imitation learning.

as well as their probabilities using beam search decoding. Based on the probabilities, we conduct Action Pruning (AP) to obtain a more compact subset of action candidates $\hat{\mathcal{A}}_t \subseteq \mathcal{A}_t$ for the RL agent. Then the RL agent considers the observation o_t and selects an action $a_t \in \hat{\mathcal{A}}_t$. To generate high-quality actions which are more context-relevant, we adapt the LM towards the target game during the RL training. Specifically, we collect and then select the past valuable trajectories τ in an additional replay buffer, to further improve the LM through a self-imitation learning manner.

4.2 Self-imitation Learning

We follow the work of Yao et al. (2020) to utilize the LM for action generation. During pre-training, given human gameplay trajectories τ, we first build the context c_t, then train the LM to minimize the expected cross-entropy loss: $\mathcal{L}_{\text{LM}} = -\mathbb{E}[\log p(a|c)]$, where $\log p(a|c) = \sum_{i=1}^{m} p(a^i|a^{<i}, c)$ for an action with m tokens. During RL, the LM will serve as a "rough" action selector to generate the top-k actions. Then the RL agent will select one action to interact with the environment.

One drawback of the previous work Yao et al. (2020) is that when facing an unseen context, the LM may generate actions with poor performance. A straightforward solution is to continuously improve the LM during RL, thus making it adapted to the target game. Since no external trajectories (e.g., from human players) are available in the RL stage, we consider resorting to the self-imitation learning (Gangwani et al., 2019), i.e., letting the LM learn from the trajectories collected during the RL interaction. One thing we should pay attention to is the quality of the trajectories – sub-optimal trajectories may adversely affect imitation learning (Hu et al., 2019; Xu et al., 2022). Text-based games, especially games originally designed for human players, may be too challenging for agents to walk through. Thus, we cannot directly obtain successful trajectories during interacting with the environment. To alleviate this problem, we build a heap-like replay buffer to store past high-quality trajectories. We regard those obtaining higher scores with fewer steps as high-quality trajectories. Specifically, we rank trajectories within the replay buffer by their game scores (i.e., the sum of collected rewards) and lengths. In addition, we also take into account the novelty, by periodically replacing the old trajectories with new ones of equivalent qualities (e.g., the same scores and lengths).

4.3 Confidence-based Action Pruning

Through the aforementioned SiL, the LM is expected to generate a more reliable action candidate set $\mathcal{A}_t$ of size N. For each action $a_{t,i} \in \mathcal{A}_t$, we then calculate its normalized probability $P(a_{t,i}|c_t)$

Table 1: Game statistics.

Game	Avg.Action Number	Avg.Action Length	Avg.Steps Per Reward	Walkthough Length	Max Score
Balances	23.29	2.99	12	122	51
Inhumane	6.96	2.36	14	123	90
Ludicorp	14.52	2.76	4	364	150
Snacktime	5.68	2.14	8	34	50
Zork1	15.96	2.75	9	400	350
Ztuu	33.93	2.96	5	84	100

according to the beam search score. The probabilities exhibit two characteristics: 1) the long-tail phenomenon in linguistics (Zhan et al., 2021), where only a few probabilities produce lots of actions; 2) the probability distribution varies greatly under different states. Given these findings, we adopt a confidence-based strategy to further prune action candidates of low values, aiming to obtain a further reduced action candidate set $\hat{\mathcal{A}}_t \subseteq \mathcal{A}_t$. Specifically, we accumulate the probabilities of top-k action candidates as the confidence value: $Conf_t(k) = \sum_{i=1}^{k} P(a_{t,i}|c_t)$. We then conduct action pruning (i.e., constraining the action space k) by bonding the confidence value to a fixed, manually determined threshold ξ: $\hat{\mathcal{A}}_t = \{a_{t,i}|Conf_t(k) \leq \xi\}$. In this way, top-$k$ action candidates are selected adaptively. For a more "familiar" context c_t (e.g., it is similar to a context that LM has encountered before), the LM is supposed to be able to obtain correct actions from the training data, and the probability distribution will be centralised to top-ranked actions. In contrast, for an "unfamiliar" c_t, the actions' probabilities might be more uniformly-distributed. In this case, the size of action candidates (e.g., k) will be expanded to ensure a high confidence value.

5 Experiments

5.1 Experimental Setup

We conduct experiments upon six games provided by the Jericho Game Suite (Hausknecht et al., 2020). These games have diverse themes and genres, and each of them can represent a type of task. Different from those generated through pre-defined simple rules (Côté et al., 2018), the games we use are more complex, making them even challenging for human players. Some games contain nonstandard actions (e.g., the spells), which are unlikely to be understood by the language model pre-trained with commonsense knowledge. Table 1 shows the game statistics calculated from the walkthrough of each game.

5.2 Baselines

Our work focus on the challenge of combinatorial action space in text-based games. Thus, we compare CSM with two baselines:

- **CALM** (Yao et al., 2020), which is a pioneer work in LM-guided action generation.
- **DRRN** (He et al., 2016), which assumes access to the "oracle" action set (i.e., the valid action handicap provided by the environment).

Of these methods, CALM is the previous state-of-the-art model without the availability of "oracle" action sets, while the DRRN agent with "oracle" action sets can be regarded as our "upper bound".

5.3 Implementation Details

Training We implement CSM upon CALM's released code, including a pre-trained GPT-2 LM [†]. Both CSM and CALM adopt DRRN as the RL agent, except that $\mathcal{A}_t$ is obtained by LM. We set the step limit of an RL episode as 100, and train the RL agent on 8 parallel running environments for 100k steps. For each step, we train the RL agent with a batch size of 64, using an Adam optimizer with a learning rate of 1e-4. We set the first 20k steps as the warm-up phase, and start self-imitation learning as well as action pruning after this phase. For SiL, we use a trajectory buffer with a size of 50. For every 500 steps, we update the LM for 1 epoch with a batch size of 8, using an Adam optimizer with a learning rate of 2e-5. If there are no fresh trajectories as the training progresses, we conduct SiL using existing trajectories within the buffer. For AP, we use beam search decoding with a beam size of 40 to generate actions and choose

[†] https://github.com/princeton-nlp/calm-textgame

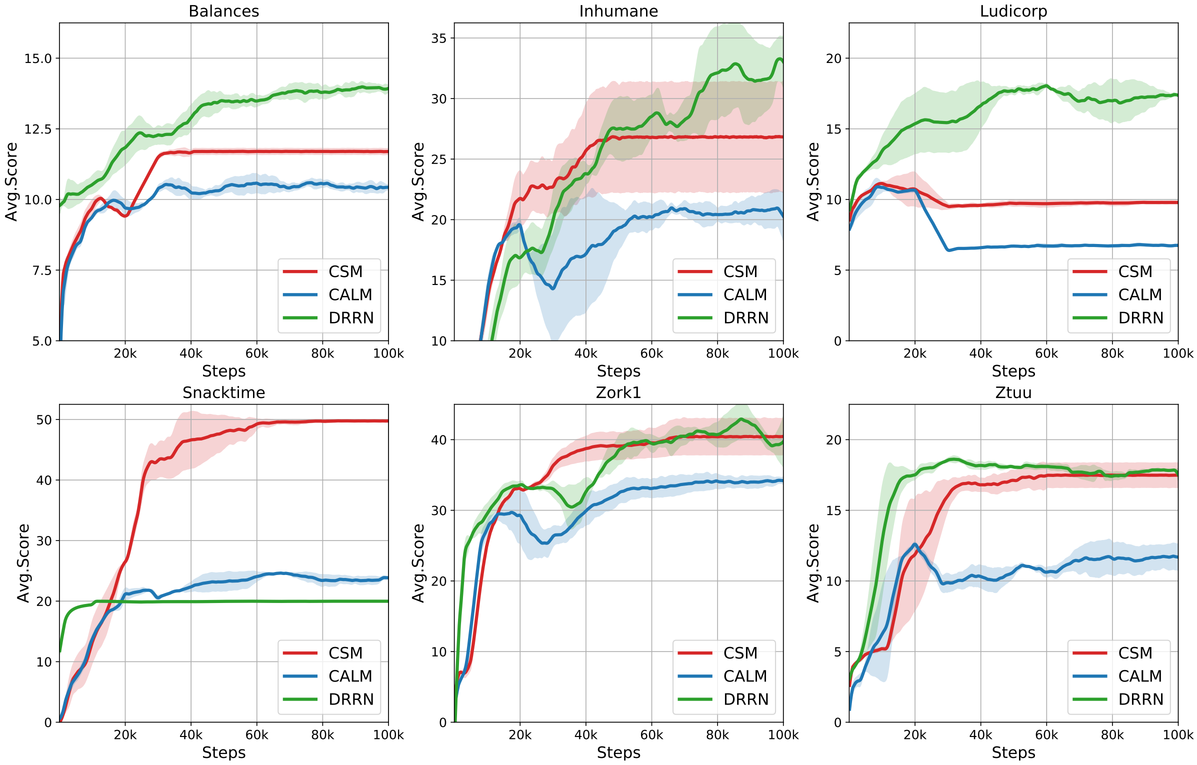

Figure 3: The performance of CSM compared to baselines (CALM and DRRN) throughout training. Shaded areas indicate one standard deviation. Our CSM outperforms CALM while getting much closer to DRRN. Besides, it successfully solves the game "Snacktime".

Table 2: The performance of CSM compared to baselines (CALM and DRRN) after training. The result with † is from Hausknecht et al. (2020). In six environments, our method obtains significant improvement compared to the CALM model, with an average normalized game score of 31.4%.

Game	*Generated* $\mathcal{A}_t$		*Oracle* $\mathcal{A}_t$	Max
	CSM	**CALM**	**DRRN**	
Balances	**11.7**	10.5	14.0	51
Inhumane	**27.0**	20.6	33.6	90
Ludicorp	**9.8**	6.8	17.5	150
Snacktime	**49.8**	24.0	20.0	50
Zork1	**40.6**	34.3	40.0	350
Ztuu	**17.5**	11.7	21.6†	100
Avg.Norm	**31.4%**	19.6%	24.9%	

the top 30 actions, i.e., $N = 30$. Then, we use the proposed confidence-based strategy to keep top-k highest-scoring action candidates (k<30). We set ξ as 0.6, and bound k to be no lower than 10. Following previous works, we define the score as the sum of rewards collected within an episode, and report the score averaged over the last 100 finished episodes.

LM For both CSM and CALM, we use the pre-trained GPT-2 model provided by Yao et al. (2020) as the LM module. The LM consists of 12 layers, 768 hidden sizes, and 12 attention heads. This module is first pre-trained on the WebText corpus (Radford et al., 2019), then re-trained on the ClubFloyd dataset (Yao et al., 2020), which consists of 426 human game playing transcripts on 590 games (note that the Jericho-supported games that we experiment with are not included).

RL Both CSM and CALM adopt the DRRN as the RL agent, except that the action candidate set is generated by the LM module. Given the current observation o_t, and a set of currently admissible actions $\mathcal{A}_t$, the RL agent first encodes o_t to build the state representation, then pairs it with each ac-

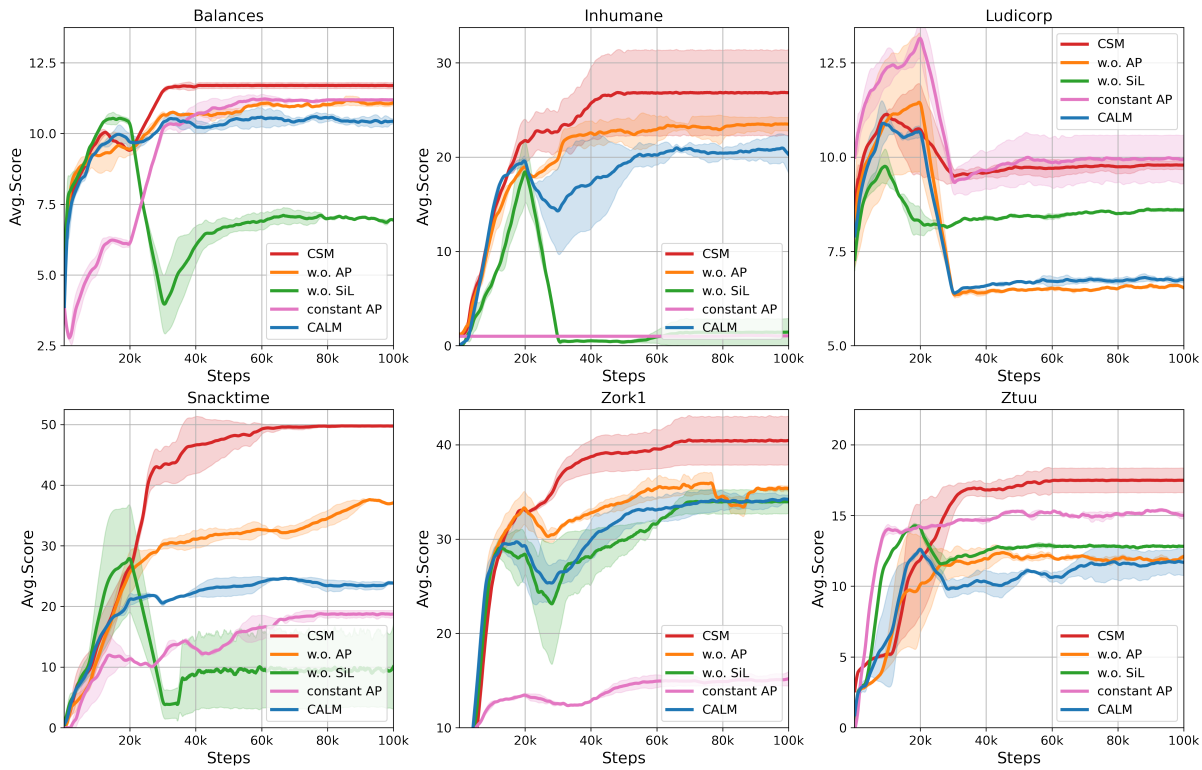

Figure 4: Average episode score throughout training for ablation models. Shaded areas indicate one standard deviation.

Table 3: Average episode score after training for ablation models. Overall, both the SiL and AP are crucial to our framework.

Game	CSM	w.o. AP	w.o. SiL	constant AP	CALM
Balances	**11.7**	11.1	7.0	11.2	10.5
Inhumane	**27.0**	23.6	1.5	1.1	20.6
Ludicorp	9.8	6.6	8.6	**10.0**	6.8
Snacktime	**49.8**	37.1	10.5	18.8	24.0
Zork1	**40.6**	35.6	34.1	15.2	34.3
Ztuu	**17.5**	12.1	12.9	15.1	11.7
Avg.Norm	**31.4%**	24.8%	10.8%	14.5%.	19.6%

tion candidate $a_{t,i} \in \mathcal{A}_t$ to compute the Q-value, which will be used as the probability for sampling the action a_t.

Warm-up Since this work does not address the RL exploration problem, we equip both CSM and CALM with a warm-up phase to facilitate training at the very beginning. During this phase, we follow Yao et al. (2020) to filter inadmissible actions from $\mathcal{A}_t$ through a pre-trained fast-text module, without applying SiL or AP. Then after this phase, the fast-text module will be discarded, and the LM has to generate the reliable $\mathcal{A}_t$ by itself. Note that this module is not essential, and could be replaced by other exploration strategies such as Zha et al. (2021); Yao et al. (2021). We leave such integration as a future direction.

5.4 Results

Fig. 3 shows the average episode score throughout training for the baselines, and Table 2 shows the average episode score after training for the baselines. Our CSM demonstrates its effectiveness by significantly outperforming the backbone CALM in all of the six games, with an average normalized game score of 31.4%. Given that DRRN has access to the "oracle" action set $\mathcal{A}_t$, its performance can be regarded as our "upper bound". We observe that the performance of CSM is much closer to DRRN, and even surpasses DRRN in two games. In particular, while DRRN gets stuck in the game "Snacktime", CSM solves this game, making its average normalized score among the highest of all. In Sec. 5.6, we further discuss this case by analyzing

Context: [CLS] scratch man [SEP] You scratch your pet, just gently, not to hurt him or anything. Hmm, your pet seems to have turned that into part of his dream or something, because he sure didn't move this time when you scratched him.right now you aren't carrying anything. (it's not like you need a lot of stuff, anyway.) [SEP] **CSM**: **jump on man**, kiss pet, lick pet, kiss man, lick man, push pet, move arm, pull pet, pull arm, push man **CALM**: pull man, push man, **jump on man**, push arm, pull pet, push pet, pull arm, lick man, kiss man, kiss pet, lick pet, smell, smell man, sniff, south, smell pet, wait, listen, east, northwest, southwest, get pet, down, jump, hug man, lick it, pet pet pet, up, hug pet, in, take pet, take all, eat pet, talk to pet, sleep, search pet, drop all, stand, open chest, open door, pet dog, pet pet dog, out, pull up on stool, west, north **DRRN**: kiss on pet, **jump on him**, north, west
Context: [CLS] push wand [SEP] Once again you chomp down to change what's inside the box. This time there are these guys poking at each other, kinda fighting and joking around and all. This is the sort of thing your pet seems to enjoy watching sometimes, people getting hit with food and slipping and falling down and stuff. Everybody is laughing and nobody ever seems to really get hurt, so you guess it's OK.held carefully between your teeth is a magic wand. [SEP] **CSM**: use wand on box, bite wand, push wand, **chew wand**, jump on man, take wand, get wand, read wand, pull wand, lick wand **CALM**: push wand, **chew wand**, wait, northeast, south, southwest, northwest, east, rub wand, down, give wand to pet, pull wand, open box, take magic wand, put wand in box, get wand, pull lever, give wand to guy, take all, get magic wand, west, take wand, up, out, eat wand, give book to pet, enter box, read book, use wand on wand, drop wand, north **DRRN**: examine legged, **push wand**, take inventory, get in all, take off legged, north, west

Figure 5: Sample gameplay from the game "Snacktime" along with the generated action candidates, and the action chosen by the RL agent (coloured with blue).

the underlying reasons.

5.5 Ablation Studies

In order to evaluate the contribution of the two components in CSM, we compare our model with two variants with either SiL ("w.o. AP") or AP ("w.o. SiL"). In order to demonstrate the effectiveness of confidence-based AP, we also employ constant AP. We set k to 12, which is the average number of actions selected by the confidence-based strategy. Fig. 4 shows the average episode score for the ablation models throughout training, and Table 3 shows the average episode score for the ablation models after training.

In general, adapting the LM with respect to the target game helps ("w.o. AP" v.s., "CALM"), while reducing the action space upon it further boosts the performance ("CSM" v.s., "w.o. AP"). Solely reducing the action space $\mathcal{A}_t$, in contrast, leads to poor performance ("CSM" v.s., "w.o. SiL" v.s., "CALM"). Also, simply utilizing the constant AP together with SiL results in a considerable performance drop. ("CSM" v.s., "constant AP"). Without SiL, the LM has a greater chance of incorrectly filtering actions that are essential to go through the target game.

5.6 Qualitative Analysis

To demonstrate the efficacy of the proposed framework, we present two gameplay examples from the game "Snacktime". Fig.5 shows the generated action candidates and the action chosen by the RL agent, where "Context" denotes c_t, "CSM" and "CALM" denote the actions generated by CSM and CALM respectively, "DRRN" denotes the "oracle" action set used by DRRN. In the first example, all models generate and select the correct action "jump on him", which leads to a +10 reward. Compared with CALM, CSM successfully reduces the action set from 30 to 10, relieving the burden for the RL agent. In the second example, both CSM and CALM generate action sets with the correct action "chew wand" being included. We found that the "oracle" action set provided by the environment is not always perfect, which explains why DRRN gets stuck here[‡]. It shows that our model is capable of generating high-quality, context-relevant actions, and further limits the action space while keeping key actions that may lead to higher scores in the games. Appendix B provides the detail interaction log of CSM on the game "Snacktime".

6 Conclusion

In this work, we studied reinforcement learning in solving the text-based game. We proposed the CSM framework to generate a set of action candidates for the RL agent, which alleviates the issue of combinatorial action space. During RL training, we

[‡]Similar phenomenon has also been reported in some other games (Tuyls et al., 2022)

collected and exploited past high-quality trajectories and utilised self-imitation learning to improve the language model. In addition, a confidence-based action pruning strategy was proposed to further restrict the action space. We evaluate our method using the Jericho benchmark. In a variety of text-based games, our method significantly improves the performance compared with the strong contemporary method, and even overcomes the challenging bottleneck in the game "Snacktime".

Limitations

In terms of limitations, text-based games are still far from being solved. Even if the agent has access to admissible actions, sparse rewards, language semantics and partial observability remain challenging obstacles for the existing game agent. In this study, we develop an effective framework to solve the issue of combinatorial action space. Future work can integrate our framework with methods that better leverage linguistic signals in order to make further progress in solving text-based games.

Acknowledgements

We thank the anonymous EACL reviewers for their insightful comments and constructive feedback.

References

Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and William L Hamilton. 2020. Learning dynamic knowledge graphs to generalize on text-based games. *arXiv preprint arXiv:2002.09127*.

Leonard Adolphs and Thomas Hofmann. 2020. Ledeepchef deep reinforcement learning agent for families of text-based games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7342–7349.

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

Prithviraj Ammanabrolu, Renee Jia, and Mark Riedl. 2022. Situated dialogue learning through procedural environment generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8099–8116.

Prithviraj Ammanabrolu and Mark Riedl. 2019. Playing text-adventure games with graph-based deep reinforcement learning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, volume 1, pages 3557–3565.

Prithviraj Ammanabrolu, Ethan Tien, Zhaochen Luo, and Mark O Riedl. 2020. How to avoid being eaten by a grue: Exploration strategies for text-adventure agents. *arXiv preprint arXiv:2002.08795*.

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Austin, Texas. Association for Computational Linguistics.

Timothy Atkinson, Hendrik Baier, Tara Copplestone, Sam Devlin, and Jerry Swan. 2019. The text-based adventure ai competition. *IEEE Transactions on Games*, 11(3):260–266.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2018. Textworld: A learning environment for text-based games. *arXiv preprint arXiv:1806.11532*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. 2015. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*.

Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 595–605.

Nancy Fulda, Daniel Ricks, Ben Murdoch, and David Wingate. 2017. What can you do with a rock? affordance extraction via word embeddings. *arXiv preprint arXiv:1703.03429*.

Simona Gandrabur, George Foster, and Guy Lapalme. 2006. Confidence estimation for nlp applications. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(3):1–29.

Tanmay Gangwani, Qiang Liu, and Jian Peng. 2019. Learning self-imitating diverse policies. In *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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7755–7765.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. 2020. Interactive fiction games: A colossal adventure. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 7903–7910.

Matthew Hausknecht, Ricky Loynd, Greg Yang, Adith Swaminathan, and Jason D Williams. 2019. Nail: A general interactive fiction agent. *arXiv preprint arXiv:1902.04259*.

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 Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1621–1630.

Hengyuan Hu, Denis Yarats, Qucheng Gong, Yuandong Tian, and Mike Lewis. 2019. Hierarchical decision making by generating and following natural language instructions. *Advances in Neural Information Processing Systems (NeurIPS)*, 32:10025–10034.

Vishal Jain, William Fedus, Hugo Larochelle, Doina Precup, and Marc G Bellemare. 2020. Algorithmic improvements for deep reinforcement learning applied to interactive fiction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 4328–4336.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Vincent Micheli and Francois Fleuret. 2021. Language models are few-shot butlers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9312–9318.

Keerthiram Murugesan, Mattia Atzeni, Pavan Kapanipathi, Pushkar Shukla, Sadhana Kumaravel, Gerald Tesauro, Kartik Talamadupula, Mrinmaya Sachan, and Murray Campbell. 2021. Text-based rl agents with commonsense knowledge: New challenges, environments and baselines. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 9018–9027.

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

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. 2018. Self-imitation learning. In *International Conference on Machine Learning (ICML)*, pages 3878–3887. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Dongwon Ryu, Ehsan Shareghi, Meng Fang, Yunqiu Xu, Shirui Pan, and Reza Haf. 2022. Fire burns, sword cuts: Commonsense inductive bias for exploration in text-based games. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 515–522.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Zijing Shi, Meng Fang, Yunqiu Xu, Ling Chen, and Yali Du. 2023. Stay moral and explore: Learn to behave morally in text-based games. In *International Conference on Learning Representations (ICLR)*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Ishika Singh, Gargi Singh, and Ashutosh Modi. 2021. Pre-trained language models as prior knowledge for playing text-based games. *arXiv preprint arXiv:2107.08408*.

Jens Tuyls, Shunyu Yao, Sham M. Kakade, and Karthik R Narasimhan. 2022. Multi-stage episodic control for strategic exploration in text games. In *International Conference on Learning Representations (ICLR)*.

Yunqiu Xu, Ling Chen, Meng Fang, Yang Wang, and Chengqi Zhang. 2020a. Deep reinforcement learning with transformers for text adventure games. In *IEEE Conference on Games (CoG)*, pages 65–72.

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 2021*, 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 (ACL)*, pages 538–560.

Yunqiu Xu, Meng Fang, Ling Chen, Yali Du, Joey Tianyi Zhou, and Chengqi Zhang. 2020b. Deep reinforcement learning with stacked hierarchical attention for text-based games. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 16495–16507.

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

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

Xingdi Yuan, Marc-Alexandre Côté, Jie Fu, Zhouhan Lin, Christopher Pal, Yoshua Bengio, and Adam Trischler. 2019. Interactive language learning by question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2796–2806.

Xingdi (Eric) Yuan, Marc-Alexandre Côté, Alessandro Sordoni, Romain Laroche, Remi Tachet des Combes, Matthew Hausknecht, and Adam Trischler. 2018. Counting to explore and generalize in text-based games. In *European Workshop on Reinforcement Learning (EWRL)*.

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 (NeurIPS)*, volume 31, pages 3562–3573.

Daochen Zha, Wenye Ma, Lei Yuan, Xia Hu, and Ji Liu. 2021. Rank the episodes: A simple approach for exploration in procedurally-generated environments. In *International Conference on Learning Representations (ICLR)*.

Zhiqiang Zhan, Jianyu Zhao, Yang Zhang, Jiangtao Gong, Qianying Wang, Qi Shen, and Liuxin Zhang. 2021. Grabbing the long tail: A data normalization method for diverse and informative dialogue generation. *Neurocomputing*, 460:374–384.

Appendix

The appendix is organized as follows: Sec. A shows more experiment results. Sec. B provides the interaction log of CSM on the game "Snacktime".

A More Results

Reproduction of DRRN Fig. 6 shows the reproducing result of the DRRN baseline, where "DRRN - Ours" denotes the "DRRN" used in the main paper. The dashed lines "DRRN - Official" denote the results reported in Hausknecht et al. (2020) and Yao et al. (2020). According to Tuyls et al. (2022), the action candidate set $\mathcal{A}_t$ provided by the environment is not always perfect, so that they manually augmented the environment-provided $\mathcal{A}_t$ with actions from the game walkthrough which are required for making progress[§]. We follow their setting to modify the environment and rerun the DRRN baseline, yielding much better performance than the official results except one game "Ztuu", which we use the official result in Table 2.

Reproduction of CALM Fig. 7 shows the reproducing result of the CALM baseline, where "CALM 20% WU - Ours" denotes the "CALM" used in the main paper. The dashed lines "CALM 100% WU - Official" denote the results reported in Yao et al. (2020). In terms of the original CALM, our replication results are comparable with or better than the official results ("CALM 100% WU" v.s., "CALM 100% WU - Official"). The original CALM adopts a fast-text model to filter out the inadmissible actions from $\mathcal{A}_t$ throughout the RL training process (i.e., they conduct warm-up for 100k steps), heavily alleviating the problem of generating inadmissible actions ("CALM 100% WU" v.s., "CALM w.o. WU"). However, obtaining this fast-text model requires prior knowledge, such as the additional training data and annotations. In our work, we would like to reduce the requirement of such external knowledge, and let the LM to conduct action pruning by itself. For all LM-based models, we only conduct warm-up for the first 20k steps, and discard the fast-text model afterwards ("CALM 20% WU"). As a future direction, we would like to consider more advanced warm-up strategies (Zha et al., 2021), thus eliminating the need for pre-training the fast-text model.

More results Besides the episode score, we provide more results for further analyzing self-imitation learning and action pruning. Regarding SiL, Fig. 8 and Fig. 9 show the average score and length of the trajectories collected in the ranked buffer, respectively. There's no doubt that the average score grows higher as the agent makes progress. Diverse trends could be observed in terms of the average length, since a newly-added trajectory might have both high score and more steps. Regarding AP, Fig. 10 shows the number of LM generated actions k, i.e., $|\hat{\mathcal{A}}_t|$, where it could be observed k gets close to the lower bound after pruning. Fig. 11 shows the LM probability of the top-1 generated action, and Fig. 12 shows the LM probability sum of the top-5 generated actions. After self-imitation learning, the top actions account for a larger proportion of the probability, making it safer for filtering those with low probabilities.

[§] https://github.com/princeton-nlp/XTX

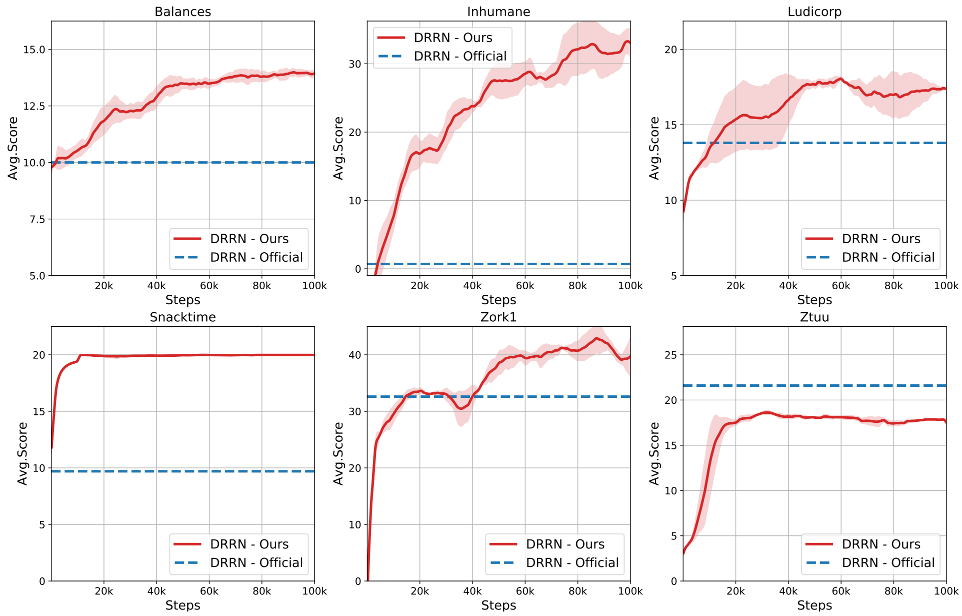

Figure 6: The reproducing result of the DRRN baseline, where "DRRN - Ours" denotes the "DRRN" used in the main paper. The dashed lines denote the results reported in Hausknecht et al. (2020) and Yao et al. (2020).

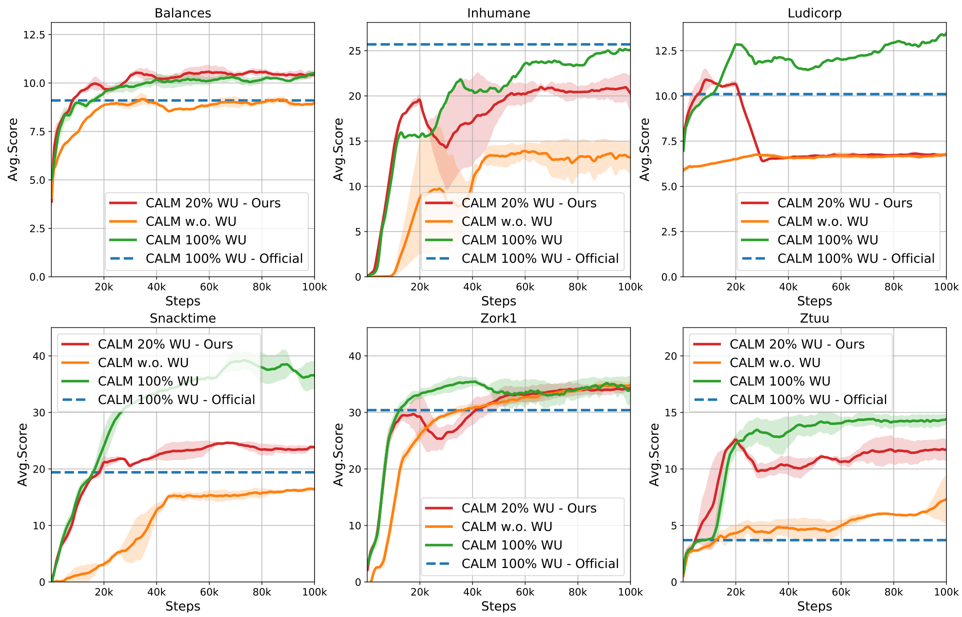

Figure 7: The reproducing result of the CALM baseline, where "CALM 20% WU - Ours" denotes the "CALM" used in the main paper. The dashed lines denote the results reported in Yao et al. (2020).

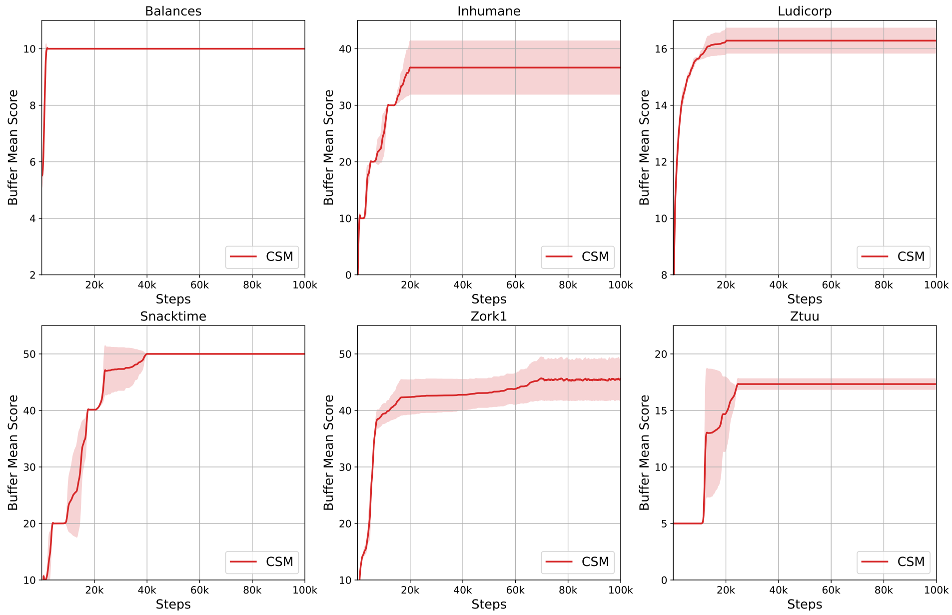

Figure 8: The average score of trajectories in the ranked buffer.

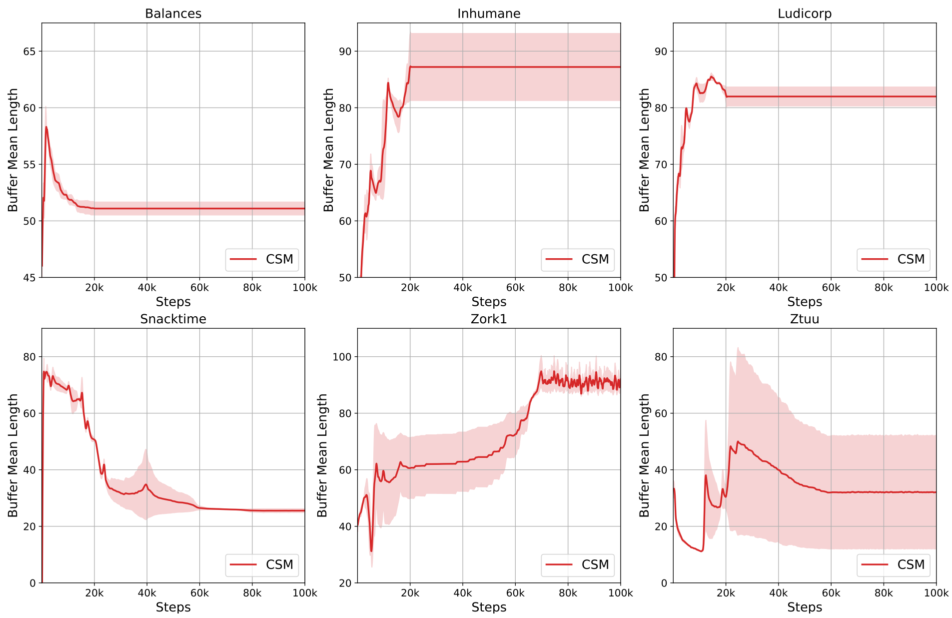

Figure 9: The average length of trajectories in the ranked buffer.

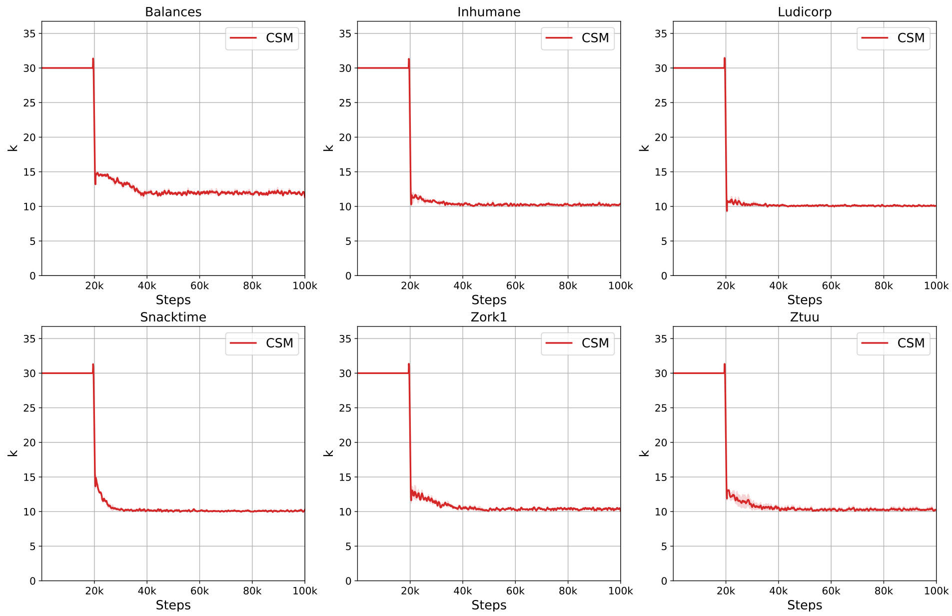

Figure 10: The number of LM generated actions k, i.e., $|\hat{\mathcal{A}}_t|$.

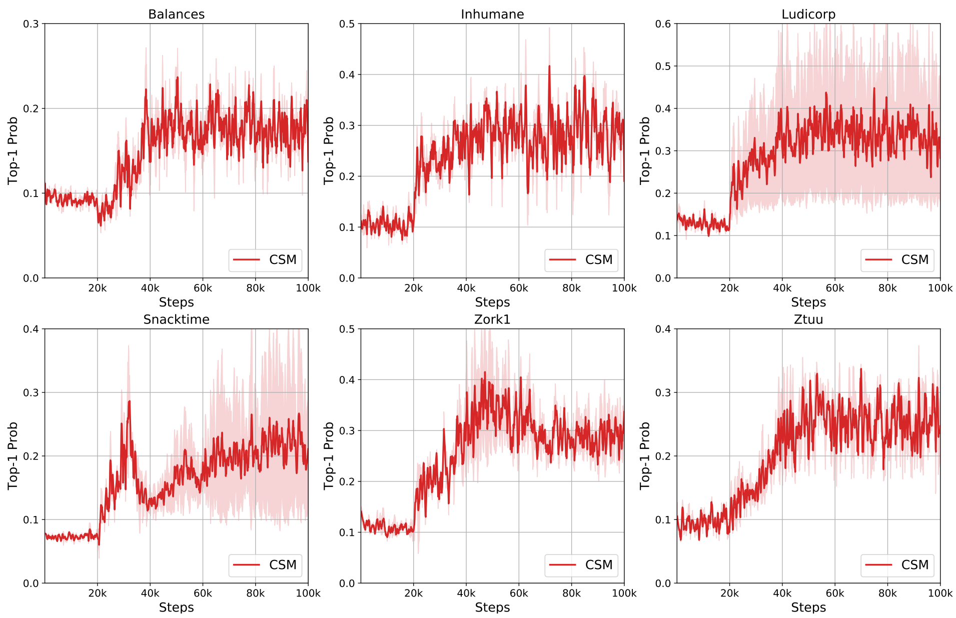

Figure 11: The LM probability of the top-1 generated action.

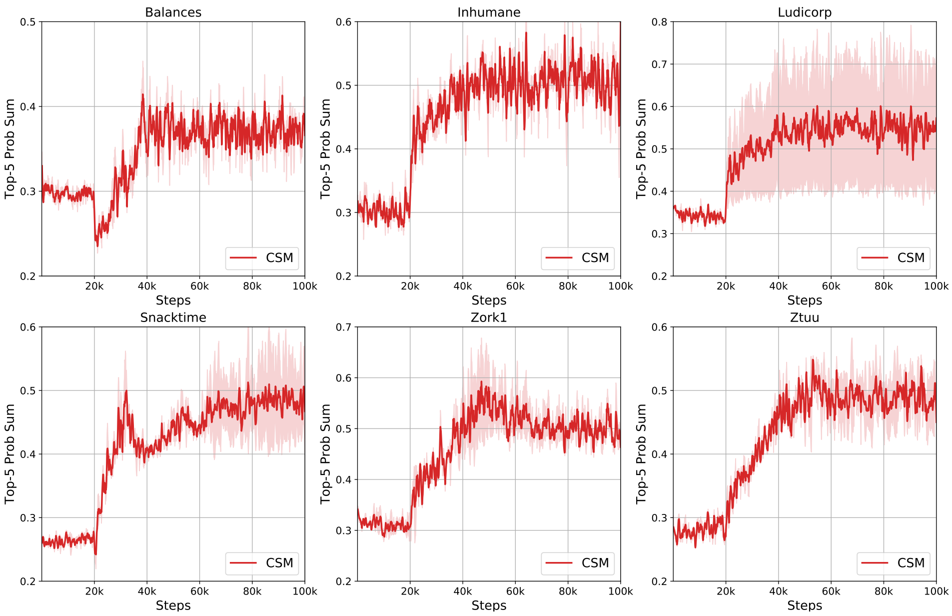

Figure 12: The LM probability sum of the top-5 generated actions.

B Interaction Example

We provide the interaction log of our CSM on the game "Snacktime", where "Context" denotes c_t, "Actions" denotes $\hat{\mathcal{A}}_t$, "Action" denotes a_t, "Reward" denotes r_t after executing a_t, "Score" denotes the episodic sum of the collected rewards, and "Done" indicates whether this episode is finished.

```
Context56826: [CLS] reset [SEP]   <RRROWWGRROWL> Your stomach
   growls.  You've been vegging out in front of that flickering
   screen for hours, your pet by your side. Now your complaining
   stomach sets you up on your feet, and on a mission.  Snack Time!
   An Interactive Break for a Bite by Hardy the Bulldog (with help
   from Renee Choba) Release 1 / Serial number 080929 / Inform 7
   build 5U92 (I6/v6.31 lib 6/12N)  (For more information please type
    ABOUT)  Sitting Room This is the room where you sit a lot. Well,
   you sleep here sometimes too. But there is a different room that
   is just for sleeping, and it is to the north. There's another room
    to the west. It's the room with the food. That is a good room.
   Even though this is the sitting room you can't sit on everything.
   There
Actions56826: ["'s no [cls]  [cl", "'s [cls]  [cls", "'s no [cls]  [
   cls", 'are [cls]  [cls]', 'is no [cls]  [cls', '[cls]  [cls', 'are
    no [cls]  [cls', 'are [cls]  [cls']
>> Action56826: are no [cls]  [cls
Reward56826: 0, Score 0, Done False

Context56827: [CLS] are no [cls]  [cls [SEP]  That's not a trick you
    know.right now you aren't carrying anything. (it's not like you
   need a lot of stuff, anyway.) [SEP]
Actions56827: ['move arm', 'scratch pet', 'lick pet', 'kiss pet', '
   jump on man', 'scratch man', 'pull man', 'pull pet', 'push pet', '
   pull arm']
>> Action56827: scratch man
Reward56827: 0, Score 0, Done False

Context56828: [CLS]  That's not a trick you know. [SEP]  scratch man
   [SEP]  You scratch your pet, just gently, not to hurt him or
   anything.  Your pet twitches a little and stirs, then is still
   again. It's a start at waking him up.right now you aren't carrying
    anything. (it's not like you need a lot of stuff, anyway.) [SEP]
Actions56828: ['move arm', 'lick man', 'chew man', 'jump on man', '
   push pet', 'push man', 'lick pet', 'pull man', 'pull pet', 'pull
   arm']
>> Action56828: push pet
Reward56828: 0, Score 0, Done False

Context56829: [CLS]  push pet [SEP]  You tuck your head up under your
    pet and push back and forth to shake him.  Your pet gets one eye
   open and half heartedly pets at you. You're making progress.right
   now you aren't carrying anything. (it's not like you need a lot of
    stuff, anyway.) [SEP]
Actions56829: ['pull man', 'jump on man', 'lick man', 'scratch pet',
   'lick pet', 'scratch man', 'pull pet', 'pull arm', 'move arm', '
   push pet']
```

```
>> Action56829: scratch man
Reward56829: 0, Score 0, Done False

Context56830: [CLS] scratch man [SEP] You scratch your pet, just
   gently, not to hurt him or anything. Hmm, your pet seems to have
   turned that into part of his dream or something, because he sure
   didn't move this time when you scratched him.right now you aren't
   carrying anything. (it's not like you need a lot of stuff, anyway
   .) [SEP]
Actions56830: ['jump on man', 'kiss pet', 'lick pet', 'kiss man', '
   lick man', 'push pet', 'move arm', 'pull pet', 'pull arm', 'push
   man']
>> Action56830: jump on man
Reward56830: 10, Score 10, Done False

Context56831: [CLS] jump on man [SEP] You jump up and get your paws
   on your pet for a second before sliding back down. Your pet
   shifts and stretches, blinking. He moves around until he is mostly
   sitting up. "Hey Hardy," he mumbles. It might take a little
   more to get him up and moving, but at least now his eyes are open!
   [Your score has just gone up by ten points.]right now you aren't
   carrying anything. (it's not like you need a lot of stuff, anyway
   .) [SEP]
Actions56831: ['kiss man', 'south', 'in', 'out', 'southeast', '
   northwest', 'northeast', 'southwest', 'east', 'down']
>> Action56831: down
Reward56831: 0, Score 10, Done False

Context56832: [CLS] down [SEP] If you want to go to the food room,
   it's to the west, and the sleeping room is to the north.right now
   you aren't carrying anything. (it's not like you need a lot of
   stuff, anyway.) [SEP]
Actions56832: ['search soft thing', 'search couch', 'search sofa', '
   search door', 'search sleeping', 'search box', 'search arm', '
   search pet', 'search man', 'search fridge']
>> Action56832: search soft thing
Reward56832: 0, Score 10, Done False

Context56833: [CLS] search soft thing [SEP] You tug and pull at the
   little soft things until finally one breaks free from the bunch.
   As you dislodge it, something clatters to the floor. Ah ha! You've
   found the magic wand! Those little soft things were trying to eat
   it! "So that's where that got away to," says your pet, snatching
   up the magic wand. He points it at the box of light and makes it
   even louder.held carefully between your teeth is a square soft
   thing. [SEP]
Actions56833: ['west west', 'west west west west west west west west
   west', 'west west west west west west west west west west', 'west
   west west west west west west west', 'west west west west west
   west west', 'west west west west west west', 'west west west', '
   west west west west west', 'west west west west', 'west']
```

```
>> Action56833: west west west west west west west west west west
Reward56833: 0, Score 10, Done False

Context56834: [CLS]  west west west west west west west west west
   west [SEP]   Food Room Food room, glorious food room!  Your bowl
   for food and water is here, but that's only the beginning. There's
    also a lot of secret places where your pet keeps food and things
   that go with food and things that make food better. There's a
   thing that gets hot and a box that stays cold, and that is your
   favorite because you know there is always food inside there. A
   cloth hangs down from the door of the cold box.held carefully
   between your teeth is a square soft thing. [SEP]
Actions56834: ['open cold box', 'open box', 'open door', 'get cloth',
    'open fridge', 'take cloth', 'pull cloth', 'open refrigerator', '
   use wand on box', 'push cloth']
>> Action56834: open refrigerator
Reward56834: 0, Score 10, Done False

Context56835: [CLS]  open refrigerator [SEP]  This is a good trick!
   (As you open your mouth, you drop the square soft thing.)  You
   pull on the cloth with your teeth and back up as the door opens.
   Inside you can see various leftovers, some jars, some sandwich
   fixings and one shiny can.right now you aren't carrying anything.
   (it's not like you need a lot of stuff, anyway.) [SEP]
Actions56835: ['get can', 'take can', 'get all from fridge', 'take
   all from fridge', 'take all', 'get all', 'get all from
   refrigerator', 'take all from refrigerator', 'get everything', '
   take everything']
>> Action56835: take all from fridge
Reward56835: 0, Score 10, Done False

Context56836: [CLS]  take all from fridge [SEP]  leftovers: All the
   food in the cold box is up too high for you or closed away. Looks
   like you will need your pet to help you get a snack. jars: Jars
   are no good for you, they just tease you by letting you see what's
    inside. But you can't get them open without your pet's help
   anyway. sandwich fixings: All the food in the cold box is up too
   high for you or closed away. Looks like you will need your pet to
   help you get a snack. shiny can: You snatch up the shiny can with
   your mouth and hold on tight.held carefully between your teeth is
   one shiny can. [SEP]
Actions56836: ['east east', 'east', 'east east east', 'east east east
    east', 'east east east east east', 'east east east east east east
   ', 'east east east east east east east', 'east east east east east
    east east east', 'east east east east east east east east east',
   'east east east east east east east east east east']
>> Action56836: east east
Reward56836: 0, Score 10, Done False

Context56837: [CLS]  east east [SEP]  Before you leave the food room,
    you push the cold box door with your nose to close it. (It's
```

```
   important to conserve energy.)  Sitting Room Your pet is here,
   staring at the box of light, holding on tight to the magic wand.
   held carefully between your teeth is one shiny can. [SEP]
Actions56837: ['feed pet', 'feed man', 'give can to pet', 'give can
   to man', 'wait', 'push wand', 'east', 'feed feed man', 'take can',
    'down']
>> Action56837: give can to man
Reward56837: 10, Score 20, Done False

Context56838: [CLS]  give can to man [SEP]  Your pet smiles as he
   reaches for the can. "Good boy, Hardy," he says, scratching you
   behind the ears. Your tail starts wagging; you just can't help it.
    (You've always been a fool for a good behind the ears scratch.)
   The can hisses with releasing air as the top pops, and then your
   pets takes a few good gulps. He must have been really thirsty,
   because soon he stands up and walks off to the other room.  From
   the other room, you hear your pet banging around and yelling about
    the dino. Your pet must have almost tripped on it. He's always
   almost tripping on little stuff like that.  After a second, you
   hear a click as the door to the water room closes. Now you have
   the long soft thing all to yourself!  [Your score has just gone up
    by ten points.]right now you aren't carrying anything. (it's not
   like you need
Actions56838: ['to move', 'all', 'pet', 'your pet', 'much', 'a pet',
   'anything', 'you', 'it', 'any', 'anymore']
>> Action56838: pet
Reward56838: 0, Score 20, Done False

Context56839: [CLS]  pet [SEP]  That's not a trick you know.right now
    you aren't carrying anything. (it's not like you need a lot of
   stuff, anyway.) [SEP]
Actions56839: ['get all', 'take all', 'take everything', 'get
   everything', 'take pet', 'get pet', 'take all from refrigerator',
   'take all from fridge', 'get can', 'take can']
>> Action56839: get everything
Reward56839: 0, Score 20, Done False

Context56840: [CLS]  That's not a trick you know. [SEP]  get
   everything [SEP]  magic wand: You snatch up the magic wand with
   your mouth and hold on tight.held carefully between your teeth is
   a magic wand. [SEP]
Actions56840: ['use wand on box', 'push wand', 'bite wand', 'chew
   wand', 'take wand', 'get wand', 'read wand', 'pull wand', 'lick
   wand', 'east']
>> Action56840: chew wand
Reward56840: 0, Score 20, Done False

Context56841: [CLS]  chew wand [SEP]  Making sure you are facing the
   box of light, you carefully press down on the wand with your teeth
    and are delighted when the box of light flickers and changes!
   Now you see a bunch of tiny people running around chasing a little
```

```
    bitty ball. Uh oh, someone got pushed. Ugh, the faces got all
   huge, and they don't look happy.held carefully between your teeth
   is a magic wand. [SEP]
Actions56841: ['use wand on box', 'bite wand', 'push wand', 'chew
   wand', 'scratch wand', 'take wand', 'get wand', 'read wand', 'pull
    wand', 'lick wand']
>> Action56841: push wand
Reward56841: 0, Score 20, Done False

Context56842: [CLS]  push wand [SEP]  Once again you chomp down to
   change what's inside the box. This time there are these guys
   poking at each other, kinda fighting and joking around and all.
   This is the sort of thing your pet seems to enjoy watching
   sometimes, people getting hit with food and slipping and falling
   down and stuff. Everybody is laughing and nobody ever seems to
   really get hurt, so you guess it's OK.held carefully between your
   teeth is a magic wand. [SEP]
Actions56842: ['use wand on box', 'bite wand', 'push wand', 'chew
   wand', 'jump on man', 'take wand', 'get wand', 'read wand', 'pull
   wand', 'lick wand']
>> Action56842: chew wand
Reward56842: 10, Score 30, Done False

Context56843: [CLS]  chew wand [SEP]  You've got the hang of this
   trick now for sure! One more bite on the wand brings up something
   new.  This is more like it! Inside the box now is this lady
   putting together some tasty looking food! You're almost drooling
   just watching. This has got to make your pet think about eating!
   [Your score has just gone up by ten points.]held carefully between
    your teeth is a magic wand.  you finally hear a familiar rush of
   water from the little water room and, after a second, the sound of
    the door opening. [SEP]
Actions56843: ['use wand on box', 'south', 'southeast', 'northwest',
   'southwest', 'out', 'northeast', 'west', 'take wand', 'east']
>> Action56843: northwest
Reward56843: 0, Score 30, Done False

Context56844: [CLS]  northwest [SEP]  If you want to go to the food
   room, it's to the west, and the sleeping room is to the north.
   You finally hear a familiar rush of water from the little water
   room and, after a second, the sound of the door opening.held
   carefully between your teeth is a magic wand.  from the other room
   , you hear your pet banging around and yelling about the dino.
   your pet must have almost tripped on it. he's always almost
   tripping on little stuff like that.  your pet walks in. he starts
   to sit down on the long soft thing but stops when he sees what's
   in the box of light. that lady cooking is doing her job! you pet
   stands and watches, transfixed. he must be thinking of food! [SEP]
Actions56844: ['out', 'southeast', 'down', 'south', 'northwest', '
   southwest', 'northeast', 'wait', 'east', 'take wand']
>> Action56844: northeast
```

```
Reward56844: 0, Score 30, Done False

Context56845: [CLS]  northeast [SEP]  If you want to go to the food
   room, it's to the west, and the sleeping room is to the north.
   From the other room, you hear your pet banging around and yelling
   about the dino. Your pet must have almost tripped on it. He's
   always almost tripping on little stuff like that.  Your pet walks
   in. He starts to sit down on the long soft thing but stops when he
    sees what's in the box of light. That lady cooking is doing her
   job! You pet stands and watches, transfixed. He must be thinking
   of food!held carefully between your teeth is a magic wand.  your
   pet makes his way to the food room.  your pet starts rattling
   around in the food room. yes! he must be making a snack! [SEP]
Actions56845: ['west', 'southeast', 'down', 'northwest', 'south', '
   southwest', 'east', 'northeast', 'out', 'wait']
>> Action56845: southwest
Reward56845: 0, Score 30, Done False

Context56846: [CLS]  southwest [SEP]  If you want to go to the food
   room, it's to the west, and the sleeping room is to the north.
   Your pet makes his way to the food room.  Your pet starts rattling
    around in the food room. Yes! He must be making a snack!held
   carefully between your teeth is a magic wand.  from the other room
    you can hear the happy sound of the cold box being opened. [SEP]
Actions56846: ['down', 'south', 'southeast', 'wait', 'northwest', '
   east', 'take wand', 'northeast', 'out', 'southwest']
>> Action56846: east
Reward56846: 0, Score 30, Done False

Context56847: [CLS]  east [SEP]  If you want to go to the food room,
   it's to the west, and the sleeping room is to the north.  From the
    other room you can hear the happy sound of the cold box being
   opened.held carefully between your teeth is a magic wand.  from
   the other room you can hear the sound of secret places opening and
    closing. [SEP]
Actions56847: ['southeast', 'northwest', 'south', 'out', 'northeast',
    'take wand', 'down', 'east', 'southwest', 'read wand']
>> Action56847: read wand
Reward56847: 0, Score 30, Done False

Context56848: [CLS]  read wand [SEP]  A magical, bumpy thing your pet
    uses to control the box of light.  From the other room you can
   hear the sound of secret places opening and closing.held carefully
    between your teeth is a magic wand.  you don't hear anything from
    the food room. but then, making a sandwich is not a very noisy
   job. [SEP]
Actions56848: ['take wand', 'southeast', 'northwest', 'south', '
   northeast', 'down', 'out', 'east', 'read wand', 'southwest']
>> Action56848: east
Reward56848: 0, Score 30, Done False
```

```
Context56849: [CLS]  east [SEP]  If you want to go to the food room,
   it's to the west, and the sleeping room is to the north.  You don'
   t hear anything from the food room. But then, making a sandwich is
    not a very noisy job.held carefully between your teeth is a magic
    wand.  you don't hear anything from the food room. but then,
   making a sandwich is not a very noisy job. [SEP]
Actions56849: ['south', 'down', 'take wand', 'southeast', 'northwest
   ', 'out', 'northeast', 'southwest', 'read wand', 'east']
>> Action56849: down
Reward56849: 0, Score 30, Done False

Context56850: [CLS]  down [SEP]  If you want to go to the food room,
   it's to the west, and the sleeping room is to the north.  You don'
   t hear anything from the food room. But then, making a sandwich is
    not a very noisy job.held carefully between your teeth is a magic
    wand.  from the other room, you hear the sound of the cold box
   closing. you're sure that in just a minute your pet will be
   leaving the food room with the sandwich! [SEP]
Actions56850: ['northwest', 'southeast', 'south', 'down', 'northeast
   ', 'southwest', 'out', 'read wand', 'east', 'take wand']
>> Action56850: northwest
Reward56850: 0, Score 30, Done False

Context56851: [CLS]  northwest [SEP]  If you want to go to the food
   room, it's to the west, and the sleeping room is to the north.
   From the other room, you hear the sound of the cold box closing.
   You're sure that in just a minute your pet will be leaving the
   food room with the sandwich!held carefully between your teeth is a
    magic wand. [SEP]
Actions56851: ['west west west west west  west', 'west west west west
     west', 'west west west  west', 'west west  west', 'west west
   west west west west west', 'west west west west west west', 'west
   west west west west', 'west west west west', 'west west west', '
   west west', 'west west west  west  west', 'west west  west  west']
>> Action56851: west west west
Reward56851: 0, Score 30, Done False

Context56852: [CLS]  west west west [SEP]   Food Room Your pet is
   here, working on making a snack.  You can also see a square soft
   thing here.  Your pet seems to have finished cleaning up after
   making the snack. He turns to leave. Now is the moment of truth.
   held carefully between your teeth is a magic wand.  sandwich in
   hand, your pet heads for the sitting room.  you follow your pet as
    he makes his way to the long soft thing and starts munching away
   on the sandwich. in a moment, it is gone, along with your dream of
    the perfect snack. looks like you'll have to wait until dinner to
    eat. poor hungry puppy!           you have missed your chance to
    snack       in that game you scored 30 out of a possible 50, in
   25 turns.   would you like to restart, restore a saved game or
   quit [SEP]
```

```
Actions56852: ['give sandwich to pet', 'give sandwich sandwich to pet
   ', 'give sandwich sandwich sandwich to pet', 'give sandwich
   sandwich sandwich sandwich sandwich sandwich sandwich to pet', '
   give sandwich sandwich sandwich sandwich sandwich sandwich to pet
   ', 'give sandwich sandwich sandwich sandwich sandwich to pet', '
   give sandwich sandwich sandwich sandwich to pet', 'give sandwich
   sandwich sandwich sandwich sandwich sandwich sandwich sandwich
   sandwich', 'give sandwich sandwich sandwich sandwich sandwich
   sandwich sandwich sandwich to']
>> Action56852: give sandwich sandwich sandwich sandwich sandwich
   sandwich sandwich to pet
Reward56852: 20, Score 50, Done True
```