Story Shaping: Teaching Agents Human-Like Behavior with Stories

Xiangyu Peng*, Christopher Cui*, Wei Zhou, Renee Jia, Mark Riedl

Georgia Institute of Technology
{xpeng62, ccui46, wzhou322, rjia35}@gatech.edu, riedl@cc.gatech.edu

Abstract

Reward design for reinforcement learning agents can be difficult in situations where one not only wants the agent to achieve some effect in the world but where one also cares about *how* that effect is achieved. For example, we might wish for an agent to adhere to a tacit understanding of commonsense, align itself to a preference for how to behave for purposes of safety, or take on a particular role in an interactive game. Storytelling is a mode for communicating tacit procedural knowledge. We introduce a technique, *Story Shaping*, in which a reinforcement learning agent infers tacit knowledge from an exemplar story of how to accomplish a task and intrinsically rewards itself for performing actions that make its current environment adhere to that of the inferred story world. Specifically, Story Shaping infers a knowledge graph representation of the world state from observations, and also infers a knowledge graph from the exemplar story. An intrinsic reward is generated based on the similarity between the agent's inferred world state graph and the inferred story world graph. We conducted experiments in text-based games requiring commonsense reasoning and shaping the behaviors of agents as virtual game characters.

1 Introduction

Reinforcement Learning (RL) is a class of techniques whereby an agent learns how to carry out a sequential task through repeated interaction with the environment. The agent is given a reward for achieving certain states in the environment, executing certain actions, or causing the world to transition between particular pairs of states. RL is thus a process of learning to maximize expected future rewards. The design of the reward signal—when the reward (or penalty) is given and how much—determines what the optimal behavior for the agent should be.

Reward design can be especially difficult in situations where one not only wants the agent to achieve some effect in the world but where we also care about *how* that effect is achieved. For example, we may want the agent to carry out the task in a way that adheres to social norms during execution. This can make it safer for humans to work with AI-driven systems because they conform to our expectations and preferences (Frazier et al. 2020; Nahian et al. 2021; Ammanabrolu et al. 2022). We may wish for our agents to demonstrate commonsense knowledge during execution (Dambekodi et al. 2020). In computer game environments, we may wish to have AI-driven agents role-play different types of characters with different interests or ways of accomplishing things (Urbanek et al. 2019; Ammanabrolu et al. 2021).

Figure 1: Excerpt from the text game, `9:05`, with actions from vanilla reinforcement learning agent represented by 🤖, and RL agent with Story Shaping by 🤖. 🤖 takes action "*go west*", which only maximizes the expected rewards and ignores the commonsense knowledge in the environment. 🤖 instead chooses to do something more similar to what a human would do—eat a Pop-Tart—based on a story written in natural language, which may be provided or created automatically by our system.

We introduce a new technique, **Story Shaping**, for specifying preferences over an agent's behavior. In Story Shaping, a reward designer specifies *how* to perform a task by providing an example story. Stories are efficient means

*These authors contributed equally.

for transferring tacit procedural knowledge between people. Storytelling is a mode of communication wherein details are abstracted away under the assumption that the recipient shares a common base of knowledge with which to reconstruct details as necessary for comprehension (Hedlund, Antonakis, and Sternberg 2002).

Story Shaping is a process whereby the agent reverse-engineers implicit state information from the exemplar story and constructs a rich, *intrinsic* reward signal that guides it toward behavior that makes its environment more closely resemble the implicit world of the story. Specifically, the agent infers a *knowledge graph* from the story, consisting of $\langle subject, relation, object \rangle$ triples for every relation that can be inferred from the story. The agent also extracts relations from its observations to construct a knowledge graph summarizing the operating environment. The agent rewards itself for how similar its current world state representation is from the desired representation derived from the exemplar story.

Story shaping is related to *Learning from Demonstrations* (LfD). In LfD, the agent is provided with a set of traces, typically enacted by humans in the same environment. The learning agent must reconstruct the latent policy that human demonstrators were following. Demonstrations are typically assumed to be complete (no missing steps) and performed in the same or very similar environment that the agent also operates in. In contrast, Story Shaping does not assume the stories are complete nor executable, and does not assume they reference the same environment. Instead, stories may reference the essential objects, locations, or activities but leave out details. Additionally, because stories do not reference the exact environment the agent inhabits, objects and locations may differ or be missing, and actions may not be executable.

We experiment with Story Shaping in text games where observations and actions are presented entirely in text. Text games are partially observable environments that have large state and action spaces and often involve puzzles requiring long-range causal dependencies (Hausknecht et al. 2020a). Text games have also been demonstrated to transfer to visual and real-world domains (Wang et al. 2022; Shridhar et al. 2021). We conduct experiments across three game platforms that either (a) require commonsense, or (b) showed that our agent is able to role-play in a more human-like manner. Additionally, we show that story-shaped agents can adapt their behavior to different character preferences.

2 Background and Related Work

Text games are turn-based games where the player must read human-written natural language (typically English) descriptions of the local environment and respond with short textual action descriptions. A text game can be defined as a partially-observable Markov Decision Process: $\langle S, P, A, O, \Omega, R, \gamma \rangle$, representing the set of environment states, conditional transition probabilities between states, the vocabulary or words used to compose text commands, observations, observation conditional probabilities, reward function, and discount factor, respectively. The transition and observation probabilities, P and Ω, are typically unknown to the agent. Observations are text sequence, and actions are composed of one to five tokens from a vocabulary. The RL agent attempts to learn a policy $\pi(o) \rightarrow a$ that maximize future expected reward.

Knowledge Graphs for Text Games. A knowledge graph is a set of $\langle subject, relation, object \rangle$ tuples. Knowledge-graph based reinforcement learning agents have been shown to be state-of-the-art in text-based games (Ammanabrolu and Riedl 2018; Ammanabrolu et al. 2020a; Ammanabrolu and Hausknecht 2020; Ammanabrolu et al. 2020b; Xu et al. 2020; Peng, Riedl, and Ammanabrolu 2022). These agents infer objects and relations from text observations and use this knowledge graph as a long-term memory of the world state as a means of handling partial observability. We build off the KG-A2C (Ammanabrolu and Hausknecht 2020) agent architecture, which uses the ALBERT (Lan et al. 2019) language model to infer objects and relations from the text observation, and a graph attention network to generate action sequences. Whereas KG-A2C uses the knowledge graph to represent world state and filter actions, our Story Shaping approach also uses the KGs to compute a dense reward signal, showing that KG-based reinforcement learning has additional untapped potential.

Intrinsic rewards. Intrinsic rewards provide qualitative guidance (Schmidhuber 1991; Oudeyer, Kaplan, and Hafner 2007; Barto 2013) for exploration and push an agent to get a specific behavior without any direct feedback from the environment. These rewards can take many forms, such as a comparison between the agent's predictions and reality (Stadie, Levine, and Abbeel 2015; Burda et al. 2018; Kim et al. 2019), or the performance on self-generated goals (Vezhnevets et al. 2017; Levy, Platt, and Saenko 2018; Nachum et al. 2018; Nair et al. 2018; Pong et al. 2019). Ammanabrolu *et al.* (2020b) intrinsically rewards a text-game playing agent for adding nodes and edges to a knowledge graph. Our technique intrinsically motivates the agent to explore states that add nodes and edges that are anticipated by the exemplar story. Related, the *learning from stories* technique by Harrison *et al.* (2016) uses stories to guide RL agents. However, it requires dozens of exemplar stories and each event is treated as a goal in a modular hierarchical policy; we only require a single story and generate a single unified policy.

3 Story Shaping

Story Shaping facilitates a RL agent's ability to learn implicit knowledge from an exemplar story about a task and reward itself for actions that bring the operating environment more in alignment with the inferred story world. Our technique starts with a given exemplar story (which can also be automatically generated), which the agent transforms into a knowledge graph, referred as *Story KG* (§3.1). During RL training, as the agent explores the game world, it builds an internal state knowledge graph, called the *World KG* (§3.2). Then the agent is updated using intrinsic rewards, calculated based on the similarity between the World KG and the Story KG (§3.3). The technique is overviewed in Figure 2.

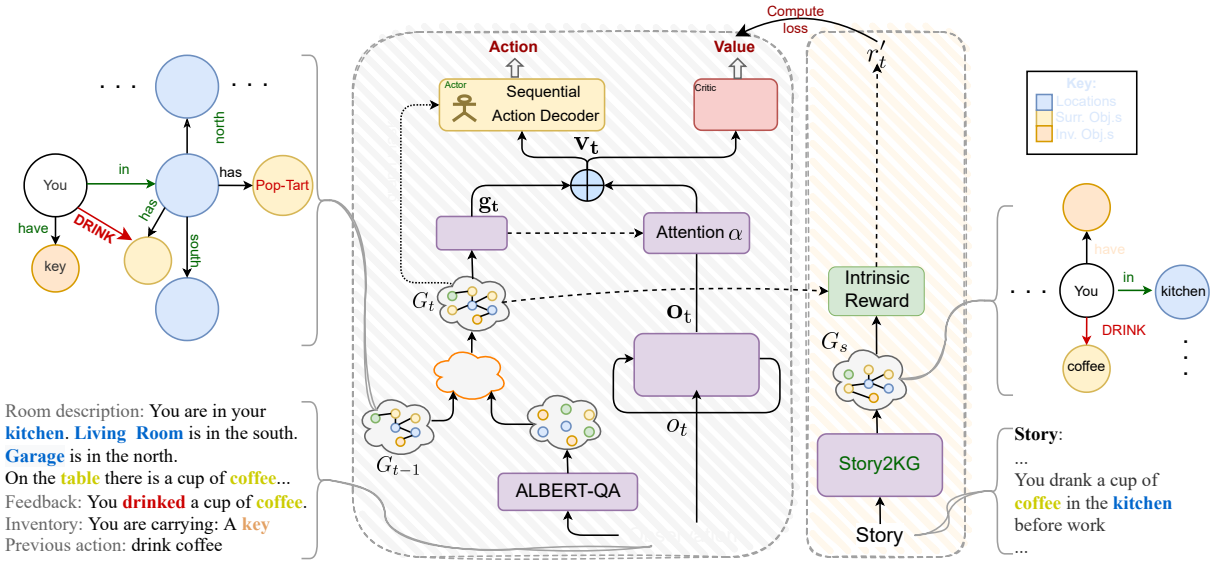

Figure 2: Knowledge graph extraction and the architecture of RL agent with Story Shaping at step t.

Game: `9:05`
First observation:
I wake up in the morning. A bathroom lies to the south, while a door to the east leads to the living room. On the end table are a telephone, a wallet and some keys. The phone rings.
Goal: GO TO WORK.

Human-written story about a routine:
Upon waking in the morning, I start my day with a *Pop-Tart* breakfast, followed by a *shower* before commencing work.

Human-written story about a Persona (`refined man`):
Upon waking, I first tend to my personal hygiene by taking a *shower* and using the *toilet*. After, I change into appropriate *clothes* before having *breakfast*. I then leave my home to begin my workday.

Automatically generated story by ChatGPT:
I will likely take a *shower* in the bathroom to the south, get *dressed*, and check my wallet and *keys* to make sure I have everything I need for the day. I may also *take a cup of coffee* before leaving my home to go to work.

Table 1: Exemplar stories constructed by human or ChatGPT. Words underlined indicate actions and entities that can be taken in the game. Words with a wavy underline indicate actions or entities that are invalid or not allowed in the game.

3.1 The Exemplar Story

A natural language story is provided as an exemplar of the behavior the agent is to enact. Table 1 shows some possible story exemplars for "going to work in the morning". We assume that a person is providing the story as a high-level description of what the agent should attempt to achieve or to describe how a task should be achieved. The story can also be generated by an automated story generation system (Peng et al. 2022b; Goldfarb-Tarrant et al. 2020; Peng et al. 2022a; Lin and Riedl 2021) or by prompting a large language model such as ChatGPT to describe a typical way of doing something.

Similar to the last example in Table 1, stories can leave details about the environment out. The person or entity that provides the story, whether human or large language model, may be unaware of the exact parameters of the operating environment of the agent. For example, "coffee" is not an entity in the game "9:05", but can be present in the exemplar example. This is an intentional benefit of using story exemplars; stories can be thought of as compact descriptions that focus on salient details with the assumption that the recipient shares common knowledge with the storyteller and can thus reconstruct the more fine-grained details.

The purpose of the exemplar story, however, is not to be a demonstration, but to generate an intrinsic reward signal that guides the reinforcement learning agent to act in ways that align the operating environment with the implicit world depicted in the story, thereby "shaping" the agent's behavior. Before the agent begins training, the exemplar story is converted into a knowledge graph—called the *Story KG*—taking the form of RDF triples (Consortium et al. 2014) of $\langle subject, relation, object \rangle$. Story KG is an explicit and persistent memory of entities mentioned from the story. This knowledge graph contains the entities and relations directly extracted from the exemplar story's text, as well as additional world details inferred from the events in the story.

To acquire the Story KG, we train a Semantic Role Labeling (SRL) (Gildea and Jurafsky 2002) model on VerbAtlas (Di Fabio, Conia, and Navigli 2019) to provide the automatic identification and labeling of argument structures of the story. VerbAtlas is a linguistic resource that provides semantic annotations on sentences based on the verb and how it is used. Verbs are important parts of stories because they convey action that change the state of the story world. For example,"*I drink coffee in the*

kitchen" is firstly processed by VerbAtlas SRL to obtain "{'verbatlas': 'DRINK', 'description': '[ARG0: You] [drink.01: drink] [ARG1: coffee] [ARGM-LOC: in the kitchen]}". Then the themes and attributes are used as entities and VerbAtlas frames are used as edges, such as ⟨You, DRINK, coffee⟩. We also incorporate location and time data into the knowledge graph (i.e., ⟨You, in, kitchen⟩). Training details can be found in Appendix A.1. Incorporating the VerbAtlas frame name of the verb as a relation satisfies three needs. First, it acts as a placeholder for the commonsense effects when they are not otherwise known. For example, the coffee is in the state of having been drunk. Second, it acts as a record of actions that have been taken in the world—the world is one in which those actions occurred. Third, some actions don't have *positive* effects that create relations, e.g., destroying something removes objects and relations because the object ceases to exist.

Story KG captures the positive relations at the end of the exemplar story, including the record of events as a placeholder for implicit effects. An important attribute of this approach is that it intentionally does not capture the order of events. The agent will learn through trial-and-error whether there are ordering constraints. There may be events in the story that can be carried out in different orders or must be carried out in different orders depending on environmental conditions. Events may be missing completely (e.g., the story doesn't explicitly say to move from the bedroom to the kitchen). Furthermore, some events may be impossible (e.g., there is no coffee in the kitchen). Instead, the story KG provides the key elements that the agent should encounter, and the agent receives more rewards for encountering and doing as many of them as possible.

3.2 The Reinforcement Learning Agent

We consider the standard reinforcement learning setting where an agent interacts with a text game environment over a number of discrete time steps. State-of-the-art approaches to RL in text environments use a knowledge graph as an external, persistent memory of the world state (Ammanabrolu and Riedl 2018; Ammanabrolu and Hausknecht 2020; Ammanabrolu et al. 2020b). As the agent explores the game world, a knowledge graph—called the *World KG*—is constructed and used as state representation. The RL agent is trained via the Advantage Actor Critic (A2C) (Mnih et al. 2016) to maintain a policy $\pi(a_t \mid s_t; \theta)$ and an estimate of the value function $V(s_t; \theta_v)$, where s_t is the game state, a_t is the action at time t. The RL agent maximizes long-term expected reward in the game in a manner and uses the same mix of n-step returns to update the policy and the value function at the same time.

In text games, actions are strings of tokens. We do not query the game environments for admissible actions—those that are guaranteed to have an effect. However, we do simplify the action space by using templates based on the verb (Ammanabrolu and Hausknecht 2020). Templates are composed of interchangeable verbs phrases (VP), optionally followed by prepositional phrases (VP PP), e.g. ($[drink/eat]$ __) and ($[apply/ask/put]$ __ $[on/about/down]$ __), where the verbs and prepositions within [.] are aliases. The agent generates actions by first sampling a template and then sampling the word from the game's vocabulary to fill in the blanks.

As the agent explores the game world, we build an internal World KG state representation. This knowledge graph is distinct from the Story KG. Following Ammanabrolu *et al.* (2020), we consider the process of building a knowledge graph to be a question-answering task. We finetune the ALBERT model (Lan et al. 2019) on the JerichoQA (Hausknecht et al. 2020b) dataset, which is specifically designed for question answering in text games. This allows the model to answer questions like "What am I carrying?" and "Where am I?". We use the answers as a set of candidate vertices V_t for the current step and the questions as a set of relations R_t. We then combine V_t and R_t with the game knowledge graph from the previous step G_{t-1} to update the game knowledge graph to G_t. More details can be found in Appendix A.2.

The ALBERT-QA technique for building the *World KG* has been shown to improve RL agent performance because it is trained to the particulars of text games (Ammanabrolu and Hausknecht 2020). The *Story KG*, on the other hand, is constructed using the VerbAtlas SRL model because exemplar stories draw from a different text distribution—they are expected to be less verbose about world details, intentionally leaving more implicit. Further, the Story KG only needs to extract a few key details to bias the agent toward certain actions and locations in the environment.

Putting it all together, at each step of training, a total score R_t and an natural language observation o_t is received from the game environment—consisting of $\left(o_{t_{\text{desc}}}, o_{t_{\text{game}}}, o_{t_{\text{inv}}}, a_{t-1}\right)$ corresponding to the room description, game feedback, inventory, and previous action. An example is depicted in the left side of Figure 2. As introduced in Section 3.2, the World KG G_t at time step t is also updated. Each component of $o_t \in \mathbb{R}^{d_o \times c}$ is processed using a GRU-based encoder to obtain $\mathbf{o}_t$ and World KG, G_t, is processed via Graph Attention Networks (GATs) (Veličković et al. 2017) followed by a linear layer to get the graph representation $\mathbf{g_t} \in \mathbb{R}^{d_o}$; c is the number of $\boldsymbol{o}_t$'s components. Then we calculate the attention α between $\boldsymbol{o}_t$ and $\mathbf{g_t}$,

$$\boldsymbol{\alpha} = \text{softmax}\left(\boldsymbol{W}_{\text{l}}\boldsymbol{h} + \boldsymbol{b}_{\text{l}}\right) \tag{1}$$

$$\boldsymbol{h} = \tanh\left(\boldsymbol{W}_{\text{o}}\boldsymbol{o}_{\text{t}} \oplus \left(\boldsymbol{W}_{\text{g}}\mathbf{g}_{\text{t}} + \boldsymbol{b}_{\text{g}}\right)\right) \tag{2}$$

where $\oplus$ denotes the addition of a matrix and a vector and $\odot$ denotes dot-product. $\boldsymbol{W}_{\text{l}} \in \mathbb{R}^{d_o \times d_o}$, $\boldsymbol{W}_{\text{g}} \in \mathbb{R}^{d_o \times d_o}$, $\boldsymbol{W}_{\text{o}} \in \mathbb{R}^{d_o \times d_o}$ are weights and $\boldsymbol{b}_{\text{l}} \in \mathbb{R}^{d_o}$, $\boldsymbol{b}_{\text{o}} \in \mathbb{R}^{d_o}$ are biases. Finally, the overall representation vector $\mathbf{v_t}$ is updated by $\mathbf{v_t} = \mathbf{g_t} + \sum_i^c \boldsymbol{\alpha}_i \odot \mathbf{o}_{\text{t},i}$

3.3 Rewarding the Agent

After obtaining the overall representation $\mathbf{v_t}$ above, we incorporate two intrinsic rewards into the RL agent's training, in order to motivate it to act in a manner that to be more closely resemble the implicit world of the story.

The *KG intrinsic reward* is determined by comparing the similarity between the agent's World KG and Story KG.

This reflects how closely the agent's actions align the actual world with the world that should exist according to the story. When new edges are added to the World KG, $\mathbf{G}_t$, we verify if the corresponding triples already exist in the Story KG. Each newly discovered triple results in a positive intrinsic reward $r_t^{\text{s}} = n \times \rho > 0$, where n is the number of same triples. An example can be found in Figure 2 (the red edges in the knowledge graphs). For example, the agent performs an action that resembles the implicit world of the story, such as "drink coffee", a new edge, $\langle$`You`, `DRINK`, `coffee`$\rangle$, will be added to the World KG. This triple is identified as being identical to one triple in the Story KG, so the *KG intrinsic reward* $r_t^{\text{s}} = 1 \times \rho$ at this step. The edge $\langle$`You`, `in`, `kitchen`$\rangle$ (the green edge in the knowledge graphs) is not considered here as we only take into account new edges added during the current round of the game.

Inspired by Ammanabrolu *et al.* (2020b), we also encourage the agent to explore more locations and scenarios by defining a *exploration intrinsic reward* $r_t^{\text{e}} = \Delta\left(\mathbf{G}_{\text{global}} - \mathbf{G}_t\right)$, where $\mathbf{G}_{\text{global}} = \bigcup_{i=1}^{t-1} \mathbf{G}_i$ is the set of all edges that the agent has ever had in its knowledge graph. When the agent learns more information about the world, it will expand the size of its World KG, increasing the likelihood of reward and success.

The overall intrinsic reward received at time step t is:

$$r'_t = r_t + \alpha \times r_t^{\text{s}} + \beta \times r_t^{\text{e}} \tag{3}$$

where α and β are scaling factors; r_t is the game score; r'_t is the reward provided to the agent on time step. The rest of the training methodology is unchanged from Ammanabrolu *et al.* (2020b).

4 Experiments

We conducted experiments in three phases:

1. We train agents to play text games where the agent must successfully navigate some normative everyday routines. Agents are provided with different exemplar stories about the tasks. In this set of experiments, we evaluate whether Story Shaping facilitates the expression of commonsense and social norm knowledge (§ 4.1).
2. We train agents to play an open-ended role-playing game, using different exemplar quests to create different personas. In this set of experiments, we evaluate whether Story Shaping is capable of shaping the agent's behaviors in a way recognizable to humans. It also demonstrates that Story Shaping can learn trope knowledge, which is knowledge particular to different storytelling conventions (§ 4.2).
3. We train agents to play games in which the exemplar stories differ from the operating environment either by referencing objects that do not exist or actions that cannot be performed. This set of experiments evaluates whether Story Shaping is robust to different assumptions between the provider of the story and the agent's environment. These differences can arise for a number of reasons, one of which being that the story is generated by another system, such as ChatGPT(§ 4.3).

Baselines We implement the RL agents below with and without Story Shaping:

- **Q*BERT** (Ammanabrolu et al. 2020b), a state-of-the-art RL agent for text games designed to work with Jericho games and TextWorld. It constructs its World KG by answering questions with ALBERT-QA.
- **KG-A2C** (Ammanabrolu and Hausknecht 2020), which uses Stanford's Open Information Extraction (Angeli, Premkumar, and Manning 2015) to build its World KG because ALBERT-QA is tuned on the JerichoQA dataset. KG-A2C is used for LIGHT experiments.

4.1 Commonsense and Social Norm Knowledge

We first seek to understand whether Story Shaping enhances the expression of common sense and social norm knowledge in Reinforcement Learning agents.

Games. We implement three slice-of-life text games on two game platforms:

- *9:05*: a game in which the agent must successfully navigate the normative routine of getting ready to work, implemented in Jericho (Hausknecht et al. 2020b), a framework for interacting with text games, as the interface connecting learning agents with interactive fiction games.
- *Shopping*: a game in which the agent must successfully purchase the clothes, developed in TextWorld (Côté et al. 2018), an open-source, extensible engine that both generates and simulates text games.
- *See Doctor*: a game in which the agent gets sick and must seek medical treatment, also developed in TextWorld.

Details for *9:05* can be found in Appendix B.1. Details for *Shopping* and *See Doctor* can be found in Appendix B.2.

Training. For each game, we train two agents. *Q*BERT-S* uses Story Shaping with stories written by humans. The baseline, *Q*BERT* is the same agent but without Story Shaping. We evaluate these two trained agents by running test games over 20 random seeds. Training details are in Appendix A.3.

Automatic Evaluation. We automatically evaluate the agents' expression of common sense and social norm knowledge[1] by:

- **Win rate**: the winning rate of trained agents on test games over 20 random seeds.
- **Avg steps**: the average number of steps that each agent takes to win the game. The game will automatically end over 50 steps.
- **Avg Commonsense score**: This is only intended for testing the agents and does not exist in the training process. Trained agents are tested on the game environments with CS (common sense) scores. Details about this test game environment can be found in Appendix A.4. A higher score indicates the agent takes more actions that express commonsense and social norm knowledge.
- **Avg game score**: the average score of each agent on the test games, which reflects how far toward the win con-

[1] Automatic metrics are used as ablation study evaluations.

Game	Agents	Win Rate %	Avg Steps	Avg CS Score	Max CS Score	Avg Game Score	Max Game Score
9:05	Q*BERT-S	100	16.30	**3.90**	4	5.00	5
	Q*BERT	100	7.25	0.40		5.00	
Shopping	Q*BERT-S	100	12.35	**3.70**	4	5.00	5
	Q*BERT	100	6.30	0.90		5.00	
Doctor	Q*BERT-S	95	19.15	**6.70**	8	4.75	5
	Q*BERT	95	14.30	0.70		4.75	

Table 2: Automatic evaluation results across 20 independent runs comparing Q*BERT-S to baseline Q*BERT Each system is trained under the same game environment.

Game	Commonsense/Social Norm			Understanding		
	Shaped %	Base %	Tie %	Shaped %	Base %	Tie %
9:05	**63.63***	9.09	27.27	36.36	**45.45**	18.18
Shopping	**66.67****	8.33	25.00	**41.67**	33.33	25.00
Doctor	**53.85**	23.08	23.08	**46.15**	38.46	15.38

Table 3: Human evaluation results showing the percentage of participants who preferred Story Shaped Q*BERT-S to baseline Q*BERT, or thought the systems were equal. Each system is trained under the same game environment. * indicates human evaluation results are significant at $p < 0.05$ confidence level; ** at $p < 0.01$ using a Wilcoxan sign test on win-lose pairs.

dition the agent made it, irrespective of *how* the agent reached the farthest point.

Results are shown in Table 2. The win rates and the average game scores are identical between agents with and without Story Shaping. However, our agent's "Avg Commonsense Score" is significantly higher than the baseline agent's, indicating that the Story Shaping agent demonstrates superior ability in expressing common sense and social norm knowledge. The larger "Avg Steps" value for our agent also suggests that it takes more actions before winning the game, which further highlights that it is not seeking the shortest possible trajectory.

Human Evaluation. We recruited 30 participants[2] using the Cloud Research platform and Amazon Mechanical Turk (Litman, Robinson, and Abberbock 2017). We screened for participants that were generally not familiar with text games. Each participant reads the winning goal of a randomly chosen game. Then they read a pair of game transcript which played by Q*BERT-S and Q*BERT, specifically. Each transcript includes game observations and the corresponding actions. Then they are given the following metrics and asked to choose which game transcript they prefer for that metric:

- This sequence of actions expresses more common sense thinking (social norm knowledge) on the action choice.
- This sequence of actions makes you understand why the agent takes these actions given what you know about the goal.

Table 3 shows the percentage of times stories from each system are preferred for each metric. In the same game environment, Q*BERT-S performs significantly better than Q*BERT on the dimension of "Common and Social Norm Sense". We can conclude that Story Shaping facilitates the expression of commonsense and social norm knowledge of the trained RL agents significantly. On the dimension of "Understanding", we would expect Story Shaped agents to be no less understandable than the baseline, indicating that longer trajectories are not random. Q*BERT-S achieves comparable results with Q*BERT. Participants who favored Q*BERT mentioned that they found the shorter game paths easier to comprehend. In Table 2, Q*BERT takes fewer steps to complete the game, making it more straightforward for human participants to follow. Our system is thus shown to improve the expression of commonsense and social norm knowledge of agents while preserving comprehensibility.

4.2 Persona Understanding

We assess whether Story Shaping has the ability to shape the agent's behaviors in a way that is identifiable to humans. We develop a role-playing game in the LIGHT (Urbanek et al. 2019) environment, which is a large-scale fantasy text adventure game research platform for training agents that can both talk and act, interacting either with other models or with humans. We provided different exemplar quests for four personas: `thief`, `bum`, `adventurer` and `thug`. We provide a LIGHT world that provides a rich set of locations and objects for all personas to make use of, or ignore. Details about this game can be found in Appendix B.3. In these experiments, we compare two KG-A2C agents utilizing Story Shaping but with different human-written stories.

We recruited an additional 29 participants[2]. Participants will read a winning goal and be told the agent's persona (thief, adventurer, thug, bum), then read two game transcripts played by versions of KG-A2C with Story shaping. One version is trained using a story about the given persona, the other using a random story selected from the remaining personas. Participants were asked to indicate which of the two agents had the given persona. At least 10 participants evaluate each game.

Table 4 displays the percentage of participants who chose the game transcript generated by the agent with the exemplar story for the given persona, versus a randomly chosen exemplar story. The percentage reflects the effectiveness of Story Shaping in shaping the agent's behaviors in a way that is recognizable to humans. The results indicate that RL agent with Story Shaping is able to comprehend exemplar quests for different personas and generate actions that align with the given persona. Additionally, all the versions of KG-A2C with Story Shaping attain a win rate of 100%, regardless of which exemplar story. It demonstrates the capability of Story Shaping in shaping the agent's behaviors in a way that is recognizable to humans while at the same time do not lose any

[2]Participants are required to provide detailed explanations for their choices in each comparison, using at least 50 characters of free text. Each game trajectory pair is evaluated by a minimum of 10 participants. Our study was approved by our Institutional Review Board, and we payed participants the equivalent of $15/hr.

Game	Given Persona	Participant Choice Correct%	Incorrect%	Can't tell%	Win Rate
LIGHT	Thief	**58.06***	19.35	22.58	100
	Adventurer	**62.50***	15.62	21.88	100
	Thug	**72.73****	12.12	15.15	100
	Bum	**64.71***	23.53	11.76	100

Table 4: The percentage of participants who preferred the agent utilizing Story Shaping with the specific story of the given persona, the agent using Story Shaping with a story from a different persona, or believed the systems were equivalent when the game goal and persona were provided to human participants. "Win Rate" is the winning rate of trained agents using Story Shaping with the specific story of the given persona on test games over 20 random seeds. The symbols (* and **) used for indicating significance in as in Table 3.

Game	Persona	Common/Social GPT%	Base%	Tie%	Understanding GPT%	Base%	Tie%
9:05	-	**71.43***	14.29	14.29	**57.14**	42.86	0.00
Shop.	-	**62.50***	25.00	12.50	**37.50**	25.00	**37.50**
LIGHT	Thief	**57.14**	28.57	14.29	**42.86**	28.57	28.57
	Adv	**83.33****	16.67	0.00	**50.00**	16.67	33.33
	Thug	**64.71***	11.76	23.53	**52.94**	35.29	11.76

Table 5: The percentage of participants who favored the agent with Story Shaping using the ChatGPT-generated story over the baseline, or deemed the systems as indistinguishable. The symbols (* and **) are used for indicating significance as in Table 3.

performance.

4.3 Robustness

Our final experiment investigates the robustness of Story Shaping to variations in assumptions between the exemplar story provider and the agent's environment. Instead of using stories designed by humans based on their understanding of the environment, we use automatically generated stories by ChatGPT. Notably ChatGPT has no familiarity with the particular game world and these exemplar stories diverge substantially from the agent's game environment, such as referencing objects that do not exist or providing actions that are not available.

We prompt ChatGPT with a description of the first game state, as well as an optional character personality to obtain the guiding story that can then be converted to a knowledge-graph triple format to be used in guiding the agent. We then proceed to replicate the training methodology outlined in Section 4.1. We follow the same evaluation process and recruited 34 participants[2] on a crowd sourcing platform to answer the same questions with Section 4.1.

Table 5 shows the preference percentage for the stories from each system in each metric. The high preference percentage shows that Story Shaping with the automatically generated story can also guide RL agents to express more common sense and social norm knowledge, even though some information in the automatically generated story's knowledge graph is unattainable. For example, in Table 6, the ChatGPT generated exemplar story involves an entity that does not exist in the game—"directory listings". Despite the mismatch, our technique allows for the flexibility needed for our agent to successfully complete tasks by utilizing other elements of the exemplar story and using trial-and-error to fill in the rest. Agents have a 100% game completion rate in all the test game environments.

Game: `shopping`
First observation: I am in front of a mall. A cafe lies to the south, while a way to the east leads to the mall.
Goal: BUY CLOTHES.

Exemplar story from ChatGPT:
To buy clothes, you *enter the mall* and navigate to the store that sells the clothing you are interested. Many malls have directory listings near the entrances, which can help you *find the specific stores* you are looking for. Once locating the store, you can *browse* and *try on the clothing*, and then *make a purchase* at the register.

Example action sequence by Q*BERT-Story Shaping:

Location: *Street*	Action: *go east*
Location: *Mall*	Action: *go north*
Location: *Store*	Action: *examine clothes; try clothes*
	Action: *give money to staff*
	Action: *buy clothes; take clothes*

Example action sequence by Q*BERT:

Location: *Street*	Action: *go east*
Location: *Mall*	Action: *go north*
Location: *Store*	Action: *give money to staff*
	Action: *buy clothes; take clothes*

Table 6: Example action sequence in the game `shopping`. Words underlined indicate actions that can be taken in the game environment. Words with a wavy underline indicate actions that are invalid or not allowed in the game.

5 Conclusions

Story Shaping is a straightforward approach to the challenge of reward design where one wishes to not only reward an agent for completing a task, but reward the agent for *how* it accomplishes the task. This might mean aligning an agent's behavior with human preferences and expectations, teaching the agent commonsense reasoning and social norms, or shaping character personas in a game.

Our technique allows one to provide a high-level exemplar story from which the agent automatically extracts knowledge about important objects, locations, and actions. It then self-rewards when these objects, and locations are encountered, and actions are performed. Because stories are high-level abstractions, Story Shaping can fill in missing details and is robust to situations where the story cannot be exacted as given. We have shown that Story Shaping has a significant and human-observable impact on agent behavior without compromising task completion.

A Implementation Details

A.1 Semantic Role Labeling Using VerbAtlas

The SRL model automatically identifies and labels the argument structures of stories. For example, it extracts `'verbatlas': 'EXIST_LIVE', 'args_words': {'Theme': 'Jenny', 'Attribute': 'Georgia'}` from "*Jenny lived in Georgia*". Verbs in the story will be represented as the VerbAtlas frame. For example, ``live'' is represented as ``EXIST_LIVE''.

We use a fine-tuned transformer model for semantic role labeling (SRL), which is a BERT (Devlin et al. 2019) model with a linear classification layer trained on the Ontonotes 5.0 dataset to predict PropBank(Palmer, Gildea, and Kingsbury 2005) SRL. This model, proposed by Shi (2019), is currently the state-of-the-art for English SRL. We use an open-source implementation[1], which is based on the official AllenNLP BERT-SRL model[2]. Trained with the following hyperparameters: Batch size: 32; Dropout for the input embeddings: 0.1; Learning rate: $5e^{-5}$; Optimizer: Adam; Total Epochs: 15.

Then, we use the mappings from Propbank frames to VerbAtlas (Di Fabio, Conia, and Navigli 2019) to return the correct corresponding VerbAtlas classes instead of Propbank's (Palmer, Gildea, and Kingsbury 2005). We can directly map VerbAtlas classes to PropBank frames because for every VerbAtlas class, there is only one PropBank frame. This allows us to use the rich content from VerbAtlas with the same model that was initially trained to predict PropBank.

A.2 Knowledge Graph Representation QA Model

The question answering network based on ALBERT (Lan et al. 2019) uses the hyperparameters listed in the original paper. These hyperparameters have been shown to work well on the SQuAD 2.0 (Rajpurkar, Jia, and Liang 2018) dataset. We did not do any additional tuning of the hyperparameters.

A.3 RL Agents With Story Shaping

Further details of what is found in Figure 2. The sequential action decoder consists two GRUs that are linked together as seen in Ammanabrolu (2020). The first GRU decodes an action template and the second decodes objects that can be filled into the template. These objects are constrained by a *graph mask*, i.e. the decoder is only allowed to select entities that are already present in the knowledge graph.

Same with Ammanabrolu (2020), the loss consists of template loss, object loss, value loss, actor loss and entropy loss. The template loss given a particular state and current network parameters is applied to the decoder. Similarly, the object loss is applied across the decoder is calculated by summing cross-entropy loss from all the object decoding steps. Entropy loss over the valid actions is designed to prevent the agent from prematurely converging on a trajectory. The following hyperparameters are taken from the original paper and are known to work well on text games. 1. discount factor: 0.9; 2. entropy coefficient: 0.03; 3. value coefficient: 9; 4. template coefficient : 3; 5. object coefficient: 9.

[1]https://github.com/Riccorl/transformer-srl

[2]https://demo.allennlp.org/semantic-role-labeling

A.4 Test Game with Human-likeness Score

In order to automatically evaluate the expression of commonsense and social norm knowledge of the trained RL agents, we develop a test game mode for each text game. During training, the player can only earn a score if they win the game. However, when evaluating the RL agents, we test both the baseline and our agents in a test game that has the same map and actions as the training game, but a different scoring system. Specifically, the test game environment not only gives a score when the player wins the game, but also gives scores when the agent reaches specific states. For example, in the game `9:05`, the test game environment will give a score of 2 when the agent eats a pop-tart in the kitchen, whereas in the training game, this action would not earn any points. Our test game mode allows us to automatically evaluate the human-like qualities of our trained RL agents. We believe that a higher score in this game mode indicates a greater degree of the expression of commonsense and social norm knowledge of the trained RL agents.

B Game Design

B.1 Jericho Game

9:05 is a text adventure game by Adam Cadre. The game is designed for a casual audience, including those new to the genre. It is a short and simple game that follows a branching narrative in which a man wakes up and receives a call telling him to go to work. Our agent can be tested effectively in this game due to the numerous optional commonsense actions available to the player. In the game, the player has the option to leave their home and go directly to work in order to win the game immediately, or they can choose to take some time to get ready by changing clothes, taking a shower, and having a toast before starting their day. All of these actions are optional and do not affect the outcome of the game.

Exemplar story we use for Section 4.1: " Upon waking in the morning, I start my day with a Pop-Tart breakfast, followed by a shower before commencing work." "

We designed a game with optional *CS scores* for evaluating the human-likeness of our agents. CS scores designed are as follows, Score 2: `shower` is used; Score 2: `Pop-Tart` is consumed.

B.2 TextWorld

We designed two games in TextWorld (Côté et al. 2018) text game engine.

Shopping is a game where the goal is to purchase clothing at a mall. The player starts out on the street and has the option to stop at various cafes and restaurants on the way to the mall. They may also encounter NPC characters who can provide information on obtaining coupons. The player has multiple routes they can take to reach the clothing stores and can choose to engage in optional activities such as purchasing a coffee or using coupons. The layout of the game is shown in Fig.3. The game score is only obtained when the player finish the game. When the player has "`clothes`" in his inventory, the game ends with a game score: `5`.

Exemplar story for experiments in Section 4.1 is: " To save money, I need to obtain a coupon. Once I have tried

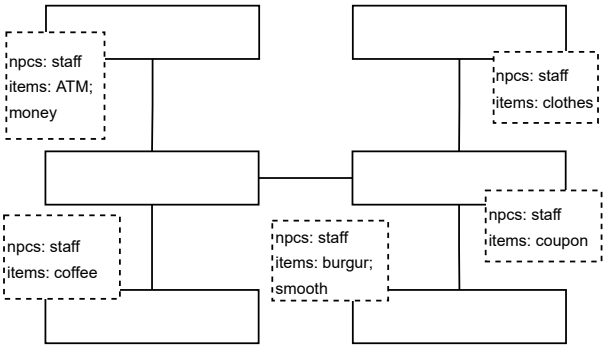

Figure 3: Layout of game "shopping".

on the clothes, I will purchase them. " CS scores designed for Section 4.1 are as follows,Score 2: `coupon` is applied; Score 2: `clothes` is tried.

See Doctor is a game where the objective is to obtain the medicine. The player starts at home with a cup of hot water and money. The player can make hot soup, buy hot coffee, visit the doctor at the hospital, or go to a drug store to get the medicine. The player has multiple paths they can take to obtain the medicine, and they can also take various optional actions such as buying a coffee or drinking hot water. The game score is only obtained when the player finish the game. When the player has "medicine" in his inventory, the game ends with a game score: 5. You can see the layout of this game in Fig.4

Exemplar Story for experiments in Section 4.1 is: " I caught a cold and drank hot water, but it didn't help after taking a shower. I went to the hospital to see the doctor and get a prescription to buy medicine. " The *CS scores* designed are as follows, Score 2: `water` is consumed; Score 2: `shower` is used; Score 2: `doctor` is seen; Score 2: `prescription` is in inventory.

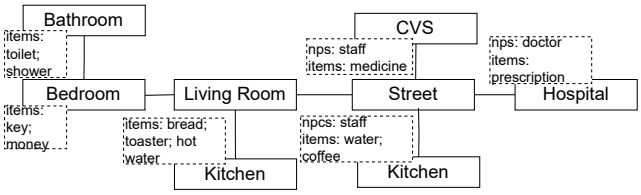

Figure 4: Layout of game "see doctor".

B.3 LIGHT

LIGHT (Urbanek et al. 2019) is a text adventure game research platform for training agents that can both talk and act. For our experiments, we developed a large map to serve as a sandbox. We design a role-playing game, "gold", where the player can play different personas, collect items, and then finally "go to Meadow". The player starts out in the Simple Town and has the option to visit various locations in the town such as the wealthy area of town or the Armory and collect different items at these locations. The player has multiple routes they can take to reach the Meadow and can choose to engage in optional activities such as visiting the Armory, or picking up old prayer books in the Sermon Hall. The layout of the game is shown in Fig. 5. The game score is only obtained when the player finish the game. When the player enters the *Meadow*, the game ends with a game score: 5.

Additionally, we also demonstrate that our system is capable of generalizing to other guiding narratives, so long as they are capable of being navigated within the environment. To this end, we provide a fourth guiding narrative with a different goal that is still achievable within the game.

Exemplar story of persona `Adventurer`: " I am a brave Adventurer. I know there is a Dungeon with valuable treasure. I go to the armory and get a sword, a shield, armor and a bow to defend myself. Once I am properly equipped, I go to the Dungeon. I get the gold, jewelry, gold cups and a golden goblet. I then leave through the Meadow. " CS scores designed for Section 4.2 are as follows, Score 10: `sword` is obtained; Score 10: `armor` is obtained; Score 10: `shield` is obtained; Score 10: `bow` is obtained; Score 10: `gold` is obtained; Score 10: `jewelry` is obtained; Score 10: `gold cups` is obtained; Score 10: `golden goblet` is obtained.

Exemplar story of persona `Thief`: " I am a cowardly Thief. I go to the wealthy area of town to search for valuables. I enter the Hillside Manor and get the gold bars there. I stealthily go to the Sermon Hall, and get the small sack of gold. I then leave through the Meadow. " CS scores designed for Section 4.2 are as follows, Score 10: `gold bars` are obtained; Score 10: `small sack of gold` is obtained.

Exemplar story of persona `Bum`: " I am a lazy bum. I wish to do as little as possible to get some coins and leave. I immediately exit the Simple Town. I only stop at the Town Square to get the donations before leaving through the Meadow. " CS scores designed for Section 4.2 are as follows, Score 10: `donations` are obtained.

Exemplar story of persona `Thug`: " I am a Thug. I wish to get revenge on the watch maker who scammed me. I find the watchmaker in the Sermon Hall. I hit the watch maker, and he falls to the floor dead. I then leave through the Meadow. " CS scores designed for Section 4.2 are as follows, Score 5: `watch maker` is hit.

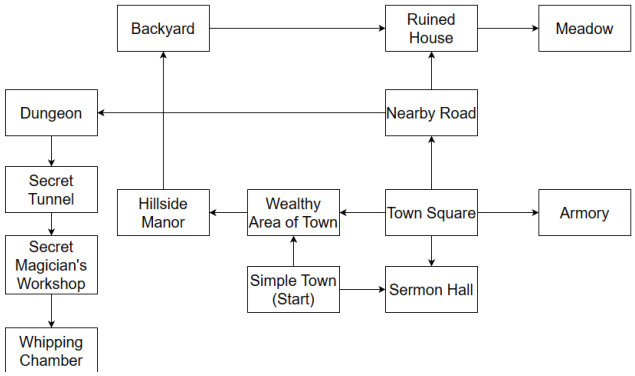

Figure 5: Layout of LIGHT Map used in testing

References

Ammanabrolu, P.; Cheung, W.; Tu, D.; Broniec, W.; and Riedl, M. 2020a. Bringing stories alive: Generating interactive fiction worlds. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, 3–9.

Ammanabrolu, P.; and Hausknecht, M. 2020. Graph constrained reinforcement learning for natural language action spaces. *arXiv preprint arXiv:2001.08837*.

Ammanabrolu, P.; Jiang, L.; Sap, M.; Hajishirzi, H.; and Choi, Y. 2022. Aligning to social norms and values in interactive narratives. *arXiv preprint arXiv:2205.01975*.

Ammanabrolu, P.; and Riedl, M. O. 2018. Playing text-adventure games with graph-based deep reinforcement learning. *arXiv preprint arXiv:1812.01628*.

Ammanabrolu, P.; Tien, E.; Hausknecht, M.; and Riedl, M. O. 2020b. How to avoid being eaten by a grue: Structured exploration strategies for textual worlds. *arXiv preprint arXiv:2006.07409*.

Ammanabrolu, P.; Urbanek, J.; Li, M.; Szlam, A.; Rocktäschel, T.; and Weston, J. 2021. How to Motivate Your Dragon: Teaching Goal-Driven Agents to Speak and Act in Fantasy Worlds. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 807–833. Online: Association for Computational Linguistics.

Angeli, G.; Premkumar, M. J. J.; and Manning, C. D. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 344–354.

Barto, A. G. 2013. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, 17–47. Springer.

Burda, Y.; Edwards, H.; Pathak, D.; Storkey, A.; Darrell, T.; and Efros, A. A. 2018. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*.

Consortium, W. W. W.; et al. 2014. RDF 1.1 concepts and abstract syntax.

Côté, M.-A.; Kádár, A.; Yuan, X.; Kybartas, B.; Barnes, T.; Fine, E.; Moore, J.; Hausknecht, M.; Asri, L. E.; Adada, M.; et al. 2018. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, 41–75. Springer.

Dambekodi, S.; Frazier, S.; Ammanabrolu, P.; and Riedl, M. O. 2020. Playing Text-Based Games with Common Sense. In *Proceedings of the NeurIPS Wordplay workshop*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186.

Di Fabio, A.; Conia, S.; and Navigli, R. 2019. VerbAtlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. 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)*, 627–637.

Frazier, S.; Nahian, M. S. A.; Riedl, M. O.; and Harrison, B. 2020. Learning Norms from Stories: A Prior for Value Aligned Agents. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*.

Gildea, D.; and Jurafsky, D. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3): 245–288.

Goldfarb-Tarrant, S.; Chakrabarty, T.; Weischedel, R.; and Peng, N. 2020. Content planning for neural story generation with aristotelian rescoring. *arXiv preprint arXiv:2009.09870*.

Harrison, B.; and Riedl, M. O. 2016. Towards learning from stories: An approach to interactive machine learning. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*.

Hausknecht, M.; Ammanabrolu, P.; Côté, M.-A.; and Yuan, X. 2020a. Interactive Fiction Games: A Colossal Adventure. In *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*.

Hausknecht, M.; Ammanabrolu, P.; Côté, M.-A.; and Yuan, X. 2020b. Interactive fiction games: A colossal adventure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7903–7910.

Hedlund, J.; Antonakis, J.; and Sternberg, R. 2002. *Tacit Knowledge and Practical Intelligence: Understanding the Lessons of Experience (ARI Research Note 2003-04)*. Washington, D.C.: United States Army Research Institute for the Behavioral and Social Sciences.

Kim, Y.; Nam, W.; Kim, H.; Kim, J.-H.; and Kim, G. 2019. Curiosity-bottleneck: Exploration by distilling task-specific novelty. In *International Conference on Machine Learning*, 3379–3388. PMLR.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Levy, A.; Platt, R.; and Saenko, K. 2018. Hierarchical reinforcement learning with hindsight. *arXiv preprint arXiv:1805.08180*.

Lin, Z.; and Riedl, M. O. 2021. Plug-and-blend: a framework for plug-and-play controllable story generation with sketches. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 17, 58–65.

Litman, L.; Robinson, J.; and Abberbock, T. 2017. TurkPrime. com: A versatile crowdsourcing data acquisition platform for the behavioral sciences. *Behavior research methods*, 49(2): 433–442.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937. PMLR.

Nachum, O.; Gu, S. S.; Lee, H.; and Levine, S. 2018. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31.

Nahian, M. S. A.; Frazier, S.; Harrison, B.; and Riedl, M. O. 2021. Training Value-Aligned Reinforcement Learning Agents Using a Normative Prior. *arXiv:2104.09469*.

Nair, A. V.; Pong, V.; Dalal, M.; Bahl, S.; Lin, S.; and Levine, S. 2018. Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31.

Oudeyer, P.-Y.; Kaplan, F.; and Hafner, V. V. 2007. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2): 265–286.

Palmer, M.; Gildea, D.; and Kingsbury, P. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1): 71–106.

Peng, X.; Li, S.; Wiegreffe, S.; and Riedl, M. 2022a. Inferring the Reader: Guiding Automated Story Generation with Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 7008–7029.

Peng, X.; Riedl, M.; and Ammanabrolu, P. 2022. Inherently explainable reinforcement learning in natural language. *Advances in Neural Information Processing Systems*, 35: 16178–16190.

Peng, X.; Xie, K.; Alabdulkarim, A.; Kayam, H.; Dani, S.; and Riedl, M. 2022b. Guiding Neural Story Generation with Reader Models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 7087–7111.

Pong, V. H.; Dalal, M.; Lin, S.; Nair, A.; Bahl, S.; and Levine, S. 2019. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 784–789. Melbourne, Australia: Association for Computational Linguistics.

Schmidhuber, J. 1991. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, 222–227.

Shi, P.; and Lin, J. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Shridhar, M.; Yuan, X.; Côté, M.-A.; Bisk, Y.; Trischler, A.; and Hausknecht, M. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Stadie, B. C.; Levine, S.; and Abbeel, P. 2015. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*.

Urbanek, J.; Fan, A.; Karamcheti, S.; Jain, S.; Humeau, S.; Dinan, E.; Rocktäschel, T.; Kiela, D.; Szlam, A.; and Weston, J. 2019. Learning to speak and act in a fantasy text adventure game. *arXiv preprint arXiv:1903.03094*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Vezhnevets, A. S.; Osindero, S.; Schaul, T.; Heess, N.; Jaderberg, M.; Silver, D.; and Kavukcuoglu, K. 2017. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, 3540–3549. PMLR.

Wang, R.; Jansen, P.; Côté, M.-A.; and Ammanabrolu, P. 2022. ScienceWorld: Is your Agent Smarter than a 5th Grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11279–11298. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Xu, Y.; Fang, M.; Chen, L.; Du, Y.; Zhou, J. T.; and Zhang, C. 2020. Deep reinforcement learning with stacked hierarchical attention for text-based games. *Advances in Neural Information Processing Systems*, 33.