BABYAI++: TOWARDS GROUNDED-LANGUAGE LEARNING BEYOND MEMORIZATION

Tianshi Cao*, **Jingkang Wang***, **Yining Zhang***, **Sivabalan Manivasagam***
Department of Computer Science
University of Toronto & Vector institute
{jcao,wangjk,ynz,manivasagam}@cs.toronto.edu

ABSTRACT

Despite success in many real-world tasks (e.g., robotics), reinforcement learning (RL) agents still learn from *tabula rasa* when facing new and dynamic scenarios. By contrast, humans can offload this burden through textual descriptions. Although recent works have shown the benefits of *instructive* texts in goal-conditioned RL, few have studied whether *descriptive* texts help agents to generalize across dynamic environments. To promote research in this direction, we introduce a new platform, *BabyAI++*, to generate various dynamic environments along with corresponding descriptive texts. Moreover, we benchmark several baselines inherited from the instruction following setting and develop a novel approach towards visually-grounded language learning on our platform. Extensive experiments show strong evidence that using descriptive texts improves the generalization of RL agents across environments with varied dynamics. Code for *BabyAI++* platform and baselines are available online: https://github.com/caotians1/BabyAIPlusPlus

1 INTRODUCTION

Reinforcement learning (RL) has recently witnessed tremendous success in various applications such as game-playing (Mnih et al., 2015; Silver et al., 2017) and robotics (Chatzilygeroudis et al., 2017; Quillen et al., 2018). However, RL is often sample inefficient - requiring large number of roll-outs to train and thus difficult to apply to real world settings. RL agents also have difficulties when generalizing to environments that differ from the training environment (Cobbe et al., 2019), thereby requiring even more data to cover variations in the environment. Integrating prior knowledge into the RL agent is a general strategy for improving sample efficiency and generalization. Hierarchical RL (Bacon et al., 2016; Osa et al., 2019), imitation learning (Ho & Ermon, 2016; Ross et al., 2011), and meta-RL (Rusu et al., 2015; Duan et al.; Finn et al., 2017; Mishra et al., 2017) can all be viewed as ways to incorporate prior knowledge (e.g. task structure, expert solution, and experience on similar tasks) into the learner.

Research in cognitive science have shown that humans have a deeply integrated representation of the visual world via language association which serves as a prior for human learning (Snow, 1972). For example, we can associate the word *ball* with different images of balls (basketball, baseball, etc.), and then associate common properties with it (round, can bounce, and be thrown). Motivated by human learning, the following question naturally arises: *Can RL agents also leverage human prior knowledge of tasks through structured natural language?*

Most prior work focused on leveraging instruction and goal-based text to improve RL agent efficiency (Luketina et al., 2019; Branavan et al., 2012; Chevalier-Boisvert et al., 2019; Goyal et al., 2019). Joint representations of visual features and instructive text have been shown to be successful for vision-and-language navigation (Chaplot et al., 2018; Hu et al., 2019; Anderson et al., 2018; Hermann et al., 2019) and question answering (Yu et al., 2018). However, we believe that leveraging *descriptive* text about the dynamics of the environment can allow for RL agents that generalize across different environments and dynamics. Learning with this type of text has only been explored in limited scope with hand-engineered language features and few environments (Branavan et al., 2012; Narasimhan

---

et al., 2017). A critical roadblock hindering the study of RL with descriptive text is the absence of interactive, dynamic and scalable environments for this type of task. Prior works have provided environments with some required elements (Kolve et al., 2017; Wu et al., 2018; Chevalier-Boisvert et al., 2019), but none have the full set of features required for the training and evaluation of the RL agents' ability to leverage descriptive language.

To answer whether descriptive text improves generalization and sample efficiency for RL, we propose a novel RL platform *BabyAI++* to generate various dynamic environments along with descriptive texts. Our platform is built upon a popular instruction-following RL environment, BabyAI (Chevalier-Boisvert et al., 2019). In contrast to existing public RL platforms, BabyAI++ is *dynamic*, *scalable*, and incorporates *descriptive text* (see Tab. 1). We also adapt and implement several baseline methods for the proposed task setting.

Environments	Inst. Text	Desc. Text	State Manipulate[1]	Var. Dynamics[2]	Procedural Env[3]	Multi-task
Kolve et al. (2017)			✓	✓	✓	
Wu et al. (2018)	✓				✓	
Narasimhan et al. (2017)	✓	✓	✓			
Chaplot et al. (2018)	✓				✓	
Chevalier-Boisvert et al. (2019)	✓		✓		✓	✓
BabyAI++ (Ours)	✓	✓	✓	✓	✓	✓

Table 1: Comparing our proposed environment with other available environments. 1: The agent can manipulate the state to achieve goals; 2: Variable dynamics at different episode; 3: Generate a wide variety of scenarios to be used for task learning.

2 PROBLEM FORMULATION AND BACKGROUND

In this section, we define our problem of RL with language descriptions and provide an overview of the BabyAI platform (Chevalier-Boisvert et al., 2019) upon which our work is built.

Natural Language-aided RL We consider the learning tasks in a standard RL problem with the additional enhancement that a description of the model dynamics of the environment is available to the agent. An environment is uniquely defined by the environment tuple $\mathbf{E} = \{\mathcal{S}, \mathcal{A}, \mathcal{F}, \rho_0\}$ which consists of the collection of states $\mathcal{S}$, collection of actions $\mathcal{A}$, state transition density function $\mathcal{F}(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$, and initial state distribution ρ_0. Then, a task can be defined as an environment equipped with a reward function and related parameters $\mathbf{T} = \{\mathbf{E}, R_E(s), \gamma, H\}$, where $R(s) : \mathcal{S} \mapsto \mathbb{R}$ is the reward function, γ is the discount factor, and H is the time horizon of the task. In this work, we focus on the setting where the task is augmented with a description of the environment, $\mathbf{T}^d = \{\mathbf{T}, \mathbf{D}_T\}$.

BabyAI The BabyAI platform is a configurable platform for procedurally-generated grid-world style environments (based on MiniGrid) and tasks. Environments in BabyAI consist of grid-world maps with single or multiple rooms. Rooms are randomly populated with objects that can be picked up and dropped by the agent, and doors that can be unlocked and opened by the agent. The number of rooms, objects in the rooms, and connectivity between rooms can be configured and randomized from episode to episode. Hence, the BabyAI environment naturally requires adaptability from the agent. Observations of the environment consist of a 7x7x3 symbolic representation of the map around the agent, oriented in the current direction of the agent (hence the environment is only partially observable). Tasks in BabyAI are procedurally generated from the map, involving going to a specific object, picking up an object, placing an object next to another, or a combination of all three (to be executed in sequence). The task is communicated to the agent through Baby-Language: a compositional language that uses a small subset of English vocabulary. Similar to the map, tasks can be randomly generated in each episode based on presence of objects.

3 BABYAI++ PLATFORM

While the BabyAI platform incorporates many challenging RL scenarios such as multi-task learning, partial-observability, and instruction following, it does not involve variable/undetermined environment dynamics, which are a common occurrence in real applications. Hence, we introduce an augmented platform BabyAI++ which is designed to evaluate an RL agents' ability to use descriptive language.
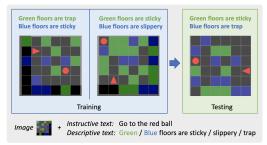
Figure 1: `GoToRedBall` task in *BabyAI++*.

Environment	N tiles	Distractor	trap	slippery	flipLeftRight	flipUpDown	sticky	magic	Partial Text
GoToRedBall-v1	2		✓	✓			✓		
GoToRedBall-v2	3		✓	✓	✓	✓	✓	✓	
PutNextLocal	2	✓	✓	✓	✓				
GoToObj	3	✓	✓	✓	✓	✓	✓	✓	
GoToObj-Partial	3	✓		✓	✓	✓	✓	✓	✓

Table 2: Properties of levels used in experiments. "N tiles" is the number of types of tiles that are used simultaneously used in one episode. "Distractor" is whether distractor objects are used.

Environment Dynamics   Various types of floor tiles are added to the map to create different kinds of state transition dynamics. For example, stepping on a *"trap"* tile will cause the episode to end with zero reward, and attempting forward movement on a *"flipUpDown"* tile will cause the agent to move backwards. These tiles are distinguished from normal tiles by their color. Similar to objects, the tiles are randomly placed on the map. More details of the dynamics can be found in Appendix A.

Descriptive Text   Descriptive text about the model dynamics is provided alongside the instructions as observation to the agent. In BabyAI++, we use text to describe which types of tiles are in use and what color is matched to each tile type. Since the pairing between color and tile type is randomized, the agent must understand the description for it to properly navigate the map. By default, each color to dynamic pair is described by a sentence in the description, but we also provide more challenging configurations such as partial descriptions. Fig. 1 provides an example BabyAI++ train/test task.

BabyAI++ Levels   We build BabyAI++ levels (see partial list in Tab. 2) upon BabyAI levels. By enabling different variable dynamics - one BabyAI level can be extended into multiple BabyAI++ levels. To evaluate language grounding, we partition every level into *training* and *testing* configurations. In the training configuration, the agent is exposed to all tile and colors types in the level, but some combinations of color-type pairs are held-out (see Tab. A1 in Appendix A.2). In the testing configuration, all color-type pairs are enabled. Hence, the agent needs to use language grounding to associate the type of the tile to the color when encountering new color-type pairs at test time.

4   REINFORCEMENT LEARNING BASELINES

We implement four baseline RL models on our new descriptive text RL setting BabyAI++. The **Image Only** RL model uses only the scene representation as input. In contrast, **Image+Text** models takes additional text descriptions of the model dynamics, and, when applicable, a text instruction (depending on the BabyAI++ level).

We study three architectures for processing descriptive text and scene representation into a combined visual+text embedding: **concat-fusion**, **FiLM**, and **attention-fusion**. In the **concat-fusion** model, we concatentate the scene embedding and text embeddings together to generate the final output embedding vector. We implement a **FiLM** (Perez et al., 2018) based model, which uses the text embedding to calculate a linear transformation that is applied to each image embedding feature. We refer readers to Appendix B for additional details.

Finally, we propose a novel baseline model **attention-fusion** which uses an attention mechanism to assign relevant text embeddings to locations on the scene embedding feature map (see Fig. 2). This takes advantage of the structured grid-world environment to explicitly ground descriptions of the environment with the observed scene by predicting which description should be assigned to each tile. More specifically, for each description sentence s_i, a description embedding d_i is computed. These descriptions are concatenated to form a description dictionary D. Then, an attention CNN processes the image embedding F and outputs "attention-probabilities" tensor W of size $7 \times 7 \times k$, where k is the number of description sentences. These attention weights are then used to obtain a linear combination of the description embeddings d_i, which are then spatially concatenated to each tile in the image embedding (i.e. a different weighted feature embedding is assigned to each tile in the
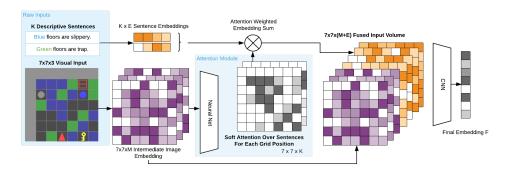
Figure 2: Diagram of the attention-fusion baseline model proposed in this paper.

Setting	Model	Training			Testing		
		Succ.	$\mathcal{R}_{avg}$	N_{epi}	Succ.	$\mathcal{R}_{avg}$	N_{epi}
GoToRedBall-v1 Image Only	Baseline	0.958 ± 0.006	0.865 ± 0.006	7.015 ± 0.149	$\underline{\textbf{0.937}} \pm 0.008$	$\underline{\textbf{0.841}} \pm 0.007$	$\textbf{7.346} \pm 0.171$
GoToRedBall-v1 Image + D.Texts	concat-fusion	0.964 ± 0.006	0.872 ± 0.006	6.996 ± 0.162	0.923 ± 0.008	0.822 ± 0.008	9.285 ± 0.348
	FiLM (Perez et al., 2018)	$\textbf{0.986} \pm 0.004$	$\textbf{0.896} \pm 0.004$	$\underline{6.755} \pm 0.139$	0.855 ± 0.011	0.739 ± 0.010	13.096 ± 0.475
	attention-fusion (ours)	$\underline{0.975} \pm 0.005$	$\underline{0.883} \pm 0.005$	$\textbf{6.800} \pm 0.130$	$\textbf{0.942} \pm 0.007$	$\textbf{0.847} \pm 0.007$	$\underline{7.580} \pm 0.199$
PutNextLocal Image Only	Baseline	0.461 ± 0.016	0.286 ± 0.011	80.905 ± 1.551	$\underline{\textbf{0.521}} \pm 0.016$	$\underline{\textbf{0.342}} \pm 0.012$	$\textbf{71.521} \pm 1.571$
PutNextLocal Image + D.Texts	concat-fusion	0.427 ± 0.016	0.267 ± 0.011	81.403 ± 1.597	0.459 ± 0.016	0.308 ± 0.012	74.918 ± 1.603
	FiLM (Perez et al., 2018)	$\underline{0.518} \pm 0.016$	$\underline{0.322} \pm 0.011$	$\underline{75.397} \pm 1.561$	0.508 ± 0.016	0.324 ± 0.011	72.029 ± 1.549
	attention-fusion (ours)	$\textbf{0.654} \pm 0.015$	$\textbf{0.415} \pm 0.011$	$\textbf{66.287} \pm 1.442$	$\textbf{0.683} \pm 0.015$	$\textbf{0.444} \pm 0.011$	$\textbf{61.152} \pm 1.418$

Table 3: Comparison of four models with/without descriptive texts on BabyAI++. Succ. and $\mathcal{R}_{avg}$ denote the success rate and average reward, the higher the better. N_{epi} denotes the average steps taken in each episode, the lower the better. The performance is evaluated and averaged for 1,000 episodes on training and testing configurations. The **best** and **second-best** values are highlighted.

visual observation embedding). The combined embedding is processed with another CNN to produce the final embedding:

$$F^{final} = \text{CNN}([F, W * D])$$

When in certain task settings an instruction is also provided, the output embedding of the **attention-fusion** network is modified with a FiLM layer that takes as input the instruction text.

5 EXPERIMENTS

We evaluate the proposed models (1) image-only, 2) concat-fusion, 3) FiLM, and 4) attention-fusion on three levels in BabyAI++: GoToRedBall-v1, GoToRedBall-v2, and PutNextLocal (see Tab. 2). All three levels are on an 8×8 grid with increasing task difficulty and training steps {5M, 10M, 50M}. Due to page limit, we leave details on experiment set-up and more results (ablation study) in Appendix C.

Benefits of descriptive texts in language grounding Fig. 3 and Tab. 3 show training curves and quantitative comparisons of four baseline models. In general, the attention-fusion model achieves the best overall performance on both training and testing environments. For the most difficult PutNextLocal level, our model holds 13.6% and 16.2% improvement over the second-best model in success rate for training and testing configurations, respectively. Moreover, we observe that although FiLM obtains better performance on the training environments, it generalizes worse on unseen testing environments in comparison to other fusion methods and the image-only baseline. We hypothesize that FiLM overfits to the training dynamics by memorizing the sentences that appear during training rather than learning proper word-level meaning. This reminds us that current techniques are not effective for language grounding and dynamic environments like BabyAI++.

Learning from instructive and descriptive texts Apart from descriptive text, instructions also play an important role because the targets for more advanced levels (e.g., GotoObj) vary at each episode. From BabyAI and Table 3, FiLM is effective in learning conditioned instructions but cannot deal with descriptive texts well. Consequently, we propose a hybrid model that deploys the attention-

Figure 3: Comparison of proposed image-only and image+text models `PutNextLocal` during the training. Supplementary figures for other environments are provided in Appendix C.1

Setting	Model	Texts	Training			Testing		
			Succ.	$\mathcal{R}_{avg}$	N_{epi}	Succ.	$\mathcal{R}_{avg}$	N_{epi}
Image Only	Baseline	no texts	0.648 ± 0.015	0.497 ± 0.013	57.061 ± 1.725	0.631 ± 0.015	0.486 ± 0.013	58.842 ± 1.748
Image + Texts	concat-fusion	instructive	0.657 ± 0.015	0.507 ± 0.013	55.867 ± 1.685	0.632 ± 0.015	0.487 ± 0.013	59.193 ± 1.703
		descriptive	0.653 ± 0.015	0.516 ± 0.013	54.898 ± 1.720	0.632 ± 0.015	0.498 ± 0.013	57.622 ± 1.755
		all texts	0.640 ± 0.015	0.487 ± 0.013	62.142 ± 1.753	0.641 ± 0.015	0.486 ± 0.013	62.675 ± 1.766
	FiLM Perez et al. (2018)	descriptive	$\underline{0.723} \pm 0.014$	0.567 ± 0.012	52.108 ± 1.601	0.673 ± 0.015	0.526 ± 0.013	56.841 ± 1.699
		all texts	0.716 ± 0.014	$\underline{0.569} \pm 0.013$	$\underline{51.902} \pm 1.654$	$\underline{0.697} \pm 0.015$	$\underline{0.552} \pm 0.013$	$\underline{53.549} \pm 1.661$
	att-fusion + FiLM (ours)	all texts	$\mathbf{0.761} \pm 0.013$	$\mathbf{0.622} \pm 0.012$	$\mathbf{48.210} \pm 1.646$	$\mathbf{0.732} \pm 0.014$	$\mathbf{0.610} \pm 0.013$	$\mathbf{48.758} \pm 1.682$

Table 4: Comparison of proposed models with different types of texts on `GoToObj`, where the targets and dynamics of environments are altered at each episode. **Best**, **Second Best**.

fusion model to ground the descriptive language and then conditions with the task instructions using FiLM. Tab. 4 shows proposed hybrid model surpasses other baselines models by a large margin.

6 CONCLUSION

We augment BabyAI with variable dynamics and descriptive text and formulate tasks and benchmarks that evaluate RL agents in this setting. We adapt existing instruction-following baselines and propose a fusion approach baseline for leveraging descriptive text in grid-world environments. Our results show descriptive texts are useful for agents to generalize environments with variable (or even unseen) dynamics by learning language-grounding. We believe the proposed BabyAI++ platform, with its public code and baseline implementations, will further spur research development in this area.

7 ACKNOWLEDGEMENTS

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pp. 3674–3683. IEEE Computer Society, 2018.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture, 2016.

SRK Branavan, David Silver, and Regina Barzilay. Learning to win by reading manuals in a monte-carlo framework. *Journal of Artificial Intelligence Research*, 43:661–704, 2012.

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. Gated-attention architectures for task-oriented language grounding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Konstantinos I. Chatzilygeroudis, Roberto Rama, Rituraj Kaushik, Dorian Goepp, Vassilis Vassiliades, and Jean-Baptiste Mouret. Black-box data-efficient policy search for robotics. In *IROS*, pp. 51–58. IEEE, 2017.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. In *ICLR (Poster)*. OpenReview.net, 2019.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL http://arxiv.org/abs/1406.1078.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1282–1289, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/cobbe19a.html.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl2: Fast reinforcement learning via slow reinforcement learning. arxiv, 2016. *arXiv preprint arXiv:1611.02779*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Prasoon Goyal, Scott Niekum, and Raymond J. Mooney. Using natural language for reward shaping in reinforcement learning. In *IJCAI*, pp. 2385–2391. ijcai.org, 2019.

Karl Moritz Hermann, Mateusz Malinowski, Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, and Raia Hadsell. Learning to follow directions in street view. *CoRR*, abs/1903.00401, 2019.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *NIPS*, 2016.

Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. Are you looking? grounding to multiple modalities in vision-and-language navigation. In *ACL (1)*, pp. 6551–6557. Association for Computational Linguistics, 2019.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob N. Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. A survey of reinforcement learning informed by natural language. In *IJCAI*, pp. 6309–6317. ijcai.org, 2019.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. Grounding language for transfer in deep reinforcement learning. *arXiv preprint arXiv:1708.00133*, 2017.

Takayuki Osa, Voot Tangkaratt, and Masashi Sugiyama. Hierarchical reinforcement learning via advantage-weighted information maximization. *ArXiv*, abs/1901.01365, 2019.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, pp. 3942–3951. AAAI Press, 2018.

Deirdre Quillen, Eric Jang, Ofir Nachum, Chelsea Finn, Julian Ibarz, and Sergey Levine. Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. In *ICRA*, pp. 6284–6291. IEEE, 2018.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–, October 2017.

Catherine E Snow. Mothers' speech to children learning language. *Child development*, pp. 549–565, 1972.

Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment, 2018.

Haonan Yu, Haichao Zhang, and Wei Xu. Interactive grounded language acquisition and generalization in a 2d world. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=H1UOm4gA-.

A BABYAI++ DETAILS

A.1 DYNAMIC FLOORS

A BabyAI++ level that uses dynamic tiles need to specify parameters such as which types of tiles can appear, how many types of tiles can appear simultaneously, and the frequency by which the tiles appear (in place of a normal tile). The descriptive dynamics of a level can be described by a set of tile colors $\mathcal{C}$ and the set of dynamic properties $\mathcal{P}$. Each instance of the level initializes a new many to one mapping from color to property $\mathcal{M}(c) : \mathcal{C} \mapsto \mathcal{P}$, as well as the color mapping of a grid location $\mathcal{G}(x, y) : \mathbb{Z}^2 \mapsto \{\mathcal{C}, 0\}$ where 0 represents no color. The following tile properties are currently implemented:

- *trap*: ends episode with zero reward upon enter;
- *slippery*: walking over slippery tile increments the time counter by just a half, thereby increasing reward;
- *flipLeftRight*: swap left and right action when agent is on this tile;
- *flipUpDown*: causes agent to move backwards when attempting forward movement;
- *sticky*: agent need to take 3 actions to leave the tile;
- *magic*: if the agent spends more than 1 turn on this tile, the agent is moved downward by 1.

A.2 TRAINING AND TESTING CONFIGURATIONS

To evaluate the ability of proposed models to ground descriptive texts in unseen environments, we partition our environments into *training* and *testing* configurations. The training set omits certain floor colors to environment dynamic mappings. More specifically, training instances randomly generate $\mathcal{M}(c)_{train}$ such that $(c, \mathcal{M}(c)_{train}) \in \mathcal{C} \times \mathcal{P} \setminus \mathcal{H}$ where $\mathcal{H}$ denotes the held-out (c, p) pairs in the training, $\mathcal{C}$ and $\mathcal{P}$ denote color and tile property sets.

On the contrary, all the possible color-property pairs are allowed during the testing, i.e., $(c, \mathcal{M}(c)_{test}) \in \mathcal{C} \times \mathcal{P}$. Taking `GoToRedBall-v1` for an example, green and blue floor tiles cannot be endowed with *trap* and *slippery* properties in training, but the mappings are allowed during testing. More details about the differences in color-property sets between these environments are presented in Table A1.

Environment	N tiles	Distractor	trap	slippery	flipLeftRight	flipUpDown	sticky	magic	Partial Text
GoToRedBall-v1	2		✓				✓✓		
GoToRedBall-v2	3		✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	
PutNextLocal	2	✓	✓	✓	✓✓				
GoToObj	3	✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	
GoToObj-Partial	3	✓		✓✓	✓✓✓	✓✓	✓✓	✓✓✓	✓

Table A1: Properties of levels used in experiments (*training* configurations). "N tiles" is the number of types of tiles that are used simultaneously used in one episode. "Distractor" is whether distractor objects are used. The colored ✓ denote this property is enabled for specific type of floor tiles in training (e.g., for `GoToRedBall-v1` level, the blue floors can be either trap or sticky, and the green floors can be either slippery and sticky).

B DETAILS OF RL ALGORITHMS

Image-Only Model The image-only RL model takes as input the local scene observation as viewed by the agent in the BabyAI grid world, as described in Sec. 2 (a 7x7x3 symbolic representation). A CNN-architecture is used to create the final embedding. We refer the reader to our code for additional architecture details.

The **Image + Text** based models take, along with the image observation, a text description describing the model dynamics, and, when applicable, a text instruction (depending on the BabyAI++ level). All text descriptions and instructions are first processed with a word-level embedding layer that is trained from scratch, and then fed into a Gated-Recurrent-Unit Network (GRU) (Cho et al., 2014) to encode the sentences into embeddings. Each model then differs in how the text embedding is utilized

with the observation embedding to enhance the final output embedding. We now describe the three architectures which we implement to utilize the descriptive text embedding.

Concat-Fusion In the concat-fusion model, we concatentate both embedding vectors together to generate the final output embedding vector. Note that this method does not explicitly perform language-grounding; that is, the text embeddings of the model dynamics for all the different tiles, D, are directly combined with the input observation embedding F:

$$F^{final} = [F, D]$$

FiLM We also implement a FiLM (Perez et al., 2018) based model, which uses the description embedding as input to two neural network "controllers" that output the parameters for the linear transformation (γ_i, β_i) on each image embedding feature F_i:

$$F_i^{final} = \gamma_i F i + \beta_i$$

The FiLM model is the current standard benchmark model in the BabyAI environment.

All agents for each model are trained using Advantage Actor Critic method (A2C) method (Mnih et al., 2016) with Proximal Policy Optimization (PPO) (Schulman et al., 2017). Each model differs only in the representation that is provided as input to the actor and critic networks. A two-layer CNN of width 128 is used as the feature extraction backbone for processing the state input. Policy and value networks are two-layer fully connected networks with hidden dimension of 64. The FiLM and RNN network used for processing text input also have two layers, with a hidden dimension of 128.

C SUPPLEMENTARY EXPERIMENTS

C.1 SUPPLEMENTARY RESULTS FOR TABLE 3 AND FIGURE 3

We provide supplementary results of Table 3 and Figure 3 on other environments on BabyAI++. In Figure. A1, we also show the learning curve of proposed hybrid model (att-fusion + FiLM) with both instructive and descriptive texts on `GoToObj` where the targets and dynamics will be altered at each episode. It is shown that our model (purple line) also surpasses other baseline models in training configurations.

	Image Only	Baseline	0.906 ± 0.009	0.792 ± 0.009	12.145 ± 0.492	0.829 ± 0.012	0.730 ± 0.011	13.531 ± 0.577
`GoToRedBall-v2`		concat-fusion	0.897 ± 0.010	0.784 ± 0.009	12.163 ± 0.489	$\mathbf{0.852} \pm 0.011$	0.741 ± 0.011	$\mathbf{13.343} \pm 0.553$
	Image + D.Texts	FiLM (Perez et al., 2018)	$\mathbf{0.943} \pm 0.007$	$\mathbf{0.840} \pm 0.007$	$\mathbf{9.808} \pm 0.386$	0.843 ± 0.012	$\mathbf{0.749} \pm 0.011$	13.774 ± 0.617
		attention-fusion (ours)	$\underline{\mathbf{0.933}} \pm 0.008$	$\underline{\mathbf{0.829}} \pm 0.008$	$\underline{\mathbf{10.646}} \pm 0.439$	$\underline{\mathbf{0.850}} \pm 0.011$	$\underline{\mathbf{0.745}} \pm 0.011$	$\underline{\mathbf{13.434}} \pm 0.567$

Table A2: Comparison of four models with/without descriptive texts on `GotoRedBall-v2`. Succ. and $\mathcal{R}_{avg}$ denote the success rate and average reward, the higher the better. N_{epi} denotes the average steps taken in each episode, the lower the better. For all metrics, we present the sample mean together with standard error. The performance is evaluated and averaged for 1,000 episodes on training and testing configurations. The **best** and <u>**second-best**</u> values in each setting are highlighted.

C.2 ABLATION STUDY

To further validate that the benefits are coming from the descriptive texts rather than extra model capacity, we conducted an ablation study on the texts for the FiLM model. Specifically, to show the utility of descriptive text as a vessel for knowledge transfer, we generate and replace original inputs with the following nonsensical texts:

- `Lorem Ipsum`: Generate random texts using `lorem` library[1]. The embedding dictionary is enlarged with the increase of training steps.
- `Random Texts`: Generate random texts from a pre-defined dictionary which contains the same number of irrelevant words as the descriptive texts in this environment.
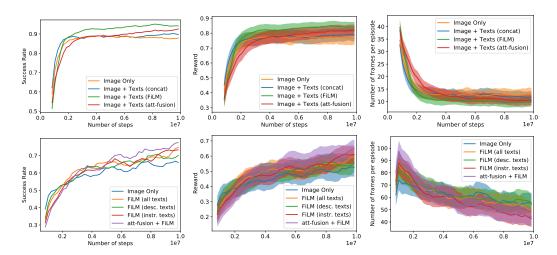
---

[1]https://pypi.org/project/lorem/

Figure A1: Comparison of proposed image-only and image+text models on `GoToRedBall-v2` (top) and `GoToObj` (bottom) during the training.

- `Shuffled Texts`: Shuffle the descriptive texts randomly at each episode. In this case, the context is broken thus the mapping for the color of tiles and their properties are difficult is difficult to learn.

Table A3 shows ablation results on `GoToRedBall-v2`. Specifically, the model with meaningful descriptive texts results in the best performance on both training and testing setting. Note that we observe that even the random texts could bring the benefits by introducing randomness and implicit exploration, which is consistent with previous literature (Branavan et al., 2012). There is still large room for more efficient utilization of descriptive texts, which is an promising direction and merits further study.

Setting	Texts	Training			Testing		
		Succ.	$\mathcal{R}_{avg}$	N_{epi}	Succ.	$\mathcal{R}_{avg}$	N_{epi}
Image Only	Baseline	0.897 ± 0.010	0.782 ± 0.009	12.398 ± 0.501	0.846 ± 0.011	0.739 ± 0.011	14.079 ± 0.575
Image + Texts	lorem ipsum	0.932 ± 0.008	0.826 ± 0.008	10.266 ± 0.413	0.839 ± 0.012	0.742 ± 0.011	12.968 ± 0.566
	random texts	0.932 ± 0.008	0.824 ± 0.008	10.276 ± 0.400	0.825 ± 0.012	0.725 ± 0.011	13.275 ± 0.564
	shuffled texts	0.925 ± 0.008	0.825 ± 0.008	$\mathbf{10.076} \pm 0.410$	0.842 ± 0.012	0.738 ± 0.011	13.786 ± 0.584
	descriptive texts	$\mathbf{0.941} \pm 0.007$	$\mathbf{0.837} \pm 0.007$	10.179 ± 0.406	$\mathbf{0.855} \pm 0.011$	$\mathbf{0.763} \pm 0.010$	$\mathbf{12.608} \pm 0.566$

Table A3: Ablation Study on the `GoToRedBall-v2` with FiLM. Here 'lorem ipsum', and 'random texts' represent generating random meaningless sentences with the same length from lorem ipsum dictionary and fixed-size pre-defined dictionary. 'shuffled texts' denotes shuffling the descriptions randomly at each episode.

C.3 VISUALIZATION OF TRAJECTORY OF MODELS

To give more intuitive understanding of the benefits of learning from descriptive texts, we visualize the trajectories (inferred path) of different models on BabyAI++ *testing* configuration. An example trajectory visualization for baseline, FiLM and attention-fusion models is provided in Figure A2. In this example, blue tiles are *slippery* (taking half time unit to pass) and green tiles are *sticky* (taking twice time unit to pass). In consequence, blue floors are "good" ones for agents to obtain higher reward. As shown in Figure A2, The attention fusion model is capable of utilizing the slippery tiles to approach the target red ball (pink line), which verifies the effectiveness of proposed method in language grounding under unseen testing environments. On the contrary, image-only baseline and FiLM models directly go to the red ball without realizing the different dynamics in testing (orange line).

D   Limitations

D.1   Environment Limitations

BabyAI++ is a good start towards more realistic environments that can help RL agents better leverage human-priors about the world to bootstrap learning and generalization. However, a critical component to a good environment is defining concrete tasks and settings that guarantees that the RL agent demonstrably learn a particular skill, such as language-grounding, to complete a task. In our current implementation of BabyAI++, because we randomly generate different tasks and model dynamics, we do not explicitly ensure that an RL agent's ability to do language grounding is a necessary requirement for completing the task. To truly show that a task sett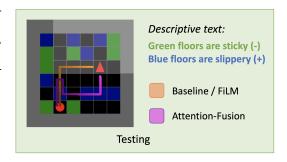ing can accurately test a particular skill, such as compositional learning, we may need to design hand-crafted expert-policies that explicitly use the provided information and see if these policies have significant difference in performance based on the information provided and how it is used. With this information, we can enhance the random generation of the task (such as grid size, number of objects, tiles, placement) to make sure the task cannot be solved without the RL agent achieving a certain skill.

Figure A2: Trajectory of proposed models on `GoToRedBall-v1`.

Another limitation of the BabyAI environment is that it is currently limited to the single-agent setting. This prevents more complex model dynamics that can make the the platform more realistic. We are interested in augmenting the environment with multiple agents and relevant text descriptions to make the environment more exciting.

D.2   Method Limitations

As depicted in our results in section 5, baseline methods such as **image-only** can complete the task only a fraction of the time. The **image+text** methods, some of which may not actually be performing language grounding, have comparable results with each other. No method is completing the tasks at 100% success rate and may still be taking more than the optimal number of steps, indicating that environment is a good test ground for new RL development that better leverages task descriptions to solve the task. Unfortunately, a limitation of the methods is that random text (Tab. 4) seems to provide similar benefits as the actual environment text descriptions. This seems to indicate that while the models which use text descriptions do perform better than no text descriptions, true language grounding has not been achieved. We need to investigate further why this might be the case.