Combined Text-Visual Attention Models for Robot Task Learning and Execution

Giuseppe Rauso[1][0009−0004−3226−8527], Riccardo Caccavale[1][0000−0001−8636−7628], and Alberto Finzi[1][0000−0001−6255−6221]

University of Naples "Federico II", Naples, Italy
{giuseppe.rauso, riccardo.caccavale, alberto.finzi}@unina.it

Abstract. In this work, we explore the interplay between text and visual attention mechanisms in a robot reinforcement learning setting, where robotic tasks are conveyed through natural language instructions. Specifically, we propose a novel approach aimed at enhancing robot task learning and execution by leveraging an integrated multimodal attention model that associates task-relevant environmental features with related words in the natural language mission text. We illustrate the overall framework architecture along with the learning process, emphasizing the interaction between textual and visual feature-based attention mechanisms. The method is trained in MiniGrid environments using the Proximal Policy Optimization algorithm, and its performance is evaluated by comparing the proposed architecture with a baseline that lacks attentional mechanisms. Experimental results demonstrate the efficacy of the approach also highlighting its potential in behavior transparency.

Keywords: Language Conditioned Reinforcement Learning · Multi-modal Attention · Behavior Transparency · Robot Task Learning.

1 Introduction

This work presents a novel approach for enhancing robot task learning and execution through the integration of combined text and visual attention models. The concept of attention, extensively studied in cognitive neuroscience, underpins various cognitive models introduced to explain different behaviors, ranging from active perception to cognitive control, with visual attention being the most examined form. Attention models and mechanisms have been also widely adopted and utilized in the field of artificial intelligence, with particular success in machine learning. This is primarily due to the ability to improve performance and accelerate training in many cases, also making the development of models more efficient. In particular, attention mechanisms in transformers have revolutionized the field of natural language processing (NLP) by enabling models to weigh the relevance of different words in sentences contextually, thus capturing long-range dependencies and interrelated patterns in texts.

In this work, we investigate the interaction between text and visual attention models in a reinforcement learning setting, where robotic agents are instructed

to accomplish tasks specified by natural language sentences. We address this issue within a language conditioned reinforcement learning setting, where joint representations of observation and textual representations are typically used to enhance policy generalization and transferability to novel/unseen environments [1, 19–21] or to enhance learning from human demonstration [8]. In this context, we assume textually specified mission goals and train the robotic agent to generate and exploit a combined multimodal attention model, where task-relevant words in the mission description are mapped into related salient visual features detected by the agent. This is achieved in a reinforcement learning setting by training the agent to generate and exploit word-related attention maps, which are suitably combined to coherently relate textual description with visual features and effectively orient the agent behavior. Such attention mechanisms are intended to support the agent's ability to focus solely on objects that correspond to the words present in the given mission, thereby enhancing the effectiveness of task learning and execution. Moreover, the alignment between salient visual features and task-relevant words aims to support the transparency of the agent's attentional behavior during task execution. We address these issues by proposing an integrated framework endowed with combined attention mechanisms, which is trained to accomplish simple tasks in MiniGrid environments using the Proximal Policy Optimization algorithm. We illustrate and discuss the overall architecture along with the learning process, emphasizing the interaction between textual and visual attention models. The approach is evaluated by comparing the proposed system with a baseline framework that lacks attentional mechanisms.The evaluation also includes assessing the quality of word-feature associations in the generated attention maps. The experimental results demonstrate the advantage of our approach in terms of efficacy and transparency.

2 Related Work

Over the years, various models of attentional mechanisms have been proposed in the context of machine learning, primarily inspired by studies derived from neuroscience [13]. Indeed, models of this type have been used in various contexts, including image and video classification and captioning [15, 23], translation [5, 26], and even in combination with text for question answering [3, 4]. Attention mechanisms have also been used in the context of reinforcement learning, as in [25], where the *Deep Attention Recurrent Q-Network* (DARQN) is proposed, adding a soft attention and a hard attention to the *Recurrent Deep Q-Network* (RDQN) [12], or in [18], where a soft attention mechanism is used to highlight the task-relevant features of the frames in combination with the *Deep Q-Network* (DQN) [16]. In this cases, the saliency maps are learned exclusively through rewards and highlight the most important visual features in a given frame that the agent needs to focus on. In [10], a multi-attention mechanism is proposed, which is used in parallel on different segments of sensory inputs for navigation in a grid environment. This approach allows the model to focus on smaller parts of the input, achieving greater sample efficiency during learn-

ing. In the literature, we also find applications of the *self-attention* mechanism in a reinforcement learning context, such as [14], where the Markovian property underlying reinforcement learning is leveraged to achieve spatio-temporal attention, or in [27], where self-attention is used to calculate the relationships between observed entities. Other works, however, have studied the application of attention mechanisms that leverage different sources, thus making it no longer a "self" attention. This is the case in [17], where the query vectors are produced from the output of an LSTM layer, while the key and value vectors are produced from the encoding of the visual observation. However, in our work, we aim to study the interactions between natural language and what the agent observes in the environment. In the field of *language conditioned reinforcement learning*, several works have explored this possibility to define goals or instructions, such as in [1,2,20,21], also leveraging gated attention mechanisms [11], and combining images and text in the calculation of attention [19]. Indeed, the idea of directly comparing the representation of text with what the agent perceives to achieve *multimodal attention*, as seen in [19], aligns closely with the concept underlying this work. However, their goals are fundamentally different; they focus on conceptual reinforcement learning where, besides maximizing reward, the objective is to extract concepts from entities in the environment based on textual scene descriptions. In our case, the text provided to the agent represents the goal, and the aim is to demonstrate how words in the task are mapped to what the agent observes in the environment. This involves creating attention maps and weights for each word, thereby enhancing both the agent's performance and the interpretability of the relationships between the text and observations developed during training using only the reward as a feedback.

3 Proposed Approach

We assume a robot task learning problem, where goals are provided in natural language. In this setting, we propose an approach based on multimodal attention mechanisms that leverages both the observed features and the words of the sentence representing the task. Our goal here is twofold: beyond enhancing the learning and execution process, our aim is to simultaneously ground the words to the associated observed features (related to objects and their characteristics, such as colors) through per-word attentional maps and weights. As a side effect, the proposed method allows for an additional level of transparency in the agent's behavior, as text attention and feature attention values are trained to be aligned and related to the task under execution. The proposed architecture is end-to-end, and both the execution of tasks and the learning of attentional maps, crucial for achieving better performance, occur solely through the reward obtained in the environment. We operate within a reinforcement learning context and employ the *Proximal Policy Optimization* (PPO) [22] algorithm for training. In the following we detail the proposed method.

3.1 Minigrid Environment

We demonstrate our approach in environments defined in *BabyAI* [8], a platform based on *MiniGrid* [9] that features grid-based simulated scenarios and tasks (see Fig. 1) formulated using a subset of a synthetic language called *Baby language*. This language is a small subset of English but is combinatorially rich; indeed, although intentionally kept simple, it contains 2.48×10^{19} possible instructions. These instructions include tasks such as reaching, picking up, opening doors, and placing objects next to others, as well as combinations like the "and" of two tasks or a sequence (before/after). In this work, we use only instructions of the type "go to <Descr>" and "pick up <Descr>," where <Descr> describes the object with an article, color (including none), and type of object. Therefore, possible sentences in the environments used for this study include phrases like "go to the red key", "pick up a box", "go to the ball".

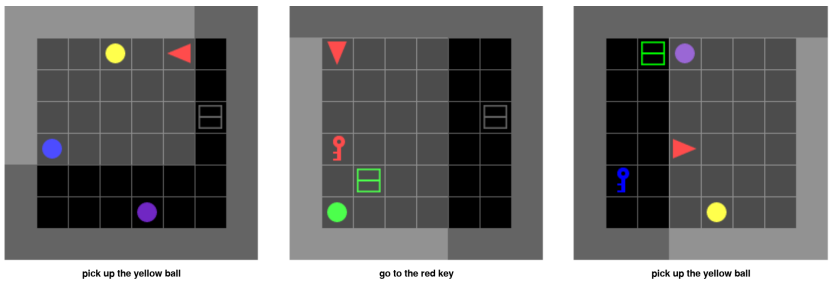

Fig. 1: Examples of MiniGrid environments used for this work, featuring "go to" and "pick up" tasks.

3.2 Background

We frame our approach in the context of a reinforcement learning problem, where the agent interacts with the environment to maximize cumulative reward. Formally, at each time step t, the agent is in some state $s_t \in S$ and chooses the next action $a_t \in A$ based on a policy π that can be deterministic, defined as $\pi : S \to A$, or stochastic, thereby determining a probability $\pi(a_t|s_t)$. The agent receives a reward r_{t+1} according to a reward function $R : S \times A \to \mathbb{R}$. Then, with a probability $p(s_{t+1}|s_t, a_t)$ it moves to the next state s_{t+1}. In particular, the environments are created based on the MiniGrid environments and associated with goal-augmented *Partially Observable Markov Decision Processes* (POMDPs), formally described by the tuple $(S, A, \Omega, p, R, G, O, \gamma)$, where Ω is the observation space, O the probabilistic observation model, G the goal space and γ the discount factor. The reward function thus becomes a goal-conditioned reward function $R : S \times A \times G \to \mathbb{R}$. We use PPO to solve the problem, which is

an on-policy policy gradient algorithm that maximizes the following objective:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \quad (1)$$

where ϵ is a hyperparameter, $\hat{A}_t$ is an estimator of the advantage function at timestep t, measuring the value of the selected action compared to the expected value of the state, $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the ratio between the new and old policies π parametrized by θ and θ_{old} respectively. This objective penalizes overly abrupt changes in the policy, aiming to keep the ratio $r_t(\theta)$ from deviating too much from 1. We rely on the original version of the approach [22] where the overall objective function is formulated as follows:

$$L_t^{\text{PPO}}(\theta) = \hat{\mathbb{E}}_t[L_t^{\text{CLIP}}(\theta) - c_1 L_t^{\text{VF}}(\theta) + c_2 S[\pi_\theta](s_t)] \quad (2)$$

with c_1, c_2, c_3 coefficients, S an entropy bonus, and $L_t^{\text{VF}} = (V_\theta(s_t) - V_t^{\text{targ}})^2$.

3.3 System Architecture and Learning Framework

The system we propose takes as input the features observed in the agent's field of view along with the mission defined in natural language and generates the associated policy exploiting world-related attention maps. The learning framework is detailed as follows (see Fig. 2). Let $O \in \mathbb{R}^{f_h \times f_w \times 3}$ and $g \in \mathbb{R}^m$ be, respectively, the portion of the grid observed by the agent (field of view) with height and width f_h and f_w, and m the maximum length (in words) of the task specification. We define $\tilde{O} = Conv(O)$ as the encoding of the observation through a convolutional network, and $\tilde{g} = Embedding(g)$ as the embedding of the task tokens. Let $\tilde{O}_{\text{flatten}}$ denote the flattening of the feature maps outputted by the convolutional network, transitioning from shape (K_{out}, f_h, f_w) to $(f_h \cdot f_w, K_{out})$, where K_{out} is number of filters used in the last convolutional layer. Drawing inspiration from the *scaled dot-product attention* mechanism with query, key, and value proposed in [26], we obtain the attention matrix

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (3)$$

where Q and K are, respectively, the projection of the embedding of the task (or mission) $\tilde{g}$ and the encoding of the observation $\tilde{O}_{\text{flatten}}$ onto a space of dimension d_k. By reshaping the rows of this matrix back to a shape of (f_h, f_w), we obtain m attention maps, one for each word. Each of these maps highlights the cells related to the corresponding word in the portion of the grid observed by the agent (see Fig. 3). However, we consider the attention maps as row vectors of A for the subsequent formalizations. To amplify or reduce the signal in relevant positions in the observed feature map based on the mission words, instead of directly multiplying A by a matrix of values (as in [26]), we propose the alternative method detailed below. We aim to derive attention weights for individual words based on what the agent is observing. Specifically, we want to determine which words are more "salient" for the portion of the grid that the agent is observing.

To this end, we compute the *Shannon entropy* [24] on the rows of the matrix A, which represent categorical distributions:

$$\forall i = 1, \ldots, m, \qquad H(A_i) = -\sum_{j=1}^{f_h \cdot f_w} A_{ij} \cdot \log A_{ij} \tag{4}$$

To obtain the attention weights for individual words, we apply the softmax to the negated entropy vector:

$$w = \text{softmax}\left(\begin{bmatrix} -H(A_1) \\ \vdots \\ -H(A_m) \end{bmatrix}\right) \tag{5}$$

Finally, to obtain the attention map, we calculate the weighted sum of the m rows of the matrix A, where the weights are the entropies obtained for each row:

$$M = \sum_{i=1}^{m} w_i \cdot A_i \tag{6}$$

This map M is applied to each feature map to highlight only the cells that represent some element in the mission text:

$$\forall i = 1, \ldots, f_h \cdot f_w, \qquad F_i = \tilde{O}^i_{\text{flatten}} \cdot M_i \tag{7}$$

Thus, by assessing the word-cell relevance, we effectively exploit a multimodal attention mechanism. The filtered feature maps are then fed into an LSTM recurrent layer, enabling the agent to operate in a partially observable environment. The output of this layer is concatenated with the output of a GRU recurrent layer that processes the mission text.

4 Empirical Evaluation

The proposed framework is assessed considering both the system performance and the quality of the word-feature grounding in the generated attention maps. We assess the effectiveness of the multimodal attention system by comparing its performance during both training and testing phases with a baseline system lacking attention mechanisms. Specifically, the baseline is created by removing the Attention Module (see the dotted box in Fig. 2) and directly passing the convolutional network encoding $\tilde{O}_{\text{flatten}}$ to the LSTM recurrent network, while concatenating the output with the text encoding from the GRU recurrent network which receives the word embeddings $\tilde{g}$.

4.1 Training

To train the agents we employed an environment defined by a single 8×8 room without walls (except for the perimeter ones) containing 4 randomly chosen and

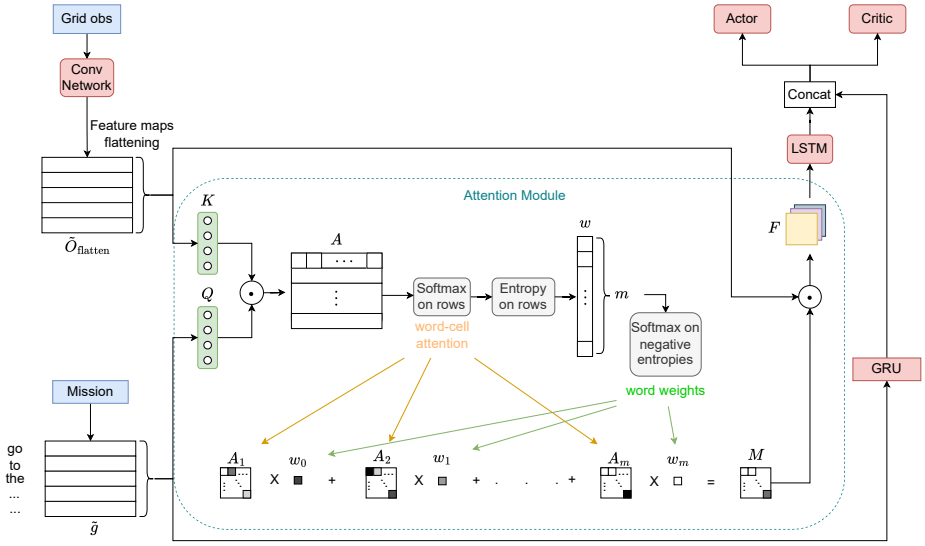

Fig. 2: System architecture. The components of the attention module are enclosed within the turquoise dotted box. The attention maps $A_1, \ldots, A_m$ are displayed as matrix form for clarity, but they are row vectors of matrix A as detailed in Section 3.3.

colored objects. For each task, one object is to be reached or picked up, while the other 3 are distractors. We use the observation encoding proposed with the MiniGrid environments, namely a 7×7 grid with 3 channels, representing the agent's field of view, where each cell of the grid is encoded as a tuple *(object id, object color, object state)*. The reward function follows the framework's default environment settings, providing a reward ranging from 0 to 1 only at the end of the episode based on the steps taken to complete the task; in case of failure, the reward is 0. We use the 'done' action to prompt the agent to recognize when it has successfully completed an episode-ending action, such as reaching a specific object (in the case of 'go to') or picking up an object (in the case of 'pick up'). However, the agent can still pick up objects without issuing a 'done' action, for example, to clear obstacles from its path. The agents were trained for 15 million steps in the described environment. The evolution of the average reward can be observed in Fig. 4, where the model without attention converges to a lower value and exhibits significant instability compared to the model with attention.

4.2 Testing

After training, we can evaluate the performance of the proposed framework and the accuracy of the trained attentional mechanisms in word-feature grounding.

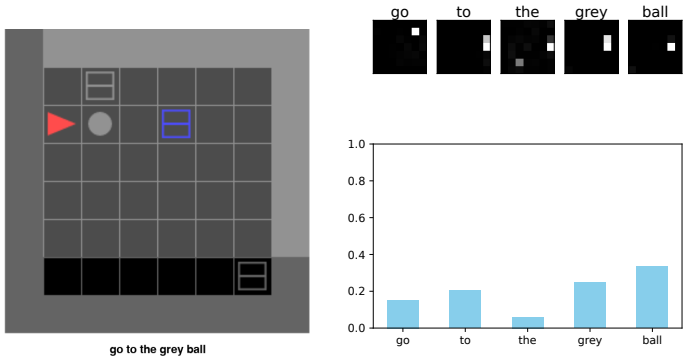

Fig. 3: Snapshot of the grid scenario (left) with the associated per-word attention maps (Top right) and the word weights w (Bottom right). The per-world maps are agent-centric with the agent positioned in the middle of the right side, facing left, with the positions of objects relative to the agent mirrored along the vertical axis. In this case, the agent must complete task described by the phrase "go to the grey ball".

Performance. The performance of the proposed framework is compared to a setup without attention. The evaluation is conducted across various environments with different dimensions and numbers of objects to assess the models' robustness in larger settings. These environments present greater challenges, as agents operate with a more limited field of view and encounter additional distractors. Fig. 6 shows the average reward and success rate over 100 episodes, averaged across 5 different seeds for both agents. The collected results demonstrate the robustness of the agent equipped with the multimodal attention mechanisms compared to the agent without attention. Indeed, the former experiences a significantly smaller decline in performance as the number of distractors increases. In the most challenging scenarios, it maintains a mean reward between 0.8 and 0.85 and around a 90% success rate. In contrast, the model without attention experiences a more significant performance drop, showing that the proposed attentional approach not only enhances performance, but also improves generalization and robustness to environmental changes.

Accuracy evaluation. To evaluate the quality of the generated attention maps, we aim to measure how well the system places higher values in the maps at the positions of the visible objects referred by the words in the mission specification. As illustrated in Sec. 3.3, each word ω in the mission text is associated with an attention map A^{ω}. For each word ω, we can also identify n_{ω} related objects in the agent's field of view (e.g., the words "red" is associated with n_{red} visible objects). We refer to $A^{\omega}_{x_i,y_i}$ as the value of the A^{ω} attention map at the coordinates (x_i, y_i), representing the actual position of the i-th visible object. Given a set of objects in the agent field of view, the accuracy of the attention map A^{ω} can be defined

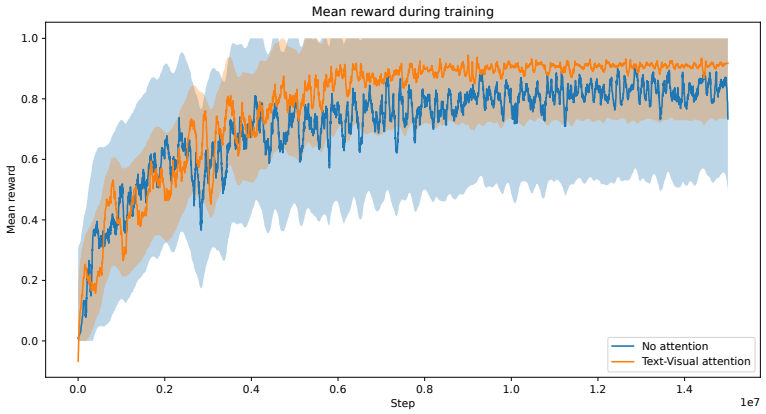

Fig. 4: Evolution of the reward for the model with attention and the one without attention during training.

as follows:

$$\mathrm{acc}_{\omega} = \frac{\sum_i^{n_\omega} \mathrm{p}_i^{\omega}}{n_\omega} \quad \text{with} \quad \mathrm{p}_i^{\omega} = \begin{cases} 1 & \text{if } A^{\omega}_{x_i,y_i} \geq \alpha \frac{1}{n_\omega} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\frac{\alpha}{n_\omega}$ is a threshold to determine whether the object is correctly detected (hit) or not (miss). Here, $\alpha \in [0, 1]$ is a parameter representing a percentage of $\frac{1}{n_\omega}$. This fraction calibrates the threshold with respect to the total weight of the map divided by the number of visible word-related objects, representing the value an object's position would have in an attention map that assigns equal weights to all word-related objects within the agent's field of view.

The accuracy measure introduced above assesses the system's ability to correlate words with the precise target positions. To relax this notion, we introduce variations of this accuracy measure accounting for hits near the target. In this regard, we introduce the accuracy measure in the neighborhood of the object's position $acc_{\omega,ng}$ that evaluates whether in the attention map there is at least one cell that exceeds the threshold in the proximity of the object position. This is defined as follows.

$$\mathrm{acc}_{\omega,ng} = \frac{\sum_i^{n_\omega} \mathrm{p}_{i,ng}^{\omega}}{n_\omega} \quad \text{with} \quad \mathrm{p}_{i,ng}^{\omega} = \begin{cases} 1 & \text{if } \exists (k,z) \in \mathrm{ng}(x_i, y_i) : A^{\omega}_{k,z} \geq \alpha \frac{1}{n_\omega} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\mathrm{ng}(x_i, y_i)$ is the set of coordinate pairs of positions surrounding the real position of the object, meaning the positions at distance 1 in each direction. However, during the experiments, we noticed biased directions for displacements in the object proximity, i.e., shifted one step forward from the actual position

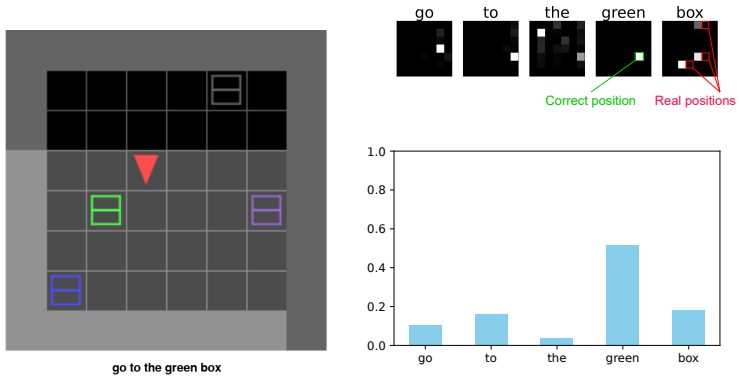

Fig. 5: During the execution of "go to the green box" the agent highlights salient task-related values aligning text salience (bottom right) and attention maps (top right). Here, the green box is correctly related and emphasized, while the box position shifted forward by one.

in the agent's visual field of view (see Fig. 5). To account for these small biases we introduce a more focused accuracy assessment in the target's neighborhood where the hits are shifted in some direction. In particular, we focus on hits with a forward shift as the bias:

$$\mathrm{acc}_{\omega,bs} = \frac{\sum_i^{n_\omega} \mathrm{p}_{i,bs}^{\omega}}{n_\omega} \quad \text{with} \quad \mathrm{p}_{i,bs}^{\omega} = \begin{cases} 1 & \text{if } A_{x_i,\overline{y_i}}^{\omega} \geq \alpha \frac{1}{\hat{n}_\omega} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where $\overline{y}$ is the shifted coordinate. Then, we can measure the accuracy $\mathrm{acc}_{\omega,\mathrm{cmb}}$ of the combined hit of the actual and shifted positions of the target objects:

$$\mathrm{acc}_{\omega,cmb} = \frac{\sum_i^{n_\omega} \mathrm{p}_{i,\mathrm{cmb}}^{\omega}}{n_\omega} \quad \text{with} \quad \mathrm{p}_{i,\mathrm{cmb}}^{\omega} = \begin{cases} \mathrm{p}_i^{\omega} \vee \mathrm{p}_{i,bs}^{\omega} & \text{if } y_i > 0 \\ \mathrm{p}_i^{\omega} & \text{otherwise} \end{cases} \tag{11}$$

Accuracy Results. To assess the system performance we focus on words related to the objects and colors using the accuracy measures introduced above. During testing, we evaluate the maps only when at least one object of the type/color mentioned in the mission sentence is in the agent's field of view. For the accuracy measures we set $\alpha = 0.5$, averaging the results over the number of evaluation step for each episode (i.e., a task mission). We further average the collected values over 100 episodes and across executions with 5 different seeds. The results are provided in Fig. 7 for the object-related maps and in Fig. 8 for the color-related maps. For these cases, we illustrate the accuracy values as the the grid dimension and the number of objects increase. The accuracy in the neighborhood $acc_{\omega,ng}$ (green line in Fig. 7 and 8), as expected, is higher than the others, ranging from over 90% to around 80%. However, the combined accuracy $acc_{\omega,cmb}$ (red line in

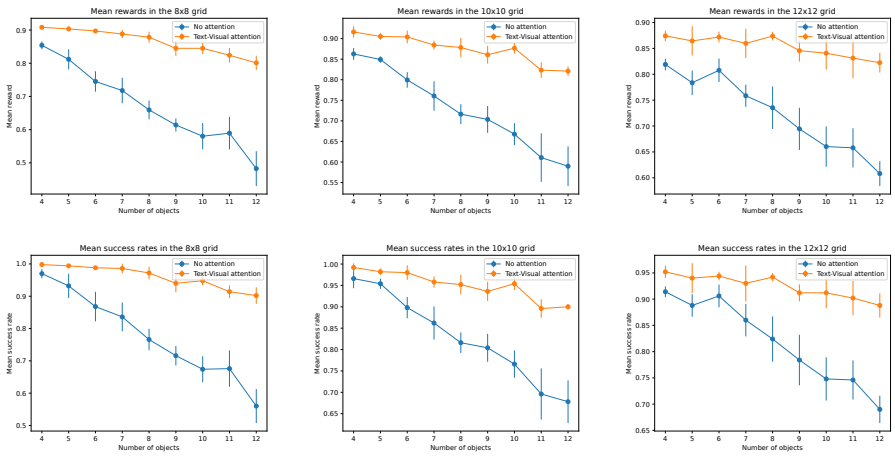

Fig. 6: Average reward (first row) and success rate (second row) with standard deviations in the test phase over 100 episodes, averaged over 5 seeds for both the model with attention and the model without attention varying with the number of objects and grid size.

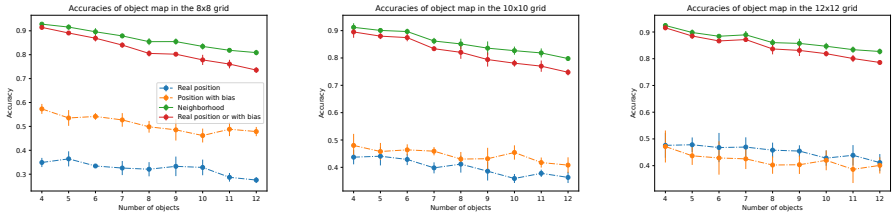

Fig. 7: Mean and standard deviation of the accuracies for the object attention maps described in Section 4.2, averaged over 100 episodes and then averaged across 5 seeds, varying with the number of objects and grid size.

Fig. 7) remains close, with a slight degradation as the number of objects -and distractors- increases. Therefore, the agent remains capable of correlating colors and objects with their associated positions in the attention maps, despite a fixed bias that occasionally shifts objects and colors forward. We can also observe that this shift is more evident for the object-related maps (see Fig. 7), where the accuracy of the actual position acc_{ω} (dotted blue line) is lower than the shifted accuracy $acc_{\omega,bs}$ (dotted orange line). On the other hand, higher acc_{ω} values can be observed for the color-related attention maps (see Fig. 8). Here the values range between 80% and 70%, with a relatively stable performance as the number of objects increases. Overall, the accuracy evaluation shows the system's capability to correlate objects, colors, and values in the word-related attentional maps. This ability is maintained even in increasingly complex scenarios not ac-

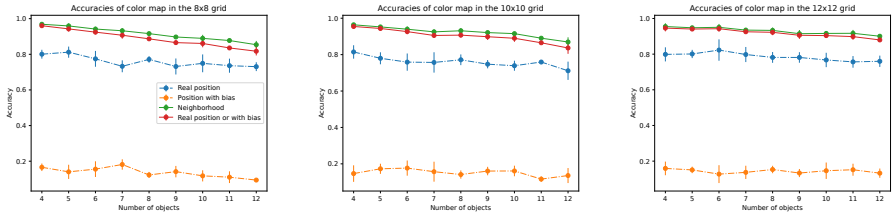

Fig. 8: Mean and standard deviation of the accuracies for the color attention maps described in Section 4.2, averaged over 100 episodes and then averaged across 5 seeds, varying with the number of objects and grid size.

counted for during training. The generated attention maps can then establish a coherent alignment between saliency in the linguistic and feature domains, providing insights about the agent's attentional focus during task execution.

5 Conclusions

We presented a novel approach to task learning, with agents instructed in natural language, that leverages multimodal attention mechanisms aligning mission words and observations relevance. Specifically, in the proposed method the agent is trained to generate per-world attention maps, thereby grounding the task words along with their relevance in the environmental observations. We show that the generated attention maps not only enhance learning and execution performance, but also provide an additional level of transparency in the agents attentional behavior, in that key words from input sentences guide the agent during task execution, enabling focused interactions with specific features and locations within the environment grid. The empirical results demonstrate the advantage of the approach in terms of average reward and success rate compared to the architecture without the proposed mechanisms. Moreover, the study on word-feature association shows the system ability to ground relevant words in the environment with high accuracy. In future work, we aim to investigate the robustness of the proposed attentional mechanisms in more complex scenarios and with more structured tasks. We plan to explore the feasibility of incremental learning, starting with simpler tasks to establish the grounding of words in the environment, as proposed in this study. Subsequently, we will leverage this capability to learn more complex tasks, in these settings, we intend to explore the integration of executive attentional mechanisms [6] suitable for flexible task orchestration [7].

Acknowledgments. This work was partially supported by the projects: EU Horizon INVERSE (grant 101136067) and euROBIN (grant 101070596), Melody (PRIN PNRR prot. P2022XALNS), SPACE IT UP (PE15 ASI/MUR).

References

1. Akakzia, A., Colas, C., Oudeyer, P.Y., Chetouani, M., Sigaud, O.: Grounding language to autonomously-acquired skills via goal generation (2021)
2. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments (2018)
3. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Learning to compose neural networks for question answering (2016)
4. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Neural module networks (2017)
5. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2016)
6. Caccavale, R., Finzi, A.: Learning attentional regulations for structured tasks execution in robotic cognitive control. Autonomous Robots **43**, 2229 – 2243 (2019)
7. Caccavale, R., Finzi, A.: A robotic cognitive control framework for collaborative task execution and learning. Topics in Cognitive Science **14**(2), 327–343 (2022)
8. Chevalier-Boisvert, M., Bahdanau, D., Lahlou, S., Willems, L., Saharia, C., Nguyen, T.H., Bengio, Y.: Babyai: A platform to study the sample efficiency of grounded language learning (2019)
9. Chevalier-Boisvert, M., Dai, B., Towers, M., de Lazcano, R., Willems, L., Lahlou, S., Pal, S., Castro, P.S., Terry, J.: Minigrid & miniworld: Modular i& customizable reinforcement learning environments for goal-oriented tasks (2023)
10. Choi, J., Lee, B.J., Zhang, B.T.: Multi-focus attention network for efficient deep reinforcement learning. ArXiv **abs/1712.04603** (2017), https://api.semanticscholar.org/CorpusID:3824441
11. Colas, C., Karch, T., Lair, N., Dussoux, J.M., Moulin-Frier, C., Dominey, P.F., Oudeyer, P.Y.: Language as a cognitive tool to imagine goals in curiosity-driven exploration (2020)
12. Hausknecht, M., Stone, P.: Deep recurrent q-learning for partially observable mdps (2017)
13. Lindsay, G.W.: Attention in psychology, neuroscience, and machine learning. Frontiers in Computational Neuroscience **14** (2020). https://doi.org/10.3389/fncom.2020.00029, https://www.frontiersin.org/articles/10.3389/fncom.2020.00029
14. Manchin, A., Abbasnejad, E., van den Hengel, A.: Reinforcement learning with attention that works: A self-supervised approach (2019)
15. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent models of visual attention (2014)
16. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M.A., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**, 529–533 (2015), https://api.semanticscholar.org/CorpusID:205242740
17. Mott, A., Zoran, D., Chrzanowski, M., Wierstra, D., Rezende, D.J.: Towards interpretable reinforcement learning using attention augmented agents (2019)
18. Mousavi, S., Schukat, M., Howley, E., Borji, A., Mozayani, N.: Learning to predict where to look in interactive environments using deep recurrent q-learning (2017)
19. Peng, S., Hu, X., Zhang, R., Guo, J., Yi, Q., Chen, R., Du, Z., Li, L., Guo, Q., Chen, Y.: Conceptual reinforcement learning for language-conditioned tasks (2023)

20. Röder, F., Eppe, M.: Language-conditioned reinforcement learning to solve misunderstandings with action corrections (2022)
21. Röder, F., Eppe, M., Wermter, S.: Grounding hindsight instructions in multi-goal reinforcement learning for robotics (2022)
22. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017)
23. Shan, M., Atanasov, N.: A spatiotemporal model with visual attention for video classification (2017)
24. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal **27**(3), 379–423 (1948). https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
25. Sorokin, I., Seleznev, A., Pavlov, M., Fedorov, A., Ignateva, A.: Deep attention recurrent q-network (2015)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2023)
27. Zambaldi, V.F., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D.P., Lillicrap, T.P., Lockhart, E., Shanahan, M., Langston, V., Pascanu, R., Botvinick, M.M., Vinyals, O., Battaglia, P.W.: Deep reinforcement learning with relational inductive biases. In: International Conference on Learning Representations (2018), https://api.semanticscholar.org/CorpusID:59233950