HELP ME EXPLORE: MINIMAL SOCIAL INTERVENTIONS FOR GRAPH-BASED AUTOTELIC AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

In the quest for autonomous agents learning open-ended repertoires of skills, most works take a *Piagetian perspective*: learning trajectories are the results of interactions between developmental agents and their *physical environment*. The *Vygotskian perspective*, on the other hand, emphasizes the centrality of the *socio-cultural environment*: higher cognitive functions emerge from transmissions of socio-cultural processes internalized by the agent. This paper acknowledges these two perspectives and presents GANGSTR, a hybrid agent engaging in both individual and social goal-directed exploration. In a 5-block manipulation domain, GANGSTR discovers and learns to master tens of thousands of configurations. In individual phases, the agent engages in *autotelic learning*; it generates, pursues and makes progress towards its own goals. To this end, it builds a graph to represent the network of discovered configurations and to navigate between them. In social phases, a simulated social partner suggests goal configurations at the frontier of the agent's current capabilities. This paper makes two contributions: 1) a minimal social interaction protocol called *Help Me Explore* (HME); 2) GANGSTR, a graph-based autotelic agent. As this paper shows, coupling individual and social exploration enables the GANGSTR agent to discover and master the most complex configurations (e.g. stacks of 5 blocks) with only minimal intervention.

1 INTRODUCTION

During early child development, open-ended play offers a learning context for exploring and mastering various skills. From solving puzzles to manipulating objects, children continuously and autonomously acquire complex behaviors driven by their intrinsic curiosity. Recently, artificial intelligence (AI) researchers have been interested in this open-ended learning framework to design artificial agents that can evolve in unbounded environments and grow repertoires of skills in a self-supervised fashion. Several argued that such artificial agents must be *autotelic*: they must be intrinsically motivated to represent, generate and pursue their own goals (Steels, 2004; Forestier et al., 2017; Colas et al., 2020b).

To this end, classic developmental approaches in AI relied on the agents' sensorimotor module to represent their goals (Eysenbach et al., 2018; Warde-Farley et al., 2018; Nair et al., 2018; 2020; Pong et al., 2019; Colas et al., 2019). This was inspired by the notion of *physical situatedness* in developmental psychology (Piaget, 1954; Needham & Libertus, 2011): children first learn to represent their goals through their embodied interactions with their surroundings, then use their intrinsic motivations to sample and pursue these goals. Unfortunately, this limits the behavioral diversity, as the distribution of *learnable* goals is highly grounded in the distribution of *previously encountered* goals. Hence, classic approaches in intrinsically motivated reinforcement learning (IMRL) usually rely on further assumptions, curricula and biases to drive the exploration process and increase behavioral diversity (Eysenbach et al., 2018; Warde-Farley et al., 2018; Nair et al., 2018; 2020; Pong et al., 2019; Colas et al., 2019).

Interestingly, developmental psychology has shown that children rely on two additional features: 1) *imaginative play* and 2) *social situatedness*. On the one hand, imaginative play allows children to imagine and target goals that they did not encounter previously; by substituting an object with another for example (Lewis et al., 1992; Stagnitti & Unsworth, 2000). The IMAGINE approach attempts to leverage this feature in IMRL by using language to imagine new goals (Colas et al., 2020a).

On the other hand, social situatedness is essential for the emergence of sophisticated forms of intelligence (Vygotsky, 1994; Bruner, 1973; Tomasello, 2009). In the *Zone of Proximal Development* (ZPD) framework for instance (Vygotsky, 1978), a social partner (SP) boosts the learner's capabilities by first ensuring the mastery of simpler and foundational skills before the more complex ones (i.e. scaffolding). More generally, verbal and nonverbal interactions help acquire shared cognitive representations that make sens for both the SP and the learner. These findings have gained popularity in AI (Lindblom & Ziemke, 2002; Sigaud et al., 2021). In particular, we set ourselves in the framework of Sigaud et al. (2021), which investigates the design of artificial agents that can be both autotelic and social: autonomous teachable agents.

To facilitate these cultural interactions, research in cognitive science hypothesizes that human learning operates over structured and symbolic programs (Chater & Oaksford, 2013; Wang et al., 2019). These high-level representations offer a handful of efficient tools for manipulating information. Specifically, symbolic representations that are independent of languages and domains (such as images, symbolic predicates, ...) facilitate generalization of underlying concepts rather than memorization of specific situations. However, using these high-level tools requires a formal syntax and a cross-cultural understanding of their semantics. Fortunately, work in developmental psychology has shown that some of these requirements, precisely spatial relational predicates, are used by infants from a very young age (Mandler, 2012). This motivated recent works in developmental AI to endow artificial agents with such symbolic inductive biases (Tellex et al., 2011; Alomari et al., 2017; Kulick et al., 2013; Asai, 2019; Akakzia et al., 2020). Beyond facilitating the learning process of autotelic agents as shown in Akakzia et al. (2020), we believe this additional semantic layer simplifies the social interactions.

Contributions. This paper makes steps towards the design of teachable autotelic agents. We consider an object manipulation benchmark with 5 different blocks for our agents to explore. Following Akakzia et al. (2020), we consider a symbolic goal space using the semantic predicates *close* and *above*, which yield a goal space of around 70.000 semantic configurations of different complexities. Furthermore, based on the notion of ZPD, a synthetic social SP assists the agent's learning process by combining two intervention schemes: 1) the SP first proposes semantic goals that are at the *frontier* of the agent's capabilities, then 2) proposes goals *beyond* that frontier. Here, our focus being on the learning agent, we make the simplifying assumption that the SP has direct access to the current agent skills. This accounts for the fact that a good teacher who designs lessons adapted to their students capabilities has a model of such skills (Bruner, 1961).

Our main contributions are twofold:

- A novel social interaction protocol, **H**elp **M**e **E**xplore (HME), which intertwines individual learning and social learning phases. In the former, the agent trains to reach known semantic goals in an autotelic fashion. In the latter, it interacts with a simulated SP that drives its exploration process towards semantic goals beyond the frontier of its current skills.
- A goal-conditioned learning architecture named GANGSTR for **G**o**A**l-conditioned **N**eural **G**raph with **S**eman**T**ic **R**epresentations. GANGSTR is based on the Soft-Actor-Critic algorithm (SAC, Haarnoja et al., 2018). Both its actor and critic are implemented with Graph Neural Networks (GNN, Scarselli et al., 2005; Gilmer et al., 2017; Battaglia et al., 2018). Furthermore, the set of discovered semantic configurations is represented with a graph where nodes are semantic configurations and edges express the predicted capacity of the agent to move between them.

We show that, when engaged in the HME protocol, GANGSTR: (i) decomposes hard goals into sequences of easier intermediate steps using its semantic goal space representation, (ii) discovers and masters a diversity of complex goals it would not have discovered alone and (iii) requires only minimal social intervention and pursues its own goals most of the time.

2 Related work

Autotelic RL. The quest for autotelic agents first emerged in the field of developmental robotics (Steels, 2004; Oudeyer & Kaplan, 2007; Nguyen & Oudeyer, 2012; 2014; Baranes & Oudeyer, 2013; Forestier & Oudeyer, 2016; Forestier et al., 2017) and recently converged with state-of-the-art deep RL research (Eysenbach et al., 2018; Warde-Farley et al., 2018; Nair et al., 2018; 2020;

Pong et al., 2019; Colas et al., 2019; 2020a;b). Such agents rely on goal-conditioned RL algorithms (Schaul et al., 2015; Andrychowicz et al., 2017; Colas et al., 2020b) and automatic curriculum mechanisms (Portelas et al., 2020). Following the Piagetian tradition in psychology, most of these approaches consider individual agents interacting with their physical environment and leveraging forms of intrinsic motivations to power their exploration and learning progress (Bellemare et al., 2016; Achiam & Sastry, 2017; Nair et al., 2018; 2020; Burda et al., 2018; Pathak et al., 2019; Colas et al., 2019; Pong et al., 2019). These approaches sample goals from the set of already-known experiences, which limits their capacity to explore.

Social Learning. The ZPD describes the set of learning situations a child can solve with the help of a *more knowledgeable other* (Vygotsky, 1978). Throughout scaffolding and supportive activities, the SP uses their knowledge to drive learners beyond what they can do alone. Interactive learning research in AI investigates and models the ways human tutors can guide their agents' learning process (Thomaz & Breazeal, 2008; Argall et al., 2009; Celemin & Ruiz-del Solar, 2015; Najar et al., 2016; Christiano et al., 2017a). A category of teaching signals requires the SP to point out sequences of actions and states using demonstrations or preferences (Argall et al., 2009; Christiano et al., 2017a; Fournier et al., 2019). Another category assists the development of autotelic agents by providing them with challenging goals or guiding instructions (Grizou et al., 2013). As we want minimal interactions with the SP who does not show the agent how to act, we focus on the latter category.

Semantic Goal Representations. Developmental psychology suggests that humans rely on intuitive mechanisms of affordances, analogy and relational patterns to reason about their goals (Gibson, 1977; Holyoak, 2012). Endowing artificial agents with such inductive biases introduces a level of abstraction that is likely to facilitate the acquisition of complex tasks and the interaction with human teachers. Our set-up and semantic representation layer is similar to that of the DECSTR agent (Akakzia et al., 2020). However, there are several important discrepancies. First, in Akakzia et al. (2020), the training process was split into 3 sequential phases: sensorimotor learning, language grounding and instruction following. Given this split, instructions from the tutor could not be used to drive the learning process of the agent. By contrast, here sensorimotor learning and instruction following are intertwined, making it possible to focus on teaching curriculum questions. Besides, in Akakzia et al. (2020), the language grounding aspect was managed by a Language Goal Generator (LGG) component translating sentences from the tutor into semantic goals. Here, the language learning aspect is outside the scope of this study, as we consider that the tutor immediately communicates a goal in a semantic representation that the agent immediately understands.

Graph-Based Exploration. Humans solve novel problems by combining familiar skills and routines. They are able to represent complex systems within their environment as compositions of entities and their interactions (Navon, 1977; Marcus, 2001; Kemp & Tenenbaum, 2008). This motivates the design of artificial agents that learn structured representations of their world, such as graphs (Džeroski et al., 2001; Toyer et al., 2018). In this paper, we aim at endowing autotelic agents with a structured representation of their goal space. Close to us, the GO-Explore algorithm uses a graph to represent a discrete archive of interesting states (Ecoffet et al., 2019; 2021). They map these states to discretized cells. They first reach a selected cell, based on heuristics such as recency or visit counts in memory, then explore from that state. We take a different approach based on the semantic character of the goals we consider. Using its goal space graph embedding, GANGSTR autonomously evaluates its performances when hopping from one goal to another. It then uses it as a proxy to determine the optimal path to reach complex semantic goal configurations.

3 METHODS

In this section, we first introduce the simulated block manipulation environment and the semantic configuration space considered in this paper (Section 3.1). Then, we describe the autotelic algorithm used by our agent (Section 3.2). Finally, we explain how the simulated SP helps the learning agent to explore and master large and complex goal spaces. (Section 3.3). The anonymous code for the environment, agent and interaction setup as well as pretrained weights can be found at https://anonymous.4open.science/r/gangstr-2C4F.

3.1 ENVIRONMENT AND SEMANTIC CONFIGURATIONS

We extend the *Fetch Manipulate* domain from Akakzia et al. (2020) to 5 blocks, itself a variant of the standard *Fetch* domains (Plappert et al., 2018). The agent embodies a Fetch robotic arm. It controls the 3D velocity of its gripper, the 1D velocity of its fingers (4D action space) and observes the Cartesian and angular positions and velocities of its gripper, fingers and of the 5 blocks it faces. In addition, it perceives a binary semantic representation of the scene similar to the one introduced in Akakzia et al. (2020). This representation asserts the presence (1) or absence (0) of the binary spatial relations *above* and *close* for each of the 10 pairs of objects. These 3 relations (*close*, *A above B*, *B above A*) applied to each of the 10 possible pairs lead to 30 binary dimensions (against $3 \times 3 = 9$ dimensions with 3 blocks, Akakzia et al., 2020). The resulting configuration space contains 2^{30} configurations, among which ~ 70.000 are physically reachable. These semantic representations are inspired by the work of Jean Mandler on a minimal set of spatial primitives children seem to be born with, or to develop early in life (Mandler, 2012).

In contrast with Akakzia et al. (2020), when resetting the setup, the 5 blocks are randomly placed in contact with the table. That is, any pair of blocks can be either far or close, but never stacked.

3.2 INTRINSICALLY MOTIVATED GOAL-CONDITIONED REINFORCEMENT LEARNING

This section introduces GANGSTR, an autotelic algorithm whose objective is to discover and master all configurations within the semantic configuration space. GANGSTR leverages the *Soft Actor-Critic* algorithm (SAC, Haarnoja et al., 2018) and *Hindsight Experience Replay* mechanisms (HER, Andrychowicz et al., 2017) to train a goal-conditioned policy and critic implemented by *Graph Neural Networks* (GNNs, Gilmer et al., 2017; Battaglia et al., 2018). In addition, the agent progressively constructs a graph of discovered configurations that is further used to decompose complex goals into sequences of easier ones. Let us now detail these components.

Object-Centered Architecture. A series of recent work argued for the importance of incorporating inductive biases in the architectures of neural function approximators (Battaglia et al., 2018; Gilmer et al., 2017; Zaheer et al., 2017; Kipf & Welling, 2016; Ding et al., 2020; Karch et al., 2020; Akakzia et al., 2020). In manipulation domains, agents need to handle separate objects, each characterized by a set of specific features. Here, what is important is the relations between objects, not their particular organisation in the agent's state vector. GNNs are particularly adapted to this case: they focus on relations between nodes in the graph, enforce permutation invariance and can handle sets of objects of different sizes.

Our implementation encodes the features of each object in a specific node and the binary semantic relations between object pairs in their edges. Further information about the agent's body (and actions for the critic) are encoded in the global features of the graph. Following Message Passing GNNs (MPGNNs, Gilmer et al., 2017), we conduct neighborhood aggregation and graph-level pooling schemes using a single network shared across nodes but separate for the actor and critic. These inductive biases facilitate the representation of spatial relations between objects and enforce a permutation invariance that boosts generalization, see details in Appendix A.2.

Semantic Graph Construction. As it interacts with the environment, the GANGSTR agent experiences new semantic configurations. This experience is used to progressively construct a knowledge graph of semantic configurations. This knowledge graph should not be confused with the GNN's graph. Here, nodes correspond to semantic configurations and directed edges encode estimations of the agent's ability to transition from the initial to the final configuration. We call these measures success rates (SRs). After each episode and if they do not exist already, the agent creates two new nodes for the initial and final configuration and one edge for their transition (initialized to $\text{SR} = 0.5$). If the edge already exists, the agent updates the SR to reflect the episode's outcome (success or failure). For each edge in the graph, the SR is computed using an exponential moving average with $\alpha = 0.01$ that emphasizes the most recent outcomes. The estimated SR of a longer path can be computed by multiplying the estimated SRs of all its transitions.

Construction Biases. As training progresses, the number of edges in the semantic graph grows and can reach more than 500k edges at convergence. This results in very few outcomes stored in each edge, i.e. sample sizes too low to hope get reliable SR estimates. Unreliable SR estimates lead to

poor choices of sub-goal decompositions which, in turn, leads to poor performance. To fight this issue, we implement two biases:

- **Attention Bias:** The number of edges in the graph can be drastically reduced if the agent only creates edges for transitions where the modified relations correspond to moving a single object. This can be seen as a focus on *small steps*, see Appendix A.3 for details.
- **Edge Permutations:** Since the GNNs of the GANGSTR agent enforce a permutation invariance which significantly boosts generalization across behaviors that are equivalent under the permutation of the objects' identities (permutation-wise equivalence), the SRs of permutation-wise equivalent transitions are expected to be close. One way to increase the sample size of our SR estimates is thus to share outcomes of all permutation-wise equivalent edges. This increases the sample size by a factor 120 (number of permutations of 5 elements). In turn, this directly results in $\sqrt{120} \approx 11$ times tighter confidence intervals for the SR estimations (Bernoulli parameter), see Appendix A.3 for details.

Note that these two biases rely on a single assumption that was already made in the design of the GNN: the agent knows which semantic features correspond to which object. Another approach could be to train a neural classifier to predict the binary outcome given the initial and final configurations and use its output preceding the sigmoid activation as a continuous estimate of SR. However, the two simple biases presented above were enough to reach good performance in our case.

Automatic Decomposition. At each individual episode, the GANGSTR agent samples a goal uniformly from the set of discovered configurations. The agent then proceeds to compute a *decomposition* of this goal into a sequence of easier, intermediate goals leading to it. This decomposition follows one of two strategies:

1. **Shortest Path:** the agent selects one of the $k = 5$ shortest paths in terms of travelled edges.
2. **Safest Path:** the agent selects the path which maximizes the SR, that is

$$path^* = \arg\max_{path} \prod_{edge \in path} \text{SR}(edge) = \operatorname*{argmin}_{path} \sum_{edge \in path} -log(\text{SR}(edge)).$$

Always taking the shortest path can be risky when it includes unreliable transitions discovered in lucky trajectories. On the other hand, always taking the safest path can lead to long detours when SR estimates are unreliable. By taking one of the k shortest paths, the agent explores short paths and collects outcomes to robustify the estimations of their SRs. This exploration of short paths improves its ability to rule among them when it optimizes for the safest path. At test time, the agent takes the safest path to maximize its probability to reach the goal configuration. In all cases, optimal paths are obtained with the Djikstra algorithm (Dijkstra, 1971).

3.3 HELP ME EXPLORE: TEACHING CURRICULUM WITH THE SIMULATED SOCIAL PARTNER

Social GANGSTR agents alternate between autotelic episodes – pursuing their own goals – and social episodes –pursuing goals suggested by the SP. The ratio of autotelic vs social episodes is a hyperparameter studied in the experiments. Let us now describe which goals the SP suggests.

Representing Social Knowledge. The SP possesses domain knowledge: it knows all the semantic configurations and how to transit between them. To mirror the structured representational systems of humans (Inhelder & Piaget, 2013; Kemp & Tenenbaum, 2008), we encode this domain knowledge into a directed knowledge graph with a similar structure as the one learned by the agent (see Section 3.2). In practice, we use the network analysis library *networkit* to construct an oracle graph where nodes represent all imaginable semantic configurations. Two semantic configurations are linked with a directed edge if it is possible to reach one from the other by moving a single object. For more details about the SP oracle graph construction process, see Appendix A.4.

Reaching the Frontier and Beyond. In line with Vygotsky's notion of ZPD (Vygotsky, 1978), the SP aims at helping the agent to reach goals that are *just beyond its current capabilities*. Let us call *root* the initial configuration node $[0]^{30}$ where all blocks are co-planar and far apart. Let us call *frontier* the set of deepest nodes in the current agent's graph, i.e. the set of nodes where a breadth-first search starting from the root would stop. Within that frontier, we can discriminate two types of

nodes: 1) the *Stepping Stones Nodes*, from which further undiscovered semantic configurations can be reached and 2) the *Terminal Nodes*, from which no new configuration can be reached. Other nodes that do not belong to the frontier can be called *Intermediate Nodes*, see a minimalist illustration in Figure 1. In addition to its domain knowledge, the SP models the agent's capabilities by keeping a list of all the configurations it discovered. During social episodes, the SP suggests goal configurations uniformly sampled from the agent's frontier. As it does so, the agent focuses on its current hardest goals, and is more susceptible to encounter new configurations lying a few steps behind this frontier. When the SP sampled a *Stepping Stone* from the frontier and the agent successfully reached it, the SP further suggests an adjacent configuration *beyond the frontier*, unknown to the agent. With this help from the SP, the agent can make progress in its ZPD.

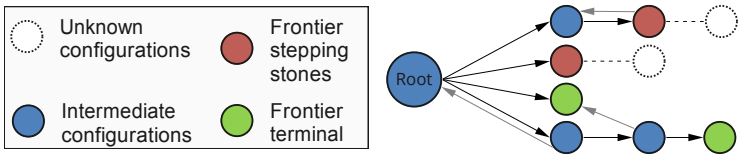


Figure 1: Illustration of a minimalist knowledge graph. As it explores, the agent progressively discovers new configurations and extends its frontier (red and green nodes).

4 EXPERIMENTS

This section studies the impact of HME social interventions on the learning trajectories and final performance of the GANGSTR agents. Section 4.1 illustrates the dynamics of HME interventions by zooming on the learning trajectory of a representative GANGSTR agent. Section 4.2 reports global measures of performance and how they vary as a function of the amount of HME interventions. Finally, Section 4.3 studies variations and ablations of the HME protocol. Let us first detail how we evaluate GANGSTR agents.

Evaluation. The subset of the 2^{30} expressible configurations the agent can physically reach is hard to compute *a priori*. To evaluate the agents, we hand-defined 11 classes of reachable configurations, each containing between 10 and 120 configurations (see Table 1 and Appendix A.6 for illustrations). This includes configurations where exactly i pairs of blocks are considered *close* (C_i), configurations containing stacks of size i (S_i), configurations containing pyramids of size 3 (P_3) and combinations of these. These classes are disjoint and their union does not cover the entire semantic configuration space (e.g. configurations $S_2 \& C_2$ are not contained). They do contain, however, classes that we consider *interesting*; in the sense that they can be easily described by humans (e.g. $P_3 \& S_2$ could be *"a tower of two and a small pyramid"*.

Evaluations are conducted each 50 cycles. During one cycle, the agent targets a goal, generates a path and follows up to 10 intermediate goals within rollouts of 40 timesteps. At test time, the agent is given 24 goal configurations uniformly sampled from each of the 11 classes (264 goals). A test episode is considered a success when the final configuration matches the goal configuration. The measure of the agent's success is the *global success rate*; the average of the 11 success rates per class. Testing episodes are conducted without exploration noise (deterministic policy) and are not added to the replay buffers (offline evaluations).

Class	C_1	C_2	C_3	S_2	S_3	$S_2 \& S_2$	$S_2 \& S_3$	P_3	$P_3 \& S_2$	S_4	S_5
# Goals	10	45	120	20	60	60	120	30	60	120	120

Table 1: Semantic classes used for evaluations. The C_i class regroups all configurations where exactly i pairs of blocks are *close*. The S_i class contains all configurations where exactly i blocks are piled up. The P_i class contains all configurations where i blocks form a pyramid. The & is a logical *AND*. All unspecified predicates are *false*, thus classes represent disjoint sets.

4.1 EFFECT OF HME ON THE LEARNING TRAJECTORY

Let us look at the dynamics induced by the HME interventions on the learning trajectory of a representative GANGSTR agent. Figure 2 considers an agent receiving goal suggestions in 20% of the

episodes (SP-20%). For four configuration classes, it presents: 1) the number of goals from that class the SP suggested (dashed lines); 2) the number of times GANGSTR reached a configuration from that class, no matter if it aimed at it or not (dotted lines) and 3) GANGSTR's success rate in that class (plain lines).

The four classes (*Stack 2&3*, *Pyr 3&Stack 2*, *Stack 4* and *Stack 5*) are considered *hard* because they involve either high piles or compositions of small structures that are unlikely to be discovered from scratch with random exploration. Indeed, the fact that dashed lines always precede dotted lines shows that GANGSTR does not discover these configurations before the SP intervenes (Figure 2).

For all difficult configurations, the dynamics follows the same pattern: 1) the agent discovers a stepping stone towards a yet-unknown configuration. This unknown configuration lies in the agent's ZPD, i.e. *just beyond* its current abilities. 2) The SP starts suggesting the stepping stone configuration as a goal and, if reached, immediately suggests the yet-unknown configuration from the ZPD (dashed line). 3) The agent discovers the configuration by exploring around the stepping stone configuration suggested by the SP; it encounters it more and more (dotted line). 4) Learning kicks in and the agents starts making progress on this configuration; it will soon master it (plain line). These figures are examples taken from the learning trajectory of a representative agent. Plots corresponding to additional configuration classes can be found in the Appendix A.6.

Agents can only progress on goals when they encountered the corresponding configuration or a similar one enough in the past. In the first case, they can learn to reach the goal with RL and hindsight learning; in the second, they rely on generalization. The SP thus acts as a facilitator, it progressively guides the agent's attention towards configurations that are just beyond those it already knows about and will soon master. In other words, it facilitates its exploration. By opening possibilities to the agent and showing it the way via the stepping stone, the SP dramatically increases the chances that the agent discovers the unknown configuration. This kind of intervention is minimal, because the SP only sporadically suggests new directions for exploration (see next section).

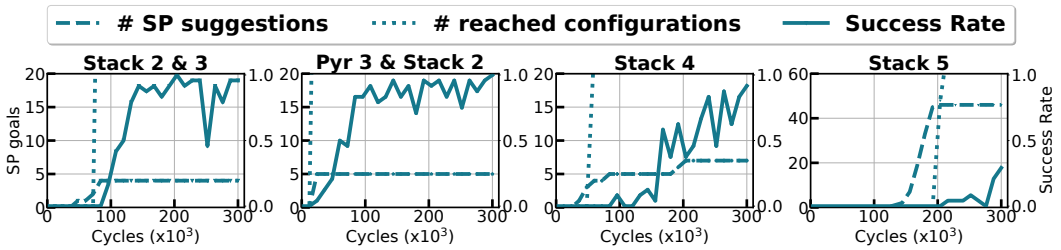


Figure 2: Learning trajectory of GANGSTR with 20% social intervention. We report the number of goals proposed by SP (dashed), the number of times GANGSTR encountered them (dotted) and its success rate (plain).

4.2 EFFECT OF THE SOCIAL LEARNING PHASES ON THE GLOBAL PERFORMANCE

Let us now look at the impact of social interventions on global performance metrics. Figure 3 presents the impacts of the entire spectrum of social interventions, from no social intervention (SP-0%) to constant social interventions (SP-100%). All social agents are found to significantly outperform the non-social agent SP-0% by more than 25%. Even 2% of social episodes is sufficient to trigger a large difference in the learning trajectories: whereas agents that were shown configurations they could eventually learn to achieve them, non-social agents could not discover them at all.

Figure 4 zooms on specific SP-0% (left), SP-2% (middle) and SP-20% (right) agents. SP-0% fails to discover stacks of 3 and more complex configurations (left), but adding 2% of social interventions is enough to open these possibilities to the agent, which quickly learns to master them (middle). Going up to 20% of social interventions is enough to discover all configuration classes (right). Additional class-specific performance can be found in Appendix A.6.

In Figure 3, SP-50% and SP-20% do not show significant differences at convergence, which suggests that constant social interventions are not required for optimal performance. Although SP-2% does not show significant difference either, this is most probably due to its high variance and to the small sample size, and should be confirmed with further experiments.

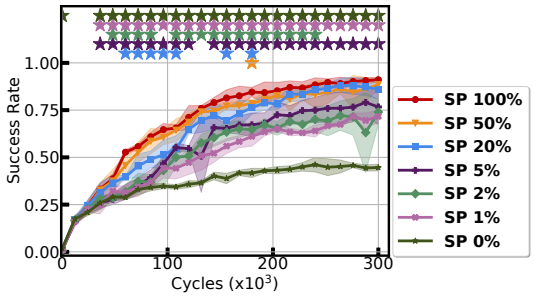


Figure 3: Global SR across training episodes for various proportion of social episodes. Mean $\pm$ standard deviation, computed over 5 seeds. Stars highlight statistical differences w.r.t. SP-100% (Welch's t-test with null hypothesis $\mathcal{H}_0$: no difference in the means, $\alpha = 0.05$).

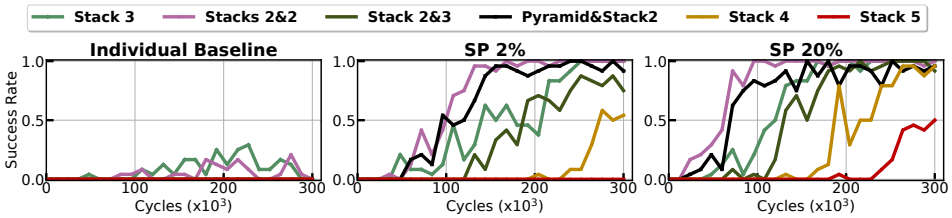
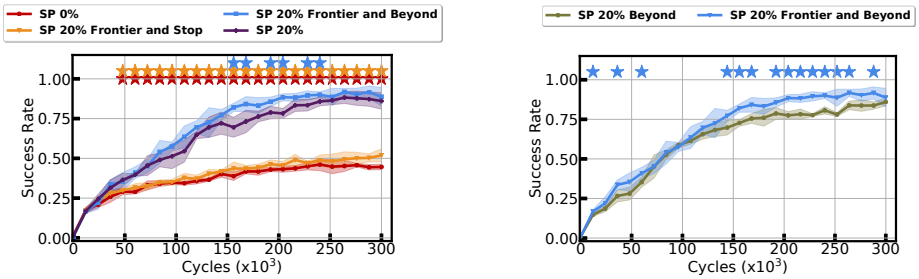


Figure 4: Success rates for each class. Ratio of SP interventions: 0% (Left), 2% (Middle) and 20% (Right).

4.3 Ablation Studies

In the HME protocol, the SP suggests a configuration at the frontier of the agent. Depending on this configuration, the interaction can take two forms: it is either a *terminal node* of the graph and the agent is simply asked to reach it, or a *stepping stone node* and the SP suggests an additional unknown node if the agent succeeds in reaching it. We call these two forms of interaction *Frontier and Stop* and *Frontier and Beyond* respectively. To investigate the relative contributions of these two forms of interaction, we ran one ablation using only *Frontier and Stop* interactions and one using only *Frontier and Beyond* interactions. Both ablations consider 20% of social episodes and are compared to SP-20% and SP-0% for reference.



(a) (b)

Figure 5: Global SR for various social interaction forms. Mean $\pm$ standard deviations, computed over 5 seeds. Stars indicate significant differences w.r.t. SP-20%. (Welch's t-test, $\alpha = 0.05$).

Figure 5a shows that the *Frontier and Stop* ablation alone does not bring significant advantages over the SP-0% condition. The *Frontier and Beyond* ablation seem to even outperform the SP-20% condition, showing that trading some of the *Frontier and Beyond* interactions with *Frontier and Stop* interactions might be slightly detrimental. The *Frontier and Stop* behaviors aimed at maintaining the attention of the agents on the hardest configurations (terminal nodes). It seems, however, that the GANGSTR agent can maintain its performance by simply sampling goal configurations uniformly.

In the *Frontier and Beyond* interactions, the SP first suggests a stepping stones, then suggests the adjacent unknown configuration. But do we really need this decomposition? The *Beyond* ablation

in Figure 5b only considers interactions where the SP directly suggests the unknown configuration without suggesting the stepping stones. Decomposing trajectories significantly speeds up the discovery and mastery of unknown configurations.

5 CONCLUSION AND FUTURE WORK

This paper makes a step towards *teachable autonomous agents*: the cognitive development of artificial agents must take its root in both social situatedness and the free exploration of the physical world (Sigaud et al., 2021). To this end, this paper introduces two contributions: 1) an autotelic agent that learns to discover and master thousands of semantic configurations, and 2) a social interaction protocol where the SP efficiently catalyzes the development of the learner. The learner embodies object-centered and relational inductive biases (GNN) and uses a knowledge graph to represent its goal space and decompose the hardest goals into sequences of intermediate steps (1). In the *Help Me Explore* interaction protocol, a simulated social partner (SP) simply suggests goals at the frontier of the agent knowledge (2). When these goals are reached and are good stepping stones for further exploration, the SP goes further and suggests unknown goal configurations—adjacent nodes just beyond that frontier. These goals represent the agent's ZPD: the set of goals it cannot reach alone (unknown configurations), but can reach with social guidance (when the SP shows the stepping stone). Although low-level exploration remains powered by undirected noise, the high-level exploration is supported by the graph itself and is extended by the suggestions of the SP. This simple exploratory curriculum unlocks the potential of the agent. With only 1% of social episodes, it can discover pyramids and stacks of 3, with 2% it can discover stacks of 4 and from 20% it reliably succeeds in stacks of 5.

Towards Better Autotelic Agents. Our autotelic agent embodies powerful inductive biases and succeeds in stacking up to 5 blocks without any demonstration. This said, it still relies on some knowledge in the semantic representations. As argued in Akakzia et al. (2020), this space is built from general predicates that are known to be used by humans early in their life (Mandler, 2012). It still relies on a limited and fixed set of predicates, which falls short of modeling the full extend of human representation capabilities. Future versions of GANGSTR should learn to construct semantic representations via their interactions with the physical and social world. One can think of unsupervised representation learning methods to learn from physical interactions (Burgess et al., 2019; Greff et al., 2019; Van Den Oord et al., 2017), and of socially-supervised representation learning methods to learn from social interactions (Christiano et al., 2017b; Stiennon et al., 2020). The selection of goals (now uniform) may also be improved by the introduction of *automatic curriculum learning* techniques. Note, however, that these automatic techniques could not replace our SP because they are limited to sample goals within the set of discovered configurations; thus they do not help the agent explore.

Towards Better Social Agents. Our current agent requires 20% social episodes to achieve all configurations (around 100,000 episodes). This remains too much to include actual humans in the loop. Hopefully, this can be significantly optimized. The ablation in Figure 5 shows that *Frontier and Beyond* interactions work well alone. Removing *Frontier and Stop* interactions would let the SP focus on them and would probably reduce the number of required social episodes. Furthermore, we found that the SP has helped the learner discover all configurations by cycle 200,000, which means it does not need to intervene any more in the last third of the cycles. Finally, the SP's guidance could be directly internalized by the agent such that each of SP's suggestions (and all their permutations) are stored in a list and re-targeted often by the agent. This way, the SP would only need to mention each (stepping-stone, unknown target) decomposition once for the agent to regularly train on all its permutations until reaching mastery.

The future of teachable autonomous agents lies in the diversity of social interactions they could learn from. Future work will aim at integrating multiple sources of social guidance using the framework of *pragmatic frames* introduced by Bruner (Bruner, 1985) and further refined in the context of robotics (Vollmer et al., 2016; Rohlfing et al., 2016). Pragmatic frames are verbal or non-verbal patterns of goal-oriented behaviors that evolve over repeated interactions between a learner and a teacher. As teachers and learners co-evolve these frames and learners learn to identify them from context, learners can integrate multiple sources of guidance in a way that will feel more natural for humans.

BROADER IMPACT STATEMENT

We believe human intervention is crucial in the quest for more explainable and safer autonomous robots. We exhibit a social protocol where our RL agents can have minimal interactions with a simulated SP and grow their repertoires of skills. Thus, this paper contributes in facilitating human teaching of artificial robots. By releasing our code and providing clear explanations of the number of seeds, error bars and statistical testing of our results, we believe we help efforts in reproducible science and allow the wider community to build upon and extend our work in the future.

ACKNOWLEDGMENTS

Anonymized for submission

REFERENCES

Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

Ahmed Akakzia, Cédric Colas, Pierre-Yves Oudeyer, Mohamed Chetouani, and Olivier Sigaud. Grounding language to autonomously-acquired skills via goal generation. *arXiv preprint arXiv:2006.07185*, 2020.

Muhannad Alomari, Paul Duckworth, David C Hogg, and Anthony G Cohn. Natural language acquisition and grounding for embodied robotic systems. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight Experience Replay. *arXiv preprint arXiv:1707.01495*, 2017.

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Masataro Asai. Unsupervised grounding of plannable first-order logic representation from images. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pp. 583–591, 2019.

Adrien Baranes and Pierre-Yves Oudeyer. Active Learning of Inverse Models with Intrinsically Motivated Goal Exploration in Robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013. Publisher: Elsevier.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29:1471–1479, 2016.

Jerome Bruner. Child's Talk: Learning to Use Language. *Child Language Teaching and Therapy*, 1 (1):111–114, 1985. Publisher: SAGE Publications Sage UK: London, England.

Jerome S Bruner. The act of discovery. *Harvard educational review*, 1961.

Jerome S. Bruner. Organization of early skilled action. *Child development*, pp. 1–11, 1973.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew Botvinick, and Alexander Lerchner. MONet: Unsupervised Scene Decomposition and Representation. *CoRR*, abs/1901.11390, 2019. URL http://arxiv.org/abs/1901.11390. _eprint: 1901.11390.

Carlos Celemin and Javier Ruiz-del Solar. Coach: Learning continuous actions from corrective advice communicated by humans. In *2015 International Conference on Advanced Robotics (ICAR)*, pp. 581–586. IEEE, 2015.

Nick Chater and Mike Oaksford. Programs as causal models: Speculations on mental programs and mental representation. *Cognitive Science*, 37(6):1171–1191, 2013.

Paul Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741*, 2017a.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep Reinforcement Learning from Human Preferences. In *Advances in neural information processing systems*, pp. 4299–4307, 2017b.

Cédric Colas, Pierre Fournier, Mohamed Chetouani, Olivier Sigaud, and Pierre-Yves Oudeyer. Curious: intrinsically motivated modular multi-goal reinforcement learning. In *International conference on machine learning*, pp. 1331–1340. PMLR, 2019.

Cédric Colas, Tristan Karch, Nicolas Lair, Jean-Michel Dussoux, Clément Moulin-Frier, Peter Ford Dominey, and Pierre-Yves Oudeyer. Language as a cognitive tool to imagine goals in curiosity-driven exploration. *arXiv preprint arXiv:2002.09253*, 2020a.

Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Intrinsically motivated goal-conditioned reinforcement learning: a short survey. *arXiv preprint arXiv:2012.09830*, 2020b.

Lisandro D Dalcin, Rodrigo R Paz, Pablo A Kler, and Alejandro Cosimo. Parallel distributed computing using python. *Advances in Water Resources*, 34(9):1124–1139, 2011.

Edsger Wybe Dijkstra. *A short introduction to the art of programming*, volume 4. Technische Hogeschool Eindhoven Eindhoven, 1971.

David Ding, Felix Hill, Adam Santoro, and Matt Botvinick. Object-based attention for spatio-temporal reasoning: Outperforming neuro-symbolic models with flexible distributed architectures. *arXiv preprint arXiv:2012.08508*, 2020.

Sašo Džeroski, Luc De Raedt, and Kurt Driessens. Relational reinforcement learning. *Machine learning*, 43(1):7–52, 2001.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Sébastien Forestier, Yoan Mollard, and Pierre-Yves Oudeyer. Intrinsically motivated goal exploration processes with automatic curriculum learning. *arXiv preprint arXiv:1708.02190*, 2017.

Sébastien Forestier and Pierre-Yves Oudeyer. Modular Active Curiosity-Driven Discovery of Tool Use. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 3965–3972. IEEE, 2016.

Pierre Fournier, Cédric Colas, Mohamed Chetouani, and Olivier Sigaud. CLIC: Curriculum learning and imitation for object control in non-rewarding environments. *IEEE Transactions on Cognitive and Developmental Systems*, 13:239–248, 2019.

James J Gibson. The theory of affordances. *Hilldale, USA*, 1(2):67–82, 1977.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

Klaus Greff, Raphaël Lopez Kaufmann, Rishab Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-Object Representation Learning with Iterative Variational Inference. *arXiv preprint arXiv:1903.00450*, 2019.

Jonathan Grizou, Manuel Lopes, and Pierre-Yves Oudeyer. Robot learning simultaneously a task and how to interpret human instructions. In *2013 IEEE third joint international conference on development and learning and epigenetic robotics (ICDL)*, pp. 1–8. IEEE, 2013.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Keith J Holyoak. Analogy and relational reasoning. 2012.

Brbel Inhelder and Jean Piaget. *The early growth of logic in the child: Classification and seriation*, volume 83. Routledge, 2013.

Tristan Karch, Cédric Colas, Laetitia Teodorescu, Clément Moulin-Frier, and Pierre-Yves Oudeyer. Deep sets for generalization in rl. *arXiv preprint arXiv:2003.09443*, 2020.

Charles Kemp and Joshua B Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692, 2008.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Johannes Kulick, Marc Toussaint, Tobias Lang, and Manuel Lopes. Active learning for teaching a robot grounded relational symbols. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

Vicky Lewis, Jill Boucher, and Arlene Astell. The assessment of symbolic play in young children: A prototype test. *International Journal of Language & Communication Disorders*, 27(3):231–245, 1992.

Jessica Lindblom and Tom Ziemke. Social situatedness: Vygotsky and beyond. In *Proceedings of the Second International Workshop on Epigenetic Robotics. Modelling cognitive development in robotic systems*, pp. 71–78. Lund University Cognitive Studies, 2002.

Jean M Mandler. On the spatial foundations of the conceptual system and its enrichment. *Cognitive science*, 36(3):421–451, 2012.

Gary Marcus. The algebraic mind, 2001.

Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *arXiv preprint arXiv:1807.04742*, 2018.

Ashvin Nair, Shikhar Bahl, Alexander Khazatsky, Vitchyr Pong, Glen Berseth, and Sergey Levine. Contextual imagined goals for self-supervised robotic learning. In *Conference on Robot Learning*, pp. 530–539. PMLR, 2020.

Anis Najar, Olivier Sigaud, and Mohamed Chetouani. Training a robot with evaluative feedback and unlabeled guidance signals. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 261–266. IEEE, 2016.

David Navon. Forest before trees: The precedence of global features in visual perception. *Cognitive psychology*, 9(3):353–383, 1977.

Amy Needham and Klaus Libertus. Embodiment in early development. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(1):117–123, 2011.

Sao Mai Nguyen and Pierre-Yves Oudeyer. Active Choice of Teachers, Learning Strategies and Goals for a Socially Guided Intrinsic Motivation Learner. *Paladyn*, 3(3):136–146, 2012. Publisher: Springer.

Sao Mai Nguyen and Pierre-Yves Oudeyer. Socially Guided Intrinsic Motivation for Robot Learning of Motor Skills. *Autonomous Robots*, 36(3):273–294, 2014. Publisher: Springer.

Pierre-Yves Oudeyer and Frederic Kaplan. What Is Intrinsic Motivation? A Typology of Computational Approaches. *Frontiers in neurorobotics*, 1:6, 2007. Publisher: Frontiers.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pp. 5062–5071. PMLR, 2019.

J. Piaget. *The construction of reality in the child*. Basic Books, New York, 1954.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.

Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic curriculum learning for deep rl: A short survey. *arXiv preprint arXiv:2003.04664*, 2020.

Katharina J. Rohlfing, Britta Wrede, Anna-Lisa Vollmer, and Pierre-Yves Oudeyer. An Alternative to Mapping a Word onto a Concept in Language Acquisition: Pragmatic Frames. *Frontiers in Psychology*, 7, April 2016. ISSN 1664-1078. doi: 10.3389/fpsyg.2016.00470. URL http://journal.frontiersin.org/Article/10.3389/fpsyg.2016.00470/abstract.

Franco Scarselli, Sweah Liang Yong, Marco Gori, Markus Hagenbuchner, Ah Chung Tsoi, and Marco Maggini. Graph neural networks for ranking web pages. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pp. 666–672. IEEE, 2005.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pp. 1312–1320. PMLR, 2015.

Olivier Sigaud, Hugo Caselles-Dupré, Cédric Colas, Ahmed Akakzia, Pierre-Yves Oudeyer, and Mohamed Chetouani. Towards teachable autonomous agents. *arXiv preprint arXiv:2105.11977*, 2021.

Karen Stagnitti and Carolyn Unsworth. The importance of pretend play in child development: An occupational therapy perspective. *British Journal of Occupational Therapy*, 63(3):121–127, 2000.

Luc Steels. The autotelic principle. In *Embodied artificial intelligence*, pp. 231–242. Springer, 2004.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to Summarize with Human Feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI magazine*, 32(4):64–76, 2011.

Andrea L Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7):716–737, 2008.

Michael Tomasello. *Constructing a language*. Harvard university press, 2009.

Sam Toyer, Felipe Trevizan, Sylvie Thiébaux, and Lexing Xie. Action schema networks: Generalised policies with deep learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Aaron Van Den Oord, Oriol Vinyals, and others. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*, pp. 6306–6315, 2017.

Anna-Lisa Vollmer, Britta Wrede, Katharina J. Rohlfing, and Pierre-Yves Oudeyer. Pragmatic Frames for Teaching and Learning in Human–Robot Interaction: Review and Challenges. *Frontiers in neurorobotics*, 10:1–10, 2016. Publisher: Frontiers.

L. S. Vygotsky. Tool and Symbol in Child Development. In *Mind in Society*, chapter Tool and Symbol in Child Development, pp. 19–30. Harvard University Press, 1978. ISBN 0674576292. doi: 10.2307/j.ctvjf9vz4.6.

Lev S. Vygotsky. Tool and symbol in child development. *The Vygotsky reader*, 1994.

Liping Wang, Marie Amalric, Wen Fang, Xinjian Jiang, Christophe Pallier, Santiago Figueira, Mariano Sigman, and Stanislas Dehaene. Representation of spatial sequences using nested rules in human prefrontal cortex. *NeuroImage*, 186:245–255, 2019.

David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. *arXiv preprint arXiv:1811.11359*, 2018.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pp. 3391–3401, 2017.

A APPENDIX

A.1 PSEUDO CODE

Algorithms 1 and 2 present the high-level pseudo-code for the individual and social learning episodes.

Algorithm 1 Individual Learning

Require Env E,
Initialize policy Π, semantic graph $\mathcal{G}_s$, path estimator PE, buffer B.
loop
  $g \leftarrow \mathcal{G}_s$.sample_node()
  $path \leftarrow PE$.sample_path($g, \mathcal{G}_s$)
  loop $g_i \in path$
    $trajectory \leftarrow E$.rollout(g_i)
    $\mathcal{G}_s$.update($trajectory$)
    PE.update($trajectory$)
    B.update($trajectory$)
  Π.update(B)
return $\Pi, PE, \mathcal{G}_s$

Algorithm 2 Social Learning

Require Env E, social partner SP
Initialize policy Π, semantic graph $\mathcal{G}_s$, path estimator PE, buffer B.
loop
  $g \leftarrow SP$.propose_goal($\mathcal{G}_s$)
  $path \leftarrow PE$.sample_path($g, \mathcal{G}_s$)
  loop $g_i \in path$
    $trajectory \leftarrow E$.rollout(g_i)
    $\mathcal{G}_s$.update($trajectory$)
    PE.update($trajectory$)
    B.update($trajectory$)
  if g is a *stepping stone* and is reached **then**
    $g_b \leftarrow SP$.propose_unknown($\mathcal{G}_s$, g)
    $path \leftarrow PE$.sample_path($g_b, \mathcal{G}_s$)
    loop $g_i \in path$
      $trajectory \leftarrow E$.rollout(g_i)
      $\mathcal{G}_s$.update($trajectory$)
      PE.update($trajectory$)
      B.update($trajectory$)
  Π.update(B)
return $\Pi, PE, \mathcal{G}_s$

A.2 OBJECT-CENTERED ARCHITECTURE

In the proposed *Fetch Manipulate* environment with 5 blocks, the GANGSTR agent perceives:

1. the low-level geometric states. Since the 5 objects share the same attributes dimensions (positions, velocities, orientations), the behavior with respect to an object should be independent from the object's attributes.
2. the high-level semantic configurations. Since the relations between all the pairs of objects share the same predicates (*close*, *above*), the behavior with respect to a binary semantic relation should be independent from the considered pair.

Thus, it is natural to encode both object-centered and relational inductive biases in our architecture. To do so, we model both the agents policies and critics as MPGNNs. We consider a graph of 5 nodes, each representing a single object. All the nodes are interconnected, yielding a compact graph of 20 directed edges. Furthermore, we consider the agent's body attributes as global features of the policy networks and both the agent's body attributes and the actions as global features of the critic networks. A single forward pass through this graph consists in three steps:

1. **Message computation** is performed for each edge. The features of the considered edge are concatenated with the features of the edge's source and target nodes before being fed to a first shared neural network NN_{edge}.
2. **Node-wise aggregation** is performed for each node. The features of the considered node are concatenated with an aggregation of the updated features of all the incoming edges. The resulting vectors are then concatenated with the global features of the graph before being fed to a second shared neural network NN_{node}.
3. **Graph-wise aggregation** is performed once for all the graph. The updated features of all the nodes of the graph are aggregated and fed to a third neural network NN_{readout}.

The aggregating function needs to be permutation invariant. We use max pooling for the *node-wise aggregation* and summation for the *graph-wised aggregation*. The final output of NN_{readout} is either the action (in the case of the actor) or the Q-value (in the case of the critic).

A.3 HIGH-LEVEL GRAPH-BASED POLICY CONSTRUCTION

Attention Bias - When exploring its environment, the agent is likely to encounter transitions that are complex to reproduce (for example while pushing one object using another). Taking these transitions into account while constructing the graph would affect the quality of the agent's high-level representation, leading to the generation of long and risky intermediate path based on inaccurate estimates of the SR. To circumvent this issue, we assume that the agent only considers *small steps* when constructing an edge between two nodes. Concretely, at the end of an episode, the agent monitors the predicates that changed when moving from the initial to the final configuration. The edge is constructed only if these predicates correspond to exactly one object.

Edge Permutation Bias - To estimate the success rates of each transition in the graph, we share outcomes across transitions that are equivalent by permutations of the objects' identities (see Section 3.2). Figure 6 illustrates this permutation bias.

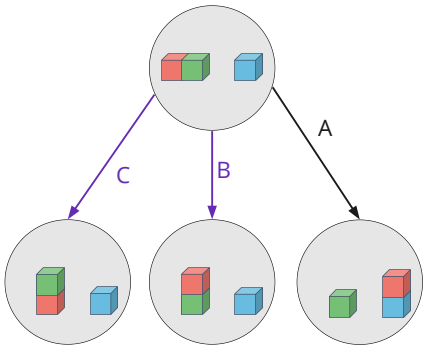


Figure 6: Estimates of SR in edges C and B can be shared.

A.4 REPRESENTING THE SOCIAL PARTNER'S KNOWLEDGE

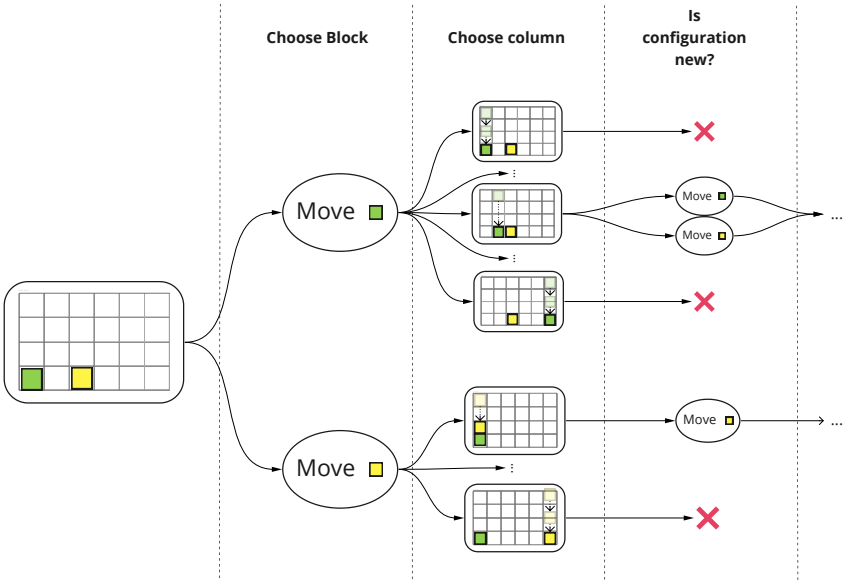


Figure 7: Simplified example with 2 blocks in a 2D grid of the oracle tree construction procedure

Similar to the agent's, we choose to represent the SP's domain knowledge as a directed semantic graph. This facilitates determining stepping stones in the agent's capabilities. However, it is impractical to manually enlist all the imaginable configurations and decide whether a pair should be linked or not in a graph. Semantic configurations can be infeasible for two reasons: 1) they are semantically impossible to obtain—e.g. two objects cannot both be above each other 2) they are physically impossible to achieve—e.g. in the case of inverted pyramids.

To avoid enlisting all semantic and physical constraints, we design a 3D grid within which each cell can contain one object. Initially, all the blocks are initialized in different columns of the grid so that they are all far from each other (*root* node). From this state, we can select one object and move it to another column. If that column already contains another object, than the first one will be stacked above the second. By doing one action at each step, we can extract the current semantic configuration and link it to the previous one in the oracle graph. Iteratively repeating this process yields a tree starting from the *root* node. See Figure 7 for a 2D illustration of the described process for two blocks. Note that this engineered process only serves to evaluate the capacities of the agent and is not used by the agent itself in any way.

A.5 IMPLEMENTATION DETAILS

This part includes details necessary to reproduce results. An anonymous under construction version of our code is available at https://anonymous.4open.science/r/gangstr-2C4F.

GNN-*based networks.* Our object-centered architecture uses two shared networks, NN_{edge} and NN_{node}, respectively for the message computation and node-wise aggregation. Both are 1-hidden-layer networks of hidden size 256. Taking the output dimension to be equal to $3\times$ the input dimension for the shared networks showed better results. All networks use ReLU activations and the Xavier initialization. We use Adam optimizers, with learning rates 10^{-3}. The list of hyperparameters is provided in Table 2.

Parallel implementation of SAC-HER. Our experiments rely on a *Message Passing Interface* (Dalcin et al., 2011) to exploit multiple processors. Each of the 24 parallel workers maintains its own replay buffer of size 10^6 and performs its own updates. Updates are summed over the 24 actors and the updated actor and critic networks are broadcast to all workers. Each worker alternates between 10 episodes of data collection and 30 updates with batch size 256. To form an epoch, this cycle is repeated 50 times and followed by the offline evaluation of the agent.

Table 2: Hyperparameters used in GANGSTR.

Hyperparam.	Description	Values.
nb_mpis	Number of workers	24
nb_cycles	Number of repeated cycles per epoch	50
$nb_rollouts_per_mpi$	Number of rollouts per worker	10
$rollouts_length$	Number of episode steps per rollout	40
$nb_updates$	Number of updates per cycle	30
$replay_strategy$	HER replay strategy	$final$
k_replay	Ratio of HER data to data from normal experience	4
$batch_size$	Size of the batch during updates	256
γ	Discount factor to model uncertainty about future decisions	0.99
τ	Polyak coefficient for target critics smoothing	0.95
lr_actor	Actor learning rate	10^{-3}
lr_critic	Critic learning rate	10^{-3}
α	Entropy coefficient used in SAC	0.2
α_{EMA}	EMA coefficient for SR edge estimation	0.01
$edge_prior$	Default value for edges' SR	0.5
$shortest_p aths$	Number of shortest paths to sample from	5
$shortest_safest_ratio$	Ratio of alternation between shortest and safest paths	0.5

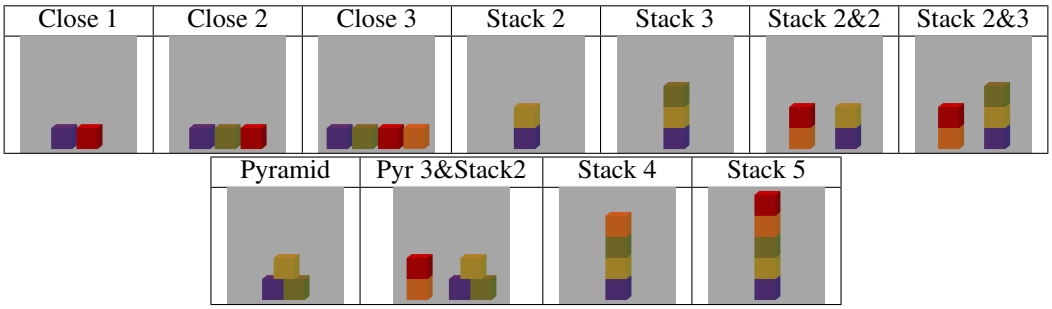


Table 3: The different semantic classes used in evaluation. The class *Close* i regroups all semantic configurations where i pairs of blocks are close.

A.6 Additional Results

Evaluation Classes. Table 3 illustrates the 11 evaluation classes presented in the main paper. For the sake of simplicity here, we only represent the blocks that are concerned by the underlying predicates. All the predicates associated with the other blocks are values to 0. We use a hard-coded function to generate a random configuration given the identifier of the considered class. We also use a dictionary where keys are configurations and values are identifiers of the classes to keep counts of either the SP proposed goals or the agent's encountered and achieved goals.

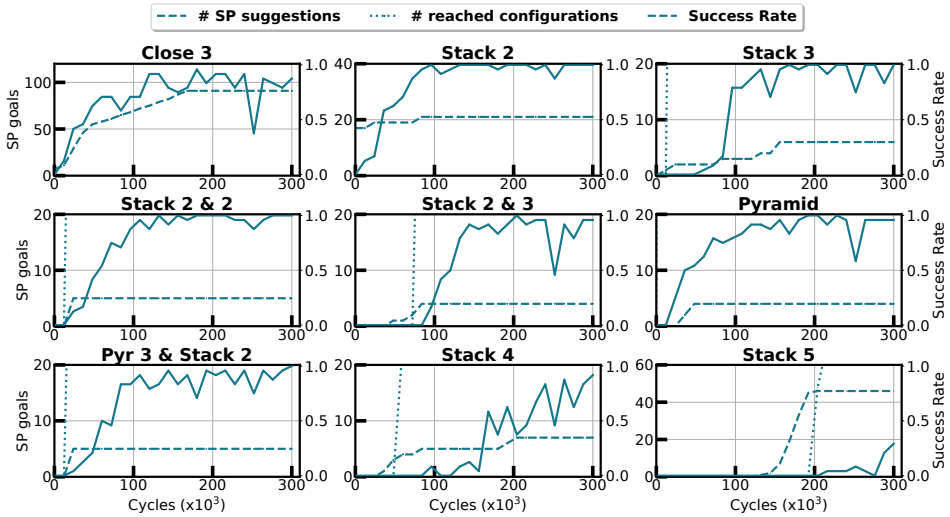


Figure 8: Learning Trajectory of GANGSTR with social intervention ratio of 20%. We report the number of goals proposed by SP (dashed), the number of times the goals were encountered (dotted) and the success rate (plain).

Additional Learning Trajectories. In Figure 8, we present additional learning trajectories of a particular GANGSTR agent SP-20%. We can see that for some classes, social intervention (dashed lines) precedes goal reaching (dotted lines). These classes are considered hard for the agent to discover on its own (for example Pyramid 3 & Stack 2, Stack 4, Stack 5). Other classes do not require social intervention to be discovered since the agent own random exploration is sufficient (for example Close 3, Stack 2, Pyramid). Once configurations are reached, the agent is able to sample them when updating its policy and learn how to achieve them when targeted (plain lines).

Addition Class-wise Performance Figure 9 depicts the success rates per class of semantic goal configurations for 3 GANGSTR agents (columns) with different ratios of social intervention (rows). These additional results follow the claims we made in the main paper. In fact, when no social intervention is provided (SP-0%), the agents only masters the goals that it is able to discovered through random explorations. This excludes goals that are hard to find, namely high stacks and

compositions of classes. Increasing the ratio of social intervention (i.e looking from up to down in Figure 9 allows to effectively grow the agents repertoires of skills.

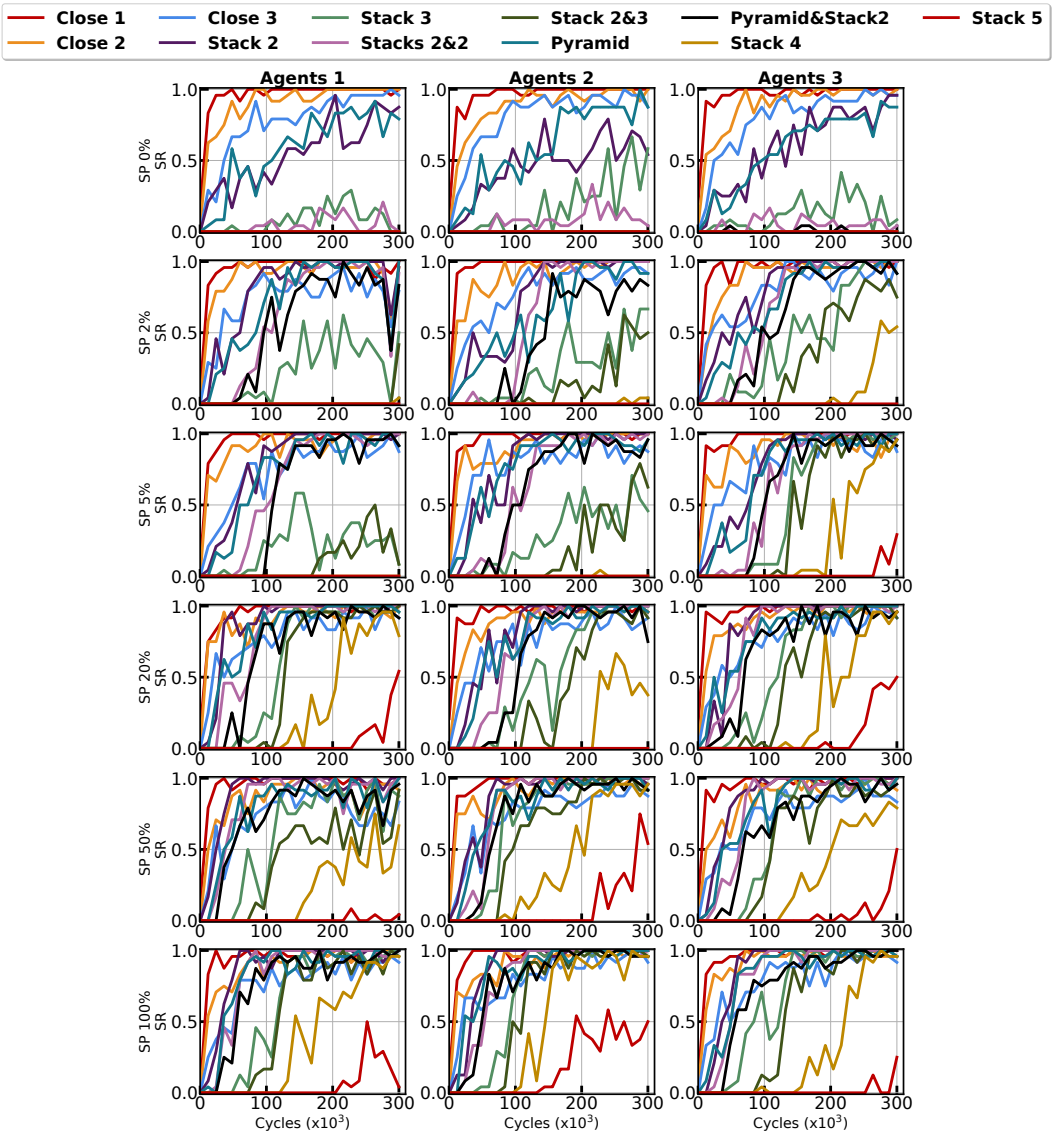


Figure 9: Success Rates per class of semantic goals configurations for (left) SP intervention at 0%, (middle) 2% and (right) 20% .