Using Natural Language to Guide Meta-Learning Agents towards Human-like Inductive Biases

Sreejan Kumar[1], Ishita Dasgupta[2], Michael Hu[1,3], Raja Marjieh[1], Robert D. Hawkins[1], Nathaniel D. Daw[1], Jonathan D. Cohen[1], Karthik Narasimhan[1], and Thomas L. Griffiths[1]

[1]Princeton University
[2]Deepmind
[3]Yobi

Abstract

Inductive biases are a key component of human intelligence, allowing people to acquire, represent, and use abstract knowledge. Although meta-learning has emerged as an approach to endowing neural networks with inductive biases, agents trained via meta-learning can use very different strategies compared to humans. We show that co-training these agents on predicting human-generated natural language task descriptions guides them toward human-like inductive biases that more appropriately capture the structure of the task distribution as humans see it. We further show that the level of abstraction at which humans write these descriptions influences the size of the effect. This work provides a foundation for investigating how to collect task descriptions at the appropriate level of abstraction to leverage for approximating human-like learning of structured representations in neural networks.

1 Introduction

Human learners are guided by strong inductive biases towards abstract knowledge (Tenenbaum et al., 2011; Griffiths et al., 2010); these biases present one of the most salient differences between humans and neural network-based learners (Lake et al., 2017). One emerging approach to bestowing human-like inductive biases on neural networks is *meta-learning* (Griffiths et al., 2019; Hospedales et al., 2020). In meta-learning paradigms, an agent is trained not just on a single task but on a *distribution* of tasks, with the aim of acquiring the underlying abstractions that these tasks have in common. However, since neural networks are not easily interpretable, it can be difficult to tell if the resulting neural networks actually acquired this abstract knowledge, or whether they have simply learned statistical artifacts correlated with abstract rules. Recently, Kumar et al. (2021) found that neural agents are biased towards learning the latter. Specifically, through the use of a task distribution generated from an abstract compositional grammar and a corresponding control task distribution with closely matched statistics, they found agents do better in the control task distribution whereas humans do better in the abstract task distribution, demonstrating a difference in inductive biases between humans and agents.

What explains such differences? One possibility is that human biases toward abstract structure are related to our language abilities (Spelke, 2003; Lupyan and Bergen, 2016). Indeed, recent work in machine learning has revealed how neural network representations can be shaped and structured through natural language supervision (Andreas et al., 2018; Luketina et al., 2019; Wong et al., 2021; Narasimhan et al., 2018; Mu et al., 2020).

In this work, we show that guiding meta-reinforcement learning agents with natural language descriptions not only increases performance on abstract task distributions, but also results in more human-like behavior: it decreases performance on control task distributions where humans perform poorly. Further, while much of language-guided RL work focuses on synthetic descriptions, we investigate different kinds of human-generated descriptions. We collect human descriptions at different levels of abstraction and find that guidance with more abstract descriptions lead to more human-like inductive biases in agents.

Our approach is to first extend and replicate the results of Kumar et al. (2021). Specifically, instead of developing an abstract task distribution using handwritten rules as in Kumar et al. (2021), we directly project human priors into a task distribution (see Fig 1B). We then test a meta-RL agent's ability to acquire this task distribution's emergent abstract priors by building a control task distribution using the same approach as Kumar et al. (2021) (see Fig 1C). We replicated the double dissociation effect seen in Kumar et al. (2021) (see Fig 1E) and then further show that we can guide the agent towards

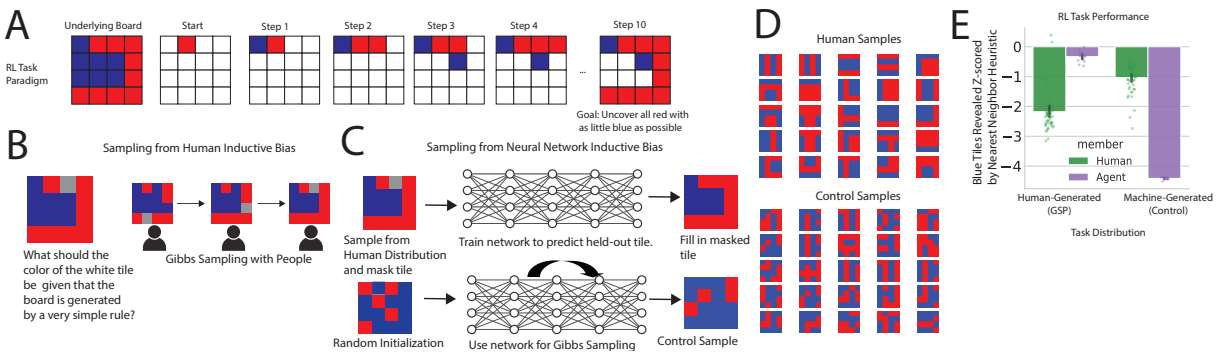

Figure 1: **Meta-RL task paradigm.** (A) In the tile-revealing task, an agent sequentially reveals tiles to uncover a picture on a 2D grid. We elicit (B) human priors and (C) control priors over the task distribution using Gibbs sampling (Geman and Geman, 1984). (D) Samples from human and control distributions. (E) Performance of (independent) humans and machine-learning agents on the tile-revealing task with human and control boards. Performance is based on number of blue tiles revealed (lower is better; see Appendix for details). Error bars are 95% confidence intervals.

learning a human-like inductive bias through the use of natural language co-supervision.

2 Methods

Tile-revealing task. We employ the tile-revealing task paradigm developed in Kumar et al. (2021) (see Fig. 1A). The observation is a 4×4 grid of tiles that are initially white except for one red tile. Actions – clicking on white tiles – reveal those tiles to be either red or blue. The episode ends when the agent reveals all the red tiles. There is a reward for each red tile revealed, and a penalty for each blue tile revealed. The goal therefore is to reveal all the red tiles while revealing as few blue tiles as possible. One "board" with a fixed configuration of red tiles defines a single task. A distribution over tasks is defined by specifying a distribution over different 4×4 grids of red and blue tiles (boards).

Eliciting human priors with Gibbs sampling. In order to elicit human inductive biases, we use a technique called Gibbs Sampling with People (GSP; Harrison et al., 2020, see Fig. 1B). We initialize a random 4×4 grid with red and blue tiles, mask out a tile, and ask a human participant to predict the color of the masked-out tile. We then change that tile to match the human's prediction and present the updated grid to another participant, masking out a different tile. This sequence of decisions implements a Markov chain; the stationary distribution of this chain is the implicit prior distribution people hold over 4×4 grid colorings (Harrison et al., 2020), There are several recognizable abstract concepts that emerged within the resulting grids, such as lines, squares, and continuous shapes (see Fig. 1D).

Constructing a control distribution. We created a control distribution, following Kumar et al. 2021, that matches the statistics of the GSP boards but uses a different underlying generative process (i.e. not produced by human decisions, see Fig. 1C). Specifically, we train a fully connected neural network to encode the conditional distributions of the GSP boards: we mask out a random tile in each board, and train the network to predict its value given the other tiles (similar to masked language models; (Devlin et al., 2018)). We then sampled boards from the network's learned conditionals with Gibbs sampling. This is the same process we used to generate the GSP boards, but using the trained neural network to generate the conditional distributions instead of human samples. We are therefore sampling from the distribution the network places over 4×4 grids (combining its own inductive bias and the data from the GSP boards). We refer to this distribution as a "control" distribution, which is comprised of tasks that are generated with different underlying generative processes but share certain statistical properties.

Collecting natural-language descriptions. We hypothesized that linguistic descriptions of the GSP boards may help guide the agent's inductive bias towards more human-like abstractions. To test this hypothesis, we collected natural-language descriptions of 500 GSP boards from a naive group of participants. There were three types of descriptions collected: two that were human-generated

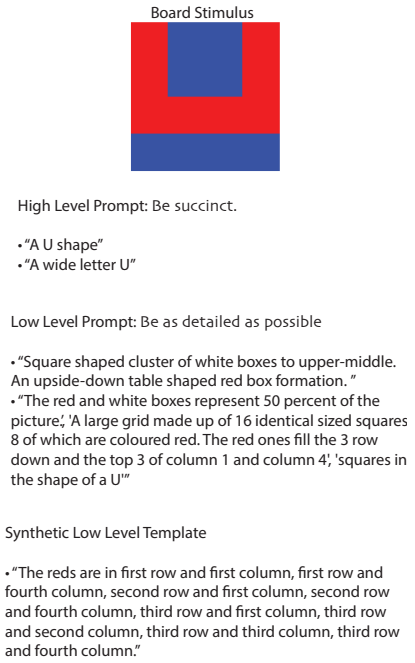

Figure 2: **Types of Text Descriptions Obtained for GSP Boards** We obtained three types of descriptions for 500 of the GSP boards: high-level, low-level, and synthetic low-level. The first two were collected directly from humans using different types of prompts that emphasized succinctness and detail respectively. The third was generated from a handmade template that verbalizes the location of red tiles. When showing participants the boards, we converted blue tiles to white tiles in order to have them focus their description on the red tiles' locations.

under different prompts and one that was synthetically generated using a template (see Fig. 2 and Appendix for exact wording of prompts). **High-level** descriptions were collected from humans who were given a prompt that encouraged succinctness in descriptions. **Low-level** descriptions were collected from humans who were given a prompt that encouraged being verbose and detailed. **Synthetic low-level** descriptions are not human generated and were obtained by using a hand-written template that verbalizes the location of all the red tiles.

Grounding agents with descriptions. We train a commonly used RNN-based meta-reinforcement learning agent (Wang et al., 2018; Duan et al., 2016) using Proximal Policy Optimization (PPO; Schulman et al. (2017)). See Fig. 3 and Appendix for more details.

In order to guide the agent to learn a human-like inductive bias, we introduce a *language grounding term* to the loss function: loss $= L^{PPO}(\theta) + c_{lang} L^{lang}(\hat{\psi}_\theta, \psi)$. Here $L^{PPO}(\theta)$ is the original

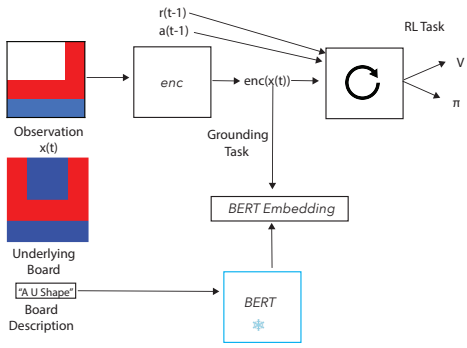

Figure 3: **Grounding architecture.** A CNN encoder observes the board state and passes it onto an LSTM policy network conditioned on the previous timestep's action and reward. We have the agent concurrently predict the BERT embedding of the corresponding language description using the encoder.

PPO loss function, c_{lang} is a hyperparameter coefficient that weights the language loss L^{lang}, ψ is the **language attribute**, and $\hat{\psi}_\theta$ is the agent's prediction of the language attribute. Optimizing for an auxiliary language task jointly with the original task has previously been found to shape the latent representations used in the original task (Mu et al., 2020; Lampinen et al., 2021).

In our study, ψ is the BERT embedding of the uncovered board's corresponding language description, obtained using the SentenceTransformer package (`https://www.sbert.net/`, based on Reimers and Gurevych (2019)). $\hat{\psi}_\theta$ is generated using a small network (two layer MLP) on top of the board encoding shared with the RL task (see Figure 3). L^{lang} is the MSE between the predicted and actual BERT embedding of the language description.

3 Results

We trained all agents on the GSP boards (see Appendix for details) and evaluated them on held-out GSP and control boards. We then compared this held-out test performance against human performance on these test boards (see Fig. 1E). Performance is based on the number of blue tiles revealed in the episode, z-scored by the performance of a nearest neighbor heuristic, so *lower is better* (see Appendix). Results are shown in Fig. 4. First, examining the performance of human participants and non-linguistic agents, we observe the same double dissociation results found by Kumar et al. (2021): humans perform better in the abstract task distribution and agents perform better in the control task

distribution. Next, we examine agents that were co-trained with a language loss L^{lang} on three different kinds of language data: low-level, high-level, and synthetic low-level (see Fig. 2 for examples of the different kinds of language data). As a final baseline, we also considered an autoencoder agent trained to predict the underlying board state (i.e., which tiles are red or blue) rather than the board's corresponding language attribute.

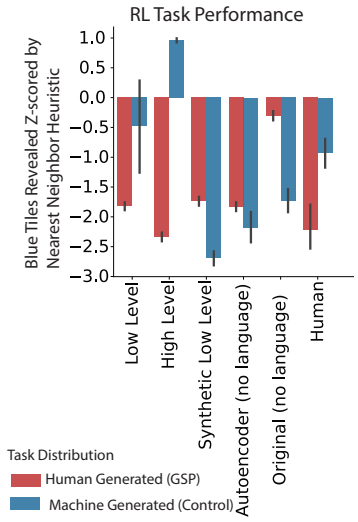

Figure 4: **Language-Grounded Agent Experiment Results** Performance of various agents on held-out tasks for each task distribution using agents co-trained with the language objective. As in Fig. 1, performance is evaluated by z-scoring the number of blue tiles revealed (*lower is better*) relative to a nearest neighbor heuristic. Error bars are 95% confidence intervals.

We set out to test whether grounding in human-generated natural language descriptions will result in our meta-RL agent producing more human-like performance. We know humans perform better on GSP boards than control boards (Fig. 1), while generic agents do the opposite. An agent performing better on the GSP boards and worse on the control boards therefore indicates more human-like behavior.

We see that **grounding on human-generated descriptions leads to a human-like inductive bias** (low and high bars of Fig. 4). Each of them perform better at the GSP boards than the control boards, just like humans do. In contrast, although the autoencoder agent (which does not use language) is substantially better on the GSP boards than the original agent, the autoencoder grounding loss also boosts its performance on the control boards, which indicates that its boost in performance relative to the original agent *is not from acquiring a human-like inductive bias* (but could be an interesting inductive bias in and of itself). We also find that **grounding in synthetic text does not seem to lead to acquiring human-like inductive biases either**, since the agent using synthetic low-level text closely matches the autoencoder and does better in the control distribution than the GSP distribution.

We also find that the **level of abstraction at which humans write their description influences the agent's acquired inductive bias**, as indicated by the differences in performance among low and high-level grounded agents. In all descriptions (even in low-level ones), humans write about abstract concepts (e.g. "squares," "boxes," "clusters," etc). These abstract concepts are most present in high-level descriptions as they let humans to be as succinct in their descriptions as possible. The agent co-trained to predict these high-level description may therefore distill these abstract concepts very strongly into the representations it learns. This could explain why the high-level agent has a "super-human" inductive bias toward abstraction, where it does best on the GSP boards (relative to all other agents and even humans) and the worst on the control boards (worse than humans, and even worse than the nearest neighbour heuristic).

4 Conclusion

In this work, we show how meta-reinforcement learning agents can be guided to have human-like inductive biases towards abstraction. To set this up, we used the task paradigm of Kumar et al. (2021) with a task distribution that directly embeds human priors through people using Gibbs Sampling with People (Fig. 1B). We used the procedure introduced in Kumar et al. (2021) to build a control task distribution (see Fig. 1D) to help benchmark for acquiring human-like inductive biases. Our results show that having the agent predict human-generated language descriptions while doing the task during training can guide the agent towards learning human-like inductive biases (Figure 4). We also manipulated the level of abstraction at which humans write their descriptions (Fig. 2) and showed that this can affect how well the learner acquires an inductive bias more consistent with human behavior. This lays the groundwork for future research in learning human-like abstract representations to move toward closing the gap between human and machine intelligence.

References

Jacob Andreas, Dan Klein, and Sergey Levine. 2018. Learning with latent language. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2166–2179.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.

Michael Chmielewski and Sarah C Kucker. 2020. An mturk crisis? shifts in data quality and the impact on study results. *Social Psychological and Personality Science*, 11(4):464–473.

JH Clark. 1924. The ishihara test for color blindness. *American Journal of Physiological Optics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. 2016. Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Stuart Geman and Donald Geman. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741.

Thomas L Griffiths, Frederick Callaway, Michael B Chang, Erin Grant, Paul M Krueger, and Falk Lieder. 2019. Doing more with less: meta-reasoning and meta-learning in humans and machines. *Current Opinion in Behavioral Sciences*, 29:24–30.

Thomas L Griffiths, Nick Chater, Charles Kemp, Amy Perfors, and Joshua B Tenenbaum. 2010. Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in cognitive sciences*, 14(8):357–364.

Thomas L Griffiths, Dylan Daniels, Joseph L Austerweil, and Joshua B Tenenbaum. 2018. Subjective randomness as statistical inference. *Cognitive psychology*, 103:85–109.

Peter Harrison, Raja Marjieh, Federico Adolfi, Pol van Rijn, Manuel Anglada-Tort, Ofer Tchernichovski, Pauline Larrouy-Maestri, and Nori Jacoby. 2020. Gibbs sampling with people. *Advances in Neural Information Processing Systems*, 33:10659–10671.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*.

Sreejan Kumar, Ishita Dasgupta, Jonathan Cohen, Nathaniel Daw, and Thomas Griffiths. 2021. Meta-learning of structured task distributions in humans and machines. In *International Conference on Learning Representations*.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

Andrew K Lampinen, Nicholas A Roy, Ishita Dasgupta, Stephanie CY Chan, Allison C Tam, James L McClelland, Chen Yan, Adam Santoro, Neil C Rabinowitz, Jane X Wang, et al. 2021. Tell me why!–explanations support learning of relational and causal structure. *arXiv preprint arXiv:2112.03753*.

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

Gary Lupyan and Benjamin Bergen. 2016. How language programs the mind. *Topics in cognitive science*, 8(2):408–424.

Jesse Mu, Percy Liang, and Noah Goodman. 2020. Shaping visual representations with language for few-shot classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4823–4830.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2018. Grounding language for transfer in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 63:849–874.

Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. 2019. Stable baselines3.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Elizabeth S Spelke. 2003. What makes us smart? core knowledge. *Language in mind: Advances in the study of language and thought*, page 277.

Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. 2011. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285.

Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. 2018. Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience*, 21(6):860–868.

Catherine Wong, Kevin M Ellis, Joshua Tenenbaum, and Jacob Andreas. 2021. Leveraging language to learn program abstractions and search heuristics. In *International Conference on Machine Learning*, pages 11193–11204. PMLR.

A Building an Abstract Task Distribution using Gibbs Sampling with People

To generate a task distribution of boards directly from humans we used Gibbs Sampling with People (GSP; Harrison et al. 2020, and a similar task for binary sequences in Griffiths et al. 2018). GSP samples internal prior distributions by putting humans "in the loop" of a Gibbs sampler. In our case, the stimulus space consisted of the space of 4×4 boards, and each of the 16 stimulus dimensions corresponded to the binary color of each tile, namely, red or blue. One of these dimensions was masked out (i.e. "greyed") for the prediction task. Each GSP trial consisted of a prediction task of predicting what color the single masked square is in the grid conditional on the colors of all other squares on the grid. Once a decision is made, the resulting stimulus is passed on to a new participant who repeats the task with another masked square and so on. A sample is generated once a full sweep through the all sixteen squares is completed, similar to the standard procedure of Gibbs sampling. In each trial, participants were presented with a board with one of its tiles covered (indicated by a white tile) as well as the following prompt "what should be the underlying color of the covered white tile such that the board is described by a very simple rule?" (Fig. 4A). They then delivered their answer by clicking on a button that corresponded to their color of choice. Overall, we ran 100 GSP chains in parallel for 15 sweeps each (24000 total possible unique boards), and chains were initialized with randomly sampled boards. The order in which tiles were masked out within each sweep was also randomized across chains to avoid potential biases. When sampling from this distribution, the probability of each board is based on how frequent it occurred during the GSP sampling process. We used the 500 most probable boards to collect language descriptions.

Participants were recruited on Amazon Mechanical Turk (AMT) and a total of 272 participants completed the study. To ensure that participants did not suffer from any color perception deficiencies, we ran the Ishihara color blindness test (Clark, 1924) as a pre-screening task. This also helped in screening out automated scripts ("bots") that masquerade as participants (Chmielewski and Kucker, 2020).

B Generating the Control Task Distribution

The same protocol in Kumar et al. 2021 was used. We trained a fully connected neural network (3 layers, 16 units each) to learn the conditional distribution of each tile given all other tiles on the GSP boards. These conditional distributions contain all the relevant statistical information about the boards. The network was given a board generated with an abstract rule that had a random tile masked out and trained to reproduce the entire board including the randomly masked tile. The loss was the binary cross-entropy between each of the predicted and actual masked tiles, summed over all tiles. The network was trained on samples from the GSP boards, and achieved an accuracy of above 99%.

We used these conditional distributions to generate samples from the distribution of boards learned using Gibbs sampling. We started with a grid in which each tile is randomly set to red or blue with probability 0.5. We then masked out one tile at a time and ran the grid through the network to extract the probability of the missing tile being red or blue from the trained conditional model. We then assign the color of this tile by sampling from this binomial probability. We repeated this by masking each tile in the 4×4 grid (in a random order) to complete a single Gibbs sweep, and repeated this whole Gibbs sweep 20 times to generate a single sample. We generate 25 such independent samples from the control distribution as held-out test data for the meta-learning agent and sample from this distribution during training (while holding out the test set).

C Testing Humans on Abstract and Control Tasks

We crowdsourced human performance on our task using Prolific (`http://www.prolific.co`) for a compensation of \$2.25 (averaging \$13.55 per hour). Participants were shown the 4×4 grid on their web browser and used mouse-clicks to reveal tiles. Each participant was randomly assigned

to either the GSP or control boards. Each participant was evaluated on the same test set of grids used to evaluate the models (24 grids from their assigned task distribution in randomized order). Note that a key difference between the human participants and model agents was that the humans did not receive direct training on any of the task distributions. Since participants had to reveal all red tiles to move on to the next grid, they were implicitly incentivized to be efficient (clicking as few blue tiles as possible) in order to finish the task quickly. We found that this was adequate to get good performance. A reward structure similar to that given to agents was displayed as the number of points accrued, but did not translate to monetary reward. There were 50 participants in each condition (GSP and control), so 100 participants in total. This was the same protocol used in Kumar et al. 2021.

D Training Meta-Reinforcement Learning Agents on the Grid Task

Following previous work in meta-reinforcement learning (Wang et al., 2018), we use an LSTM meta-learner that takes the full board as input, passes it through a convolutional layer and feeds that, along with the previous action and reward, to 120 LSTM units. The agent had 16 possible actions corresponding to choosing a tile (on the 4×4 board) to reveal. The reward function was: +1 for revealing red tiles, -1 for blue tiles, +5 for the last red tile, and -2 for choosing an already revealed tile. The agent was trained using Proximal Policy Optimization (PPO; Schulman et al. 2017 using the Stable Baselines package Raffin et al. 2019) for one million episodes. We performed a hyperparameter sweep separately for the agents without the grounding loss (i.e. original agents) and with the grounding loss, since we have to tune the new c_{lang} weight on the grounding loss jointly. We performed a hyperparameter sweep for: batch size, n_steps (number of steps to run in an environment update), gamma, learning rate, learning rate schedule (constant or linear), clip range, number of epochs, the λ for Generalized Advantage Estimate (GAE λ), max grad norm, activation function, value loss coefficient, entropy coefficient, and grounding loss coefficient for agents with the grounding loss. The hyperparameter sweep was done by sampling from the space of hyperparameter using the Tree-Structured Parzen Estimator (Bergstra et al., 2011). We evaluated 200 samples of hyperparameters from the space for all agents. Both grounding and non-grounding agents used the same hyperparameter spaces to sweep over. We initially did a separate hyperparameter sweep for different grounding agents, but we found in initial experiments that they all reached similar hyperparameter values and training reward after the search. Hyperparameters were evaluated by training on 100,000 episodes and looking at the training reward. The environments used during test time (Fig. 4) were completely held-out during this process.

E Performance Evaluation on the Task

Doing well on the task is indicated by the ability to reveal all red tiles on a grid while revealing as little blue tiles as possible. So, we measure performance by counting the number of blue tiles revealed in the episode. Since different boards will have different number of red tiles/have varying levels of difficulty, we controlled for task difficulty/length by measuring the performance relative to a "nearest neighbor" heuristic. The nearest neighbor heuristic randomly selects covered tiles that are adjacent to currently uncovered red tiles (or any covered tile if such a tile does not exist). For each board, we ran this heuristic on the board 1000 times to generate a distribution of performances (i.e., number of blue tiles revealed) and z-scored the human/agent's number of blue tiles revealed according to this distribution. A z-score of below 0 means that the human/agent did better than the mean performance of the nearest neighbor heuristic, and the specific value of the z-score reflects how many standard deviations the human/agent did better than the mean performance of the nearest neighbor heuristic.

F Natural Language Descriptions of the GSP Boards

We took 500 of the highest probability GSP boards and broke them up into sets of 25. We then randomly assigned participants on Prolific (`http://www.prolific.co`) to these sets of 25 boards. Around 10 participants did each set of the boards given the low level prompt and around 10 participants did each of the board given the high level prompt. As a result, each participant wrote descriptions for 25 boards and each board has approximately 20 descriptions, with about half being from the low level prompt and half being from the high level prompt.

The low-level prompt was: "Your goal is to de-

scribe this pattern of red squares in words. Be as detailed as possible. Someone should be able to reproduce the entire board given your description. You may be rewarded based on how detailed your description is." and the high-level prompt was "Be as general as possible in your description. Your description should use as few words as possible and focus on the pattern of red squares as a whole and not individual squares. You may be rewarded based on how concise your description is." We converted the blue tiles to white when showing the boards to the participants so that they would focus their descriptions on the red tiles.

Synthetic low level descriptions were non-human generated and used the template. The template goes through the location of every tile and says "The reds are in: Xth column and Yth row,...," for every red tile location.

G Hyperparameter Values

The following table contains the hyperparameters used.

Agent	No Grounding Loss	Grounding Loss
batch_size	16	256
n_steps	2048	8
gamma	0.9	0.9
learning_rate	0.000516501	0.000376021
lr_schedule	linear	linear
ent_coef	1.3907E-05	1.45674E-06
clip_range	0.3	0.3
n_epochs	10	5
gae_lambda	0.8	0.95
max_grad_norm	2	0.6
vf_coef	0.000914363	0.016291309
activation_fn	relu	tanh
grounding_coef	0	0.494866282

H Training Curves of Agents

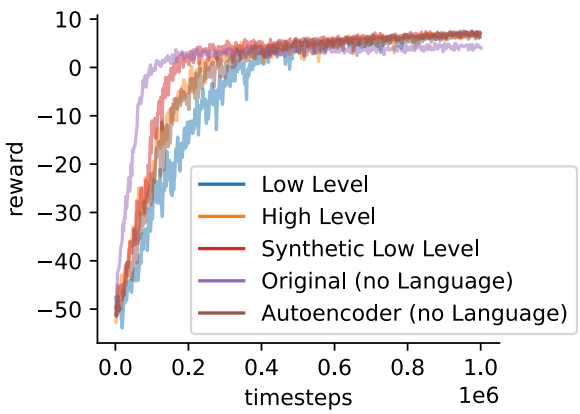

Figure 5: **Training Reward Curves for All Agents**