LEARNING TO GENERALIZE FROM SPARSE AND UNDERSPECIFIED REWARDS

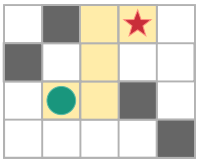
Rishabh Agarwal, Chen Liang, Dale Schuurmans, Mohammad Norouzi
{rishabhagarwal, crazydonkey, schuurmans, mnorouzi}@google.com
Google Brain

ABSTRACT

We consider the problem of learning from sparse and underspecified rewards, where an agent receives a complex input, such as a natural language instruction, and needs to generate a complex response, such as an action sequence, while only receiving binary success-failure feedback. Such success-failure rewards are often underspecified: they do not distinguish between purposeful and accidental success. Generalization from underspecified rewards hinges on discounting spurious trajectories that attain accidental success, while learning from sparse feedback requires effective exploration. We address exploration by using a mode covering direction of KL divergence to collect a diverse set of successful trajectories, followed by a mode seeking KL divergence to train a robust policy. We propose Meta Reward Learning (MeRL) to construct an auxiliary reward function that provides more refined feedback for learning. The parameters of the auxiliary reward function are optimized with respect to the validation performance of a trained policy. The MeRL approach outperforms our alternative reward learning technique based on Bayesian Optimization, and achieves the state-of-the-art on weakly-supervised semantic parsing. It improves previous work by 1.2% and 2.4% on WIKITABLEQUESTIONS and WIKISQL datasets respectively.

1 INTRODUCTION

The remarkable success of reinforcement learning (RL) (Sutton & Barto, 2018) in addressing video games (Mnih et al., 2015; Silver et al., 2017), continuous control (Lillicrap et al., 2015; Hafner et al., 2018), and robotic learning (Kalashnikov et al., 2018; Haarnoja et al., 2018) often hinges on the availability of high-quality and dense reward feedback. However, broadening the applicability of RL algorithms to real-world environments with *sparse* and *underspecified* rewards is an ongoing challenge, requiring a learning agent to generalize from limited feedback. Figure 1 illustrates two examples of contextual environments with sparse and underspecified rewards. The rewards are *sparse*, since only a few trajectories in the combinatorial space of all trajectories leads to a non-zero return. In addition, the rewards are *underspecified*, since the agent may receive a return of 1 for exploiting *spurious* patterns in the environment. We assert that the generalization performance of an



$\boldsymbol{x}$ = "Right Up Up Right"
$R(\boldsymbol{a}) = \mathbb{1}[\texttt{Execute}(\bullet, \boldsymbol{a}) = \star]$

$\boldsymbol{a}_1 = (\rightarrow, \uparrow, \uparrow, \rightarrow)$
$\boldsymbol{a}_2 = (\leftarrow, \rightarrow, \rightarrow, \uparrow, \uparrow, \rightarrow)$
$\boldsymbol{a}_3 = (\uparrow, \rightarrow, \rightarrow, \uparrow)$
$R(\boldsymbol{a}_1) = R(\boldsymbol{a}_2) = x(\boldsymbol{a}_3) = 1$

(a) Instruction Following

Rank	Nation	Gold
1	USA	10
2	GBR	9
3	CHN	8
4	RUS	2
5	GER	2
6	JPN	2
7	FRA	2

$\boldsymbol{x}$ = "Which nation won the most gold medal?" $\boldsymbol{y}$ = "USA"
$R(\boldsymbol{a}) = \mathbb{1}[\texttt{Execute}(\boldsymbol{a}) = \text{"USA"}]$

$\boldsymbol{a}_1 = \texttt{argmax\_row}\,(\text{Gold})\,.\text{Nation}$
$\boldsymbol{a}_2 = \texttt{argmin\_row}\,(\text{Rank})\,.\text{Nation}$
$\boldsymbol{a}_3 = \texttt{first\_row}\,().\text{Nation}$
$R(\boldsymbol{a}_1) = R(\boldsymbol{a}_2) = R(\boldsymbol{a}_3) = 1$

(b) Semantic Parsing

Figure 1: (a) *Instruction following* in a simple maze. A blind agent is presented with a sequence of (Left, Right, Up, Down) instructions. Given the input text, the agent (•) performs a sequence of actions, and only receives a reward of 1 if it reaches the goal ($\star$). (b) *Semantic parsing* from question-answer pairs. An agent is presented with a natural language question $\boldsymbol{x}$ and is asked to generate a SQL-like program $\boldsymbol{a}$. The agent receives a reward of 1 if execution of a program $\boldsymbol{a}$ on the relevant data table leads to the correct answer (*e.g.,* USA). The reward is underspecified because *spurious* outputs (*e.g.,* $\boldsymbol{a}_2, \boldsymbol{a}_3$) can also achieve a reward of 1.

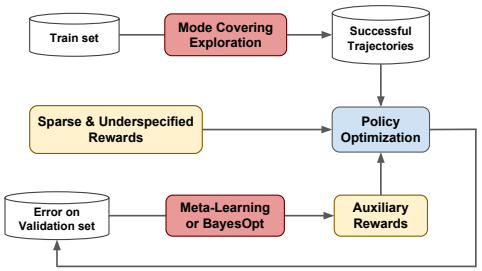


Figure 2: Overview of the proposed approach. We employ (1) mode covering exploration to collect a diverse set of successful trajectories in a memory buffer; (2) Meta-learning or Bayesian optimization to learn an auxiliary reward function to discount spurious trajectories.

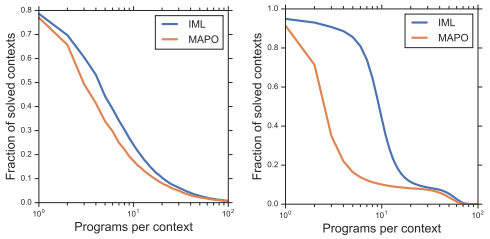


Figure 3: Fraction of total contexts for which at least k programs ($1 \leq k \leq 100$) are discovered during the course of training using the IML and MAPO (*i.e.,* RER) objectives on weakly-supervised semantic parsing datasets (a) WIKITABLEQUESTIONS and (b) WIKISQL.

agent trained in this setting hinges on (1) effective exploration to find successful trajectories, and (2) discounting spurious trajectories to learn a generalizable behavior.

To mitigate challenges of learning from sparse and underspecified rewards, this paper proposes:

- a principled exploration strategy in which a *mode covering* direction of KL divergence is used to learn a high entropy exploration policy to collect a diverse set of successful trajectories. Then, given such trajectories, a *mode seeking* direction of KL divergence is used to learn a robust policy with favorable generalization performance.
- an automatic strategy to discover a rich trajectory-level reward function to help a learning agent discount spurious trajectories and improve generalization. We utilize both gradient-based Meta-Learning (Finn et al., 2017; Maclaurin et al., 2015) and Bayesian Optimization (Snoek et al., 2012) for reward learning, where the parameters of the auxiliary reward function are optimized in an outer loop to maximize generalization performance of the trained policy.

We evaluate our overall approach (outlined in Figure 2) on the instruction following task and on two real world weakly-supervised benchmarks (Pasupat & Liang, 2015; Zhong et al., 2017). In all these experiments, we observe a significant benefit from the mode covering exploration strategy and its combination with Meta Reward Learning (MeRL) resulting in state-of-the-art performance.

2 FORMULATION

Let $\boldsymbol{x}$ denote a complex input, such as a natural language question or instruction, which places an agent in some context. Let $\boldsymbol{a}$ denote a multivariate response, such as an action trajectory that the agent should produce. Let $R(\boldsymbol{a} \mid \boldsymbol{x}, y)$[1] $\in \{0, 1\}$ denote a contextual success-failure feedback that uses some side information y to decide whether $\boldsymbol{a}$ is successful in the context of $\boldsymbol{x}$ and y. For instance, y may be some goal specification, *e.g.,* the answer (denotation) in Figure 1b, or the 2D coordinates of the goal in Figure 1a. To simplify the equations, we drop the conditioning of the return function on $\boldsymbol{x}$ and y and express the return function as $R(\boldsymbol{a})$.

Our aim is to optimize the parameters of a stochastic policy $\pi(\boldsymbol{a} \mid \boldsymbol{x})$ according to a training set in order to maximize the empirical success rate of a policy on novel test contexts. For evaluation, the agent is required to only provide a single action trajectory $\widehat{\boldsymbol{a}}$ for each context $\boldsymbol{x}$, accomplished via approximate inference: $\widehat{\boldsymbol{a}} \approx \underset{\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})}{\operatorname{argmax}} \pi(\boldsymbol{a} \mid \boldsymbol{x})$.

Let $\mathcal{A}(\boldsymbol{x})$ denote the combinatorial set of all plausible action trajectories for a context $\boldsymbol{x}$, and let $\mathcal{A}^+(\boldsymbol{x})$ denote a subset of $\mathcal{A}(\boldsymbol{x})$ comprising successful trajectories, *i.e.,* $\mathcal{A}^+(\boldsymbol{x}) \equiv \{\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x}) \,|\, R(\boldsymbol{a} \,|\, \boldsymbol{x}, y) = 1\}$.

[1]For simplicity of the exposition, we assume that $R(\boldsymbol{a} \mid \boldsymbol{x}, y)$ is deterministic, even though our results are applicable to stochastic rewards as well.

3 MODE COVERING EXPLORATION (MAPOX)

To address the problem of policy learning from binary success-failure feedback, previous work has proposed the following objective functions:

► **IML (Iterative Maximum Likelihood)** estimation (Liang et al., 2017; Abolafia et al., 2018) is an iterative process for optimizing a policy based on

$$O_{\text{IML}} = \sum_{\boldsymbol{x} \in \mathcal{D}} \frac{1}{|\mathcal{A}^+(\boldsymbol{x})|} \sum_{\boldsymbol{a}^+ \in \mathcal{A}^+(\boldsymbol{x})} \log \pi(\boldsymbol{a}^+ \mid \boldsymbol{x}) \,. \tag{1}$$

The key idea is to replace $\mathcal{A}^+(\boldsymbol{x})$ in (1) with a buffer of successful trajectories collected so far, denoted $\mathcal{B}^+(\boldsymbol{x})$. While the policy is being optimized based on (1), one can also perform exploration by drawing *i.i.d.* samples from $\pi(\cdot \mid \boldsymbol{x})$ and adding such samples to $\mathcal{B}^+(\boldsymbol{x})$ if their rewards are positive.

► **RER (Regularized Expected Return)** is the common objective function used in RL,

$$O_{\text{RER}} = \sum_{\boldsymbol{x} \in \mathcal{D}} \tau \mathcal{H}(\pi(\cdot \mid \boldsymbol{x})) + \sum_{\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})} R(\boldsymbol{a}) \pi(\boldsymbol{a} \mid \boldsymbol{x}), \tag{2}$$

where $\tau \geq 0$ and $\mathcal{H}$ denotes Shannon Entropy. Entropy regularization often helps with stability of policy optimization leading to better solutions (Williams & Peng, 1991). The MAPO (Liang et al., 2018) estimator used in our experiments is based on this objective.

The IML and RER objective can be expressed using *mode covering* and mode seeking directions of KL divergence respectively (for more details, see section A.2 in the supplementary material). Our key intuition is that for the purpose of exploration and collecting a diverse set of successful trajectories (regardless of whether they are spurious or not) the mode covering behavior of IML should be advantageous over the mode seeking behaviour of MAPO (*i.e.* RER). We conduct some experiments to evaluate this intuition, and as shown in Figure 3, we find that IML generally discovers many more successful trajectories than MAPO.

Based on these findings, we develop a novel combination of IML and MAPO, which we call MAPOX (MAPO eXploratory). The key difference between MAPO and MAPOX is in the way the initial memory buffer of programs is initialized. In addition to using random search to populate an initial buffer of programs as in (Liang et al., 2018), we also use IML to find a large set of diverse trajectories, which are passed to MAPO to select from. In our experiments, we observe a notable gain from this form of mode covering exploration combining the benefits of IML and MAPO.

4 LEARNING REWARDS WITHOUT DEMONSTRATION

For the general category of problems involving learning with underspecified rewards, our intuition is that fitting a policy on spurious trajectories is disadvantageous for the policy's generalization to unseen contexts. Accordingly, we put forward the following hypothesis: One should be able to learn an auxiliary reward function based on the performance of the policy trained with that reward function on a held-out validation set. We propose two specific approaches to implement this high level idea: (1) based on gradient based Meta-Learning (MAML) (Finn et al., 2017) (Algorithm 1) (2) using BayesOpt (Snoek et al., 2012) as a gradient-free black box optimizer (Algorithm 2). Refer to the supplementary material for a qualitative comparison of these two approaches.

Notation. $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ denote the training and validation datasets respectively. $\mathcal{B}^+_{\text{train}}$ represents the training memory buffer containing successful trajectories (based on underspecified rewards) for contexts in $\mathcal{D}_{\text{train}}$. We employ a feature-based terminal reward function $R_{\boldsymbol{\phi}}$ parameterized by the weight vector $\boldsymbol{\phi}$. For a given context $\boldsymbol{x}$, the auxiliary reward is only non-zero for successful trajectories. Specifically, for a feature vector $\boldsymbol{f}(\boldsymbol{a}, \boldsymbol{x})$ for the context $\boldsymbol{x}$ and trajectory $\boldsymbol{a}$ and the underspecified rewards $R(\boldsymbol{a} \mid \boldsymbol{x}, y)$, *i.e.*, $R_{\boldsymbol{\phi}}(\boldsymbol{a} \mid \boldsymbol{x}, y) = \boldsymbol{\phi}^T \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{x}) R(\boldsymbol{a} \mid \boldsymbol{x}, y)$.

4.1 META REWARD-LEARNING (MERL)

At each iteration of MeRL, we simultaneously update the policy parameters $\boldsymbol{\theta}$ and the auxiliary reward parameters $\boldsymbol{\phi}$. The policy $\pi_{\boldsymbol{\theta}}$ is trained to maximize the training objective O_{train} (3) computed

using the training dataset and the auxiliary rewards $R_{\boldsymbol{\phi}}$ while the auxiliary rewards are optimized to maximize the meta-training objective O_{val} (4) on the validation dataset:

$$O_{\text{train}}(\pi_{\boldsymbol{\theta}}, R_{\boldsymbol{\phi}}) = \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{train}}} \sum_{\boldsymbol{a} \in \mathcal{B}^{+}_{\text{train}}(\boldsymbol{x})} R_{\boldsymbol{\phi}}(\boldsymbol{a}) \pi_{\boldsymbol{\theta}}(\boldsymbol{a} \mid \boldsymbol{x}) + \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{train}}} \tau \mathcal{H}(\pi_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x})), \tag{3}$$

$$O_{\text{val}}(\pi) = \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{val}}} \sum_{\boldsymbol{a} \in \mathcal{B}^{+}_{\text{val}}(\boldsymbol{x})} R(\boldsymbol{a}) \pi(\boldsymbol{a} \mid \boldsymbol{x}). \tag{4}$$

An overview of MeRL is presented in Algorithm 1 in the supplementary material. MeRL requires O_{val} to be a differentiable function of $\boldsymbol{\phi}$. To tackle this issue, we compute O_{val} using only samples from the buffer $\mathcal{B}^{+}_{\text{val}}$ containing successful trajectories for contexts in $\mathcal{D}_{\text{val}}$. Since we don't have access to ground-truth programs, we use beam search in non-interactive environments and greedy decoding in interactive environments to generate successful trajectories using policies trained with the underspecified rewards only.

The validation objective is computed using the policy obtained after one gradient step update on the training objective and therefore, the auxiliary rewards only indirectly affect the validation objective via the updated policy parameters $\boldsymbol{\theta}'$ as shown in equations (5) and (6):

$$\boldsymbol{\theta}'(\boldsymbol{\phi}) = \boldsymbol{\theta} \; - \; \alpha \nabla_{\boldsymbol{\theta}} O_{\text{train}}(\pi_{\boldsymbol{\theta}}, R_{\boldsymbol{\phi}}), \tag{5}$$

$$\nabla_{\boldsymbol{\phi}} O_{\text{val}}(\pi_{\boldsymbol{\theta}'}) = \nabla_{\boldsymbol{\theta}'} O_{\text{val}}(\pi_{\boldsymbol{\theta}'}) \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}'(\boldsymbol{\phi}). \tag{6}$$

4.2 Bayesian Optimization Reward-Learning (BoRL)

At each trial in BoRL, we sample auxiliary reward parameters by maximizing the acquisition function computed using the posterior distribution over the validation objective. After sampling the reward parameters, we optimize the O_{RER} objective on the training dataset for a fixed number of iterations. Once the training is finished, we evaluate the policy on the validation dataset, which is used to update the posterior distribution. BoRL is closely related to the previous work on learning metric-optimized example weights (Zhao et al., 2018) for supervised learning. An overview of BoRL is presented in Algorithm 2 in the supplementary material.

BoRL does not require the validation objective O_{val} to be differentiable with respect to the auxiliary reward parameters, therefore we can directly optimize the evaluation metric we care about. For example, in non-interactive environments, the reward parameters are optimized using the beam search accuracy on the validation set $\mathcal{D}_{\text{val}}$. In this work, we use Batched Gaussian Process Bandits (Desautels et al., 2014) employing a Matérn kernel with automatic relevance determination (Rasmussen, 2004) and the expected improvement acquisition function (Močkus, 1975).

5 Experiments

We evaluate our approach on two weakly-supervised semantic parsing benchmarks, WikiTableQuestions (Pasupat & Liang, 2015) and WikiSQL [2] (Zhong et al., 2017). Additionally, we demonstrate the negative effect of under-specified rewards on the generalization ability of an agent in the instruction following task (refer to section 5.1). For all our experiments, we report the mean accuracy and standard deviation based on 5 runs with identical hyperparameters.

5.1 Instruction Following Task

In this task (see Figure 1a), we compare the following setups for an agent trained with the RER (2) objective using the same neural architecture with a fixed replay buffer:

► **Oracle Reward**: This agent is trained using the replay buffer containing only the gold trajectories.

► **Underspecified Reward**: For each environment, we added a fixed number of additional spurious trajectories (trajectories which reach the goal without following the language instruction) to the oracle replay buffer.

[2]Note that we only make use of weak-supervision in WikiSQL and therefore, our methods are not directly comparable to methods trained using strong supervision in the form of (question, program) pairs on WikiSQL.

Table 1: Performance of the trained agent with access to different type of rewards in the instruction following task.

Reward structure	Dev	Test
Underspecified	73.0 ($\pm$ 3.4)	69.8 ($\pm$ 2.5)
Underspecified + Auxiliary (BoRL)	75.3 ($\pm$ 1.6)	72.3 ($\pm$ 2.2)
Underspecified + Auxiliary (MeRL)	83.0 ($\pm$ 3.6)	74.5 ($\pm$ 2.5)
Oracle Reward	95.7 ($\pm$ 1.3)	92.6 ($\pm$ 1.0)

Table 2: Results on WIKITABLEQUESTIONS.

Method	Dev	Test	Improvement on MAPO
MAPO	42.2 ($\pm$ 0.6)	42.9 ($\pm$ 0.5)	–
MAPOX	42.6 ($\pm$ 0.5)	43.3 ($\pm$ 0.4)	+0.4
BoRL	42.9 ($\pm$ 0.6)	43.8 ($\pm$ 0.2)	+0.9
MeRL	43.2 ($\pm$ 0.5)	**44.1** ($\pm$ 0.2)	**+1.2**

Table 3: Results on WIKISQL using weak supervision.

Method	Dev	Test	Improvement on MAPO
MAPO	71.8 ($\pm$ 0.4)	72.4 ($\pm$ 0.3)	–
MAPOX	74.5 ($\pm$ 0.4)	74.2 ($\pm$ 0.4)	+1.8
BoRL	74.6 ($\pm$ 0.4)	74.2 ($\pm$ 0.2)	+1.8
MeRL	74.9 ($\pm$ 0.1)	**74.8** ($\pm$ 0.2)	**+2.4**
MAPO (Ens. of 5)	-	74.2	–
MeRL (Ens. of 5)	-	**76.9**	**+2.7**

Table 4: Comparison to previous approaches for WIKITABLEQUESTIONS

Method	Ensemble Size	Test
Pasupat & Liang (2015)	-	37.1
Neelakantan et al. (2016)	15	37.7
Haug et al. (2018)	15	38.7
Zhang et al. (2017)	-	43.7
MAPO (Liang et al., 2018)	10	46.3
MeRL	10	**46.9**

► **Underspecified + Auxiliary Reward**: In this case, we use the replay buffer with spurious trajectories similar to the underspecified reward setup, however, we additionally learn an auxiliary reward function using MeRL and BoRL (see Algorithm 1 and 2 respectively). For more details, refer to section A.6 in the supplementary material.

All the agents trained with different types of reward signal achieve an accuracy of approximately 100% on the training set. However, as shown in Table 1, the generalization performance of Oracle rewards $>$ Underspecified + Auxiliary rewards $>$ Underspecified rewards. Using our Meta Reward-Learning (MeRL) approach, we are able to bridge the gap between Underspecified and Oracle rewards, which confirms our hypothesis that the generalization performance of an agent can serve as a reasonable proxy to reward learning.

5.2 WEAKLY-SUPERVISED SEMANTIC PARSING

On WIKISQL and WIKITABLEQUESTIONS benchmarks, the task is to generate an SQL-like program given a natural language question such that when the program is executed on a relevant data table, it produces the correct answer. We only have access to weak supervision in the form of question-answer pairs (see Figure 1b). The performance of an agent trained to solve this task is measured by the number of correctly answered questions on a held-out test set.

We compare the following variants of our technique with the current state-of-the-art in weakly supervised semantic parsing, Memory Augmented Policy Optimization (**MAPO**) (Liang et al., 2018):

► **MAPOX**: Combining the exploration ability of IML with generalization ability of MAPO, MAPOX runs MAPO starting from a memory buffer $\mathcal{B}^{+}_{\text{train}}$ containing all the high reward trajectories generated during the training of IML and MAPO using underspecified rewards only.

► **BoRL** (MAPOX + Bayesian Optimization Reward-Learning): As opposed to MAPOX, BoRL optimizes the MAPO objective only on the highest ranking trajectories present in the memory buffer $\mathcal{B}^{+}_{\text{train}}$ based on a parametric auxiliary reward function learned using BayesOpt (see Algorithm 2).

► **MeRL** (MAPOX + Meta Reward-Learning): Similar to BoRL, MeRL optimizes the MAPO objective with an auxiliary reward function simultaneously learned with the agent's policy using meta-learning (see Algorithm 1).

Results. We present the results on weakly-supervised semantic parsing in Table 2 and Table 3. We observe that MAPOX noticeably improves upon MAPO on both datasets by performing better exploration. In addition, MeRL and BoRL both improve upon MAPOX in WIKITABLEQUESTIONS demonstrating that even when a diverse set of candidates from IML are available, one still benefits from our framework for automatic reward learning. On WIKISQL we do not see any gain from BoRL on top of MAPOX, however, MeRL improves upon MAPOX by 0.6% accuracy. Table 3 also shows that even with ensembling 5 models, MeRL significantly outperforms MAPO. Finally, Table 4 compares our approach with previous works on WIKITABLEQUESTIONS.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *ArXiv:1603.04467*, 2016.

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings of the twenty-first international conference on Machine learning*, 2004.

Daniel A. Abolafia, Mohammad Norouzi, Jonathan Shen, Rui Zhao, and Quoc V. Le. Neural program synthesis with priority queue training. *arXiv:1801.03526*, 2018.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. *Proceedings of the IEEE international conference on computer vision*, 2015.

Yoav Artzi and Luke Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association of Computational Linguistics*, 2013.

Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. Learning to understand goal specifications by modelling reward. *ICLR*, 2019.

J. Berant, D. Deutch, A. Globerson, T. Milo, and T. Wolfson. Explaining relational queries to non-experts. *International Conference on Data Engineering (ICDE)*, 2019.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. *EMNLP*, 2013.

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. Discourse-aware neural rewards for coherent text generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pp. 173–184, 2018.

David L Chen and Raymond J Mooney. Learning to interpret natural language navigation instructions from observations. *AAAI*, 2011.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: First steps towards grounded language learning with a human in the loop. *ICLR*, 2019.

Minseok Cho, Reinald Kim Amplayo, Seung-won Hwang, and Jonghyuck Park. Adversarial tableqa: Attention supervision for question answering on tables. *arXiv:1810.08113*, 2018.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 2017.

Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *The Journal of Machine Learning Research*, 2014.

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

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning*, 2017.

Justin Fu, Anoop Korattikara, Sergey Levine, and Sergio Guadarrama. From language to goals: Inverse reinforcement learning for vision-based instruction following. *ICLR*, 2019.

Adam Gleave and Oliver Habryka. Multi-task maximum entropy inverse reinforcement learning. *arXiv:1805.08882*, 2018.

Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D Sculley. Google vizier: A service for black-box optimization. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. *arXiv:1802.07245*, 2018.

Kelvin Guu, Panupong Pasupat, Evan Liu, and Percy Liang. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. *ACL*, 2017.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

Till Haug, Octavian-Eugen Ganea, and Paulina Grnarova. Neural multi-step reasoning for question answering on semi-structured tables. *ECIR*, 2018.

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. Grounded language learning in a simulated 3d world. *arXiv:1706.06551*, 2017.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in Neural Information Processing Systems*, 2016.

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen tau Yih, and Xiaodong He. Natural language to structured query generation via meta-learning. *CoRR*, abs/1803.02400, 2018.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *NIPS*, 2018.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. Neural semantic parsing with type constraints for semi-structured tables. *EMNLP*, 2017.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv:1811.07871*, 2018.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *ACL*, 2017.

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V Le, and Ni Lao. Memory augmented policy optimization for program synthesis and semantic parsing, 2018.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Shikun Liu, Andrew J Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. *arXiv preprint arXiv:1901.08933*, 2019.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. *ICML*, 2015.

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

Jonas Močkus. On bayesian methods for seeking the extremum. *Optimization Techniques IFIP Technical Conference*, 1975.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. It was the training data pruning too! *arXiv:1803.04579*, 2018.

Ofir Nachum, Mohammad Norouzi, and Dale Schuurmans. Improving policy gradient by exploring under-appreciated rewards. *ICLR*, 2017.

Anusha Nagabandi, Chelsea Finn, and Sergey Levine. Deep online learning via meta-learning: Continual adaptation for model-based rl. *arXiv:1812.07671*, 2018.

Arvind Neelakantan, Quoc V. Le, Martín Abadi, Andrew D McCallum, and Dario Amodei. Learning a natural language interface with neural programmer. *arXiv:1611.08945*, 2016.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv:1803.02999*, 2018.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. Reward augmented maximum likelihood for neural structured prediction. *NIPS*, 2016.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. *ACL*, 2015.

Panupong Pasupat and Percy Liang. Inferring logical forms from denotations. *ACL*, 2016a.

Panupong Pasupat and Percy Liang. Inferring logical forms from denotations. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016b.

Carl Edward Rasmussen. Gaussian processes in machine learning. *Advanced lectures on machine learning*, 2004.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv:1803.09050*, 2018.

Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. Towards diverse text generation with inverse reinforcement learning. *arXiv preprint arXiv:1804.11258*, 2018.

D Silver, T Hubert, J Schrittwieser, I Antonoglou, M Lai, A Guez, M Lanctot, L Sifre, D Kumaran, T Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *NIPS*, 2012.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv:1611.05763*, 2016.

Xin Wang, Wenhu Chen, Yuan-Fang Wang, and William Yang Wang. No metrics are perfect: Adversarial reward learning for visual storytelling. *arXiv preprint arXiv:1804.09160*, 2018.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992.

Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 1991.

Terry Winograd. Understanding natural language. *Cognitive psychology*, 1972.

Lijun Wu, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. Sequence prediction with unlabeled data by reward function learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3098–3104. AAAI Press, 2017.

Lijun Wu, Fei Tian, Yingce Xia, Yang Fan, Tao Qin, Lai Jian-Huang, and Tie-Yan Liu. Learning to teach with dynamic loss functions. In *Advances in Neural Information Processing Systems*, pp. 6467–6478, 2018.

Annie Xie, Avi Singh, Sergey Levine, and Chelsea Finn. Few-shot goal inference for visuomotor learning and planning. *CoRL*, 2018.

Kelvin Xu, Ellis Ratner, Anca Dragan, Sergey Levine, and Chelsea Finn. Few-shot intent inference via meta-inverse reinforcement learning. *arXiv:1805.12573*, 2018a.

Tianbing Xu, Qiang Liu, Liang Zhao, and Jian Peng. Learning to explore via meta-policy gradient. *ICML*, 2018b.

Zhongwen Xu, Hado van Hasselt, and David Silver. Meta-gradient reinforcement learning. *arXiv:1805.09801*, 2018c.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. *ACL*, 2016.

M. Zelle and R. J. Mooney. Learning to parse database queries using inductive logic programming. *Association for the Advancement of Artificial Intelligence (AAAI)*, 1996.

Yuchen Zhang, Panupong Pasupat, and Percy Liang. Macro grammars and holistic triggering for efficient semantic parsing. *ACL*, 2017.

Sen Zhao, Mahdi Milani Fard, and Maya Gupta. Metric-optimized example weights. *arXiv preprint arXiv:1805.10582*, 2018.

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. *Advances in Neural Information Processing Systems*, 2018.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv:1709.00103*, 2017.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. *AAAI*, 2008.

Haosheng Zou, Tongzheng Ren, Dong Yan, Hang Su, and Jun Zhu. Reward shaping via meta-learning. *arXiv preprint arXiv:1901.09330*, 2019.