MURAL: Meta-Learning Uncertainty-Aware Rewards for Outcome-Driven Reinforcement Learning

Kevin Li [*1] Abhishek Gupta [*1] Vitchyr Pong [1] Ashwin Reddy [1] Aurick Zhou [1] Justin Yu [1] Sergey Levine [1]

Abstract

Exploration in reinforcement learning is, in general, a challenging problem. A common technique to make learning easier is providing demonstrations from a human supervisor, but such demonstrations can be expensive and time-consuming to acquire. In this work, we study a more tractable class of reinforcement learning problems defined simply by examples of successful outcome states, which can be much easier to provide while still making the exploration problem more tractable. In this problem setting, the reward function can be obtained automatically by training a classifier to categorize states as successful or not. However, as we will show, this requires the classifier to make uncertainty-aware predictions that are very difficult using standard techniques for training deep networks. To address this, we propose a novel mechanism for obtaining calibrated uncertainty based on an amortized technique for computing the normalized maximum likelihood (NML) distribution, leveraging tools from meta-learning to make this distribution tractable. We show that the resulting algorithm has a number of intriguing connections to both count-based exploration methods and prior algorithms for learning reward functions, while also providing more effective guidance towards the goal. We demonstrate that our algorithm solves a number of challenging navigation and robotic manipulation tasks which prove difficult or impossible for prior methods.

1. Introduction

While reinforcement learning (RL) has been shown to successfully solve problems with careful reward design (Ra-

---
[*]Equal contribution [1]Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, USA. Correspondence to: Kevin Li <kevintli@berkeley.edu>, Abhishek Gupta <abhigupta@berkeley.edu>.
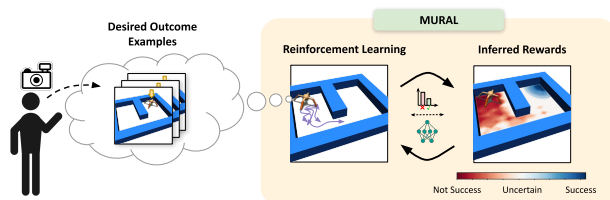
Figure 1. **MURAL:** Our method trains an uncertainty-aware classifier based on user-provided examples of successful outcomes. Appropriate uncertainty in the classifier, obtained via a meta-learning based estimator for the normalized maximum likelihood (NML) distribution, automatically incentivizes exploration and provides reward shaping for RL.

jeswaran et al., 2018), RL in its most general form, with no assumptions on the dynamics or reward function, requires solving a challenging uninformed search problem in which rewards are sparsely observed. Techniques that explicitly provide "reward-shaping" (Ng et al., 1999), or modify the reward function to guide learning, can help take some of the burden off of exploration, but shaped rewards are often difficult to provide without significant domain knowledge. Moreover, in many domains of practical significance, actually specifying rewards in terms of high dimensional observations can be extremely difficult, making it infeasible to directly apply RL to problems with challenging exploration.

Can the RL problem be made more tractable if the agent is provided with examples of successful outcomes instead of an uninformative reward function? Such examples are often easier to provide than, for example, entire demonstrations or a hand-designed reward function. However, they can still provide considerable guidance on how to successfully accomplish a task, potentially alleviating exploration challenges if the agent can successfully recognize similarities between visited states and the provided examples. In this paper, we study such a problem setting, where instead of a hand-designed reward function, the RL algorithm is provided with a set of *successful outcome examples*: states in which the desired task has been accomplished successfully. Prior work (Fu et al., 2018b; Zhu et al., 2020) aims to solve tasks in this setting by estimating the distribution over these states and maximizing the probability of reaching states that are likely under this distribution. While this can work

well in some domains, it has largely been limited to settings without significant exploration challenges. In our work, we focus on the potential for this mode of task specification to enable RL algorithms to solve more challenging tasks without the need for manual reward shaping. Intuitively, the availability of extra information in the form of explicit success examples can provide the algorithm more directed information for exploration, rather than having to simply rely on uninformed task agnostic exploration methods. This allows us to formulate a class of more tractable problems, which we refer to as outcome-driven RL.

However, in order to attain improved exploration, an outcome-driven RL agent must be able to estimate some notion of similarity between the visited states and successful outcomes, so as to utilize this similarity as a kind of automatic reward shaping. Our method addresses this challenge by training a classifier to distinguish successful states, provided by the user, from those generated by the current policy, analogously to generative adversarial networks (Goodfellow et al., 2014) and previously proposed methods for inverse reinforcement learning (Fu et al., 2018a). In general, such a classifier may not provide effective reward shaping for learning the policy, since it does not explicitly quantify uncertainty about success probabilities and can be overly pessimistic in providing reward signal for learning. We discuss how Bayesian classifiers incorporating a particular form of uncertainty quantification based on the normalized maximum likelihood (NML) distribution can incentivize exploration in outcome-driven RL problems. To understand its benefits, we connect our approach to count-based exploration methods, while also showing that it improves significantly over such methods when the classifier exhibits good generalization properties, due to its ability to utilize success examples. Finally, we propose a practical algorithm to train NML-based success classifiers in a computationally efficient way using meta-learning, and show experimentally that our method can more effectively solve a range of challenging navigation and robotic manipulation tasks.

Concretely, this work illustrates the challenges of using standard success classifiers (Fu et al., 2018b) for outcome-driven RL, and proposes a novel technique for training uncertainty aware classifiers with normalized maximum likelihood, which is able to both incentivize the exploration of novel states and provide reward shaping that guides exploration towards successful outcomes. We present a tractable algorithm for learning these uncertainty aware classifiers in practice by leveraging concepts from meta-learning. We analyze our proposed technique for reward inference experimentally across a number of navigation and robotic manipulation domains and show benefits over prior classifier-based RL methods as well as goal-reaching methods.

2. Related Work

While a number of methods have been proposed to improve exploration, it remains a challenging open problem in RL (Misra et al., 2019). Standard exploration methods either add bonuses to the reward function that encourage a policy to visit novel states in a task-agnostic manner (Wiering & Schmidhuber, 1998; Auer et al., 2002; Schaul et al., 2011; Houthooft et al., 2016; Pathak et al., 2017; Tang et al., 2017; Stadie et al., 2015; Bellemare et al., 2016; Burda et al., 2018a; O'Donoghue, 2018), or approximate Thompson sampling from a posterior over value functions (Strens, 2000; Osband et al., 2013; 2016). Whereas these techniques are uninformed about the actual task, we consider a constrained, yet still widely applicable, set of problems where the desired outcome can be specified by success examples, allowing for more efficient task-directed exploration.

Designing well-shaped reward functions can also make exploration easier, but often requires significant domain knowledge (Andrychowicz et al., 2020), access to privileged information (Levine et al., 2016) or a human in the loop providing rewards (Knox & Stone, 2009; Singh et al., 2019b). Prior work has considered specifying rewards by providing example demonstrations and inferring rewards with inverse RL (Abbeel & Ng, 2004; Ziebart et al., 2008; Ho & Ermon, 2016; Fu et al., 2018a). This requires expensive expert demonstrations to be provided to the agent. In contrast, our work has the minimal requirement of successful outcome states, which can be provided more cheaply and intuitively. This subclass of problems is also related to goal-conditioned RL (Kaelbling, 1993; Schaul et al., 2015; Zhu et al., 2017; Andrychowicz et al., 2017; Nair et al., 2018; Veeriah et al., 2018; Rauber et al., 2018; Warde-Farley et al., 2018; Colas et al., 2019; Ghosh et al., 2019; Pong et al., 2020) but is more general, since it allows for the notion of success to be more abstract than reaching a single state.

A core idea behind our method is using a Bayesian classifier to learn a suitable reward function. Bayesian inference with expressive models and high dimensional data can often be intractable, requiring strong assumptions on the form of the posterior (Hoffman et al., 2013; Blundell et al., 2015; Maddox et al., 2019). In this work, we build on the concept of normalized maximum likelihood (Rissanen, 1996; Shtar'kov, 1987), or NML, to learn Bayesian classifiers that can impose priors over the space of outcomes. Although NML is typically considered from the perspective of optimal coding (Grünwald, 2007; Fogel & Feder, 2018), we show how it can be used for success classifiers, and discuss connections to exploration in RL. We propose a novel technique for making NML computationally tractable based on meta-learning, which more directly optimizes for quick NML computation as compared to prior methods like Zhou & Levine (2020) which learn an amortized posterior.

3. Preliminaries

In this section, we discuss background on RL using successful outcome examples as well as conditional normalized maximum likelihood.

3.1. Reinforcement Learning with Outcome Examples

We follow the framework proposed by Fu et al. (2018b) and assume that we are provided with a Markov decision process (MDP) *without* a reward function, given by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, \mu_0)$, as well as successful outcome examples $\mathcal{S}_+ = \{s_+^k\}_{k=1}^K$, which is a set of states in which the desired task has been accomplished. This formalism is easiest to describe in terms of the control as inference framework (Levine, 2018). The relevant graphical model (refer to (Fu et al., 2018b)) consists of states and actions, as well as binary success variables $e_t \in \{0, 1\}$ that represent the occurrence of a particular event. The agent's objective is to cause this event to occur (e.g., a robot that is cleaning the floor must cause the "floor is clean" event to occur). Formally, we assume that the states in $\mathcal{S}_+$ are sampled from the distribution $p(s_t|e_t = 1)$ — that is, states where the desired event has taken place — and try to infer the distribution $p(e_t = 1|s_t)$ to use as a reward function. In this work, we focus on efficient methods for solving this reformulation of the RL problem by utilizing a novel uncertainty quantification method to represent $p(e_t|s_t)$.

In practice, prior methods that build on this formulation of the RL problem (Fu et al., 2018b) derive an algorithm where the reward function in RL is produced by a classifier that estimates $p(e_t = 1|s_t)$. Following the derivation in Fu et al. (2018a), it is possible to show that the correct source of *negative* examples is the state distribution of the policy itself, $\pi(s)$. This insight results in a simple algorithm: at each iteration of the algorithm, the policy is updated to maximize the current reward, given by $\log p(e_t = 1|s_t)$, then samples from the policy are added to the set of negative examples $\mathcal{S}_-$, and the classifier is retrained on the original positive set $\mathcal{S}_+$ and the updated negative set $\mathcal{S}_-$. As noted in prior work (Fu et al., 2018b), this process is closely connected to GANs and inverse reinforcement learning, where the classifier plays the role of the discriminator and the policy that of the generator. However, as we will discuss, this strategy can often face significant exploration challenges.

3.2. Conditional Normalized Maximum Likelihood

Our work utilizes the principle of conditional normalized maximum likelihood (CNML) (Rissanen & Roos, 2007; Grünwald, 2007; Fogel & Feder, 2018), which we review briefly. CNML is a method for performing k-way classification, given a model class Θ and a dataset $\mathcal{D} = \{(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)\}$, and has been shown to

provide better calibrated predictions and uncertainty estimates with minimax regret guarantees (Bibas et al., 2019). More specifically, the CNML distribution can be shown to provably minimize worst-case regret against an oracle learner that has access to the true labels, but does not know which point it will be tested on. We refer the reader to Fogel & Feder (2018); Zhou & Levine (2020) for a more complete consideration of the theoretical properties of the CNML distribution.

To predict the class of a query point x_q, CNML constructs k augmented datasets by adding x_q with a different label in each dataset, which we write as $\mathcal{D} \cup (x_q, y = i), i \in (1, 2, ..., k)$. CNML then defines the class distribution by solving the maximum likelihood estimation problem at query time for each of these augmented datasets to convergence, and normalizes the likelihoods as follows:

$$p_{\text{CNML}}(y = i|x_q) = \frac{p_{\theta_i}(y = i|x_q)}{\sum_{j=1}^k p_{\theta_j}(y = j|x_q)} \qquad (1)$$

$$\theta_i = \arg\max_{\theta \in \Theta} \mathbb{E}_{(x,y) \sim \mathcal{D} \cup (x_q, y=i)}[\log p_\theta(y|x)] \qquad (2)$$

If x_q is close to other datapoints in $\mathcal{D}$, the model will struggle to assign a high likelihood to labels that differ substantially from those of nearby points. However, if x_q is far from all datapoints in $\mathcal{D}$, then the different augmented maximum likelihood problems can easily classify x_q as any arbitrary class, providing us with likelihoods closer to uniform. We refer readers to Grünwald (2007) for an in-depth discussion of CNML and its connections to minimum description length and regret minimization. Intuitively, the CNML classifier provides a way to impose a uniform prior for uncertainty quantification, where we predict the uniform distribution on unseen inputs since they are maximally uncertain, and defer more to the maximum likelihood solution on frequently seen inputs since they are minimally uncertain.

4. Bayesian Success Classifiers for Reward Inference

As discussed in Section 3.1, a principled way of approaching outcome-driven RL is to train a classifier to determine whether a particular state is a successful outcome or not. However, while such a technique would eventually converge to the correct solution, it frequently suffers from uninformative or incorrect rewards during the learning process. For example, Figure 2 depicts a simple 2D maze scenario where the agent starts at the top left corner and the positive outcomes are at the bottom right corner of the environment. Without suitable regularization, the decision boundary may take on the form of a sharp boundary *anywhere* between the positive and negative examples in the early stages of training. As a result, the classifier might provide little to no reward signal for the policy, since it can assign arbitrarily small
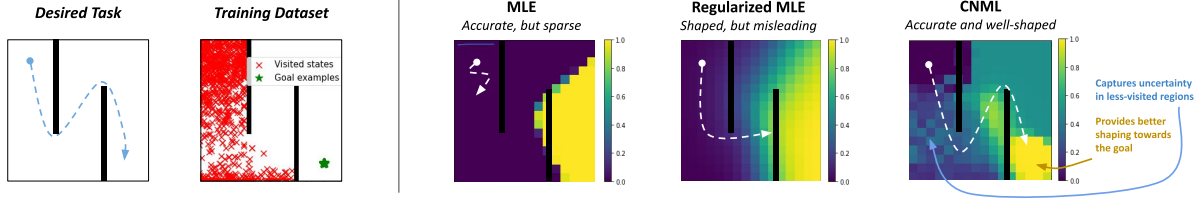
Figure 2. Comparison of rewards given by various classifier training schemes on a 2D maze example. Typical maximum likelihood (MLE) classifiers commonly suffer from either a lack of useful learning signal (if trained to convergence) or misleading local optima (if regularized using standard methods such as weight decay or early stopping), whereas CNML produces accurate and well-shaped rewards.

Algorithm 1 RL with CNML-Based Success Classifiers

1: User provides success examples $\mathcal{S}_+$
2: Initialize policy π, replay buffer $\mathcal{S}_-$, and reward classifier parameters $\theta_\mathcal{R}$
3: **for** iteration $i = 1, 2, ...$ **do**
4:    Add on-policy examples to $\mathcal{S}_-$ by executing π.
5:    Sample n_{test} points from $\mathcal{S}_+$ (label 1) and n_{test} points from $\mathcal{S}_-$ (label 0) to construct a dataset $\mathcal{D}$
6:    Assign state rewards as $r(s) = p_{\text{CNML}}(e = 1|s, \mathcal{D})$
7:    Train π with RL algorithm
8: **end for**

probabilities to the states sampled from the policy. Given that the classifier-based RL process is essentially equivalent to training a GAN (as described in Section 3.1), this issue is closely related to the challenges of GAN training as noted by Arjovsky & Bottou (2017), where an ideal maximum likelihood discriminator provides no gradient signal for training the generator.

We note that this issue is not pathological: our experiments in Section 6 show that this phenomenon of poor reward shaping happens in practice. In addition, introducing naïvely chosen forms of regularization such as weight decay, as is common in prior works, may actually provide *incorrect* reward shaping to the algorithm, making it more challenging to actually accomplish the task (as illustrated in Figure 2). This often limits classifier-based RL techniques to tasks with trivial exploration challenges. In this section, we will discuss how a simple change to the procedure for training a classifier, going from standard maximum likelihood estimation to an approach based on the principle of normalized maximum likelihood, allows for an appropriate consideration of uncertainty quantification that can solve problems with non-trivial exploration challenges.

4.1. Regularized Success Classifiers via Normalized Maximum Likelihood

It is important to note that for effective exploration in reinforcement learning, the rewards should not just indicate whether a state is a successful outcome (since this will be 0 everywhere but successful outcomes), but should instead

provide a sense of whether a particular state may be on the path to a successful outcome and should be explored further. The standard maximum likelihood classifier described in Section 3 is overly pessimistic in doing so, setting the likelihood of all intermediate states to 0 in the worst case, potentially mislabeling promising states to explore. To avoid this, we want to use a classification technique that minimizes this worst-case regret, maintaining some level of uncertainty about whether under-visited states are on the path to successful outcomes. As discussed in Section 3.2, the technique of conditional normalized maximum likelihood provides us a straightforward way to obtain such a classifier. CNML is particularly well suited to this problem since, as discussed in Zhang (2011), it essentially imposes a uniform prior over the space of outcomes. It thus avoids pathological collapse of rewards by maintaining a measure of uncertainty over whether a state is potentially promising to explore further, rather than immediately bringing its likelihood to 0 as maximum likelihood solutions would.

To use CNML for reward inference, the procedure is similar to the one described in Section 3. We construct a dataset using the provided successful outcomes as positives and on-policy samples as negatives. However, the label probabilities for RL are instead produced by the CNML procedure to obtain rewards $r(s) = p_{\text{CNML}}(e = 1|s)$ as follows:

$$r(s) = \frac{p_{\theta_1}(e = 1|s)}{p_{\theta_1}(e = 1|s) + p_{\theta_0}(e = 0|s)} \qquad (3)$$

$$\theta_0 = \arg\max_{\theta \in \Theta} \mathbb{E}_{(s_j, e_j) \sim \mathcal{D} \cup (s, e=0)}[\log p_\theta(e_j|s_j)] \qquad (4)$$

$$\theta_1 = \arg\max_{\theta \in \Theta} \mathbb{E}_{(s_j, e_j) \sim \mathcal{D} \cup (s, e=1)}[\log p_\theta(e_j|s_j)] \qquad (5)$$

This reward is then used to perform policy updates, new data is collected with the updated policy, and the process is repeated. A full description can be found in Algorithm 1.

To illustrate how this change affects reward assignment during learning, we visualize a potential assignment of rewards with a CNML-based classifier on the problem described earlier in Fig 2. When the success classifier is trained with CNML instead of standard maximum likelihood, interme-

diate unseen states would receive non-zero rewards rather than simply having vanishing likelihoods like the maximum likelihood solution, thereby incentivizing exploration. In fact, the CNML likelihood has a strong connection to count-based exploration, as we show next. Additionally, we also see that CNML is able to provide more directed shaping towards the successful outcomes when generalization exists across states, as explained below.

4.2. Relationship to Count-Based Exploration

In this section we relate the success likelihoods obtained via CNML to commonly used exploration methods based on counts. Formally, we prove that the success classifier trained with CNML is equivalent to a version of count-based exploration in the absence of any generalization across states (i.e., a fully tabular setting).

Theorem 1. *Suppose we are estimating success probabilities $p(e = 1|s)$ in the tabular setting, where we have an independent parameter for each state. Let $N(s)$ denote the number of times state s has been visited by the policy, and let $G(s)$ be the number of occurrences of state s in the set of positive examples. Then the CNML success probability $p_{CNML}(e = 1|s)$ is equal to $\frac{G(s)+1}{N(s)+G(s)+2}$. For states that are not represented in the positive examples, i.e. $G(s) = 0$, we then recover inverse counts $\frac{1}{N(s)+2}$.*

Refer to Appendix A.1 for a full proof. While CNML has a strong connection with counts as described above, it is important to note two advantages. First, the rewards are estimated without an explicit generative model, simply by using a standard discriminative model trained via CNML. Second, in the presence of generalization via function approximation, the exploration behavior from CNML can be significantly more task directed, as described next.

In most problems, when the classifier is parameterized by a function approximator with non-trivial generalization, the structure of the state space actually provides more information to guide the agent towards the successful examples than simply using counts. In most environments (Brockman et al., 2016; Yu et al., 2019) states are not completely uncorrelated, but instead lie in a representation space where generalization correlates with the dynamics structure in the environment. For instance, states from which successful outcomes can be reached more easily (i.e., states that are "close" to successful outcomes) are likely to have similar representations. Since the uncertainty-aware classifier described in Section 4.1 is built on top of such features and is trained with knowledge of the desired successful outcomes, it is able to incentivize more *task-aware* directed exploration than simply using counts. This phenomenon is illustrated intuitively in Fig 2, and demonstrated empirically in our experimental analysis in Section 6.

5. MURAL: Training Uncertainty-Aware Success Classifiers for Outcome Driven RL via Meta-Learning and CNML

In Section 4, we discussed how success classifiers trained via CNML can incentivize exploration and provide reward shaping to guide RL. However, the reward inference technique via CNML described in Section 4.1 is in most cases computationally intractable, as it requires optimizing separate maximum likelihood estimation problems to convergence on every data point we want to query. In this section, we describe a novel approximation that allows us to apply this method in practice.

5.1. Meta-Learning for CNML

We adopt ideas from meta-learning to amortize the cost of obtaining the CNML distribution. As noted in Section 4.1, computing the CNML distribution involves repeatedly solving maximum likelihood problems. While computationally daunting, these problems share a significant amount of common structure, which we can exploit to estimate the CNML distribution more efficiently. Meta-learning uses a distribution of training problems to explicitly meta-train models that can quickly adapt to new problems and, as we show next, can be directly used to accelerate CNML.

To apply meta-learning to computing the CNML distribution, we can formulate each of the maximum likelihood problems described in Equation 2 as a separate task for meta-learning, and apply a standard meta-learning technique to obtain a model capable of few-shot adaptation to the maximum likelihood problems required for CNML. While any meta-learning algorithm is applicable, we found model agnostic meta-learning (MAML) (Finn et al., 2017) to be effective. MAML aims to meta-train a model that can quickly adapt to new tasks via a few steps of gradient descent by explicitly performing a bi-level optimization. We refer readers to Finn et al. (2017) for a detailed overview.

The meta-training procedure to enable quick querying of CNML likelihoods can be described as follows. Given a dataset $\mathcal{D} = \{(\mathbf{x_0}, \mathbf{y_0}), (\mathbf{x_1}, \mathbf{y_1}), ..., (\mathbf{x_n}, \mathbf{y_n})\}$, we construct $2n$ different tasks τ_i, each corresponding to performing maximum likelihood estimation on the original dataset $\mathcal{D}$ combined with an additional point $(\mathbf{x_i}, \mathbf{y'})$, where y' is a proposed label of either 0 or 1 and $\mathbf{x_i}$ is a point from the dataset $\mathcal{D}$. Given these constructed tasks $\mathcal{S}(\tau)$, we perform meta-training as described by Finn et al. (2017):

$$\max_{\theta} \ \mathbb{E}_{\mathbf{x_i} \sim \mathcal{D}, y'. \in \{0,1\}}[\mathcal{L}(\mathcal{D} \cup (\mathbf{x_i}, \mathbf{y'}), \theta'_i)], \quad (6)$$

$$s.t \ \ \theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}(\mathcal{D} \cup (\mathbf{x_i}, \mathbf{y'}), \theta). \quad (7)$$

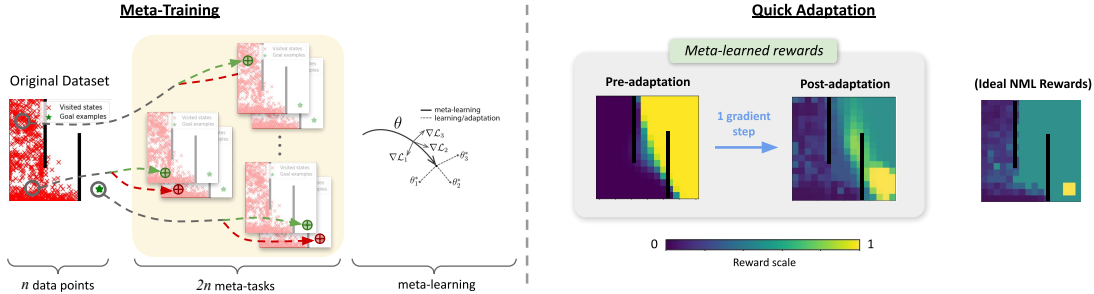This training procedure produces a set of parameters θ that

Figure 3. Diagram of using meta-NML to train a classifier. Meta-NML learns an initialization that can quickly adapt to new datapoints with arbitrary labels. At evaluation time, it approximates the NML probabilities (right) fairly well with a single gradient step.

can then be quickly adapted to provide the CNML distribution with a step of gradient descent. The model can be queried for the CNML distribution by starting from the meta-learned θ and taking one step of gradient descent for the dataset augmented with the query point, each with a different potential label. These likelihoods are then normalized to provide the CNML distribution as follows:

$$p_{\text{meta-NML}}(y|x; \mathcal{D}) = \frac{p_{\theta_y}(y|x)}{\sum_{y \in \mathcal{Y}} p_{\theta_y}(y|x)} \tag{8}$$

$$\theta_y = \theta - \alpha \nabla_\theta \mathbb{E}_{(x_i, y_i) \sim \mathcal{D} \cup (x,y)}[\mathcal{L}(x_i, y_i, \theta)]. \tag{9}$$

This process is illustrated in Fig 3, which shows how the meta-NML procedure can be used to obtain approximate CNML likelihoods with just a single gradient step.

This scheme for amortizing the computational challenges of NML (which we call *meta-NML*) allows us to obtain normalized likelihood estimates without having to retrain maximum likelihood to convergence at every single query point. A complete description, runtime analysis and pseudocode of this algorithm are provided in Appendix A.2 and A.3. Crucially, we find that meta-NML is able to approximate the CNML outputs well with just one or a few gradient steps, making it several orders of magnitude faster than standard CNML.

5.2. Applying Meta-NML to Success Classification

We apply the meta-NML algorithm described previously to learning uncertainty-aware success classifiers for providing rewards for RL in our proposed algorithm, which we call MURAL. Similarly to Fu et al. (2018b), we can train our classifier by first constructing a dataset $\mathcal{D}$ for binary classification, using success examples as positives, and on-policy samples as negatives, balancing the number of sampled positives and negatives in the dataset. Given this dataset, the classifier parameters $\theta_\mathcal{R}$ can be trained via meta-NML as described in Equation 7. The classifier can then be used to directly and quickly assign rewards to a state s according to its probabilities $r(s) = p_{\text{meta-NML}}(e = 1|s)$, and perform

standard reinforcement learning, as noted in Algorithm 2. Further details are in Appendix A.2.

6. Experimental Evaluation

In our experimental evaluation we aim to answer the following questions: (1) Can MURAL make effective use of successful outcome examples to solve challenging exploration tasks? (2) Does MURAL scale to dynamically complex tasks? (3) What are the impacts of different design decisions on the effectiveness of MURAL?

Further details, videos, and code can be found at https://sites.google.com/view/mural-rl

6.1. Experimental Setup

We first evaluate our method on maze navigation problems, which require avoiding several local optima. Then, we consider three robotic manipulation tasks that were previously covered in Singh et al. (2019a) with a Sawyer robot arm: door opening, tabletop object pushing, and 3D object picking. We also evaluate on a previously considered locomotion task (Pong et al., 2020) with a quadruped ant navigating to a target, as well as a dexterous manipulation problem with a robot repositioning an object with a multi-fingered hand. In the hand manipulation experiments, the classifier is provided with access to only the object position, while in the other tasks the classifier is provided the entire Markovian state. As we show in our results, exploration in these environments is challenging, and using naïvely chosen reward shaping often does not solve the problem at hand.

We compare with a number of prior algorithms. To provide a comparison with a previous method that uses standard success classifiers, we include the VICE algorithm (Fu et al., 2018b). Note that this algorithm is quite related to MURAL, but it uses a standard maximum likelihood classifier rather than a classifier trained with CNML and meta-learning. We also include a comparison with DDL, a technique for learning dynamical distances (Hartikainen et al., 2019). We addi-
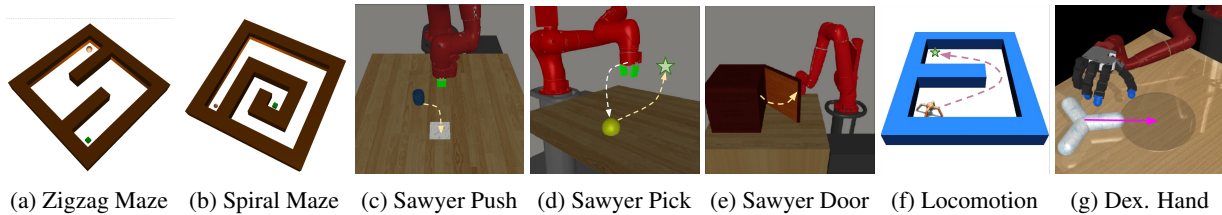
(a) Zigzag Maze (b) Spiral Maze (c) Sawyer Push (d) Sawyer Pick (e) Sawyer Door (f) Locomotion (g) Dex. Hand

Figure 4. We evaluate on two mazes, three robotic arm manipulation tasks, one locomotion task and one dexterous manipulation task: (a) the agent must navigate around an S-shaped corridor, (b) the agent must navigate a spiral corridor, (c) the robot must push a puck to location, (d) the robot must raise a randomly placed tennis ball to location, (e) the robot must open the door a specified angle. (f) the quadruped ant must navigate the maze to a particular location (g) the dexterous robotic hand must reposition an object on the table.

Algorithm 2 MURAL: Meta-learning Uncertainty-aware Rewards for Automated Outcome-driven RL

1: User provides success examples $\mathcal{S}_+$
2: Initialize policy π, replay buffer $\mathcal{S}_-$, and reward classifier parameters $\theta_{\mathcal{R}}$
3: **for** iteration $i = 1, 2, ...$ **do**
4:     Add on-policy samples to $\mathcal{S}_-$ by executing π.
5:     **if** iteration $i \mod k == 0$ **then**
6:         Sample n_{train} states from $\mathcal{S}_-$ to create $2n_{\text{train}}$ meta-training tasks
7:         Sample n_{test} total test points equally from $\mathcal{S}_+$ (label 1) and $\mathcal{S}_-$ (label 0)
8:         Meta-train $\theta_{\mathcal{R}}$ via meta-NML using Equation 7
9:     **end if**
10:    Assign state rewards via Equation 5
11:    Train π with RL algorithm
12: **end for**

---

tionally include comparisons to algorithms for task-agnostic exploration to show that MURAL performs more directed exploration and reward shaping. To provide a direct comparison, we use the same VICE method for training classifiers, but combine it with novelty-based exploration based on random network distillation (Burda et al., 2018b) for the robotic manipulation tasks, and oracle inverse count bonuses for maze navigation. We also compare to prior task-agnostic exploration techniques which use intrinsic curiosity (Pathak et al., 2017) and density estimates (Vezzani et al., 2019). Finally, to demonstrate the importance of shaped rewards, we compare to running Soft Actor-Critic (Haarnoja et al., 2018) with two naïve reward functions: a sparse reward, and a heuristic reward which uses L2 distance. More details are included in Appendix A.4 and A.6.

6.2. Comparisons with Prior Algorithms

We compare with prior algorithms on the domains described above. As we can see in Fig 5, MURAL is able to very quickly learn how to solve these challenging exploration tasks, often reaching better asymptotic performance than most prior methods, and doing so more efficiently than

VICE (Fu et al., 2018b) or DDL (Hartikainen et al., 2019). This suggests that MURAL is able to provide directed reward shaping and exploration that is substantially better than standard classifier-based methods. We provide a more detailed analysis of the shaping behavior of the learned reward in Section 6.4.

To isolate whether the benefits purely come from exploration or also from task-aware reward shaping, we compare with methods that only perform task-agnostic exploration. From these comparisons, it is clear that MURAL significantly outperforms methods that only use novelty-seeking exploration. We also compare to a heuristically-designed reward function based on Euclidean distance. MURAL generally outperforms simple manual shaping in terms of sample complexity and asymptotic performance, indicating that the learned shaping is non-trivial and adapted to the task. Of course, with sufficient domain knowledge, it is likely that this would improve. In addition, we find that MURAL scales up to tasks with challenging exploration in higher dimensional state and action spaces such as quadruped locomotion and dexterous manipulation, as seen in Fig 5.

6.3. Ablations

We first evaluate the importance of meta-learning for estimating the CNML distribution. In Figure 6, we see that naïvely estimating the CNML distribution by taking a single gradient step and following the same process as in our method, but without any meta-training, results in much worse performance. Second, we analyze whether the exploration behavior incentivized by MURAL is actually directed and task-aware or if it simply approximates count-based exploration. To that end, we modify the training procedure so that the dataset $\mathcal{D}$ consists of only the on-policy negatives, and add the inferred reward from the meta-NML classifier to the reward obtained by a standard maximum likelihood classifier (similarly to the VICE+RND baseline). We see that this performs poorly, showing that the MURAL classifier is doing more than just performing count-based exploration, and benefits from better reward shaping due to the success examples. Further ablations are available in Appendix A.5.
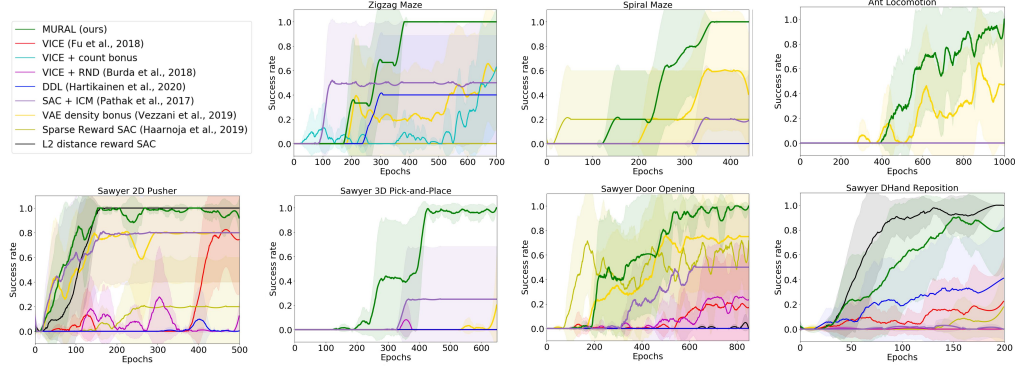
Figure 5. MURAL outperforms prior goal-reaching and exploration methods on all our evaluation environments, including ones with high-dimensional state and action spaces. MURAL also performs comparably to or better than a heuristically shaped hand-designed reward that uses Euclidean distance (black line), demonstrating that designing a well-shaped reward is not trivial in these domains. Shading indicates a standard deviation across 5 seeds. For details on the success metrics used, see Appendix A.4.

6.4. Analysis of MURAL

MURAL and reward shaping. To better understand how MURAL provides reward shaping, we visualize the rewards for various slices along the z axis on the Sawyer Pick-and-Place task, an environment which presents a significant exploration challenge. In Fig 7 we see that the MURAL rewards clearly correlate with the distance to the mean object position in successful

Figure 6. Ablative analysis of MURAL. The amortization from meta-learning and access to positive examples are both important components for performance.

outcomes, shown as a white star, thus guiding the robot to raise the ball to the desired location even if it has never reached this before. In contrast, the maximum likelihood classifier has a sharp, poorly-shaped decision boundary.
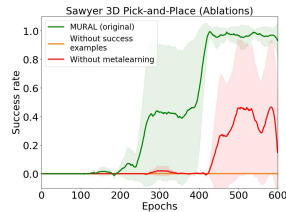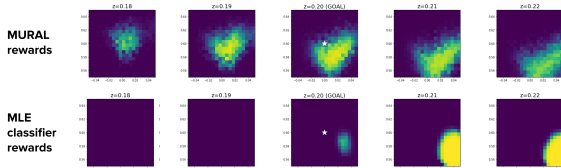

Figure 7. Visualization of reward shaping for 3D Pick-and-Place at various z values (heights). MURAL learns rewards that provide a smooth slope toward the successful outcomes, while the MLE classifier learns a sharp and poorly shaped decision boundary.

MURAL and exploration. Next, to illustrate the connection between MURAL and exploration, we compare the states visited by MURAL and by VICE (Fu et al., 2018b) in Figure 8. We see that MURAL naturally incentivizes the

agent to visit novel states, allowing it to navigate around local minima. In contrast, VICE learns a misleading reward function that prioritizes closeness to the success outcomes in xy space, causing the agent to get stuck.

Interestingly, despite incentivizing exploration, MURAL does not simply visit all possible states; at convergence, it has only covered around 70% of the state space. In fact, in Figure 8, MURAL prioritizes states that bring it closer to the success outcomes and ignores ones that don't, making use of the positive examples provided to it. This suggests that MURAL benefits from both novelty-seeking behavior and effective reward shaping.
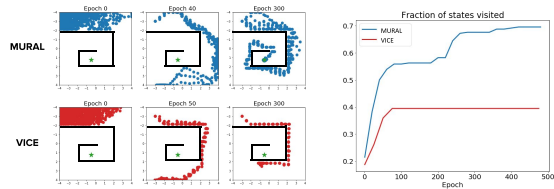

Figure 8. Plots of visitations and state coverage over time for MURAL vs. VICE. MURAL explores a significantly larger portion of the state space and is able to avoid local optima.

7. Discussion

In this work, we consider a subclass of RL problems where examples of successful outcomes specify the task. We analyze how solutions via standard success classifiers suffer from shortcomings, and training classifiers via CNML allows for better exploration to solve challenging problems. To make learning tractable, we propose a novel meta-learning approach to amortize the CNML process. While this work has shown the effectiveness of Bayesian classifiers for reward inference for tasks in simulation, it would be interesting to scale this solution to real world problems.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, O. P., and Zaremba, W. Hindsight experience replay. In *Advances in neural information processing systems*, 2017.

Andrychowicz, M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Arjovsky, M. and Bottou, L. Towards principled methods for training generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 2002.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in neural information processing systems*, pp. 1471–1479, 2016.

Bibas, K., Fogel, Y., and Feder, M. Deep pnml: Predictive normalized maximum likelihood for deep neural networks. *CoRR*, abs/1904.12286, 2019.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pp. 1613–1622, 2015.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *CoRR*, 2016.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018a.

Burda, Y., Edwards, H., Storkey, A. J., and Klimov, O. Exploration by random network distillation. *CoRR*, abs/1810.12894, 2018b.

Colas, C., Fournier, P., Chetouani, M., Sigaud, O., and Oudeyer, P.-Y. Curious: intrinsically motivated modular multi-goal reinforcement learning. In *International conference on machine learning*, pp. 1331–1340, 2019.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

Fogel, Y. and Feder, M. Universal batch learning with log-loss. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 21–25. IEEE, 2018.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adversarial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018a.

Fu, J., Singh, A., Ghosh, D., Yang, L., and Levine, S. Variational inverse control with events: A general framework for data-driven reward definition. In *Advances in Neural Information Processing Systems*, pp. 8547–8556, 2018b.

Ghosh, D., Gupta, A., Fu, J., Reddy, A., Devin, C., Eysenbach, B., and Levine, S. Learning to reach goals without reinforcement learning. *arXiv preprint arXiv:1912.06088*, 2019.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial networks. *CoRR*, abs/1406.2661, 2014.

Grünwald, P. D. *The minimum description length principle*. MIT press, 2007.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., and Levine, S. Soft actor-critic algorithms and applications. *CoRR*, abs/1812.05905, 2018.

Hartikainen, K., Geng, X., Haarnoja, T., and Levine, S. Dynamical distance learning for unsupervised and semi-supervised skill discovery. *CoRR*, abs/1907.08225, 2019.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pp. 4565–4573, 2016.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pp. 1109–1117, 2016.

Kaelbling, L. P. Learning to achieve goals. In *IJCAI*, 1993.

Knox, W. B. and Stone, P. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pp. 9–16, 2009.

Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *CoRR*, abs/1805.00909, 2018.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17:39:1–39:40, 2016.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pp. 13153–13164, 2019.

Misra, D., Henaff, M., Krishnamurthy, A., and Langford, J. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. *CoRR*, abs/1911.05815, 2019. URL http://arxiv.org/abs/1911.05815.

Nair, A. V., Pong, V., Dalal, M., Bahl, S., Lin, S., and Levine, S. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, pp. 9191–9200, 2018.

Ng, A. Y., Harada, D., and Russell, S. J. Policy invariance under reward transformations: Theory and application to reward shaping. In Bratko, I. and Dzeroski, S. (eds.), *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pp. 278–287. Morgan Kaufmann, 1999.

O'Donoghue, B. Variational bayesian reinforcement learning with regret bounds. *arXiv preprint arXiv:1807.09647*, 2018.

Osband, I., Russo, D., and Van Roy, B. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pp. 3003–3011, 2013.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, 2016.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

Pong, V. H., Dalal, M., Lin, S., Nair, A., Bahl, S., and Levine, S. Skew-fit: State-covering self-supervised reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2020.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations, 2018.

Rauber, P., Ummadisingu, A., Mutz, F., and Schmidhuber, J. Hindsight policy gradients. In *International Conference on Learning Representations*, 2018.

Rissanen, J. and Roos, T. Conditional NML universal models. In *2007 Information Theory and Applications Workshop*, pp. 337–341, 2007.

Rissanen, J. J. Fisher information and stochastic complexity. *IEEE transactions on information theory*, 42(1):40–47, 1996.

Schaul, T., Sun, Y., Wierstra, D., Gomez, F., and Schmidhuber, J. Curiosity-driven optimization. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *International conference on machine learning*, pp. 1312–1320, 2015.

Shtar'kov, Y. M. Universal sequential coding of single messages. *Problemy Peredachi Informatsii*, 23(3):3–17, 1987.

Singh, A., Yang, L., Finn, C., and Levine, S. End-to-end robotic reinforcement learning without reward engineering. In Bicchi, A., Kress-Gazit, H., and Hutchinson, S. (eds.), *Robotics: Science and Systems XV, University of Freiburg, Freiburg im Breisgau, Germany, June 22-26, 2019*, 2019a.

Singh, A., Yang, L., Hartikainen, K., Finn, C., and Levine, S. End-to-end robotic reinforcement learning without reward engineering. In *Robotics: Science and Systems (RSS)*, 2019b.

Stadie, B. C., Levine, S., and Abbeel, P. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Strens, M. A bayesian framework for reinforcement learning. In *ICML*, 2000.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pp. 2753–2762, 2017.

Veeriah, V., Oh, J., and Singh, S. Many-goals reinforcement learning. *arXiv preprint arXiv:1806.09605*, 2018.

Vezzani, G., Gupta, A., Natale, L., and Abbeel, P. Learning latent state representation for speeding up exploration. *arXiv preprint arXiv:1905.12621*, 2019.

Warde-Farley, D., Van de Wiele, T., Kulkarni, T., Ionescu, C., Hansen, S., and Mnih, V. Unsupervised control through non-parametric discriminative rewards. In *International Conference on Learning Representations*, 2018.

Wiering, M. and Schmidhuber, J. Efficient model-based exploration. In *Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior: From Animals to Animats*. MIT Press Cambridge, MA, 1998.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, 2019.

Zhang, J. Model selection with informative normalized maximum likelihood: Data prior and model prior. 2011.

Zhou, A. and Levine, S. Amortized conditional normalized maximum likelihood. *CoRR*, abs/2011.02696, 2020.

Zhu, H., Yu, J., Gupta, A., Shah, D., Hartikainen, K., Singh, A., Kumar, V., and Levine, S. The ingredients of real world robotic reinforcement learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., and Farhadi, A. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.