Autonomous Robotic Reinforcement Learning with Asynchronous Human Feedback

Anonymous Author(s)
Affiliation
Address
email

Abstract: Ideally, we would like to place a robot in a real-world environment, and keep it improving itself by gathering more experience autonomously. However, algorithms for autonomous robotic learning have been challenging to realize in the real world. While this has often been attributed to the challenge of sample complexity, even sample-efficient techniques are hampered by two major challenges - the difficulty of providing well "shaped" rewards, and the difficulty of continual reset-free training. In this work, we describe a system for real-world reinforcement learning that enables agents to show continual improvement by training directly in the real world without requiring painstaking effort to hand-design reward functions or reset mechanisms. Our system leverages occasional non-expert human-in-the-loop feedback from remote users to learn informative reward functions to guide exploration, while leveraging a simple self-supervised learning algorithm for goal-directed policy learning. We show that in the absence of resets, it is particularly important to account for the current "reachability" of the exploration policy when deciding which regions of the space to explore. Based on this insight, we instantiate a practical learning system - `GEAR`, which enables robots to simply be placed in real world environments and left to train autonomously without interruption. The system streams robot experience to a web interface only requiring occasional asynchronous feedback from remote, non-expert humans in the form of binary comparative feedback. We evaluate this system on a suite of robotic tasks in simulation and demonstrate its effectiveness at learning behaviors both in simulation and the real world.

Keywords: Autonomous Learning, Reward Specification, Reset-Free Learning

1 Introduction

Robotic reinforcement learning (RL) is a useful tool for continual improvement, particularly in unstructured real-world domains like homes or offices. The promise of *autonomous* RL methods for robotics is tremendous - simply place a robotic learning agent in a new environment, and see a continual improvement in behavior with an increasing amount of collected experience. Ideally, this would happen without significant environment specific instrumentation or algorithm design. However, the practical challenges involved in enabling real-world autonomous RL are non-trivial. While this challenge has been chalked down to sample efficiency [1, 2], we argue that the bottleneck for autonomous real-world RL is the requirement for constant human *effort* during learning. Given the episodic nature of most RL algorithms, human effort is required to provide constant resets to the system [3, 4, 5], and to carefully hand-designed reward functions [6, 7, 8] for learning. The requirement for constant human intervention to provide these rewards and resets [3], has made autonomous real world RL challenging.

The field of "autonomous RL" [3, 4, 5] has looked to study this problem of enabling uninterrupted real-world training of learning systems. A majority of these techniques have aimed to infer reward

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

functions from demonstrations or goal specifications [9, 10, 11, 12], while enabling reset-free learning by training an agent to reset itself [13, 9, 5]. However, these techniques have been challenging to scale to tasks with non-trivial exploration [14].

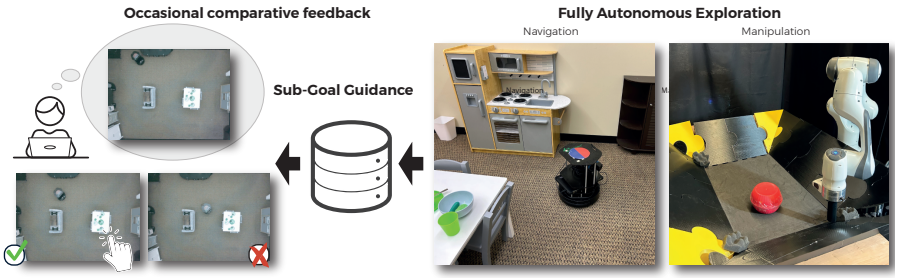

Figure 1: Problem setting in GEAR. The robot explores the world autonomously and reset-free only using cheap, occasional binary feedback from non-expert users to guide exploration. This allows for massive scaling of data experience and solving of much more challenging tasks.

To enable autonomous RL techniques to scale to complex tasks with challenging exploration, we note that directly reaching a final target goal through autonomous exploration can be difficult. However, achieving a promising intermediate sub-goal can be relatively simple. The process of reaching promising sub-goals can then be repeated until the desired target goal is accomplished, making the overall exploration process tractable, as long as sub-goals are appropriately chosen. The important question becomes - *"what are promising sub-goals to reach and how can we learn how to reach them during autonomous RL?"*

In this work, we note that a promising sub-goal is one that satisfies two criteria: (1) it is closer to the desired final goal than the current state of the system, and (2) it is reachable from the current state under the current policy. While criterion (1) is challenging to estimate without actually having the task solved beforehand, in this work, we show that asynchronously provided non-expert human feedback in the form of binary comparisons can be used to cheaply estimate state-goal proximity. This estimate of proximity can be paired with a density model for measuring reachability under the current policy, thereby selecting *promising* intermediate sub-goals that satisfy both criteria (1) and (2). Our proposed learning system - **G**uided **E**xploration for **A**utonomous **R**einforcement learning (GEAR) leverages occasionally provided comparative human feedback [15, 16] for directing exploration towards promising intermediate sub-goals, and self-supervised policy learning techniques [17] for learning goal-directed behavior from this exploration data. This is a step towards autonomous RL system that can leverage both self-collected experience and cheap human guidance to scale to a variety of complex real-world tasks with minimal effort.

2 Related Work

Autonomous Reinforcement Learning in Real World. RL has been used to learn skills in the real world through interaction with real-world environments [18, 19, 20, 21, 22, 23]. A common limitation is the requirement for episodic resets, necessitating frequent human interventions. "Autonomous reinforcement learning" aims to obviate these challenges by building learning systems that can learn with minimal interventions [4, 24, 5]. A large class of autonomous RL methods involve learning a forward policy to accomplish a task and a backward policy to reset the system [24, 25, 4, 26, 27, 5]. A different class of methods [13, 28, 29, 30] views the reset-free RL problem as a multi-task RL problem. These techniques typically require a human provided reward [5, 24, 4, 13, 28] or rely on simple techniques like goal-classifiers to provide rewards [11, 26]. These reward mechanisms fail to solve domains with challenging elements of exploration. GEAR is able to learn autonomously with minimal manual resets, while using cheap, asynchronous human feedback to guide exploration more effectively.

Learning from Human Feedback: To alleviate the challenges of reward function specification, we build on the framework of learning from binary human comparisons [15, 31, 32, 16, 33, 34, 35].

These techniques leverage binary comparative feedback from human supervisors to learn reward functions that can then be used directly with RL methods, often model-free [36]. While these techniques have often been effective in learning for language models, they have seen relatively few demonstrations in fully real-world autonomous RL at scale. The primary challenge is that these methods are too sample inefficient for real-world use, or require too much human feedback [15, 35]. In this work, we build on a recently introduced technique [37] that combines self-supervised policy learning via hindsight supervised learning [17] with learning from human feedback to allow for robust learning that is resilient to infrequent and incorrect human feedback. While [37] was largely evaluated in simulation or in simple episodic tasks, we leverage insights from [37] to build RL systems that do not require resets or careful environment setup.

Goal-conditioned reinforcement learning: In this work, we build on the framework of goal-conditioned policy learning to learn robot behaviors. Goal-conditioned RL [17, 38, 39, 40, 41] studies the sub-class of MDPs where tasks involve reaching particular "goal" states. The key insight behind goal-conditioned RL methods has been that in these problems, different tasks can be characterized directly by states in the state-space that the agent is tasked with reaching. This makes it relatively easy to determine which states in the environment have *actually* been reached. Based on this insight, a variety of self-supervised goal-conditioned RL techniques have been proposed [39, 42, 43, 44, 40, 17, 45, 46, 47] that aim to leverage the idea of "hindsight"-relabeling to learn policies. These techniques typically rely on policy generalization to show that learning on *actually* achieved goals can help reaching desired target goals. As opposed to these techniques, GEAR uses self-supervised policy learning [17, 42] for acquiring goal-directed behavior, while relying on human feedback [48] to direct exploration.

3 Preliminaries

Problem Setting. In this work, we focus on the autonomous reinforcement learning problem [4]. We model the agent's environment as a Markov decision process (MDP), symbolized by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho_0, \gamma)$, with standard notation [49]. The reward function is $r \in \mathcal{R}$, where $r : S \times A \to \mathcal{R}^1$ is unknown to us, as it is challenging to specify. As we discuss in Section 4, an approximate reward $\hat{r}(s)$ must be inferred from human feedback in the process of training. As in several autonomous RL problem settings, of particular interest is the initial state distribution $\rho_0(s)$, which is provided for evaluation, but the system is run reset-free without the ability to reset the system to initial states $s \sim \rho_0(s)$ during training [4]. The aim is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$, that maximizes the expected sum of rewards $\mathbb{E}_{\pi,\rho_0}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$ starting from the initial state distribution $\rho_0(s)$ and executing a learned policy π.

Goal-Conditioned Reinforcement Learning. Since reward functions are challenging to define in the most general case, goal-conditioned policy learning techniques considers a simplified class of MDPs where rewards are restricted to the problem of reaching particular goals. The goal-reaching problem can be characterized as $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho_0, \gamma, \mathcal{G}, p(g))$, with a goal space $\mathcal{G}$ and a target goal distribution $p(g)$ in addition to the standard MDP setup. In the episodic goal-conditioned RL setting, each episode involves sampling a goal from the goal distribution $g \sim p(g)$, and attempting to reach it. The policy, π, and reward, r, are conditioned on the selected goal $g \in \mathcal{G}$. Goal conditioned RL problems leverage a special form of the reward function as $r(s, a, g) = \mathbb{1}(s = g)$, 1 when a goal is reached, and 0 otherwise. As in standard RL, at evaluation time an agent is tasked with maximizing the discounted reward based on the goal, $\mathbb{E}_{g\sim p(g),\pi,\rho_0}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, g)]$. Note that unlike most work in goal-conditioned RL, we are in the autonomous goal-conditioned RL setting, where we do not have access to resets during training.

While goal-conditioned RL makes the reward $r(s_t, a_t, g)$ particularly easy to specify, in continuous spaces $r(s_t, a_t, g) = \mathbb{1}(s_t = g)$ is zero with high probability. Recent techniques have circumvented this by leveraging the idea of *hindsight relabeling* [43, 39, 17]. The key idea is to note that while the reward is 0 w.h.p for goals that were commanded, it can be set to 1 had the states that are actually reached, been commanded as goals, thereby providing self-supervised learning signal [43, 39, 17].

This idea of using self-supervision for policy learning has spanned both techniques for RL [39, 40] and iterated supervised learning [42, 17]. The resulting objective for supervised policy learning can be expressed as $\arg\max_\pi \mathbb{E}_{\tau\sim\mathbb{E}_g[\bar{\pi}(\cdot|g),g\sim p(g)]}\left[\sum_{t=0}^{T}\log\pi(a_t|s_t,\mathcal{G}(\tau))\right]$. Self-supervised policy learning algorithms [17] alternate between sampling trajectories by commanding the desired goals under the current policy $\bar{\pi}(\cdot|g)$ and target goal distribution $g\sim p(g)$, where $\bar{x}$ denotes stop-gradient. Policies can then be learned based on the goals reached in hindsight $g=\mathcal{G}(\tau)$, where $\mathcal{G}$ is any function for hindsight relabeling —for instance, simply choosing the last state of τ as the goal.

4 GEAR: A System for Autonomous Robotic Reinforcement Learning with Asynchronous Human Feedback

Our proposed system - **G**uided **E**xploration for **A**utonomous **R**einforcement learning (GEAR), is able to leverage a self supervised policy learning scheme with occasional human feedback to learn behaviors while requiring minimal resets or hand-specified rewards. While typical autonomous RL methods aim to reset themselves autonomously, this comes at the cost of a challenging reward design problem. The reward function has to be able to account for all eventualities that the system may find itself in, rather than behaviors from a single initial state. In this work, we instantiate a practical algorithm GEAR, that takes the perspective that cheap, asynchronous feedback in the form of binary comparisons provided remotely by humans can guide exploration in autonomous RL.

4.1 Reset-Free Learning via Goal-Conditioned Policy Learning

The problem of autonomous RL involves learning a policy to perform a task evaluated from a designated initial state distribution ρ_0, but without access to episodic resets during training. Typical paradigms for this problem [4, 26, 5] have alternated between training a "forward" and policy to solve the task, and a "reverse" policy to reset the environment. The challenge with applying this in the real world boils down to the difficulty of reward specification.

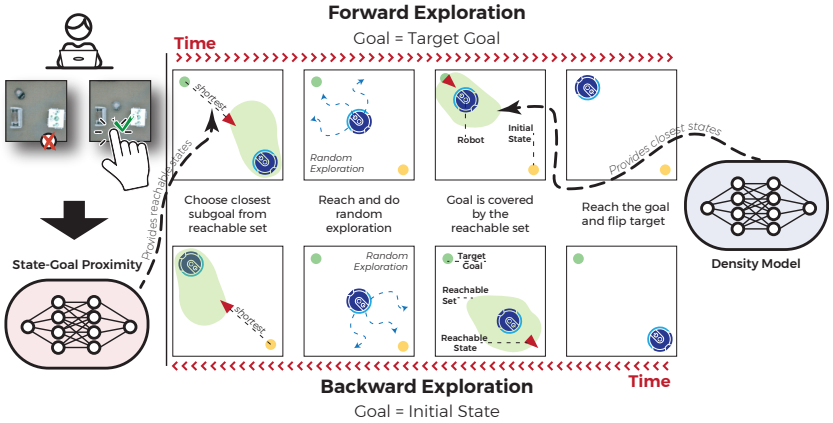

Figure 2: Depiction of autonomous exploration with GEAR - the policy alternates between trying to go to a goal state and get back to the initial state. In doing so the agent is commanded an intermediate sub-goal that is both proximal to the goal, and reachable under the current policy. When this is absent, the policy explores randomly. The resulting policy learns to go back and forth, while efficiently exploring the space.

To circumvent the problem of reward function specification, we can model this as a goal-conditioned policy learning problem and leverage self-supervised learning methods [17]. This involves learning a single goal-conditioned policy $\pi(a|s,g)$ that can perform both forward and reverse tasks. While trying to accomplish the forward goal, the policy takes $g\sim p(g)$ as it's goal, and once the goal is accomplished, the reverse process takes in $g\sim\rho_0(s)$ as it's goal, bringing the agent back to it's initial state. The policy $\pi(a|s,g)$ can be learned by self-supervised goal-conditioned policy learning methods, such as methods in [17, 42]. These methods relabel the commanded goals g to goals $\mathcal{G}(\tau)$

which were actually reached in hindsight, and use supervised learning to learn policies from this data (Section 3).

A problem with this self-supervised policy learning algorithm is that it relies purely on policy generalization for exploration; the policy π is commanded to reach $g \sim p(g)$ or $g \sim \rho_0(s)$, even if it has not actually seen valid paths to reach g. This may result in policies executing catastrophically bad trajectories when commanded with the desired target goals $g \sim p(g)$, since they are out-of-distribution for the policy training distribution. This problem is especially exacerbated in autonomous RL settings.

4.2 Guided Exploration and Policy Learning via Asynchronous Human Feedback

Rather than always commanding the initial state or the target goal as described above, human feedback can help select meaningful intermediate sub-goals to command the policy to direct exploration. In autonomous RL, meaningful intermediate sub-goals g_{sub} are those that make *progress* towards either the target goal $g \sim p(g)$ or the starting state $g \sim \rho_0(s)$ under the *current policy* π. Given a desired goal g, the intermediate sub-goal g_{sub} should be - (1) proximal to the desired goal g in terms of dynamical distance [47], and (2) reachable from the current state s under the policy π. Without knowledge of the optimal value function V^*, it is non-trivial to estimate both conditions (1) and (2). In GEAR, we rely on binary feedback provided by human users to estimate state-goal proximity (condition (1)) and on density estimation for computing state-goal reachability (condition (2)).

Proximity Estimation Using Human Feedback To estimate state goal proximity, we draw inspiration from work [37, 15, 16] in learning from human preferences. Specifically, we build directly on a recently proposed technique that uses human preferences to guide exploration in the *episodic* setting [37]. While [37] also guides exploration using human preferences to estimate state goal proximities, it has a strict episodic requirement making it unsuitable for autonomous RL.

Techniques based on comparative feedback aim to learn reward functions from binary comparisons provided by non-expert human supervisors. In the case of autonomous policy learning, human users can be asked to label which of state s_i or s_j is closer to a particular desired goal g. These preferences can then be used to infer (unnormalized) state-goal proximity functions $d_\phi(s_i, g)$ and $d_\phi(s_j, g)$ by optimizing $d_\phi(s_j, g)$ with respect to the following objective (derived from the Bradley-Terry economic model [48, 15, 33]): $\mathcal{L}_{\text{rank}}(\theta) = -\mathbb{E}_{(s_i,s_j,g),\sim\mathcal{D}}\left[\mathbb{1}_{i<j}\left[\log \frac{\exp -d_\phi(s_i,g)}{\exp -d_\phi(s_i,g)+\exp -d_\phi(s_j,g)}\right] + (1-\mathbb{1}_{i<j})\left[\frac{\exp -d_\phi(s_j,g)}{\exp -d_\phi(s_i,g)+\exp -d_\phi(s_j,g)}\right]\right]$. This suggests that if a state s_i is preferred over s_j in terms of it's proximity to the goal g, the unnormalized distance $d_\phi(s_i, g) < d_\phi(s_j, g)$.

During exploration, a promising sub-goal can be chosen by sampling a batch of visited states and scoring them according to their estimated state-goal proximity $d_\phi(s, g)$. As outlined in [37], the most promising candidate sub-goal need not be chosen to simply maximize the estimate proximity $d(s, g)$, but can instead be sampled softly from a softmax distribution $p(g_{\text{sub}}|s, g) = \frac{\exp -d_\phi(s,g)}{\sum_{s'\in\mathcal{D}} \exp -d_\phi(s',g)}$ to deal with imperfections in human comparisons.

Reachability Estimation Using Likelihood Based Density Estimation Learning proximity measures from human feedback as described above, can indicate which intermediate sub-goal g_{sub} is closest to a particular target goal g. However, this does not provide a measure of "reachability" - whether a chosen sub-goal g_{sub} is actually reachable by the current policy π, starting from the current environment state s. This is not a significant problem in an episodic setting, but in the autonomous reset-free learning case, many states that have been visited in the data buffer may not be reachable from the current environment state s using the current policy π.

In our self-supervised policy learning scheme (Section 4.1), reachability corresponds directly to marginal density - seeing (s, g_{sub}) pairs more often in the dataset is likely to indicate that g_{sub} is

reachable from a particular state s. This is a simple consequence of the fact that policies are learned via hindsight relabeling with supervised learning [17], and that a supervised learning oracle would ensure reachability for states with enough density in the training set. This suggests that the set of *reachable* intermediate sub-goals g_{sub} can be computed by estimating and then thresholding the marginal likelihood of various (s, g_{sub}) in the training data. To do so, standard maximum likelihood generative modeling technique can be used to learn a density $p_\psi(s_t, g_{\text{sub}})$ [50, 51, 52, 53, 54].

The learned density model $p_\psi(s_t, g_{\text{sub}})$ can be used to select *reachable* goals with a simple procedure - given a batch of sampled candidate sub-goals from the states visited thus far, we first filter reachable candidates by thresholding density $p_\psi(s_t, g_{\text{sub}}) \geq \epsilon$. The set of reachable candidates can then be used for sampling a proximal goal proportional to the state-goal distance d_ϕ estimated from human feedback as described above. Note that when there are no viable reachable candidates, the policy can perform random exploration. In our experimental evaluation in simulation, we estimate this density with a neural autoregressive density model [50, 51, 52] or a discretized, tabular density model.

4.3 System Overview

The overall system in GEAR alternates between trying to explore and learn a policy $\pi_\theta(a|s, g)$ to reach the target goal distribution $g \sim p(g)$ in the forward process, and reach the initial state $g \sim \rho_0(s)$ in the reverse process. In each of these processes, the agent selects appropriate intermediate sub-goals g_{sub} for exploration based on the current state and the desired target goal. The goal selection mechanism first samples a set of visited states from the replay buffer $\mathcal{D}_{\text{candidates}} = s_{i=1}^N \sim \mathcal{D}$. Then, it estimates the reachability of these sampled states from the current state s_t with an estimated density model p_ψ and the density model can filter out the unreachable states by density. Amongst the remaining reachable states $\mathcal{D}_{\text{candidates}}^{\text{filtered}}$, intermediate sub-goals are sampled g_{sub} proportional to their estimated proximity to the desired goal, according to the human-trained model $p(g_{\text{sub}}|s, g) = \frac{\exp -d_\phi(g_{\text{sub}}, g)}{\sum_{s' \in \mathcal{D}_{\text{candidates}}^{\text{filtered}}} \exp -d_\phi(s', g)}$. If no states are in the reachable set, the agent performs random exploration. This exploration process repeats until either the target goal $g \sim p(g)$(or the start state $g \sim \rho_0(s)$) is reached, following which the goal is flipped and the learning continues. Occasionally, the agent updates its model of density $p_\psi(s_t, g_{\text{sub}})$ by likelihood-based training and accepts occasional, asynchronous feedback from human supervisors to update the state-goal proximity estimate $d_\phi(g_{\text{sub}}, g)$. To warm-start training, we add a set of teleoperated data (potentially suboptimal) to the replay buffer, and simply pretrain our policy $\pi(a|s, g)$ with supervised learning on hindsight relabeled trajectories as described in Section 3 and [17, 42, 37]. Detailed pseudocode and algorithm equations can be found in Appendix A.

5 Experimental Evaluation

To evaluate our proposed learning system GEAR, we aim to answer the following questions: **(1)** Is GEAR able to learn behaviors autonomously in simulation environments without requiring resets or careful reward specification? **(2)** Is GEAR able to scale to learning robotic behaviors directly in the real world using asynchronous human feedback and autonomous RL?

5.1 Evaluation Domains

We evaluate our proposed learning system on three domains in simulation and two real world variants. Details of these environments can be found in Appendix B:

Maze navigation: As shown in Fig 3, this involves navigating a holonomic point agent around a maze environment from the starting position indicated in red to the target position indicated in green. This involves a challenging element of exploration requiring navigating through narrow passages to get to the target.

LoCoBot navigation: As shown in Fig 3, this involves navigating a non-holonomic LoCoBot robot around obstacles in a home environment. As opposed to the point navigation environment, this is

made more challenging by the presence of cluttered and non-trivial obstacles in the environment and the non-holonomic control. We study a real world variant of this task, with a TurtleBot deployed in a miniature home as shown in Fig 3. The agent is tasked with navigating through clutter to reach a particular set of goal locations. We bootstrap the TurtleBot environment in the real world with 5 offline demonstrations of 30 time-steps each to make learning more efficient, and then learn autonomously with real human feedback through the interface shown in Fig 7. Details can be found in Appendix B.

Block Pusher: As shown in Fig 3, this involves learning to push a block from one position to another in an arena. This is challenging as planar pushing requires careful low level control and the block can get trapped against the walls during autonomous learning. We study a real world variant of this task with a Franka Panda robot pushing a block to various locations as shown in Fig 3. Details can be found in Appendix B. We bootstrap the pusher environment in simulation with 20 offline demonstrations, and in the real world world with 25 offline demonstrations to make learning more efficient. For pusher experiments we used a synthetic proxy (as in [37]) to provide comparative labels in place of a real human supervisor to ensure systematic comparisons. Details can be found in Appendix B.

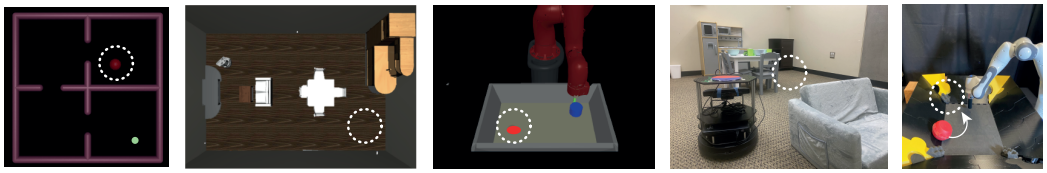

Figure 3: Evaluation Domains for GEAR. We consider a mixture of navigation and manipulation tasks both in simulation and the real world for autonomous learning. Note that the white circles are the goal locations.

5.2 Baselines and Comparisons

To test the effectiveness of GEAR, we compare with several baselines: **(1) GCSL Reset-Free [17]:** this baseline involves doing autonomous learning with purely self-supervised policy learning [17], alternating between commanding the start and the goal during exploration **(2) HUGE Reset-Free [37]:** this baseline compares with the HUGE algorithm [37], which leverages comparative human feedback asynchronously, but without accounting for policy reachability during training **(3) Classifier Based Rewards (VICE) [12, 9]:** this baseline [55, 26] performs forward-backward autonomous RL going between a start position and a goal position with the rewards provided by a classifier that is trained with goal states as positives and on-policy states as negatives, as in [26] **(4) Learning from Human Preferences [15]:** this baseline adapts the learning from human preferences paradigm [15] to the autonomous RL setting by commanding goals both forward and backward. To evaluate all methods, we follow the protocol in [4], where training proceeds reset-free but intermediate checkpoints are loaded in for evaluation from the initial state distribution. Additional visualizations are at https://sites.google.com/view/gearcorl/.

5.3 Does GEAR learn behaviors autonomously in simulation?

As evident in Fig 4, when evaluation success rates are averaged over 4 seeds, GEAR is able to successfully learn across environments in simulation. Even in domains with offline demonstrations (Pusher (Simulation)), we see a significant improvement over initialization, with autonomous training. Moreover, it is able to solve autonomous RL problems more effectively than autonomous RL methods using classifier based rewards, because it receives well shaped rewards from the comparative feedback. Note that GEAR is significantly more performant than the HUGE reset-free baseline [37], since it accounts for policy reachability while commanding goals. When ignoring this, the goal selection mechanism often selects infeasible goals that get the agent trapped. In Fig 4, we conduct experiments using both autoregressive neural density models [50, 51] and tabular densities for measuring policy reachability. While the comparisons in Fig 4 are done from the Lagrangian system

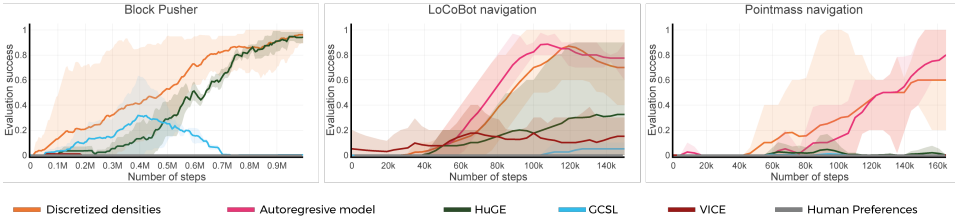

Figure 4: Success rate of autonomous training in simulation of GEAR as compared to baselines. We find that both autoregressive and tabular variants of GEAR are able to successfully accomplish all tasks, more efficiently than alternative reset-free, goal conditioned and human in the loop techniques

state, we will explore scaling this to visual inputs in future work. For the simulation experiments here, we leveraged a synthetic oracle in place of a real human for ease of comparison, but in the following section we leveraged real human supervisors to provide guidance for TurtleBot navigation. Ablations and analysis can be found in Appendix C, and experiment details in Appendix B, D.

5.4 Does GEAR learn behaviors autonomously in the real world from human feedback?

Next, we evaluated GEAR in the real world on a real-world robot navigation task with the TurtleBot (Fig 3). This task involves leaving the robot in a living room and allowing it to explore how to navigate the room autonomously to go from one location to another through obstacles, with human feedback provided through a web interface (Appendix Fig 7). The robot is provided around 150 transition tuples of teleoperated seed data, and then left to improve autonomously. Human supervisors provide occasional feedback, approximately 1900 comparative labels provided asynchronously over 6 hours. We see from the evaluation plots in Fig 5 that training improves the policies over time, learning in approximately 6 hours. We also evaluated GEAR on a real world pusher environment, shown in Fig 3. The robot is provided roughly 800 transitions tuples (25 trajectories) of teleoperated seed data, and then left to improve autonomously. We see from Fig 5 that policies improve over time, learning in approximately 20 minutes. Details are in Appendix B.

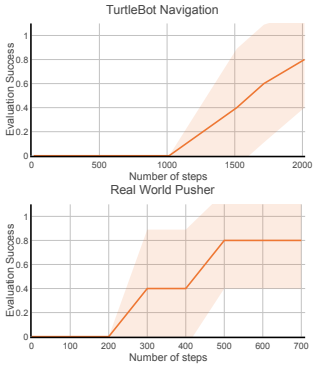

Figure 5: Evaluation in the real-world for Franka pusher and TurtleBot Navigation showing continuous improvement

6 Conclusion and Limitations

In this work, we present a framework for autonomous reinforcement learning in the real world using cheap, asynchronous human feedback. We show how a self supervised policy learning algorithm can be efficiently guided using binary human comparison feedback to select intermediate subgoals that direct exploration. The resulting learning algorithm can be deployed on a variety of tasks in simulation and the real world, learning from self-collected data with occasional human feedback.

Limitations: This work has several limitations: (1) *Safety Guarantees:* Real-world exploration with RL can be unsafe, leading to potentially catastrophic scenarios during exploration, (2) *Limitation to Binary Comparisons:* Binary comparisons provided by humans are cheap, but provide impoverished feedback since this only provides a single bit of information per comparison. (3) *Requirement for Pretraining Demonstrations:* For practical learning, the efficiency of GEAR is enhanced by using teleoperated pretraining data. This can be expensive to collect (4) *Density Model as a Proxy for Reachability:* We use density models as proxies for reachability, but this is only a valid metric in a small set of quasistatic systems. More general notions of reachability can be incorporated. (5) *Learning from low dimensional state:* the current instantiation of GEAR learns from low dimensional state estimates through a visual state-estimation system, which needs considerable tuning.

References

[1] L. M. Smith, I. Kostrikov, and S. Levine. A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *CoRR*, abs/2208.07860, 2022. doi:10.48550/arXiv.2208.07860. URL https://doi.org/10.48550/arXiv.2208.07860.

[2] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg. Daydreamer: World models for physical robot learning. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 2226–2240. PMLR, 2022. URL https://proceedings.mlr.press/v205/wu23c.html.

[3] H. Zhu, J. Yu, A. Gupta, D. Shah, K. Hartikainen, A. Singh, V. Kumar, and S. Levine. The ingredients of real world robotic reinforcement learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rJe2syrtvS.

[4] A. Sharma, K. Xu, N. Sardana, A. Gupta, K. Hausman, S. Levine, and C. Finn. Autonomous reinforcement learning: Formalism and benchmarking. *arXiv preprint arXiv:2112.09605*, 2021.

[5] B. Eysenbach, S. Gu, J. Ibarz, and S. Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=S1vuO-bCW.

[6] W. B. Knox, A. Allievi, H. Banzhaf, F. Schmitt, and P. Stone. Reward (mis)design for autonomous driving. *Artif. Intell.*, 316:103829, 2023. doi:10.1016/j.artint.2022.103829. URL https://doi.org/10.1016/j.artint.2022.103829.

[7] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. V. Wyk, A. Zhurkevich, B. Sundaralingam, Y. Narang, J. Lafleche, D. Fox, and G. State. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. *CoRR*, abs/2210.13702, 2022. doi:10.48550/arXiv.2210.13702. URL https://doi.org/10.48550/arXiv.2210.13702.

[8] P. Agrawal. The task specification problem. In A. Faust, D. Hsu, and G. Neumann, editors, *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pages 1745–1751. PMLR, 2021. URL https://proceedings.mlr.press/v164/agrawal22a.html.

[9] A. Sharma, R. Ahmad, and C. Finn. A state-distribution matching approach to non-episodic reinforcement learning. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 19645–19657. PMLR, 2022. URL https://proceedings.mlr.press/v162/sharma22a.html.

[10] A. Sharma, A. M. Ahmed, R. Ahmad, and C. Finn. Self-improving robots: End-to-end autonomous visuomotor reinforcement learning. *CoRR*, abs/2303.01488, 2023. doi:10.48550/arXiv.2303.01488. URL https://doi.org/10.48550/arXiv.2303.01488.

[11] K. Xu, Z. Hu, R. Doshi, A. Rovinsky, V. Kumar, A. Gupta, and S. Levine. Dexterous manipulation from images: Autonomous real-world RL via substep guidance. *CoRR*, abs/2212.09902, 2022. doi:10.48550/arXiv.2212.09902. URL https://doi.org/10.48550/arXiv.2212.09902.

[12] J. Fu, A. Singh, D. Ghosh, L. Yang, and S. Levine. Variational inverse control with events: A general framework for data-driven reward definition. In S. Bengio, H. M.

Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8547–8556, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/c9319967c038f9b923068dabdf60cfe3-Abstract.html.

[13] A. Gupta, J. Yu, T. Z. Zhao, V. Kumar, A. Rovinsky, K. Xu, T. Devlin, and S. Levine. Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pages 6664–6671. IEEE, 2021. doi:10.1109/ICRA48506.2021.9561384. URL https://doi.org/10.1109/ICRA48506.2021.9561384.

[14] K. Xu, S. Verma, C. Finn, and S. Levine. Continual learning of control primitives : Skill discovery via reset-games. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3472ab80b6dff70c54758fd6dfc800c2-Abstract.html.

[15] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

[16] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

[17] D. Ghosh, A. Gupta, J. Fu, A. Reddy, C. Devin, B. Eysenbach, and S. Levine. Learning to reach goals without reinforcement learning. 2019.

[18] S. Lange, M. Riedmiller, and A. Voigtländer. Autonomous reinforcement learning on raw visual input data in a real world application. In *The 2012 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2012.

[19] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[20] H. Zhu, A. Gupta, A. Rajeswaran, S. Levine, and V. Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019.

[21] M. Bloesch, J. Humplik, V. Patraucean, R. Hafner, T. Haarnoja, A. Byravan, N. Y. Siegel, S. Tunyasuvunakool, F. Casarini, N. Batchelor, et al. Towards real robot learning in the wild: A case study in bipedal locomotion. In *Conference on Robot Learning*, pages 1502–1511. PMLR, 2022.

[22] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4):1307–1319, 2020.

[23] H. Bharadhwaj, Z. Wang, Y. Bengio, and L. Paull. A data-efficient framework for training and sim-to-real transfer of navigation policies. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 782–788. IEEE, 2019.

[24] W. Han, S. Levine, and P. Abbeel. Learning compound multi-step controllers under unknown dynamics. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6435–6442. IEEE, 2015.

[25] C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

[26] A. Sharma, A. M. Ahmed, R. Ahmad, and C. Finn. Self-improving robots: End-to-end autonomous visuomotor reinforcement learning. *arXiv preprint arXiv:2303.01488*, 2023.

[27] A. Sharma, A. Gupta, S. Levine, K. Hausman, and C. Finn. Autonomous reinforcement learning via subgoal curricula. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18474–18486, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/99c83c904d0d64fbef50d919a5c66a80-Abstract.html.

[28] A. Gupta, C. Lynch, B. Kinman, G. Peake, S. Levine, and K. Hausman. Demonstration-bootstrapped autonomous practicing via multi-task reinforcement learning. *CoRR*, abs/2203.15755, 2022. doi:10.48550/arXiv.2203.15755. URL https://doi.org/10.48550/arXiv.2203.15755.

[29] K. Xu, S. Verma, C. Finn, and S. Levine. Continual learning of control primitives: Skill discovery via reset-games. *Advances in Neural Information Processing Systems*, 33:4999–5010, 2020.

[30] Z. Zhang and L. Weihs. When learning is out of reach, reset: Generalization in autonomous visuomotor reinforcement learning. *CoRR*, abs/2303.17600, 2023. doi:10.48550/arXiv.2303.17600. URL https://doi.org/10.48550/arXiv.2303.17600.

[31] E. Biyik. Learning preferences for interactive autonomy. *CoRR*, abs/2210.10899, 2022. doi:10.48550/arXiv.2210.10899. URL https://doi.org/10.48550/arXiv.2210.10899.

[32] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia. Active preference-based learning of reward functions. In N. M. Amato, S. S. Srinivasa, N. Ayanian, and S. Kuindersma, editors, *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017. doi:10.15607/RSS.2017.XIII.053. URL http://www.roboticsproceedings.org/rss13/p53.html.

[33] D. S. Brown and S. Niekum. Deep bayesian reward learning from preferences. *CoRR*, abs/1912.04472, 2019. URL http://arxiv.org/abs/1912.04472.

[34] K. Lee, L. M. Smith, and P. Abbeel. PEBBLE: feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6152–6163. PMLR, 2021. URL http://proceedings.mlr.press/v139/lee21i.html.

[35] K. Lee, L. M. Smith, A. D. Dragan, and P. Abbeel. B-pref: Benchmarking preference-based reinforcement learning. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/d82c8d1619ad8176d665453cfb2e55f0-Abstract-round1.html.

[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

[37] I. Review. Breadcrumbs to the goal: Goal-conditioned exploration from human-in-the-loop feedback. *NeurIPS*, 2023.

[38] M. Liu, M. Zhu, and W. Zhang. Goal-conditioned reinforcement learning: Problems and solutions. In L. D. Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5502–5511. ijcai.org, 2022. doi:10.24963/ijcai.2022/770. URL https://doi.org/10.24963/ijcai.2022/770.

[39] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[40] A. Levy, G. Konidaris, R. Platt, and K. Saenko. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*, 2017.

[41] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, V. Kumar, and W. Zaremba. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *CoRR*, abs/1802.09464, 2018. URL http://arxiv.org/abs/1802.09464.

[42] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020.

[43] L. P. Kaelbling. Learning to achieve goals. In *IJCAI*, volume 2, pages 1094–8. Citeseer, 1993.

[44] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time. *arXiv preprint arXiv:2210.06407*, 2022.

[45] T. Davchev, O. Sushkov, J.-B. Regli, S. Schaal, Y. Aytar, M. Wulfmeier, and J. Scholz. Wish you were here: Hindsight goal selection for long-horizon dexterous manipulation. *arXiv preprint arXiv:2112.00597*, 2021.

[46] A. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine. Visual reinforcement learning with imagined goals. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9209–9220, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/7ec69dd44416c46745f6edd947b470cd-Abstract.html.

[47] K. Hartikainen, X. Geng, T. Haarnoja, and S. Levine. Dynamical distance learning for unsupervised and semi-supervised skill discovery. *arXiv preprint arXiv:1907.08225*, 2019.

[48] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[49] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. 2018.

[50] B. Uria, M. Côté, K. Gregor, I. Murray, and H. Larochelle. Neural autoregressive distribution estimation. *CoRR*, abs/1605.02226, 2016. URL http://arxiv.org/abs/1605.02226.

[51] A. van den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves. Conditional image generation with pixelcnn decoders. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4790–4798, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/b1301141feffabac455e1f90a7de2054-Abstract.html.

[52] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In M. Balcan and K. Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1747–1756. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/oord16.html.

[53] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

[54] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

[55] J. Fu, A. Singh, D. Ghosh, L. Yang, and S. Levine. Variational inverse control with events: A general framework for data-driven reward definition. *CoRR*, abs/1805.11686, 2018. URL http://arxiv.org/abs/1805.11686.

[56] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.