Evaluating the progress of Deep Reinforcement Learning in the real world: aligning domain-agnostic and domain-specific research

Juan Jose Garau-Luis [1] **Edward Crawley** [1] **Bruce Cameron** [1]

Abstract

Deep Reinforcement Learning (DRL) is considered a potential framework to improve many real-world autonomous systems; it has attracted the attention of multiple and diverse fields. Nevertheless, the successful deployment in the real world is a test most of DRL models still need to pass. In this work we focus on this issue by reviewing and evaluating the research efforts from both domain-agnostic and domain-specific communities. On one hand, we offer a comprehensive summary of DRL challenges and summarize the different proposals to mitigate them; this helps identifying five gaps of domain-agnostic research. On the other hand, from the domain-specific perspective, we discuss different success stories and argue why other models might fail to be deployed. Finally, we take up on ways to move forward accounting for both perspectives.

1. Introduction

In the recent years, multiple research fields and industries have become interested in Deep Reinforcement Learning (DRL) frameworks as a way to enhance decision-making processes in the real world and design better autonomous systems. The range of domains is large (Li, 2017; Naeem et al., 2020), including—but not limited to—robotics (Polydoros & Nalpantidis, 2017), communications (Luong et al., 2019), drug discovery (Elton et al., 2019), fluid mechanics (Garnier et al., 2021), autonomous vehicles (Talpaert et al., 2019), and recommender systems (Afsar et al., 2021).

Different motivations trigger the interest of domain-specific research in DRL:

- In some industries systems are scaling and presenting new degrees of freedom, which makes them harder to operate, especially in time-sensitive settings. In those cases DRL becomes a new approach to fast decision-making. An example is the work within the satellite communications community, in a moment when constellations are getting larger and more flexible (Deng et al., 2020; Ferreira et al., 2019).
- Some systems require control policies that leverage raw signals such as image, sound, or brain activity. DRL and its representation capabilities are thus studied to achieve better performance. Some robotics (Polydoros & Nalpantidis, 2017; Ibarz et al., 2021) and healthcare applications (Esteva et al., 2019; Yu et al., 2019) fall in this category.
- When supervision is costly or not possible, DRL offers a way to learn policies by encoding system goals into the reward function and leveraging exploration during training. This is the case of NP-hard combinatorial optimization problems (Drori et al., 2020) or drug design studies (Olivecrona et al., 2017; Popova et al., 2018), where DRL is able to provide approximate solutions and good candidate molecules, respectively.
- DRL is a framework that can account for long-term dependencies in decision-making, which is especially interesting for recommender platforms and other interaction-based systems (Afsar et al., 2021; Zhang et al., 2019).

Despite these motivations and the extensive research efforts to prove the usefulness of DRL in real-world contexts, the successful deployment in real environments is a test many of the proposed models in the literature still need to pass. This is mostly due to the added complexities of the real world compared to current experimental DRL settings.

From a domain-specific perspective, concrete tasks, problems, and environments in the real world are hard to fully characterize and training in real environments is not always possible or preferred. In addition, the nature of these tasks or problems—regardless of the domain—entails dealing with certain challenges that make learning more difficult: non-stationarity, high-dimensionality, sparse reward, etc.

[1]Engineering Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: Juan Jose Garau-Luis <garau@mit.edu>.

Reinforcement Learning for Real Life (RL4RealLife) Workshop in the 38^{th} International Conference on Machine Learning, 2021.

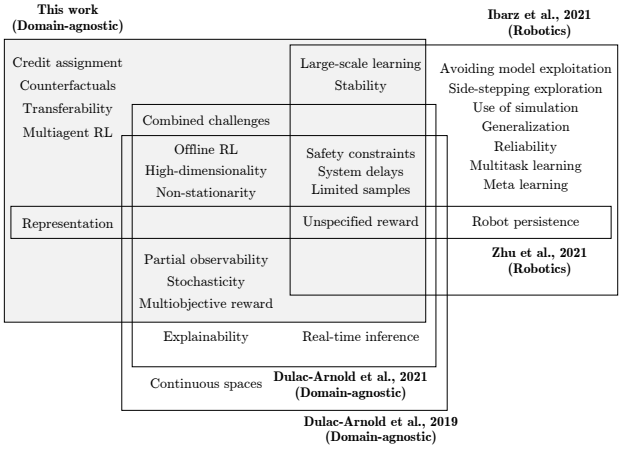

Figure 1. List of challenges of real-world DRL considered in different studies. Some of the challenges might interact or overlap in specific settings.

Reinforcement Learning (RL) researchers have identified a large number of those challenges (Dulac-Arnold et al., 2019; 2021; Ibarz et al., 2021; Zhu et al., 2020) and many solutions have been proposed to mitigate each of them. While the results are positive, testing is mostly limited to simulated environments. A key question remains often unanswered: how do the proposed models work in a real-world setting? Without that feedback it is not clear how much we have progressed in the path towards DRL-based autonomy.

To try to shed some light on this issue, in this paper we attempt to summarize and evaluate the progress of real-world-oriented DRL research from the perspective of both domain-agnostic and domain-specific research. We start by reviewing the domain-agnostic challenges of real-world DRL and compiling the solutions proposed in the literature. Based on this review, we identify five gaps that we deem necessary to address moving forward: a bias towards robotics use cases, not enough research on combined challenges, lack of real-world follow-through, little understanding of design tradeoffs, and operation being ignored.

In addition to the focus of the RL community on the problem, we argue domain-specific operators play an important role in adopting the technology. Therefore, we next analyze the same problem—the lack of real-world deployments—from the domain-specific viewpoint and try to align both efforts. We first highlight some examples of success stories, then discuss why other deployments might fail, and finally address ways to move forward.

2. Challenges of real-world DRL

Different studies have tried to summarize the challenges of real-world DRL, these are pictured in Figure 1. The work in (Dulac-Arnold et al., 2019; 2021) offers a comprehensive review of nine different domain-agnostic challenges and their impact on trained agents. Within the context of robotics, (Ibarz et al., 2021) identifies twelve issues based on real case studies and discusses possible mitigation strategies for each. Also focusing on robotics, (Zhu et al., 2020) highlights three real-world challenges and proposes a model taking those into account on a set of dexterous robotic manipulation tasks.

While the set of challenges is diverse, the mitigation strategies to address them are not unique to one challenge but sometimes the same method or mechanism is proposed in different contexts. To summarize the data, in this work we assume a domain-agnostic perspective and extend the challenges identified in (Dulac-Arnold et al., 2021). In the following section, we address each challenge independently and list the mitigation approaches found in the literature.

The challenges we consider are: offline DRL, learning from limited samples, large-scale learning, high-dimensionality, safe DRL, partial observability, non-stationarity, unspecified reward, multiobjective reward, system delays, representation, transferability, long trajectories and credit assignment, stochasticity, multiagent DRL, counterfactual reasoning, stability, and the combination of many of these challenges.

Note these challenges do not necessarily need to be uncorrelated, in some contexts specific challenges might be consequences of other challenges being present (e.g., non-stationarity during learning might be a cause of a partially observable environment). There are two other challenges some authors have highlighted that are not included in our analysis as isolated challenges: continuous spaces and real-time inference. In our view these challenges are hard to be found being addressed in isolation and therefore are excluded from our list.

Finally, designing, implementing, and operating a DRL model in the real world might also entail other considerations that go beyond difficult learning setups and are directly connected with the societal implications of the technology. Here we refer to interpretability (identified as one of the nine challenges in (Dulac-Arnold et al., 2021)), reproducibility (discussed in detail in (Henderson et al., 2018)), reliability, fairness, and privacy, as other challenges that are out of the scope of this work but remain important on a societal level.

3. Challenge mitigation strategies

In this section we aim to summarize the approaches different researchers have proposed to address the DRL challenges introduced in the previous section. Table 1 offers a summary of the high-level approaches and the contexts in which they are applied. This summary is based on our view of the challenges and the reviewed literature; it is open to different interpretations.

Table 1. Summary of the different approaches that have been considered by the RL community to address the specific challenges of real-world DRL. Each approach has been considered for different challenges independently.

Approach	Description	Examples
Meta learning	An outer loop learner changes meta parameters to better adapt to the challenge	Meta learning for offline DRL, meta learning for multi-objective reward, Meta RL for transferability, replacing action maximization by neural network search in high-dimensional spaces
Mathematical guarantee	Derive equations and theorems that support the challenge fulfillment	Lyapunov functions and primal-dual methods for safe DRL, assume uncertainty matrices to address non-stationarity
Neural network architectures	Rely on Deep Learning advances to increase robustness against the challenge	RNNs for partial observability, attention mechanisms for credit assignment, network ensembles for offline DRL
RL theory	Adapt theoretical RL frameworks to DRL settings and the use of neural networks	Off-policy algorithms, POMDPs, Delay-Aware MDPs, Constrained MDPs, Maximum-entropy RL
Embeddings and latent spaces	Address challenge problems by relying on robust intermediate embeddings	High- to low-dimensional embeddings, latent variables for non-stationarity, multimodal and contrastive learning-based representations, unsupervised learning
Reward estimation or modification	Try to overcome the challenge by directly modifying the reward structure and/or function	Reward shaping in long trajectories, reward shaping for safe DRL, reward redistribution, distributional RL against stochasticity
Deriving a model	Instead of learning a policy, learn models of the environment and use them to plan	Model-based RL, imitation learning, inverse RL
Pruning and masking	The learning process involves deciding, among different learning signals, how important each of them is and eliminating the unnecessary ones	Batch-constrained methods for offline DRL, distributed training in large-scale settings, action elimination in high-dimensional spaces, divide-and-conquer methods in stochastic environments
Use auxiliary tasks	Provide the agent with auxiliary tasks that, combined, increase robustness against the challenge	Multitask learning and online learning for data efficiency, self-supervised learning for unspecified reward, hierarchical RL in long trajectories
Data augmentation	Rely on different data augmentation and data wrangling techniques	Randomization to transfer better, data augmentation to address non-stationarity and stochasticity
Heuristics	Use human-crafted rules or processes to address the challenge	Scalarization of multiple objectives, hyperparameter tuning
Population-based methods	Have multiple agents with slightly different parameters/objectives and search for the best ones	Multiagent populations in partially observable environments, multiobjective populations
Multiagent specific	Solutions specific to the multiagent challenge that can not be mapped to other challenges	Independent Q-learning, decentraliced actors and centralized critic, hybrid mechanisms

Offline DRL Some systems might require to learn from offline logs instead of directly interacting with the environment, as that might be costly or not possible. An extensive review on the subject is presented in (Levine et al., 2020). To address this issue, different **off-policy algorithms** such as DDPG (Lillicrap et al., 2015) or D4PG (Barth-Maron et al., 2018) can be used in some cases. Other authors propose **batch-constrained RL** approaches (Fujimoto et al., 2019; Kumar et al., 2019; Siegel et al., 2020; Wu et al., 2019), where the learned policy is constrained based on the state-action distribution from the dataset and the extrapolation error is accounted for. Then, the work in (Agarwal et al., 2020) considers an **ensemble** or convex combination of Q-functions to leverage data in a replay buffer. Finally, **model-based RL** (Sutton & Barto, 2018) constitutes another research area in the context of offline DRL (Yu et al., 2020c; Kidambi et al., 2020).

Learning from limited samples Sometimes an agent must learn from a small number of samples, either because acquiring experience is slow or costly, or rapid adaptation to a new context is needed. While the representations chosen or learned can impact the learning speed (Srinivas et al., 2020), multiple methods have been proposed to directly address data efficiency. One alternative is to **learn a model** of the world and use that model to plan (Chua et al., 2018; Buckman et al., 2018). In the context of learning specific tasks, if **expert demonstrations** (Duan et al., 2017) or behavioral priors (Singh et al., 2020) are available, the agent can bootstrap from those to increase data efficiency. If the goal of the agent is **multitask learning**, the tasks can be learned concurrently taking into account multiple gradient inputs (Yu et al., 2020a), or if the tasks are to be learned

sequentially, **meta learning** algorithms, especially few-shot methods, offer a way to learn new tasks faster (Finn et al., 2017; Li et al., 2017; Sung et al., 2018; Lee et al., 2019b). Finally, in **online learning** contexts, where new tasks need to be learned fast and on-the-fly, different approaches have been proposed to promote forward and backward transfer (Schwarz et al., 2018; Mallya & Lazebnik, 2018; Chaudhry et al., 2018; Nagabandi et al., 2018) and avoid catastrophic forgetting (Kirkpatrick et al., 2017).

Large-scale learning In specific settings an agent should be able to capitalize on massive amounts of data fast, either because experience comes at a high frequency or multiple independent agents can collect experience simultaneously. For the latter case, when environments can be parallelized (e.g., self-driving cars, recommender systems), **distributed training with importance and priority mechanisms** has been proposed in different works (Adamski et al., 2018; Horgan et al., 2018; Espeholt et al., 2018).

High-dimensionality Some agents might need to operate in high-dimensional or combinatorial state and action spaces (e.g., natural language, molecular space). Here one approach is to operate with lower dimensional **embeddings** of the spaces (Dulac-Arnold et al., 2015; Robine et al., 2020). Other authors propose **action elimination** mechanisms to determine which actions not to take first (Zahavy et al., 2018). In the context, of Q-learning (Sutton & Barto, 2018) over large action spaces, (de Wiele et al., 2020) proposes **replacing the maximization** operation for a neural network. Finally, the use of **canonical spaces** can help reduce the state space by encapsulating redundant spaces together (Wu et al., 2017).

Safe DRL While exploration plays an important part in the success of RL, agents acting in real-world environments should account for safety constraints and be able to evaluate risks. One common approach to that end is to encode constraints as part of the reward function (Garcia & Fernández, 2015), but that might not be always desirable (Achiam et al., 2017). Some studies propose adding learnable safety layer on top of the policy in order to **prune or correct unsafe actions** (Dalal et al., 2018). The work in (Tessler et al., 2018) explores **reward shaping** and proposes a method that substracts constraint-violation penalties to the reward. Then, **Lyapunov functions** have been proposed to certify the stability and safety of different RL-based controllers (Chow et al., 2018; Berkenkamp et al., 2017). Constraint satisfaction can also be guaranteed by means of **primal-dual methods**, as shown in (Qin et al., 2021). Finally, an agent can also learn to **trade rewards and costs** by specifying constraints as costs with state-dependent and learnable Lagrangian coefficients (Bohez et al., 2019).

Partial observability Many environments in the real world are partially observable. In the context of DRL, some authors initially proposed incorporating **past observations** to the state (Mnih et al., 2015) or use **recurrent neural network** architectures (Hausknecht & Stone, 2015). Inspired by the theory on POMDPs (Cassandra et al., 1994), works like (Igl et al., 2018) propose training a variational autoencoder to learn latent representations encoding **belief states**. In the case the agent competes against other non-fully-observable agents, (Jaderberg et al., 2019) shows that **training populations of agents** eventually leads to best agents finding suitable policies for the environment. If the agent must cooperate with the other agents, the use of **shared experience replay** helps mitigating the effect of partial observability (Omidshafiei et al., 2017).

Non-stationarity A robust policy should be effective in non-stationary environments, where the underlying transition dynamics change over time due to various factors such as noise. In these contexts, one alternative is the use of **latent variables** that encode environment representations robust to noisy cues (Xie et al., 2020). A well-established approach is to **assume uncertainty in the transition matrices and derive robust algorithms** that consider worst-case scenarios (Mankowitz et al., 2019) or pursue soft-robustness (Derman et al., 2018). Bayesian optimization-based methods can be also derived from this latter idea (Derman et al., 2020). Finally, **data augmentation and randomization** during training can also lead to policies that adapt to real-world environments and generalize better (Peng et al., 2018).

Unspecified reward Sometimes agents must learn skills without reward signals, due to unavailable human feedback, complex exploration dynamics, or long horizon tasks. If there is no reward function but expert demonstrations are available, **Inverse RL** is an approach to learn reward signals (Fu et al., 2017). The work in (Hansen et al., 2020) proposes a method to train policies by means of **self-supervised learning** when deploying in environments without reward information. Another alternative is to learn a goal-conditioned policy via **unsupervised learning**, maximizing the similarity between visited states and a goal state (Warde-Farley et al., 2018). In the context of multitask learning, in (Eysenbach et al., 2018) an agent is shown to learn a diverse set of distinguishable skills by **maximizing entropy**. These skills can be then used to beetter adapt to new tasks.

Multiobjective reward Several tasks in the real world require accounting for multiple objectives and an agent must learn to reason about them. To that end, many works rely on **scalarization** approaches that combine the different objectives into a weighted reward function. This approach can be hard to tune if there are changes on the individual rewards' scale or their priorities over time. To have a better control over the objectives, (Abdolmaleki et al., 2020) proposes training **individual policies for each objective** and then,

instead of combining rewards in the reward space, combine policies in the distribution space. Another alternative is to train a different **policy per preference over objectives** (Xu et al., 2020; Yang et al., 2019), which leads to dense Pareto-optimal sets of policies that trade the different objectives following the operator's preferences. Finally, **meta learning** methods have also been proposed to learn new preferences in a few-shot fashion (Chen et al., 2019).

System delays DRL experimental setups generally assume negligible delay when acting, observing the new state, or receiving the reward. That might not be the case in the real world. To address this issue, the framework of **Delay-Aware MDPs** was introduced in (Chen et al., 2020) to account for delayed dynamics. A similar idea is proposed in (Derman et al., 2021), where the delayed-Q algorithm leverages a forward dynamics model to predict delayed states. In the context of recommendation systems, the method in (Mann et al., 2018) exploits **intermediate observations/symbols** to mitigate the effect of delays.

Representation In certain environment the challenge lies in encoding all information relevant to the problem or task efficiently, leveraging the sufficient amount of domain knowledge or inductive biases (Hessel et al., 2019). Trade-offs are present, e.g., learning policies from physical state-based features might be more sample-efficient—although not always possible—than learning from pixels (Tassa et al., 2018). The question "what makes a good representation for RL?" is studied in (Singh et al., 2020). A simple approach is to design different representations for the same environment and turn the specific chosen representation into a **hyperparameter** that can be tuned based on the scenario (Kim & Ha, 2020). Different environment encodings can be also combined into **multimodal representations** (e.g., image and sound in video-based environments) (Tsai et al., 2018; Tian et al., 2019). Helpful representations can be also learned, for instance by means of **contrastive learning** frameworks (Wu et al., 2018; Srinivas et al., 2020). Then, (Zhang et al., 2020) proposes learning **invariant representations** by means of lossy autoencoders that capture only task-relevant elements. Finally, representation problems can be also regarded from the perspective of the reward; better reward functions might be devised following **reward shaping** methods (Faust et al., 2019; Chiang et al., 2019).

Transferability Policies should be able to be transferred to different instances of the system and/or environment regardless of their low-level features, without posing a considerable challenge. To that end, **randomization** strategies can be used to increase robustness against transfer (Lee et al., 2019a; Tobin et al., 2017). **Meta RL** methods serve as another way to achieve transferability, by parametrizing specific elements of the DRL framework and using an outer loop learner trained on multiple environments (Oh et al., 2020; Houthooft et al., 2018; Kirsch et al., 2019; Alet et al., 2020). Learning common **invariant latent spaces** could be another approach to consider in some contexts (Gupta et al., 2017).

Long trajectories and credit assignment When trajectories are long and/or rewards are sparse, learning efficient behaviors can be hard; the agent must discover a long sequence of correct actions. **Hierarchical RL** poses a possible solution, by considering a hierarchy of auxiliary tasks with known reward structure in order for the agent to reason at different levels of temporal resolution (Riedmiller et al., 2018; Nachum et al., 2018; Vezhnevets et al., 2017). Other works propose **attention mechanisms** to ease credit assignment over long timescales (Wayne et al., 2018; Hung et al., 2019). Then, the method presented in (Arjona-Medina et al., 2019) tackles the problem by **redistributing reward**, i.e., creating a return-equivalent MDP that redistributes reward more uniformly. Finally, **reward shaping** methods are also studied for this type of contexts (Su et al., 2015; Chiang et al., 2019).

Stochasticity In some occasions, real-world environments can be too stochastic, which might lead to high variance gradient estimates that hamper learning. To make sure the agent is trained over a wide distribution of states, using **data augmentation** strategies and **randomization** are proposed by some authors (Lee et al., 2019a; Tobin et al., 2017). The method presented in (Ghosh et al., 2017) suggests partitioning the initial state distribution and train different policies later to be merged in a **divide-and-conquer** fashion. In highly-stochastic environments, the final policy might be better if the agent does not learn based on the average return but on a **distribution over returns** (Dabney et al., 2018; Bellemare et al., 2017).

Multiagent DRL In many real-world environments (e.g., robot swarms, autonomous cars), a team of agents must align their behavior while acting in a decentralized way (Rashid et al., 2018); leveraging experience from multiple agents is not always straightforward and other challenges such as partial-observability and non-stationarity might also come into play. An extended review on the subject can be found in (Nguyen et al., 2020). To address this challenge, one approach is to have **each agent learn independently** (Tampuu et al., 2017), which decentralizes training but might originate stability problems (Foerster et al., 2017). On the opposite side, (Foerster et al., 2018) explores the framework of **multiple decentralized actors and a single centralized critic**. Inspired by Value Decomposition Networks (Sunehag et al., 2018), the work in (Rashid et al., 2018; Son et al., 2019) proposes **hybrid mechanisms** to combine per-agent Q-functions into a single centralized Q-function. **Multiagent Policy Gradient** algorithms introduce a similar concept focused on continuous spaces (Lowe

et al., 2017; Li et al., 2019), which can be also combined with attention (Iqbal & Sha, 2019).

Counterfactual reasoning The ability to reason about actions not taken and "what-ifs" is necessary in some real-world systems, especially when constraints or risks are hard to capture. This is a relevant problem in healthcare applications (Prasad et al., 2017). This challenge partly overlaps with offline RL, since extrapolation techniques can be useful in some contexts, especially when there is correlation between state-action pairs inside and outside databases (Fujimoto et al., 2019). While some studies might touch on this concrete challenge, we did not find any work specifically focusing on counterfactuals and real-world DRL. Facebook's platform Horizon (Gauci et al., 2018), one of the success stories of real-world DRL, leverages work on Counterfactual Policy Evaluation (Wang et al., 2017) to evaluate policies without deploying them online.

Stability Once deployed, agents should maintain the desired behavior for indefinite time, even when new experience is collected. This challenge has a link with other considerations: online learning and the problem of catastrophic forgetting, autonomous resets ((Ibarz et al., 2021) identifies autonomous resets as one of the specific challenges in the context of robotics), and the general issue of reliability. While this is a challenge directly related to the post-deployment or operation phase, we did not find specific works directly tackling this issue for real-world DRL.

Combined challenges Finally, as pointed out in (Dulac-Arnold et al., 2021), real-world DRL challenges usually do not appear in isolation but combined. The literature specifically addressing multiple challenges simultaneously is scarce. For instance, the work in (Jaderberg et al., 2019) focuses on both multiagent settings and partial observability, although they are commonly related. We have not been able to find works tackling numerous challenges at the same time.

4. Domain-agnostic research's gaps

As seen in Table 1, different strategies have been adopted to address specific challenges of real-world DRL; these can be grouped into thirteen types. Based on this literature review, we deem there is a good understanding of each individual challenge, and the provided references demonstrate that novel methods are able to reach new levels of robustness in the test environments. The review has also allowed us to identify the following five gaps that we believe are important to consider when evaluating the progress of real-world DRL.

1. Bias towards robotics use cases Most of the frameworks, use cases, and test environments are focused on control applications, specifically robotics. There are multiple problems in the real world that consist of optimization, design, or recommendation tasks. In some cases their underlying systems are simpler than highly-actuated robots and thus considering them as additional benchmarks could be beneficial moving forward. Extending the focus to these systems might be an opportunity to achieve new successful deployments.

2. Not enough research on combined challenges The majority of the presented studies focus on one challenge at a time and ignore combined challenges analyses. Real-world environments display multiple challenges simultaneously, often with high degrees of interaction. Combined effects are studied in (Dulac-Arnold et al., 2021); their paper proves a simple interaction of a few challenges can substantially hamper the policy performance. They also provide a benchmark task to study this issue in more depth.

3. Lack of real-world follow-through Most of the studies cited in this work make use of the same test environments: MuJoCo (Todorov et al., 2012), DeepMind Control Suite (Tassa et al., 2018), the Arcade Learning Environment (Bellemare et al., 2013), DeepMind Lab (Beattie et al., 2016), Behaviour Suite for RL (Osband et al., 2019), Alchemy (Wang et al., 2021), Meta-World (Yu et al., 2020b), or PyBullet (Coumans & Bai, 2016). These environments are still simulations; in most cases real-world testing is left as future work. While it is highly difficult to create real-world benchmarks all researchers can use, presenting results on real systems would add value to the community.

4. Little understanding of design tradeoffs Comparisons of different approaches generally tie loosely to the design perspective and tradeoffs. We believe domain-specific operators looking to introduce DRL into their systems might not find obvious which approaches to try and use when designing their models. Currently, there are little efforts to align all research directions alongside that goal.

5. Operation is generally ignored The vast majority of works ignore operation after deployment, which is an essential piece of information for potential domain-specific operators. In that sense, an open question is: what test scenarios and/or Key Performance Indicators (KPIs) should we use to guarantee operability over time and identify possible performance degradations?

5. Domain-specific research perspective

In the previous sections we outlined the contributions and gaps of RL research when it comes to real-world DRL. In general, though, many of the papers proposing DRL to address concrete problems or applications come from domain-specific communities, mainly following the motivations presented in Section 1. In that sense, the adoption is positive. However, in the majority of cases the path from proofs of concept and prototypes to real deployments and

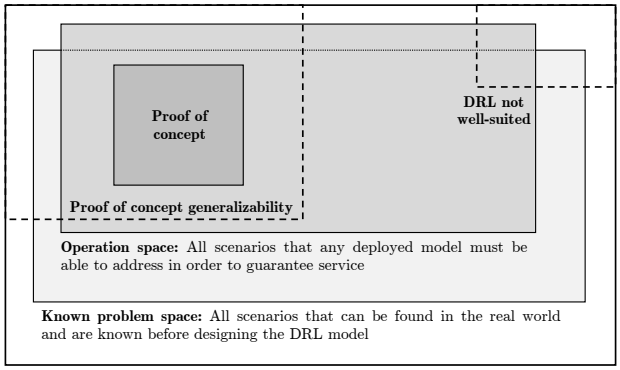

Figure 2. Representation of a real-world problem in terms of a problem space, known problem space, and operation space.

system integrations is still to be traversed. In this section we focus on this issue by, first, reviewing some examples of successful deployments; then, arguing what is missing in other proofs of concept; and finally, outlining possible ways of achieving new deployments that benefit from both RL and domain-specific research.

5.1. Successful deployments in the real world

One of the main roadblocks in real-world DRL is training in simulation before deploying. The *sim-to-real gap* is a well-known problem that many applications face. Still, the work in (OpenAI et al., 2019) proved a robot could learn manipulation skills and solve a Rubik's Cube only from high-quality simulation training. The agent relies on a simple algorithm, PPO (Schulman et al., 2017), to learn the policy. Training in simulation also offered the advantage to easily randomize different environment properties, which helped learning more robustly. (Osinski et al., 2020) and (Tai et al., 2017) are examples that follow the same framework for autonomous driving and robot mapless navigation, respectively. However, the deployment context in both applications is more complex than solving a Rubik's Cube, therefore the authors only limit themselves to very controlled test conditions in real settings. These differences suggest that the specific real-world problem addressed plays an important role in determining the deployability of a certain DRL model. This motivates our discussion in the following parts of this section.

In some cases, agents might be able to train in the real world and not need to rely on simulations. The work in (Haarnoja et al., 2018) focuses on the ability to learn in a real setting, specifically proposing a method for a quadrupedal robot to learn locomotion skills. By maximizing both return and entropy, the robot acquires a stable gait from scratch in about two hours. Entropy maximization might become essential in real-world training as a means to explore more and better.

In many industries, deployment involves integration within a company's operations. A good example of an industrial deployment is Google Loon's DRL-based station-keeping mechanism (Bellemare et al., 2020). The company replaced its previous controller by a DRL agent that is better at keeping balloons close to base stations. A key feature of this example is that the agent's actions do not alter the environment (i.e., wind currents) and therefore the learning process has less interactions to capture.

Another relevant industrial deployment is Facebook's Horizon (Gauci et al., 2018), which the company uses to decide when to send notifications to users. In this latter case, the possibility to gather massive amounts of data from millions of users has made DRL a successful decision-maker. This mirrors the usefulness of self-play in videogame applications such as AlphaZero (Silver et al., 2018) or AlphaStar (Vinyals et al., 2019).

5.2. Missing links between proofs of concept and deployments

We have presented a collection of real-world examples in which DRL is making a difference. Still, the quantity of proofs of concept in the literature substantially outnumbers the quantity of reported deployments. We argue this is due to a combination of different factors inherent to domain-specific research. Throughout this section we use Figure 2 to better understand them.

The premise of the majority of domain-specific papers entails picking a concrete decision-making problem or task of the domain and adapting the DRL framework to address it. In its broadest sense, this problem (e.g., autonomous driving, warehouse management, designing a new drug) can be defined by a *problem space* that captures all possible scenarios that can be encountered in the real world. For example, in the case of autonomous driving, this would correspond to all types of roads, traffic, localizations, etc. Many of these possible scenarios are known beforehand, and therefore constitute the *known problem space*, represented in Figure 2. We also assume there is a space, whose size varies depending on the application, that encapsulates the scenarios that are unknown to model designers and operators until the model is deployed and/or operated in the real world.

Then, almost all real-world tasks are embedded in business frameworks in which quality of service considerations come into play. This originates an *operation space*, which basically defines what a successful deployment is, especially in industrial contexts. This space is not necessarily completely contained within the known problem space. We argue that in the literature, generally, all proofs of concept of a DRL model addressing a real-world problem in a specific domain are based on scenarios within both the operation and known

problem spaces. However, the generalizability of such models is not enough to capture the complete operation space, although it might capture scenarios outside of the operation and the known problem spaces.

The difference between the operation space and the proofs of concept's generalizability is what is slowing the deployment of DRL models in the real world. Operators might be reluctant to integrate DRL models into the systems until this gap is reduced or completely covered for an individual problem. For instance, in the case of the Rubik's Cube addressed in (OpenAI et al., 2019), the operation space is almost certainly covered by the proposed model, resulting in a successful deployment in real life.

5.3. Possible considerations to move forward

The framework presented in Figure 2 is intrinsic to domain-specific research, where methods are outlined within a real problem's context and there is a good understanding of this problem. In contrast, as discussed in Section 4, many of the domain-agnostic studies presented have strong generalizability but lack real-world follow-through, i.e., an operation space as background. When acknowledging both an operation space and the generalizability of proofs of concept, three ideas follow to achieve new deployments: pushing the generalizability boundaries of current proofs of concept, designing additional models that aim to cover remaining areas of the operation space, and considering if certain areas can't be covered by DRL-based methods at all. We weigh in on each of them in the following paragraphs.

A straightforward direction to cover the operation space is to design models that generalize better; this has been a central issue in the RL community. Broadly speaking, the challenges presented in Section 2 hamper the generalizability of DRL models in the real-world, and the mitigation strategies discussed in Section 3 aim to increase it. However, in domain-specific communities success is in many cases measured by the performance on specific scenarios, ignoring generalizability. Hence, model designers tend to follow different routes and tailor their solutions to very specific scenarios in order to surpass other state-of-the-art methods, usually hand-crafting many elements of the DRL framework in the process. While these dynamics increase engagement in DRL, they generalize poorly and hardly fulfill expectations from the operation perspective. Operators likely care more about generalizability and might want to trade it for performance.

Even though pursuing solutions that generalize better is interesting, it is also fair to assume that a single DRL model might not be sufficient to cover the entire operation space, especially if there are certain quality of service requirements in place. Therefore, another idea to consider is implementing additional proofs of concept so that the overall union covers the operation space. The operator would then need to decide which model to use in which scenario. We believe in some cases this approach is easier to implement than attempting to design a model capable of covering the whole operation space. This is especially easier to ponder from domain-specific research, where an operation space is usually acknowledged. However, current design practices focused on tailoring DRL models to very specific scenarios make this approach hard to scale. The cost of covering the operational space of complex problems by implementing multiple proofs of concept could be unaffordable in some cases.

Finally, we should consider those problems in which no DRL model or combination of models is able to cover the entire operation space. There might be certain operational scenarios that are too complex for a DRL agent (see Figure 2); we might want to rely on other types of decision-making approaches in those cases. Acknowledging the possible existence of this space could avoid many futile attempts to achieve deployments solely based on DRL, especially if model designers have already explored pushing generalizability boundaries and adding new models. We believe in some cases DRL will not work in the real world only by itself, but in combination with other decision-making technologies.

6. Conclusion

In this work we have focused on real-world-oriented Deep Reinforcement Learning (DRL) research from both domain-agnostic and domain-specific perspectives. We have offered our view on why there is a lack of real-world deployments of DRL models despite the numerous efforts from different research communities, and identified different directions to move forward. On one hand, we have provided a comprehensive review of the domain-agnostic challenges of real-world DRL and summarized which are the different approaches being taken to address them. Thanks to this review, we have identified five gaps in domain-agnostic research: a bias towards robotics use cases, not enough research on combined challenges, a lack of real-world follow-through, little understanding of the design tradeoffs, and an omission of operation considerations. On the other hand, we have explained the motivations and success stories of domain-specific research when it comes to DRL. Still, the number of deployments is low. We attribute this to a misalignment between the generalizability of proofs of concept in the literature and operation requirements. Finally, we have discussed possible ways to increase the operability and robustness of domain-specific DRL models and how those can benefit from the research on how to mitigate real-world DRL challenges.

References

Abdolmaleki, A., Huang, S., Hasenclever, L., Neunert, M., Song, F., Zambelli, M., Martins, M., Heess, N., Hadsell, R., and Riedmiller, M. A distributional view on multi-objective policy optimization. In *International Conference on Machine Learning*, pp. 11–22. PMLR, 2020.

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International Conference on Machine Learning*, pp. 22–31. PMLR, 2017.

Adamski, I., Adamski, R., Grel, T., Jędrych, A., Kaczmarek, K., and Michalewski, H. Distributed Deep Reinforcement Learning: Learn How to Play Atari Games in 21 minutes. pp. 370–388. 2018. doi: 10.1007/978-3-319-92040-5_19.

Afsar, M. M., Crump, T., and Far, B. Reinforcement learning based recommender systems: A survey. jan 2021. URL http://arxiv.org/abs/2101.06286.

Agarwal, R., Schuurmans, D., and Norouzi, M. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pp. 104–114. PMLR, 2020.

Alet, F., Schneider, M. F., Lozano-Perez, T., and Kaelbling, L. P. Meta-learning curiosity algorithms. mar 2020. URL http://arxiv.org/abs/2003.05325.

Arjona-Medina, J. A., Gillhofer, M., Widrich, M., Unterthiner, T., Brandstetter, J., and Hochreiter, S. Rudder: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, pp. 13566–13577, 2019.

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., TB, D., Muldal, A., Heess, N., and Lillicrap, T. Distributed Distributional Deterministic Policy Gradients. apr 2018. URL http://arxiv.org/abs/1804.08617.

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., Schrittwieser, J., Anderson, K., York, S., Cant, M., Cain, A., Bolton, A., Gaffney, S., King, H., Hassabis, D., Legg, S., and Petersen, S. DeepMind Lab. dec 2016. URL http://arxiv.org/abs/1612.03801.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47:253–279, jun 2013. ISSN 1076-9757. doi: 10.1613/jair.3912.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458. PMLR, 2017.

Bellemare, M. G., Candido, S., Castro, P. S., Gong, J., Machado, M. C., Moitra, S., Ponda, S. S., and Wang, Z. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836): 77–82, dec 2020. ISSN 0028-0836. doi: 10.1038/s41586-020-2939-8.

Berkenkamp, F., Turchetta, M., Schoellig, A. P., and Krause, A. Safe model-based reinforcement learning with stability guarantees. *arXiv preprint arXiv:1705.08551*, 2017.

Bohez, S., Abdolmaleki, A., Neunert, M., Buchli, J., Heess, N., and Hadsell, R. Value constrained model-free continuous control. feb 2019. URL http://arxiv.org/abs/1902.04623.

Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In *Advances in Neural Information Processing Systems*, pp. 8224–8234, 2018.

Cassandra, A. R., Kaelbling, L. P., and Littman, M. L. Acting optimally in partially observable stochastic domains. In *Aaai*, volume 94, pp. 1023–1028, 1994.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient Lifelong Learning with A-GEM. dec 2018. URL http://arxiv.org/abs/1812.00420.

Chen, B., Xu, M., Li, L., and Zhao, D. Delay-Aware Model-Based Reinforcement Learning for Continuous Control. may 2020. URL http://arxiv.org/abs/2005.05440.

Chen, X., Ghadirzadeh, A., Bjorkman, M., and Jensfelt, P. Meta-Learning for Multi-objective Reinforcement Learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 977–983. IEEE, nov 2019. ISBN 978-1-7281-4004-9. doi: 10.1109/IROS40897.2019.8968092.

Chiang, H.-T. L., Faust, A., Fiser, M., and Francis, A. Learning Navigation Behaviors End-to-End With AutoRL. *IEEE Robotics and Automation Letters*, 4(2):2007–2014, apr 2019. ISSN 2377-3766. doi: 10.1109/LRA.2019.2899918.

Chow, Y., Nachum, O., Duenez-Guzman, E., and Ghavamzadeh, M. A Lyapunov-based Approach to Safe Reinforcement Learning. may 2018. URL http://arxiv.org/abs/1805.07708.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pp. 4754–4765, 2018.

Coumans, E. and Bai, Y. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016.

Dabney, W., Ostrovski, G., Silver, D., and Munos, R. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pp. 1096–1105. PMLR, 2018.

Dalal, G., Dvijotham, K., Vecerik, M., Hester, T., Paduraru, C., and Tassa, Y. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

de Wiele, T., Warde-Farley, D., Mnih, A., and Mnih, V. Q-Learning in enormous action spaces via amortized approximate maximization. *arXiv preprint arXiv:2001.08116*, 2020.

Deng, B., Jiang, C., Yao, H., Guo, S., and Zhao, S. The Next Generation Heterogeneous Satellite Communication Networks: Integration of Resource Management and Deep Reinforcement Learning. *IEEE Wireless Communications*, 27(2):105–111, apr 2020. ISSN 1536-1284. doi: 10.1109/MWC.001.1900178.

Derman, E., Mankowitz, D. J., Mann, T. A., and Mannor, S. Soft-Robust Actor-Critic Policy-Gradient. mar 2018. URL http://arxiv.org/abs/1803.04848.

Derman, E., Mankowitz, D., Mann, T., and Mannor, S. A bayesian approach to robust reinforcement learning. In *Uncertainty in Artificial Intelligence*, pp. 648–658. PMLR, 2020.

Derman, E., Dalal, G., and Mannor, S. Acting in Delayed Environments with Non-Stationary Markov Policies. jan 2021. URL http://arxiv.org/abs/2101.11992.

Drori, I., Kharkar, A., Sickinger, W. R., Kates, B., Ma, Q., Ge, S., Dolev, E., Dietrich, B., Williamson, D. P., and Udell, M. Learning to Solve Combinatorial Optimization Problems on Real-World Graphs in Linear Time. jun 2020.

Duan, Y., Andrychowicz, M., Stadie, B., Ho, O. J., Schneider, J., Sutskever, I., Abbeel, P., and Zaremba, W. One-shot imitation learning. In *Advances in neural information processing systems*, pp. 1087–1098, 2017.

Dulac-Arnold, G., Evans, R., van Hasselt, H., Sunehag, P., Lillicrap, T., Hunt, J., Mann, T., Weber, T., Degris, T., and Coppin, B. Deep Reinforcement Learning in Large Discrete Action Spaces. dec 2015. URL http://arxiv.org/abs/1512.07679.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. Challenges of Real-World Reinforcement Learning. *arXiv preprint arXiv:1904.12901*, 2019.

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., and Hester, T. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, apr 2021. ISSN 0885-6125. doi: 10.1007/s10994-021-05961-4. URL https://link.springer.com/10.1007/s10994-021-05961-4.

Elton, D. C., Boukouvalas, Z., Fuge, M. D., and Chung, P. W. Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering*, 4(4):828–849, 2019. ISSN 2058-9689. doi: 10.1039/C9ME00039A.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., and Kavukcuoglu, K. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. feb 2018.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., and Dean, J. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, jan 2019. ISSN 1078-8956. doi: 10.1038/s41591-018-0316-z.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is All You Need: Learning Skills without a Reward Function. feb 2018. URL http://arxiv.org/abs/1802.06070.

Faust, A., Francis, A., and Mehta, D. Evolving Rewards to Automate Reinforcement Learning. may 2019. URL http://arxiv.org/abs/1905.07628.

Ferreira, P. V. R., Paffenroth, R., Wyglinski, A. M., Hackett, T. M., Bilen, S. G., Reinhart, R. C., and Mortensen, D. J. Reinforcement Learning for Satellite Communications: From LEO to Deep Space Operations. *IEEE Communications Magazine*, 57(5):70–75, may 2019. ISSN 0163-6804. doi: 10.1109/MCOM.2019.1800796.

Finn, C., Abbeel, P., and Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. mar 2017. URL http://arxiv.org/abs/1703.03400.

Foerster, J., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H. S., Kohli, P., and Whiteson, S. Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning. feb 2017. URL http://arxiv.org/abs/1702.08887.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Fu, J., Luo, K., and Levine, S. Learning Robust Rewards with Adversarial Inverse Reinforcement Learning. oct 2017. URL http://arxiv.org/abs/1710.11248.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062. PMLR, 2019.

Garcia, J. and Fernández, F. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Garnier, P., Viquerat, J., Rabault, J., Larcher, A., Kuhnle, A., and Hachem, E. A review on deep reinforcement learning for fluid mechanics. *Computers & Fluids*, 225:104973, 2021. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2021.104973.

Gauci, J., Conti, E., Liang, Y., Virochsiri, K., He, Y., Kaden, Z., Narayanan, V., Ye, X., Chen, Z., and Fujimoto, S. Horizon: Facebook's Open Source Applied Reinforcement Learning Platform, nov 2018. URL http://arxiv.org/abs/1811.00260.

Ghosh, D., Singh, A., Rajeswaran, A., Kumar, V., and Levine, S. Divide-and-Conquer Reinforcement Learning. nov 2017. URL http://arxiv.org/abs/1711.09874.

Gupta, A., Devin, C., Liu, Y., Abbeel, P., and Levine, S. Learning invariant feature spaces to transfer skills with reinforcement learning. *arXiv preprint arXiv:1703.02949*, 2017.

Haarnoja, T., Ha, S., Zhou, A., Tan, J., Tucker, G., and Levine, S. Learning to Walk via Deep Reinforcement Learning. dec 2018. URL http://arxiv.org/abs/1812.11103.

Hansen, N., Jangir, R., Sun, Y., Alenyà, G., Abbeel, P., Efros, A. A., Pinto, L., and Wang, X. Self-Supervised Policy Adaptation during Deployment. jul 2020. URL http://arxiv.org/abs/2007.04309.

Hausknecht, M. and Stone, P. Deep Recurrent Q-Learning for Partially Observable MDPs. jul 2015. URL http://arxiv.org/abs/1507.06527.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018. ISBN 9781577358008.

Hessel, M., van Hasselt, H., Modayil, J., and Silver, D. On Inductive Biases in Deep Reinforcement Learning. jul 2019. URL http://arxiv.org/abs/1907.02908.

Horgan, D., Quan, J., Budden, D., Barth-Maron, G., Hessel, M., van Hasselt, H., and Silver, D. Distributed Prioritized Experience Replay. mar 2018. URL http://arxiv.org/abs/1803.00933.

Houthooft, R., Chen, Y., Isola, P., Stadie, B., Wolski, F., Ho, O. J., and Abbeel, P. Evolved policy gradients. In *Advances in Neural Information Processing Systems*, pp. 5400–5409, 2018.

Hung, C.-C., Lillicrap, T., Abramson, J., Wu, Y., Mirza, M., Carnevale, F., Ahuja, A., and Wayne, G. Optimizing agent behavior over long time scales by transporting value. *Nature Communications*, 10(1):5223, dec 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-13073-w.

Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., and Levine, S. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, apr 2021. ISSN 0278-3649. doi: 10.1177/0278364920987859.

Igl, M., Zintgraf, L., Le, T. A., Wood, F., and Whiteson, S. Deep variational reinforcement learning for POMDPs. In *International Conference on Machine Learning*, pp. 2117–2126. PMLR, 2018.

Iqbal, S. and Sha, F. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 2961–2970. PMLR, 2019.

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castañeda, A. G., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A., Sonnerat, N., Green, T., Deason, L., Leibo, J. Z., Silver, D., Hassabis, D., Kavukcuoglu, K., and Graepel, T. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, may 2019. ISSN 0036-8075. doi: 10.1126/science.aau6249.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. MOReL : Model-Based Offline Reinforcement Learning. may 2020. URL http://arxiv.org/abs/2005.05951.

Kim, J. T. and Ha, S. Observation Space Matters: Benchmark and Optimization Algorithm. nov 2020. URL http://arxiv.org/abs/2011.00756.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., and Others. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Kirsch, L., van Steenkiste, S., and Schmidhuber, J. Improving Generalization in Meta Reinforcement Learning using Learned Objectives. oct 2019. URL http://arxiv.org/abs/1910.04098.

Kumar, A., Fu, J., Tucker, G., and Levine, S. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. jun 2019. URL http://arxiv.org/abs/1906.00949.

Lee, K., Lee, K., Shin, J., and Lee, H. Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning. oct 2019a. URL http://arxiv.org/abs/1910.05396.

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019b.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, S., Wu, Y., Cui, X., Dong, H., Fang, F., and Russell, S. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4213–4220, 2019.

Li, Y. Deep Reinforcement Learning: An Overview. *arXiv preprint arXiv:1701.07274*, 2017.

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. jul 2017. URL http://arxiv.org/abs/1707.09835.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Abbeel, O. P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, pp. 6379–6390, 2017.

Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y.-C., and Kim, D. I. Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. *IEEE Communications Surveys & Tutorials*, 21(4):3133–3174, 2019. ISSN 1553-877X. doi: 10.1109/COMST.2019.2916583.

Mallya, A. and Lazebnik, S. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Mankowitz, D. J., Levine, N., Jeong, R., Shi, Y., Kay, J., Abdolmaleki, A., Springenberg, J. T., Mann, T., Hester, T., and Riedmiller, M. Robust Reinforcement Learning for Continuous Control with Model Misspecification. jun 2019. URL http://arxiv.org/abs/1906.07516.

Mann, T. A., Gowal, S., György, A., Jiang, R., Hu, H., Lakshminarayanan, B., and Srinivasan, P. Learning from Delayed Outcomes via Proxies with Applications to Recommender Systems. jul 2018. URL http://arxiv.org/abs/1807.09387.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., and Others. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529–533, 2015.

Nachum, O., Gu, S. S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning. In *Advances in neural information processing systems*, pp. 3303–3313, 2018.

Naeem, M., Rizvi, S. T. H., and Coronato, A. A Gentle Introduction to Reinforcement Learning and its Application in Different Fields. *IEEE Access*, 8:209320–209344, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.3038605. URL https://ieeexplore.ieee.org/document/9261348/.

Nagabandi, A., Finn, C., and Levine, S. Deep Online Learning via Meta-Learning: Continual Adaptation for Model-Based RL. dec 2018. URL http://arxiv.org/abs/1812.07671.

Nguyen, T. T., Nguyen, N. D., and Nahavandi, S. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Transactions on Cybernetics*, 50(9):3826–3839, sep 2020. ISSN 2168-2267. doi: 10.1109/TCYB.2020.2977374.

Oh, J., Hessel, M., Czarnecki, W. M., Xu, Z., van Hasselt, H., Singh, S., and Silver, D. Discovering Reinforcement Learning Algorithms. jul 2020. URL http://arxiv.org/abs/2007.08794.

Olivecrona, M., Blaschke, T., Engkvist, O., and Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48, dec 2017. ISSN 1758-2946. doi: 10.1186/s13321-017-0235-x.

Omidshafiei, S., Pazis, J., Amato, C., How, J. P., and Vian, J. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*, pp. 2681–2690. PMLR, 2017.

OpenAI, Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., Schneider, J., Tezak, N., Tworek, J., Welinder, P., Weng, L., Yuan, Q., Zaremba, W., and Zhang, L. Solving Rubik's Cube with a Robot Hand. oct 2019. URL http://arxiv.org/abs/1910.07113.

Osband, I., Doron, Y., Hessel, M., Aslanides, J., Sezener, E., Saraiva, A., McKinney, K., Lattimore, T., Szepesvari, C., Singh, S., Van Roy, B., Sutton, R., Silver, D., and Van Hasselt, H. Behaviour Suite for Reinforcement Learning. aug 2019. URL http://arxiv.org/abs/1908.03568.

Osinski, B., Jakubowski, A., Ziecina, P., Milos, P., Galias, C., Homoceanu, S., and Michalewski, H. Simulation-Based Reinforcement Learning for Real-World Autonomous Driving. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6411–6418. IEEE, may 2020. ISBN 978-1-7281-7395-5. doi: 10.1109/ICRA40945.2020.9196730.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810. IEEE, may 2018. ISBN 978-1-5386-3081-5. doi: 10.1109/ICRA.2018.8460528.

Polydoros, A. S. and Nalpantidis, L. Survey of Model-Based Reinforcement Learning: Applications on Robotics. *Journal of Intelligent & Robotic Systems*, 86(2):153–173, may 2017. ISSN 0921-0296. doi: 10.1007/s10846-017-0468-y.

Popova, M., Isayev, O., and Tropsha, A. Deep reinforcement learning for de novo drug design. *Science Advances*, 4 (7):eaap7885, jul 2018. ISSN 2375-2548. doi: 10.1126/sciadv.aap7885.

Prasad, N., Cheng, L.-F., Chivers, C., Draugelis, M., and Engelhardt, B. E. A Reinforcement Learning Approach to Weaning of Mechanical Ventilation in Intensive Care Units. apr 2017. URL http://arxiv.org/abs/1704.06300.

Qin, Z., Chen, Y., and Fan, C. Density Constrained Reinforcement Learning, 2021. URL https://openreview.net/forum?id=jMc7DlflrMC.

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. mar 2018. URL http://arxiv.org/abs/1803.11485.

Riedmiller, M., Hafner, R., Lampe, T., Neunert, M., Degrave, J., Van de Wiele, T., Mnih, V., Heess, N., and Springenberg, J. T. Learning by Playing - Solving Sparse Reward Tasks from Scratch. feb 2018. URL http://arxiv.org/abs/1802.10567.

Robine, J., Uelwer, T., and Harmeling, S. Discrete Latent Space World Models for Reinforcement Learning. oct 2020. URL http://arxiv.org/abs/2010.05767.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Schwarz, J., Luketina, J., Czarnecki, W. M., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. Progress & Compress: A scalable framework for continual learning. may 2018. URL http://arxiv.org/abs/1805.06370.

Siegel, N. Y., Springenberg, J. T., Berkenkamp, F., Abdolmaleki, A., Neunert, M., Lampe, T., Hafner, R., Heess, N., and Riedmiller, M. Keep Doing What Worked: Behavioral Modelling Priors for Offline Reinforcement Learning. feb 2020. URL http://arxiv.org/abs/2002.08396.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., and Others. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

Singh, A., Liu, H., Zhou, G., Yu, A., Rhinehart, N., and Levine, S. Parrot: Data-Driven Behavioral Priors for Reinforcement Learning. nov 2020. URL http://arxiv.org/abs/2011.10024.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. may 2019. URL http://arxiv.org/abs/1905.05408.

Srinivas, A., Laskin, M., and Abbeel, P. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. apr 2020. URL http://arxiv.org/abs/2004.04136.

Su, P.-H., Vandyke, D., Gasic, M., Mrksic, N., Wen, T.-H., and Young, S. Reward Shaping with Recurrent Neural Networks for Speeding up On-Line Policy Learning in Spoken Dialogue Systems. aug 2015. URL http://arxiv.org/abs/1508.03391.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V. F., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., and Others. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *AAMAS*, pp. 2085–2087, 2018.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

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

Tai, L., Paolo, G., and Liu, M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 31–36. IEEE, sep 2017. ISBN 978-1-5386-2682-5. doi: 10.1109/IROS.2017.8202134.

Talpaert, V., Sobh, I., Kiran, B. R., Mannion, P., Yogamani, S., El-Sallab, A., and Perez, P. Exploring applications of deep reinforcement learning for real-world autonomous driving systems. jan 2019. URL http://arxiv.org/abs/1901.01536.

Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., Aru, J., and Vicente, R. Multiagent cooperation and competition with deep reinforcement learning. *PLOS ONE*, 12(4):e0172395, apr 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0172395.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T., and Riedmiller, M. DeepMind Control Suite. jan 2018. URL http://arxiv.org/abs/1801.00690.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward Constrained Policy Optimization. may 2018. URL http://arxiv.org/abs/1805.11074.

Tian, Y., Krishnan, D., and Isola, P. Contrastive Multiview Coding. jun 2019.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30. IEEE, sep 2017. ISBN 978-1-5386-2682-5. doi: 10.1109/IROS.2017.8202133.

Todorov, E., Erez, T., and Tassa, Y. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, oct 2012. ISBN 978-1-4673-1736-8. doi: 10.1109/IROS.2012.6386109.

Tsai, Y.-H. H., Liang, P. P., Zadeh, A., Morency, L.-P., and Salakhutdinov, R. Learning Factorized Multimodal Representations. jun 2018. URL http://arxiv.org/abs/1806.06176.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pp. 3540–3549. PMLR, 2017.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., and Others. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, pp. 1–5, 2019.

Wang, J. X., King, M., Porcel, N., Kurth-Nelson, Z., Zhu, T., Deck, C., Choy, P., Cassin, M., Reynolds, M., Song, F., Buttimore, G., Reichert, D. P., Rabinowitz, N., Matthey, L., Hassabis, D., Lerchner, A., and Botvinick, M. Alchemy: A structured task distribution for meta-reinforcement learning. feb 2021. URL http://arxiv.org/abs/2102.02926.

Wang, Y.-X., Agarwal, A., and Dud\ik, M. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning*, pp. 3589–3597. PMLR, 2017.

Warde-Farley, D., Van de Wiele, T., Kulkarni, T., Ionescu, C., Hansen, S., and Mnih, V. Unsupervised Control Through Non-Parametric Discriminative Rewards. nov 2018. URL http://arxiv.org/abs/1811.11359.

Wayne, G., Hung, C.-C., Amos, D., Mirza, M., Ahuja, A., Grabska-Barwinska, A., Rae, J., Mirowski, P., Leibo, J. Z., Santoro, A., Gemici, M., Reynolds, M., Harley, T., Abramson, J., Mohamed, S., Rezende, D., Saxton, D., Cain, A., Hillier, C., Silver, D., Kavukcuoglu, K., Botvinick, M., Hassabis, D., and Lillicrap, T. Unsupervised Predictive Memory in a Goal-Directed Agent. mar 2018. URL http://arxiv.org/abs/1803.10760.

Wu, C., Kreidieh, A., Vinitsky, E., and Bayen, A. M. Emergent behaviors in mixed-autonomy traffic. In *Conference on Robot Learning*, pp. 398–407. PMLR, 2017.

Wu, Y., Tucker, G., and Nachum, O. Behavior Regularized Offline Reinforcement Learning. nov 2019. URL http://arxiv.org/abs/1911.11361.

Wu, Z., Xiong, Y., Yu, S., and Lin, D. Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination. may 2018. URL http://arxiv.org/abs/1805.01978.

Xie, A., Harrison, J., and Finn, C. Deep Reinforcement Learning amidst Lifelong Non-Stationarity. jun 2020. URL http://arxiv.org/abs/2006.10701.

Xu, J., Tian, Y., Ma, P., Rus, D., Sueda, S., and Matusik, W. Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Yang, R., Sun, X., and Narasimhan, K. A Generalized Algorithm for Multi-Objective Reinforcement Learning and Policy Adaptation. aug 2019. URL http://arxiv.org/abs/1908.08342.

Yu, C., Liu, J., and Nemati, S. Reinforcement Learning in Healthcare: A Survey. aug 2019. URL http://arxiv.org/abs/1908.08796.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient Surgery for Multi-Task Learning. jan 2020a. URL http://arxiv.org/abs/2001.06782.

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. In *Conference on Robot Learning*, pp. 1094–1100. PMLR, 2020b.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J., Levine, S., Finn, C., and Ma, T. MOPO: Model-based Offline Policy Optimization. may 2020c. URL http://arxiv.org/abs/2005.13239.

Zahavy, T., Haroush, M., Merlis, N., Mankowitz, D. J., and Mannor, S. Learn what not to learn: Action elimination with deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 3562–3573, 2018.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning Invariant Representations for Reinforcement Learning without Reconstruction. jun 2020. URL http://arxiv.org/abs/2006.10742.

Zhang, S., Yao, L., Sun, A., and Tay, Y. Deep Learning Based Recommender System. *ACM Computing Surveys*, 52(1):1–38, feb 2019. ISSN 0360-0300. doi: 10.1145/3285029.

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. *arXiv preprint arXiv:2004.12570*, 2020.