Guided Reinforcement Learning – A Review and Evaluation for Efficient and Effective Real-World Robotics

Julian Eßer, Nicolas Bach, Christian Jestel, Oliver Urbann, and Sören Kerner.

***Abstract*— Recent successes aside, reinforcement learning still faces significant challenges in its application to the real-world robotics domain. Guiding the learning process with additional knowledge offers a potential solution, thus leveraging the strengths of data- and knowledge-driven approaches. However, this field of research encompasses several disciplines and hence would benefit from a structured overview.**

In this paper, we propose the concept of *guided reinforcement learning* that provides a systematic approach towards accelerating the training process and improving the performance for real-world robotic settings. We introduce a taxonomy that structures guided reinforcement learning approaches and shows how different sources of knowledge can be integrated into the learning pipeline in a practical way. Based upon this, we describe available approaches in this field and quantitatively evaluate their specific impact in terms of efficiency, effectiveness, and sim-to-real transfer within the robotics domain.

I. INTRODUCTION

Reinforcement learning (RL) is a promising approach for solving decision-making problems in a human-like fashion through trial and error interactions with the environment [121]. In recent years, reinforcement learning has demonstrated remarkable progress on a variety of challenging tasks, from classic strategy and real-time computer games [14] to the robotics domain [5]. It has been applied to continuous control problems [74], including legged locomotion [64], [113], [119], robot navigation [43], [23], [54], and dexterous manipulation [98], [17], [52]. These success stories built upon the data-driven trial and error nature of the approach to freely explore the search space.

However, learning control policies in such a way naturally requires many interactions with the environment. This emphasizes the importance of both collecting high quality samples and exploring the search space in a sample-efficient manner. While directly learning on real robots is appealing, it comes along with substantial challenges such as high sample cost, partial observability, and safety constraints [28]. Hence, simulators are often adopted as scalable training environments avoiding safety issues as found in the real world. Training robots in simulation is faster, cheaper and safer, but deploying these policies to the physical robot can fail due to a mismatch between the simulated and real world, also known as reality gap [144].

Combining data- and knowledge-driven approaches in a hybrid fashion can be a potential solution to address these challenges. Von Rueden et al. [126] propose an abstract concept for informed machine learning, where prior knowledge is directly integrated into learning systems. They introduce a taxonomy as classification framework in this field that considers the knowledge source, its representations, and its integration into the machine learning pipeline. Building on this work, hybrid approaches may also be a promising avenue to explore for reinforcement learning in the real-world robotics application domain.

Related to robotics, several lines of research have emerged towards more efficient exploration of the search space and effective policy deployment for real-world systems. For instance, dedicated algorithms have been developed that lead to improved sample-efficiency [4], [10], [116]. Demonstration data has been used to accelerate reinforcement learning approaches [17], [131], [68]. Carefully selecting task-specific state representations, reward functions, and action spaces can improve both time to convergence and performance [86], [119], [82]. Reinforcement learning approaches also can be combined with classical control to learn in state spaces of lower complexity [27], [136]. Finally, integrating knowledge about the learning task structure has been found to improve performance and accelerate convergence [96], [139]. The high variety of approaches resulting from different disciplines impedes a wide-ranging understanding of the state-of-the-art in learning control policies for real-world robotics and highlights the necessity of a structured overview.

Recent surveys provide partial overviews of the field. For example, [142] highlights strategies to improve the sample efficiency in reinforcement learning in a general manner, while [61] focuses its application to the robotics domain. Von Rueden et al. [127] analyze how machine learning and simulation can be combined to a hybrid modeling approach. Another survey [144] puts special emphasis on sim-to-real transfer methods for robotics. Dulac-Arnold et al. [28] outline unique challenges for real-world reinforcement learning. Finally, a recent case study [47] provides valuable hands-on insights for successful real-world policy deployment. Our work supplements the above by providing a systematic overview of integrating knowledge into the reinforcement learning pipeline to increase both efficiency and effectiveness for real-world robotics.

In this work we propose the concept for guided reinforcement learning that provides an intuitive approach to accelerate the training process and improve the performance for real-world robotics settings. We introduce a taxonomy that classifies guided reinforcement learning approaches and shows how different sources of knowledge can be integrated

All authors are with the Lamarr Institute for Machine Learning and Artificial Intelligence and the department of AI and Autonomous Systems at the Fraunhofer Institute for Material Flow and Logistics (IML), Dortmund, Germany. Corresponding author: julian.esser@iml.fraunhofer.de

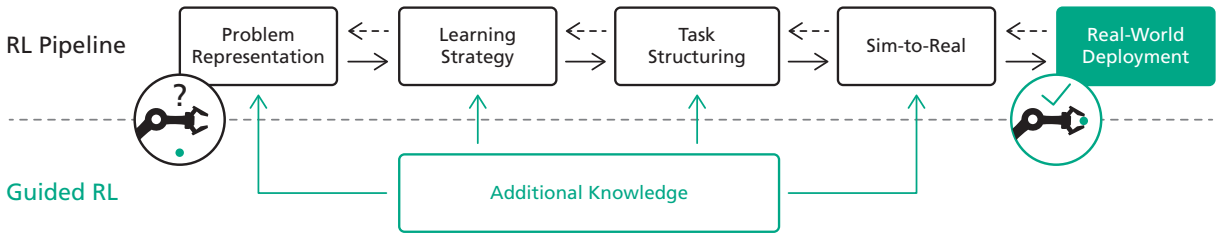


Fig. 1: Pipeline of *guided reinforcement learning*. Adapted from the concept of informed machine learning [126], additional knowledge can be integrated at all levels of the reinforcement learning pipeline to accelerate the training process and improve the performance for real-world robotic settings.

into the learning pipeline in a practical way. Furthermore, we describe available approaches in this field and quantitatively evaluate their specific impact in terms of efficiency, effectiveness, and sim-to-real transfer within the robotics domain.

The paper is structured as follows: In Section II, we introduce our concept of *guided reinforcement learning* and provide a connection to related areas. Section III presents the taxonomy and its central building blocks on a conceptual level. Based on this taxonomy, we classify a large amount of recent research papers in Section IV. Section V presents a quantitative evaluation of the most common methods used in guided reinforcement learning. Finally, we discuss challenges and future directions in Section VI and conclude in Section VII.

II. Concept of Guided Reinforcement Learning

In this section, we present our concept of *guided reinforcement learning* with its definition, the overall goal for efficient and effective real-world robotics deployment, and a link to adjacent research areas.

A. Definition

Guided reinforcement learning describes the integration of additional knowledge into the learning process to accelerate and improve the success for real-world robotics deployment. Figure 1 shows the information flow of guided reinforcement learning. The additional knowledge can be integrated at different stages of the reinforcement learning pipeline: the problem representation, the learning strategy, task structuring, or sim-to-real transfer methods. For a detailed discussion of the pipeline see Section III-C.

B. Efficient and Effective Learning

Accelerating the success for real-world robotics deployment involves both learning in an *efficient* and *effective* manner and forms the central goal of guided reinforcement learning, as shown in Fig. 2. Based on the metrics frequently used in the literature [82], [32], [113], [43], [106], [111], [52], we adopt the following definitions:

Definition II.1 (Efficiency)**.** A training process is considered more *efficient* if it requires fewer interactions with the environment or less time to converge than the baseline.

Definition II.2 (Effectiveness)**.** A training process is considered more *effective* if the performance of a policy in terms of total return or success rate is higher compared to the baseline.

Definition II.3 (Sim-to-Real)**.** A training process is considered as *sim-to-real* if a simulation is adopted for training policies or evaluating methods before real-world deployment.

While efficient and effective policy training as well as real-world robotics deployment form the natural dimensions of guided reinforcement learning, combining these three dimensions constitutes the key motivation. For an in-depth evaluation of available approaches in this direction (see Section V), we finally introduce the following term:

Definition II.4 (Guided RL Compliance)**.** A training process is considered fully *Guided RL compliant* when improvements are achieved across all the three dimensions of efficiency, effectiveness, and sim-to-real.

C. Related Areas

This study focuses on guided reinforcement learning, which integrates prior knowledge directly into the learning pipeline to accelerate the success for real-world robotics. Hence, this review paper is located at the intersections of deep reinforcement learning, robotics, and simulation.

There are several related lines of research, which we do *not* explicitly consider in the context of this study. Selecting model-based or off-policy algorithms has been found to improve the sample-efficiency compared to on-policy algorithms [47]. Also, tuned hyperparameters of the learning algorithm tend to improve the overall policy performance [45]. Furthermore, learning several tasks at once as done in multi-task learning can lead to more efficient training [56]. In the same manner, research effort in the field of meta learning aims at solving unseen tasks fast and efficiently [34]. However, all the methods we distilled in the field of guided reinforcement learning are agnostic with respect to the choice of algorithm (e.g. [117], [41], [74], [89], [32]) and learning task such as locomotion, navigation, or manipulation.

In particular, we are not about to define a strict distinction between efficient, effective, and guided reinforcement learning. Instead, our central motivation is to review existing approaches and distill a structured overview of recent research directions that hopefully strengthens the connection between the RL and robotics communities.

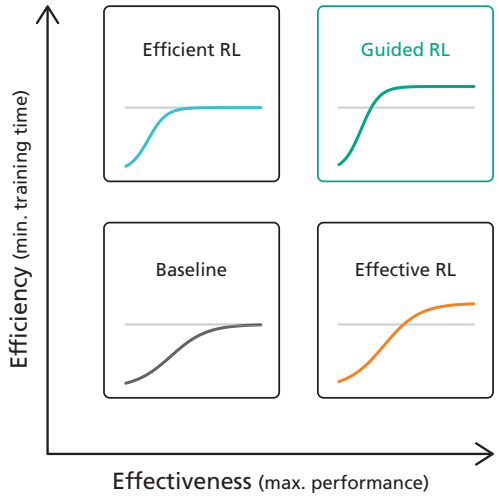


Fig. 2: Overall goal of *guided reinforcement learning*. Guided reinforcement learning approaches aim to simultaneously accelerate the training process and maximize the performance for the real robotic system.

III. TAXONOMY

In this section, we introduce a taxonomy for *guided reinforcement learning* (Fig. 3). Based upon the concept of informed machine learning [126], we structure the taxonomy according to the *knowledge source*, *methodical representation*, and *integration into the pipeline*. Here we introduce the central building blocks of the taxonomy on a conceptual level, while an extensive categorization of approaches will be presented in Section IV.

A. Knowledge Source

Three types of prior knowledge form the basis for most Guided RL methods. These knowledge sources can be roughly categorized into scientific knowledge, world knowledge, and expert knowledge. As detailed by [126], the sources range from formalized to intuitive knowledge and will be briefly described in the following.

1) Scientific Knowledge: is formalized and has its origin e.g. in physics, biology, or engineering. This type of knowledge can be validated through experiments or empirical analysis. Scientific knowledge can be used to develop realistic simulators or to integrate findings from biology into the learning process, for instance.

2) World Knowledge: is either formalized or intuitive and considers facts from e.g. everyday life. Consequently, this knowledge is held by a large group of people. In the context of Guided RL, for instance, world knowledge may be used to design intuitive observation and action spaces, or integrate a natural structure of the learning task.

3) Expert Knowledge: is available to a special group of experienced professionals, with a strong connection to the robotics and RL domains. Such knowledge is rather informal and typically plays a key role in engineering design decisions. For example, expert knowledge is integrated when formalizing a RL problem or may be used to design an overall learning strategy.

B. Guided RL Methods

This category is the key component of our taxonomy, as it connects directly to the RL pipeline (see Fig. 1) and hence the robotic applications. Here we provide a first conceptual overview of these methods, while Section IV provides a detailed description of the most frequent approaches.

1) State Representation (IV-A): describes the observable space for the model, where approaches typically aim to transform or extend the state into more instructive representations.

2) Reward Design (IV-B): includes techniques to induce knowledge by means of designing appropriate dense reward functions or automatic learning approaches.

3) Abstract Learning (IV-C): describes the selection of a task-specific action space for a robotics problem that potentially can be hybridized with model-based approaches.

4) Offline RL (IV-D): focuses on using offline data and tries to efficiently learn policies with RL from recorded training sets.

5) Parallel Learning (IV-E): deals with the parallelization of the algorithmic components while balancing scalabilty and robustness of the learning process.

6) Learning from Demonstration (IV-F): leverages example trajectories, both online and offline and focuses on distilling them into the trained policy.

7) Curriculum Learning (IV-G): is based on the idea of structuring a complex task by iteratively solving simpler tasks with increased levels of difficulty.

8) Hierarchical RL (IV-H): exploits the hierarchical structure underlying the learning task to solve different subtasks or deploy high- and low-level policies.

9) Perfect Simulator (IV-I): aims at building more realistic simulation environments in terms of accurate robot models, physics computation, and environment representation.

10) Domain Randomization (IV-J): strives to policies more robust by highly randomizing the simulation in terms of either visual or dynamics properties.

11) Domain Adaptation (IV-K): approaches typically condition an adaptation module to transfer observations between the simulated and real world, or vice versa.

C. RL Pipeline

From our extensive literature review we find that an applied RL pipeline for real-world robotics can be structured according to four components, namely problem representation, learning strategy, task structuring, and sim-to-real methods (see Fig. 1). Within each of these iterative pipeline steps, additional knowledge can be integrated by means of Guided RL methods.

1) Problem Representation: Representing a real-world robotics problem into the formal description underlying RL, typically requires a large amount of knowledge. A key challenge is to appropriately select observations, define a reward function, and to specify the action space of an agent for a desired learning task. Moreover, choosing suitable

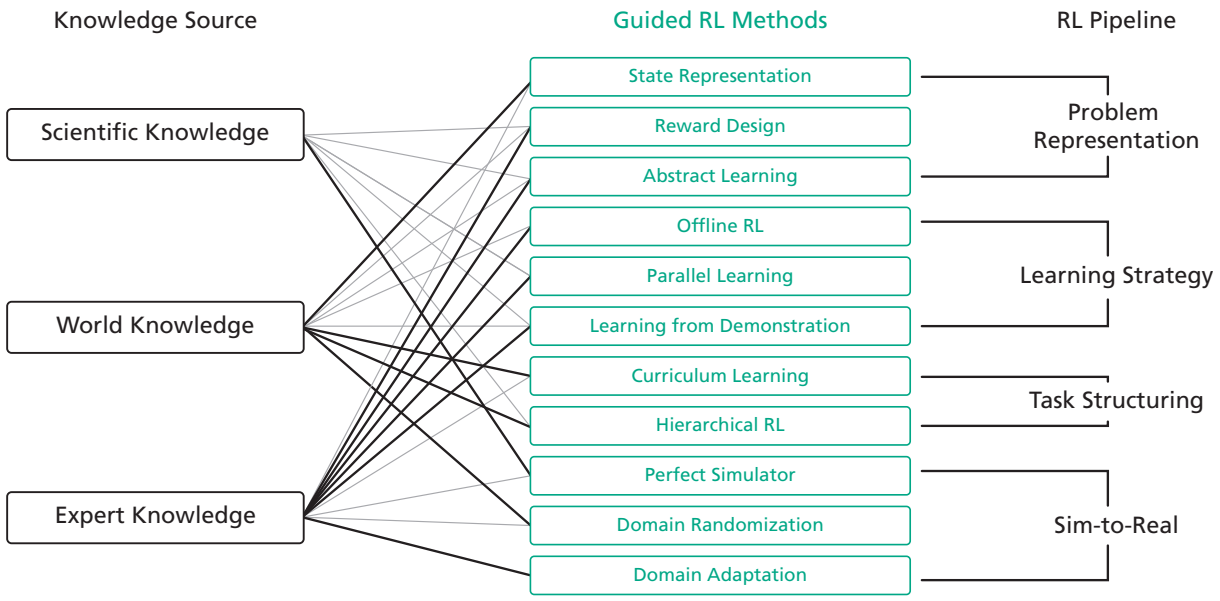


Fig. 3: Taxonomy of *guided reinforcement learning*. Inspired from [126], it structures *guided reinforcement learning* research according to the underlying knowledge source, specific methodical approaches and their integration into the learning pipeline. For each method, the relevance of the knowledge sources is indicated by the thickness of the connecting lines. The goal of this taxonomy is to provide a structured overview for sorting recent research activities (see Table I).

training data requires a careful assessment between real-world and synthetic data.

2) Learning Strategy: Integrating expert knowledge into the learning strategy can be done by deploying parallel learning architectures for a given problem, casting the problem as an online or offline problem, or utilizing real or synthetic demonstration samples.

3) Task Structuring: Depending on the complexity of the real-world robotics problem, further knowledge can be integrated in the sense of meaningfully structuring the learning task. For instance, a complex task could be learned sequentially with increased levels of difficulty or by decomposing it into several subtasks.

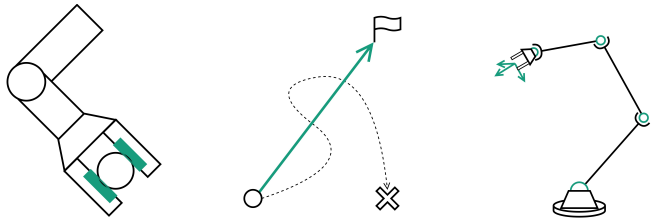
4) Sim-to-Real Methods: Finally, additional knowledge can be used to accelerate the success for real-world robotics deployment by reducing the discrepancies between the simulated and real world. For example, scientific knowledge can be used to tune the simulation environment, or world knowledge by means of domain randomization could robustify the policy training process.

IV. DESCRIPTION OF METHODS

In this section, we provide a detailed overview of the guided reinforcement learning approaches we identified in our literature review. We will structure our description according to the methods of the introduced taxonomy (Section III), since they form the natural connection between the knowledge sources and the practical applications.

A. State Representations (SR)

Choosing the right state representation is an important aspect of solving learning tasks since it defines the observable



(a) State Representation (b) Reward Design (c) Abstract Learning

Fig. 4: Guided RL methods integrated as *problem representation*. (a) Example of an extended state representation with tactile sensor information (see Section IV-A). (b) Using a dense reward function to guide the policy to convergence (see Section IV-B). (c) Abstract learning in different action spaces, e.g. joint or end-effector space (see Section IV-C).

space of an agent. Designing the observation space in a task-specific manner with measurable sensor data can significantly enhance the efficiency of the training process (see Fig. 4a).

Melnik et al. [86] extend the shadow-dexterous-hand with tactile-sensor information and the additional sensor information result led to improved RL agent performance. Some work includes other sensory modalities, such as Church et al. [24] presenting tactile-based RL agents, which learn from tactile information represented as depth images. A zero-shot policy transfer is achieved through a GAN capable of translating real tactile images to simulated depth images. Ning et al. [97] introduce an autonomous robotic ultrasound imaging system where the observation is a concatenated latent vector of two conventional autoencoders. Chen et al. [22]

TABLE I: References classified by *guided reinforcement learning* methods and knowledge sources.

Guided RL Methods	**Source**		
	Scientific Knowledge	World Knowledge	Expert Knowledge
State Representation		[86], [24], [87], [97], [53]	[22], [138], [55]
Reward Design	[119], [38]	[93], [135], [18]	[54], [23], [33], [31]
Abstract Learning	[27]	[106], [107]	[82], [136], [134], [3], [16]
Offline RL		[26]	[39], [133], [1], [116], [140], [20], [63]
Parallel Learning	[48], [114]		[88], [32], [44], [11], [79], [113], [58], [80]
Learning from Demonstration	[7], [35]	[19]	[21], [131], [68], [2], [51], [65]
Curriculum Learning		[111], [83], [36], [118], [78], [29]	[60]
Hierarchical RL	[139], [81]	[103], [95], [66], [94], [71], [129]	
Perfect Simulator	[40], [109], [90], [77]		[141], [46], [42], [137]
Domain Randomization		[84], [108], [98], [124]	[123], [102], [91], [128]
Domain Adaptation			[17], [52], [143], [75], [43], [104], [110], [64]

examine feasibility to convey ambient sounds information about 3D scene structures. Xu et al. [138] create a dataset of 15000 transparent objects and present TransparentNet to estimate depth images despite light refraction and absorption. Lastly, Ji et al. [55] propose a state estimator for quadrupedal locomotion to extend the state representation.

Other work combines present sensor modalities. For instance, Miki et al. [87] introduce a state representation combining proprioception and exteroception for the quadrupedal ANYmal, which enables locomotion on various complex real-world terrain with occasional sensor degradation. Then again, Jangir et al. [53] present a manipulator setup with two cameras, including a transformer-based cross-view attention mechanism to extract correlated features.

B. Reward Design (RD)

Reward design mainly addresses adjusting reward functions in their terms and parameters, as well as automatically learning them from given data.

Reward design, on the one hand, is an effective method to incorporate expert knowledge into RL. For complex tasks, where off-the-shelf RL algorithms typically fail to converge, designing appropriate dense reward functions can lead to increased sample efficiency and performance (see Fig. 4b). Research that incorporated reward shaping in their work are, e.g. Jestel et al. [54], who present a robust policy for multi-robot navigation, which learned emergent behavior in multi-robot scenarios such as swapping, intersections, and constrictions and possess the ability to recover from dead ends. Siekmann et al. [119] designed a parametric reward function for all common bipedal gaits such walking or running that is proven to allow a successful transfer of the policy to the real robot Cassie. Fu et al. [38] propose a bio-inspired reward function for locomotion, which is based on reducing the energy consumption while walking and generates different natural gaits depending on the command velocity. Eteke et al. [33] presents a skill learning framework that learns rewards from very few demonstrations. The rewards are learned by a hidden markov model from deep perceptual features, which leads to better performance than a sparse reward signal. Chiang et al. [23] used AutoRL [100] to automatically apply a reward shaping technique, navigating a mobile robot in long indoor environments.

Reward learning, on the other hand, is an approach often found in human-robot-interaction, where users rate given agent trajectories to learn a reward function as done by Myers et al. [93] introducing a multi-modal reward learning approach where users only need to rank a set of given trajectories. Wilde et al. [135] propose a new feedback mode where users rate trajectories based on a slider bar to get scaled feedback. Cabi et al. [18] propose reward sketching to efficiently gather dense human feedback that can be used to train reward models. Escontrela et al. [31] use a adverserial RL approach [105] adjusting the reward to distill the walking style of a real dog into a robot.

C. Abstract Learning (AL)

Apart from carefully selecting the observation space, a key role in representing the learning problem is given by the choice of the action space (see Fig. 4c). Most approaches that utilize an abstract action space demonstrate improvements in sample efficiency, while some ideas also deploy hybrid learning and model-based approaches. As an introductional read we refer to Varin et al. [125] who give a comparison over classical used action spaces in various manipulation tasks. In manipulation, some work improves over classically used end-effector space. For instance, Martin-Martin et al. [82] who introduce variable impedance control in the end-effector space (VICES) to simplify exploration and improve robustness to disturbances. Wong et al. [136] introduce OSCAR, a data-driven version of Operational Space Control [59] adaptive to changes in the dynamics of a manipulation setting. Bogdanovic et al. [16] propose a policy learning impedance and desired position in joint space and compare this approach to torque control and fixed gain pd controller. Duan et al. [27] propose a task space for bipedal locomotion. The policy learns selecting setpoints for the feet and an inverse dynamics controller transfers these setpoints to joint level control. Other approaches use learned action spaces to alleviate the learning problem, such as Pertsch et al. [106], [107] leveraging offline datasets to learn latent space representations of sequences of actions (skills) along with prior distributions over these skills. On new downstream tasks, they show that the priors can be used to guide policy learning, enabling agents to sample-efficiently solve long-horizon tasks, such as robotic manipulation tasks. Whitney et al. [134] and Allshire et al. [3] propose latent representations

of actions for manipulation, which robustly handle the dynamics of manipulation settings. They show improvements in sample efficiency and performance in pixel-based continuous control environments.

D. Offline RL (OL)

Offline reinforcement learning, also called batch RL, can potentially improve the sample efficiency of other RL approaches detrimentally, as it is a data-driven paradigm that trains policies from offline data. Due to the novelty of the field, researchers are mainly focused on developing algorithms to produce high-performance policies. Offline data sets are often collected from previous training runs of online RL training methods, but could also be preprocessed recordings of real-world sensor data (see Fig. 5a). For further information and oversight over the field of offline RL we refer to [69].

Offline reinforcement learning suffers from the so-called extrapolation-problem, where the policy produces out-of-distribution actions, wrongly overestimated by the value function [39]. There are several algorithms that are robust to this problem, such as Batch-Constrained Deep Q-leaning [39], Random Ensemble Mixture [1], Critic Regularized Regression [133], Implicit Q-learning [63], and MuZero Unplugged [116].

Another approach to offline RL is to leverage techniques developed in machine learning, such as Chen et al. [20] who utilize the transformer architecture conditioned on the desired reward, past states, and action to produce future actions that achieve a desired return. Other work addresses offline data itself. Yarats et al. [140], e.g. propose using reward-free unsupervised data first and then annotating the reward to learn a RL policy.

There are also some efforts to produce large offline data sets, such as Dasari et al. [26] who introduce an open data base to learn models for vision-based robotic manipulation, which consists of 15 million video frames for 7 different robot manipulators.

E. Parallel Learning (PL)

Parallel Learning deals with utilizing one or possibly more heterogeneous hardware resources in the most efficient way by means of parallelization (see Fig. 5b). Furthermore, it addresses how to implement scalability and the necessary robustness to handle different sizes of a learning process.

There are several formulations of parallel learning architectures that were developed throughout the last years, such as A3C [88], IMPALA [32], Ape-X [44], D4PG [11], and R2D2 [58]. All of the mentioned architectures are robust to parallel deployment and show large improvements in sample-efficiency and performance of the final policy over the baselines.

There are also some adjacent optimization paradigms, such as evolutionary strategies [114] that can be scaled to large proportions and are capable of producing competitive policies in comparison to RL-based ones. Other work by Mania et al. [80] proposes using a variant of random search, augmented random search (ARS), that is also very scalable and derives near optimal policies.

Other work makes use of hardware accelerators to permit massive parallelization and thus facilitates the training process [73]. In this vein, Makoviychuk et al. [79] present Isaac Gym, a fully GPU-based simulator for RL that is capable of simulating a high number of environments with a single GPU. Building upon this, Rudin et al. [113] use Isaac Gym to train a quadruped to walk in minutes over progressively complex terrain.

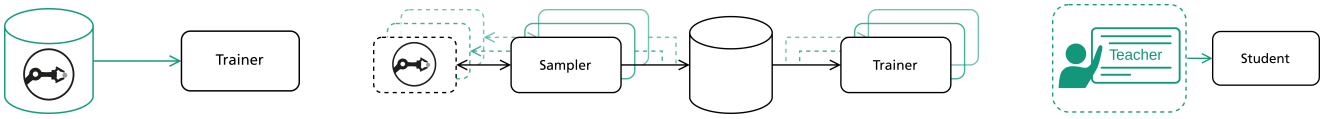
Lastly, the combination of other optimization techniques with RL seems to be a promising approach. For example, Jaderberg et al. [48] present a two-level optimization evolutionary process targeting a population of RL agents. This framework enables agents, conditioned on pixels only, to play a complex 3D game matching human performance.

F. Learning from Demonstration (LfD)

Although learning from demonstration [115] is a research field on its own, many RL approaches make use of such techniques. As an introduction to the field, we refer to Osa et al. [99] and Billard et al. [15]. Aside from standard techniques, such as behavior cloning [9] (learning offline data in a supervised way) or DAgger (train the policy to mimic an expert in an online fashion) [112], novel approaches are presented by Florence et al. [35] proposing to use implicit behavior cloning and let the policy be presented by an energy-based model [67]. Laskey et al. [65] present the DART algorithm that collects demonstrations with injected noise while adjusting the noise level according to the trained policy.

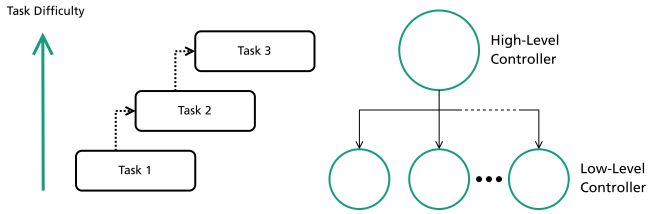
Other approaches first train a teacher policy on unchanging task setups via e.g. RL and distill a policy capable of interpolating between different task setups, such as [7] who choose neural dynamical policies (NDP) [8] to present their teacher and students. Others learn teacher policies on true state information to then derive a student policy conditioned on a reduced or substituted input space, e.g. Chen et al. [21], whose final vision-based policies are able to reorient objects in the shadow hand domain or Lee et al. [68] who also distill a vision-based policy and test it in their RGB-stacking benchmark. Other work makes the use of classical optimization methods to represent the expert and distill their demonstrations into trainable policies, such as Wang et al. [131] who use a hybrid learning process of RL and imitation learning with the OMG planner [130] as expert.

Other than using a teacher, some work utilizes human demonstrations. Akbulut et al. [2] introduce a new framework called ACNMP, combining supervised learning and reinforcement learning to conserve old skills learned from robot demonstrations while being adaptive to new environments. James and Davison [51] present a coarse-to-fine discrete RL algorithm to solve sparse reward manipulation tasks using only a small amount of demonstration and exploration data (work extended by [49], [50]). Celemin et al. [19] include human corrective advice in the action domain through a LfD approach, while an RL algorithm guides the learning process



(a) Offline RL (b) Parallel Learning (c) Learning from Demonstration

Fig. 5: Guided RL methods integrated as *learning strategy*. (a) Offline RL from a recorded dataset (see Section IV-D). (b) Parallel learning with multiple samplers and trainers (see Section IV-E). (c) Learning from Demonstration with distilled student policies (see Section IV-F).



(a) Curriculum Learning (b) Hierarchical RL

Fig. 6: Guided RL methods integrated as *task structuring*. (a) Curriculum learning for tasks of increasing complexity (see Section IV-G). (b) Hierarchical RL with dedicated low-level policies for different subtasks (see Section IV-H).

by filtering out human feedback that does not maximize the reward.

G. Curriculum Learning (CL)

In the context of Reinforcement Learning, curriculum learning [13] provides a framework for increasing the sample efficiency through task structuring, where the policy for a complex task is learned by solving simpler tasks with gradually increased levels of difficulty (see Fig. 6a). This can reduce convergence time on the one hand, but also may help solving problems that are too difficult to learn from scratch [96], [120]. Most approaches rely on either expert knowledge, to gradually increase the difficulty of a target task, or data-driven strategies for automatic curricula generation.

Matiisen et al. [83] introduce a framework for automatic CL called Teacher-Student Curriculum Learning (TSCL), where a teacher automatically chooses appropriate subtasks based on the students progress of learning a complex task. Klink et al. [60] introduce Self-Paced Contextual Reinforcement (SPRL) that gives the agent the freedom to control the intermediate task distribution. Florensa et al. [36] present an approach for reverse curriculum generation, where the robot gradually learns to reach more distant goals starting from goals near the start states. In the same vein, Sharma et al. [118] generate a curriculum of initial states, where the agent learns to reset to generated subgoals based on its performance. Rodriguez and Behnke [111] introduce an approach to learn omnidirectional locomotion for humanoid robots using CL, by gradually increasing the task difficulty with scheduled target velocities.

Finally, some approaches use CL to address complex tasks involving multiple goals or multiple robots. For instance, Luo et al. [78] exploit a curriculum that gradually adjusts the precision requirements for multi-goal reach experiments and show it improves performance in a faster way. Eoh et al. [29], on the other hand, employ a curriculum learning approach for challenging multi-robot object transportation tasks that gradually increases both the transportation distance and number of robots involved. Leyendecker et al. [70] propose a combination of reward-curriculum and domain randomization to develop a robust sim-to-real transferable policy to execute a manipulation task in an industrial setup.

H. Hierarchical RL (HL)

Another way to improve the training efficiency and effectiveness via task structuring is hierarchical RL [12], [101], which is based on the idea of decomposing complex tasks into a hierarchy of subtasks (see Fig. 6b). Typically, these subtasks are addressed by dedicated low-level policies, orchestrated by a more general high-level policy, and thus potentially can be reused in a sample-efficient manner. One standard formulation of hierarchies is introduced by [122] with the option framework, where high-level policies choose options instead of actions, which are presented by closed-loop low-level policies that output actions for a certain amount of time, enabling temporal abstraction. Bacon et al. [6] extends this work with an option formulation of the critic.

Some work formulates hierarchical algorithms, such as Yang et al. [139] who propose hierarchical-deep deterministic policy gradient (h-DDPG) for continuous robotic control tasks, where compound and basic skills are learned simultaneously by two levels of hierarchy. Nachum et al. [95] present a general and data-efficient hierarchical RL algorithm, called HIRO, to learn complex robotic behaviours. Their approach consists of low-level controllers that are supervised with goals generated automatically by high-level controllers.

Others deploy hierarchies in their policies, e.g. Peng et al. [103] introducing a two-level hierarchical control framework for learning a variety of locomotion skills for a physically simulated bipedal robot. Le et al. [66] introduce a hierarchical guidance framework that also effectively leverages expert feedback. Instead of merely giving a subtask decomposition, a high-level expert is deployed to focus the low-level learner on relevant parts of the state space. Margolis et al. [81] address the problem of dynamic locomotion over discontinuous terrain by using a high-level controller

to produce a trajectory based on visual inputs that is then tracked by a low-level controller. Nachum et al. [94] employ a hierarchy to learn low-level goal reaching skills coordinated by a high-level controller for coordinated multi-agent object manipulation. Wang et al. [129] applies a hierarchical policy in a 6D-closed cluttered scene grasping setting that learns an embedding space on expert plans and chooses sampled plans via critic, as well as appropriate options [122] via option classifier. Finally, Li et al. [71] adopt a hierarchical structure for interactive navigation tasks, where a high-level policy generates subgoals and selects low-level policies returning task phase specific robot actions.

I. Perfect Simulator (PS)

One intuitive path towards effective real-world deployment is to build a realistic simulator that minimizes the reality gap (see Fig. 7a). Simulators that accurately capture the real-world physics are appealing since they potentially allow to directly transfer the trained models in a zero-shot fashion into the real-world [144]. Increasing the realism of the simulated environment includes better robot models, physics computation, and environment representations, respectively.
System identification [76] is about building a precise mathematical model of a physical system. In the context of robotics simulation, carefully tuning the physical parameters such as friction, weight, or elasticity can significantly increase the realism of the simulator. Moreover, machine learning approaches can be applied either offline [57] or as presented by Yu et al. in an online fashion by predicting the dynamics model parameters in real-time [141].
Accurately simulating the complex dynamics of modern robots also imposes high demands on the choice of physics engine. Erez et al. [30] analyze quantitative measures of simulation performance and speed related to solving the numerical challenges of multi-body dynamics present in robotics. Besides choosing an appropriate physics engine, the physics simulator has to also support the need of the robotics use-case. As [25] conclude, for each robotics sub-domain different simulators are preferred, depending on the relevance of e.g. sensors, dynamic contacts, or friction modeling. Muratore et al. [90] apply dynamics randomization and use a newly developed algorithm to switch parameters of the DR stopping overfitting to simulator dynamics. Lowrey et al. [77] leverage real world robot data to carefully identified robot parameters [62] enabling an RL trained policy to transfer directly from simulation into reality. Heiden et al. [42] propose a hybrid simulator with learned neural networks that switches between analytical and learned computation of physical effects. Xia et al. [137] present the Gibson environment capable of realistic visual perception for active agents based on real world data.
Finally, an accurate representation of the environment can significantly reduce the reality gap. Ramos et al. [109] present BayesSim, a framework that offers adaptive Bayesian estimates for simulation parameters via simulation-based inference, while Golemo et al. [40] introduce neural-augmented simulation (NAS), a method for augmenting robotic simulators with real robot trajectories. Hwangbo et al. [46] present a neuronal net trained on real data for the robot ANYmal to convert policy action into torque value for simulation model.

J. Domain Randomization (DR)

The idea of domain randomization is to highly randomize the simulation along a wide range of parameter distributions (see Fig. 7b). Instead of carefully modeling the real-world parameters in simulation, the real world simply appears as just another variation of these distributions [144]. Depending on the parameters to be randomized, common approaches deploy either randomization of visual or dynamics components. For a more thorough survey on randomization simulations, the reader is referred to [92].

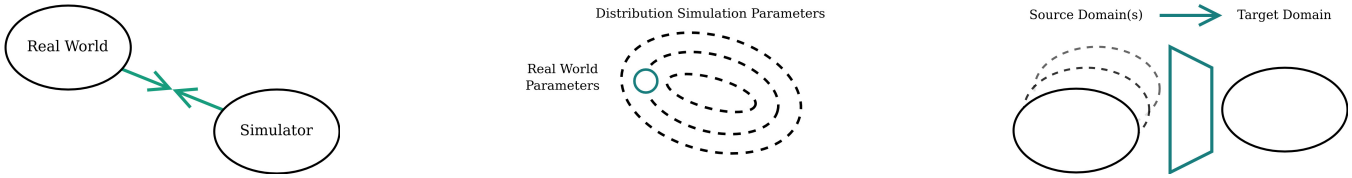
Tobin et al. [123] first introduced the idea of randomizing rendering in the simulator to transfer neural networks to reality for the purpose of robotic control. Mehta et al. [84] propose active domain randomization (ADR), which learns a parameter sampling strategy to leverage randomization ranges that are most informative. OpenAI et al [98] present Automatic Domain Randomization (ADR) which adjusts the domain randomization environment parameters depending on the policy success for solving Rubik's Cube with a real robot hand. Prakash et al. [108] present structured domain randomization (SDR) that creates context-aware synthetic data by taking into account the structure of the scene. Instead of randomizing visual components of the simulator, Peng et al. [102] introduce dynamics randomization that include parameters such as link masses, joint damping, or PD-gains, respectively. Muratore et al. [91] introduce neural posterior domain randomization (NPDR), which adapts the simulator's parameters using only few real-world rollouts to match the observed dynamics. Tsai et al. [124] leverage a single human demonstration to identify the simulator's distribution over dynamics parameters and adapt the domain randomization to reduce the sim-to-real gap.

Ideas similar to visual and dynamics randomization have been adopted in other works, where perturbances are introduced to obtain more robust agents. For example, Wang et al. [128] consider noisy rewards while other recent works apply noisy sensor signals [111], or random external forces [113] for effective policy deployment in the real world.

K. Domain Adaptation (DA)

Domain adaptation techniques aim to minimize the reality gap by training adaptation modules, often represented by autoencoders, capable of projecting one domain into another, e.g. real-world camera images to simulation-look-a-likes (see Fig. 7c). The goal-domain can be either simulated environments, the real world, or abstract latent spaces. Wang et al. [132] presents a comprehensive survey on this field of research.

Domain adaptation based on vision was done by several researchers. Bousmalis et al. [17] implement the former idea via GraspGAN, where an adaptation module



(a) Perfect Simulator (b) Domain Randomization (c) Domain Adaptation

Fig. 7: Guided RL methods integrated for *sim-to-real*. (a) Perfect simulator to minimize the reality gap (see Section IV-I). (b) Domain randomization to match the real world parameter distribution (see Section IV-J, based on [144]). (c) Domain adaptation for matching source and target domains (see Section IV-K).

is trained to convert synthetic images taken from simulation to more photo-realistic observations. James et al. [52] present randomized-to-canonical adaptation networks (RCANs) which learn to project synthetic images derived from randomized simulations into the style of the canonical simulation. Rao et al. [110] present RL-CycleGAN that convert synthetic images into more realistic images. Liu et al. [75] introduce an approach called Real-Sim-Real (RSR) that adapts the real-world state into a simplified one by a segmentation model. Zhang et al. [143] propose adaptation modules, which are trained independent of the DRL agent and can be deployed for different scenarios, e.g. indoor, or outdoor navigation. Hoeller et al. [43] introduce a navigation policy for the quadrupedal robot ANYmal that can navigate in cluttered environments with static and dynamic obstacles.

Other work investigates using an adaption module to handle environmental factors. Peng et al. [104] introduce a framework for training quadrupedal robots to imitate agile locomotion skills from animals, where the learned policies can then be transferred from simulation to the real world through a sample efficient domain adaptation process. Kumar et al. [64] present the rapid motor adaptation (RMA) algorithm that adapts in real-time to unseen real-world scenarios.

V. EVALUATION OF APPROACHES

In this section, we present a systematic evaluation of guided reinforcement learning approaches. As we will show here, combining multiple methods and especially specific combinations lead to improvements in all three Guided RL dimensions, namely efficiency, effectiveness, and sim-to-real transfer. In the following, we first describe the methodical approach and then present the key insights for both individual methods and combinations of those.

A. Methodical Approach

Table II shows an overview of the Guided RL approaches discussed in Section IV. According to the three dimensions of Guided RL (Section II), we identify for each of the discussed dimensions if (i) the overall training time has been reduced (efficiency), if (ii) an improved policy performance has been achieved (effectiveness), or whether (iii) the trained policy has been deployed to the real world (sim-to-real). Specifically, we adopt the achievements claimed by the authors themselves along the three dimensions, verified by means of figures, tables, or specific text passages, respectively. Furthermore, for more in-depth analysis, the last column of the table shows which specific Guided RL methods were used in each of the approaches. Consequently, Table II provides a structured overview of the approaches both in terms of achievements along the three dimensions and used Guided RL methods.

Based upon those classified references, Fig. 8 shows the normalized contribution of the respective methods in terms of efficiency, effectiveness, and sim-to-real. For instance, among the covered papers adopting hierarchical RL, many approaches have shown improvements in terms of policy performance and hence this method seems to contribute significantly towards increasing the effectiveness of the learning approach. To increase the statistical significance of the evaluation not only papers of the corresponding method are considered, but also all papers adopting that method (see column "Guided RL Methods" of Table II).

B. Key Insights on Individual Methods

As the quantitative evaluation of references has shown (Fig. 8), particular methods tend to lead to improvement in terms of efficiency, effectiveness, or sim-to-real. The following key insights can support selecting individual methods to increase the probability of an approach being either more efficient, more effective, or reaching real-world deployment, respectively.

1) Improving the Efficiency: As our findings show, parallel learning architectures, abstract learning, and learning from demonstration data in particular often lead to accelerating the reinforcement learning training process. First, an efficient parallelization of the algorithmic components allows scaling the learning problem to different sizes [88], [32], [44]. Second, simplifying the learning task by means of task-specific action spaces or hybrid model-based and model-free approaches can improve the overall efficiency. Lastly, training based on expert demonstrations tends to be rich in information and hence can accelerate policy training [7], [131], [124]. Furthermore, the efficiency can likely be improved by employing more instructive state representations, applying a curriculum to gradually tackle difficult learning tasks, or utilizing accurate simulation environments.

2) Improving the Effectiveness: In terms of effectiveness, in particular offline RL, hierarchical RL, and curriculum learning seem to have a significant impact on the overall

TABLE II: References classified by *guided reinforcement learning* methods in terms of efficiency, effectiveness, and sim-to-real. Approaches marked with an asterisk achieved simultaneous improvements along all three dimensions.

	Ref	Efficiency	Effectiveness	Sim-to-Real	Guided RL Methods
State Representation	[86]	✓	✓		SR (PL,DR)
	[87]			✓	SR (RD,LfD,CL,DR)
	[24]	✓		✓	SR (PL,DR,DA)
	[22]				SR
	[138]				SR
	[97]			✓	SR
	[53]		✓	✓	SR (DR)
	[55]		✓	✓	SR (CL,DR)
Reward Design	[119]			✓	RD (DR)
	[54]			✓	RD
	[93]	✓		✓	RD
	[23]		✓	✓	RD (CL)
	[33]*	✓	✓	✓	RD (LfD)
	[38]			✓	RD (LfD,DR,DA)
	[135]		✓	✓	RD
	[18]		✓		RD (OL)
	[31]		✓	✓	RD (PL,LfD,DR)
Abstract Learning	[82]*	✓	✓	✓	AL (RD,CL,DR,DA)
	[27]*	✓	✓	✓	AL (RD,PS)
	[106]	✓	✓		AL (LfD)
	[136]		✓		AL (PS,DR)
	[107]	✓	✓		AL (HL,LfD)
	[134]	✓	✓		AL (SR,LfD)
	[3]	✓			AL (LfD)
	[16]		✓	✓	AL (RD)
Offline RL	[39]		✓		OL
	[1]		✓		OL
	[63]	✓	✓		OL
	[133]		✓		OL
	[26]				OL (PS)
	[116]	✓	✓		OL (LfD)
	[140]		✓		OL (LfD)
	[20]		✓		OL
Parallel Learning	[88]	✓	✓		PL
	[32]	✓	✓		PL
	[44]	✓	✓		PL
	[11]		✓		PL
	[58]	✓	✓		PL
	[48]		✓		PL
	[79]	✓			PL (PS,DR)
	[113]	✓		✓	PL (SR,RD,CL,PS,DR)
	[80]	✓	✓		PL
	[114]	✓			PL
Learning from Demonstration	[21]				LfD (SR, PL, CL,DR)
	[7]*	✓	✓	✓	LfD (AL,PL)
	[131]		✓	✓	LfD (SR,AL,PL,DR)
	[35]		✓	✓	LfD
	[68]		✓	✓	LfD (SR,RD,AL,OL,CL,DR)
	[2]*	✓	✓	✓	LfD (SR,AL)
	[51]*	✓	✓	✓	LfD (SR,AL)
	[19]*	✓	✓	✓	LfD (AL)
	[65]*	✓	✓	✓	LfD (AL,DR)
Curriculum Learning	[111]			✓	CL (RD,PS,DR)
	[83]		✓		CL
	[36]		✓		CL
	[78]*	✓	✓	✓	CL
	[118]	✓	✓		CL
	[29]	✓	✓		CL
	[60]*	✓	✓	✓	CL (DR)
	[70]*	✓	✓	✓	CL (RD,AL,PL,PS,DR)
Hierarchical RL	[103]		✓		HL (RD,LfD)
	[139]		✓		HL
	[95]	✓	✓		HL
	[66]	✓	✓		HL (LfD)
	[129]		✓	✓	HL (SR,LfD)
	[81]		✓	✓	HL (RD,AL,PL,LfD,DR)
	[94]		✓	✓	HL (DR)
	[71]		✓		HL (RD)
Perfect Simulator	[141]		✓		PS
	[42]	✓		✓	PS (LfD)
	[77]			✓	PS (PL)
	[40]		✓	✓	PS (DA)
	[109]				PS (DR)
	[90]			✓	PS (DR)
	[46]			✓	PS (DR)
	[137]				PS (DA)
Domain Randomization	[123]			✓	DR
	[84]*	✓	✓	✓	DR
	[108]				DR
	[102]		✓	✓	DR
	[91]			✓	DR
	[98]		✓	✓	DR (PL,LfD,CL)
	[124]		✓	✓	DR (RD)
	[128]		✓		DR
Domain Adaptation	[17]*	✓	✓	✓	DA (DR)
	[52]*	✓	✓	✓	DA (DR)
	[75]*	✓	✓	✓	DA
	[143]		✓	✓	DA
	[43]*	✓	✓	✓	DA (SR,PL,CL,DR)
	[104]			✓	DA (DR)
	[110]*	✓	✓	✓	DA (DR)
	[64]		✓	✓	DA (SR,RD,CL,DR)

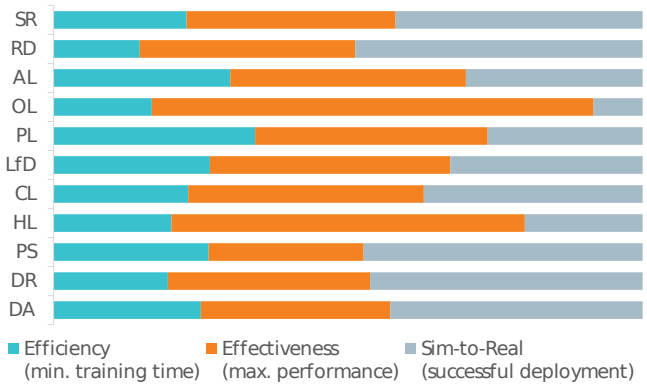


Fig. 8: Evaluation of *guided reinforcement learning* methods. Based on our literature review (see Table II), quantitative results show the relative contribution of the respective methods for accelerating the training process (efficiency), improving the overall policy performance (effectiveness), and successful real-world deployment (sim-to-real).

policy performance. On the one hand, training policies based on recorded data sets can be a valuable path to effectiveness, since the full range of potential information can be extracted from the samples [39], [133], [18]. On the other hand, utilizing curricula or hierarchical learning schemes turn out to be viable approaches to improve the policy performance and to tackle even more complex robotic tasks [111], [113], [83]. Besides this, multiple other methods can contribute to enhance the overall task performance such as meaningfully formulating the overall learning problem, incorporating demonstration data, or deploying parallel learning structures.

3) Improving Sim-to-Real Transfer: As our evaluation results show, domain randomization, domain adaptation, and perfect simulator are methods often deployed for transferring policies trained in simulation to the real-world. First, domain randomization turns out to be a popular method often adopted for successful sim-to-real transfer [84], [102], [113], which is likely due to the simplicity of implementation that can be easily adopted for most robotics problems. Second, domain adaptation in terms of adaptation modules is often employed to successfully transfer between the simulated and the real world, and vice versa [17], [52], [43]. Finally, many approaches strive to reduce the reality gap by improving the realism of the simulators in terms of better robot models, better physics computation, or better environment representation in order to close the sim-to-real gap [141], [30], [64].

C. Key Insights on Guided RL Compliance

While efficient and effective policy training as well as real-world robotics deployment form the natural dimensions of Guided RL, combining these three dimensions constitutes the overall goal (see Fig. 2). For this purpose, we analyze correlations between exactly those papers that were able to achieve improvements in all three dimensions, described in the following as *Guided RL compliant* (Table II, marked with asterisk). Overall, we identify three common patterns among

the Guided RL compliant papers, which simultaneously accelerate the training process (efficiency), improve the policy performance (effectiveness), and transfer the policies to the real-world (sim-to-real).

1) Using Multiple Guided RL Methods: First, we find that the Guided RL compliant papers tend to use a variety of Guided RL methods. For instance, [27], [7], [51] utilize at least three Guided RL approaches, while [82], [70], [43] deploy five or even more approaches to obtain improvements in all three Guided RL dimensions.

2) Combining Particular Guided RL Methods: Second, we note that not only the number of used methods tends to be an important factor, but also that combining *particular* Guided RL methods can improve the probability of simultaneously improving efficiency, effectiveness, and sim-to-real transfer. By analyzing the Pearson correlation coefficients, we observed high positive correlations between DA-DR for sim-to-real transfer, and PL-DR for data-driven scalability. Moreover, our analysis has shown that AL-LfD are often combined by means of reduced action spaces, and RD-PS to incorporate further knowledge for both simulation and reward design.

3) Exploiting Multiple Levels of the Guided RL Pipeline: Finally, we observe that the Guided RL compliant papers also tend to specifically incorporate multiple levels of the Guided RL pipeline (see Fig. 1). For instance, [33], [65], [60] employ Guided RL methods for two or three pipeline stages. Moreover, several of the Guided RL compliant papers [43], [70] even integrate knowledge into all four levels of the Guided RL pipeline to simultaneously accelerate the training process (efficiency), improve the policy performance (effectiveness), and achieve sim-to-real transfer.

VI. DISCUSSION OF CHALLENGES AND DIRECTIONS

In this section, we outline potential challenges and future directions in the field of guided reinforcement learning. We start the discussion by looking at specific approaches first (see Table III) and then distill common challenges and directions.

A. Methodical Challenges & Directions

We summarize our findings of the main approaches of guided reinforcement learning on a high-level in Table III. For each approach, it provides the taxonomy, its main motivation, the central idea, potential challenges, and our perspective on current and future research directions. Details on the methods themselves and corresponding papers can be found in Section IV, while the challenges and directions of these approaches are discussed in more detail in the following along the Guided RL pipeline levels.

1) Problem Formulation: A key challenge with state representation turns out to be balancing the richness of the observable space according to the computational effort [75], [22], [138]. A potential way to mitigate this challenge is to deliberately combine multi-modal sensor information, such as additional tactile sensors for touch information [86], [24]. With reward design, a potential challenge can be selecting reward terms and reward parameters that represent the target task in an accurate way [54], [93]. Potential directions include bio-inspired reward shaping [119], parameter optimization [100], and inverse RL [37] for automatically finding appropriate reward functions. Lastly, selecting an abstracted action space instead of a complex one can largely simplify the training task (e.g., [106], [136]). We see a potential challenge in choosing appropriate levels and methods of abstractions, such as joint space and task space [82]. Two potential directions include latent action spaces [106] and hybrid strategies [27], where model-free RL and model-based approaches are interleaved meaningfully.

2) Learning Strategy: A key challenge with Offline RL seems to be effectively processing the collected training data in order to extract the necessary information [39], [1], [18]. Current and future directions in this domain focus on introducing offline RL data sets [1], [26] or proposing novel algorithms to improve the data usage [133], [39], [1]. With parallel learning, a main challenge is to design the information flow of the components to be parallelized [88], [32], [44]. Directions include improving the robustness of the parallel learning scheme [32], [11] while scaling to larger architectures [88], [44], [11]. In case of learning from demonstration, potential challenges include accurate behavior modeling and extrapolating the behavior to new situations [7], [35], [124]. Current directions include generalizing the behavior beyond specific demonstrations and creating sophisticated benchmarks for evaluating trained policies [2], [21], [68].

3) Task Structuring: Potential challenges with curriculum learning include selecting the right sequence of sub-tasks to be trained and suitable levels of difficulty [36], [113]. One current direction is effective progression of the learning tasks, for example with increased locomotion velocities [111] or challenging terrains [113]. Another direction is to automate the process of task generation, e.g. via multiple competing or cooperating policies [83]. With hierarchical RL on the other hand, a potential challenge includes designing an appropriate hierarchical structure with task-specific responsibilities of the individual policies [139], [95], [66]. Current directions include deploying a hierarchical learning structure to solve more complex and long-horizon tasks [103], [72].

4) Sim-to-Real: Since the realty gap effects all parts of the simulator, potential challenges with perfect simulators include both the realism of robot models and environments as well as the physics computation accuracy [141], [64], [85]. Directions include successful zero-shot transfer without retraining on the real system [64]. Moreover, we see high potential in further integrating real data into the real-to-sim loop [102]. With domain randomization, a challenge often found is determining the most expedient randomization parameters and ranges. A current direction is to leverage randomization ranges that are most informative [84] and provide context for the randomization [108]. Another direction is to improve the robustness of agents by introducing perturbances of e.g. noisy rewards [128] or random exter-

TABLE III: Primary methodological approaches of *guided reinforcement learning*. Based on the introduced taxonomy (Section III), we summarize for each method its key motivation (Section V), the fundamental idea (Section IV) and potential challenges and directions (Section VI).

Taxonomy (See Sec. III)			**Key Motivation**	**Fundamental Idea**	**Potential Challenges**	**Potential Directions**
Pipeline	Method	Source	(See Sec. V)	(See Sec. IV)	(See Sec. VI)	(See Sec. VI)
1. Problem Formulation	State Representation (Sec. IV-A)	World Knowledge	Effectiveness, Efficiency	Employ states with more instructive representations	Balance state richness and computing effort	Combine multi-modal sensor information
	Reward Design (Sec. IV-B)	Expert Knowledge	Effectiveness, Efficiency	Shape or learn dense reward function	Select task-specific reward terms and parameters	Bio-inspired shaping, Inverse RL
	Abstract Learning (Sec. IV-C)	Expert Knowledge	Efficiency Effectiveness	Substitute complex actions spaces by task-specific ones	Choose appropriate levels of abstraction	Hybrid RL & model-based approaches
2. Learning Strategy	Offline RL (Sec. IV-D)	Expert Knowledge	Effectiveness, Efficiency	Learn policies from recorded data set	Process the collected training data effectively	Novel offline RL algorithms
	Parallel Learning (Sec. IV-E)	Expert Knowledge	Efficiency, Effectiveness	Deploy parallelization of the learning algorithm	Design information flow of parallel components	Robust learning along with scalability
	Learning from Demonstration (Sec. IV-F)	Expert Knowledge	Efficiency, Effectiveness	Train a policy based on example trajectories	Accurate behavior modelling and extrapolation	Generalize behavior beyond specific demonstrations
3. Task Structuring	Curriculum Learning (Sec. IV-G)	World Knowledge	Effectiveness, Efficiency	Iteratively solve more complex tasks	Select task sequence and subtask difficulty	Effective difficulty progression, automatic task generation
	Hierarchical RL (Sec. IV-H)	World Knowledge	Effectiveness, Efficiency	Decompose complex task into hierarchy of subtasks	Design appropriate hierarchical structure	More complex tasks, long-horizon tasks
4. Sim-to-Real	Perfect Simulator (Sec. IV-I)	Scientific Knowledge	Sim-to-Real, Efficiency	Build more realistic training environment	Accurately model robots, physics and environment	Successful zero-shot transfer
	Domain Randomization (Sec. IV-J)	World Knowledge	Sim-to-Real, Effectiveness	Randomize visual or dynamics parameters	Determine randomization parameters and ranges	Informative randomization, automatic adjustment
	Domain Adaptation (Sec. IV-K)	Expert Knowledge	Sim-to-Real, Effectiveness	Transfer observations between domains	Select domains & design adaptation module	No overlap between source and target domain

nal forces [113]. Finally, with domain adaptation, potential challenges lie in appropriately selecting the source and target domains, and designing the adaptation module, respectively [17], [52], [143], [43]. Current research directions include the identification of useful source and target domains, as well as alternative generative models to correctly represent the target domains.

B. Common Challenges & Directions

Based upon our evaluation study (Section V) and methodological analysis (Section VI-A), we identify three common challenges in the field of guided reinforcement learning and outline potential directions in the following.

1) Sample-Efficient Policy Training: Learning control policies in a data-driven fashion naturally requires many interactions with the environment and hence constitutes one of the key challenges to accelerate the training process. Specific limitations include the amount and quality of available training data on the one hand, and the ability to efficiently process such data on the other. In particular, parallel learning approaches, task-specific action spaces, and leveraging expert demonstrations represent potential ways to improve the training efficiency.

2) Complex and Long-Horizon Tasks: Another major challenge is to effectively train policies for high performance on complex and long-horizon robotic tasks, e.g. complex object-stacking, combined locomotion & manipulation, and multi-agent scenarios. In particular, such tasks turn out to be challenging since interacting deliberately with objects in the environment requires advanced reasoning capabilities. Potential ways to circumvent this challenge are to train on recorded offline data sets or to deliberately apply task structuring approaches (e.g. curriculum learning).

3) Real-World Robotics Deployment: When simulation environments are adopted for training policies grounded on synthetic training data, bridging the reality gap is a key challenge for successful real-world robotics deployment. As our evaluation has shown, improving the realism of the simulator, randomizing simulation parameters, and training adaptation modules can lead to zero-shot transfer of policies trained solely in simulation. Alternative promising directions include learning directly on the real system, or leveraging real-world data in an offline RL fashion to circumvent the reality gap.

VII. CONCLUSION

In this paper, we presented a taxonomy for integrating different types of knowledge into reinforcement learning to enhance the efficiency and effectiveness for real-world robotics, which we describe using the term *guided reinforcement learning*. Based on a systematic and comprehensive literature review, we presented a description of available approaches in this field. Moreover, we quantitatively evaluated the relations between these and find that (i) using *multiple* methods (ii) combining *particular* methods and (iii) exploiting *various*

methods, can significantly improve the training process. Finally, we hope this conceptual clarification and review of guided reinforcement learning helps other RL and robotics researchers to accelerate the training process and improve the performance for real-world robotic tasks.

ACKNOWLEDGMENT

This research has received funding from the Federal Ministry of Education and Research of Germany as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence (LAMARR22B), DynaFoRo (01IS22074), and from the European Union's Horizon 2020 research and innovation programme (101017151). The authors would like to thank Laura von Rueden and Christian Bauckhage for fruitful discussions on Sections I to III, and in particular also thank the more than 60 corresponding authors on the basis of whose feedback Sections IV to VI have evolved in their present form.

REFERENCES

[1] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 104–114, 2020.

[2] Mete Akbulut, Erhan Oztop, Muhammet Yunus Seker, Hh X, Ahmet Tekden, and Emre Ugur. Acnmp: Skill transfer and task extrapolation through learning from demonstration and reinforcement learning via representation sharing. In *Proceedings of the 2020 Conference on Robot Learning*, pages 1896–1907, 2021.

[3] Arthur Allshire, Roberto Martín-Martín, Charles Lin, Shawn Manuel, Silvio Savarese, and Animesh Garg. Laser: Learning a latent action space for efficient reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6650–6656, 2021.

[4] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. In *Advances in Neural Information Processing Systems*.

[5] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[6] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[7] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Hierarchical neural dynamic policies. *Robotics: Science and Systems*.

[8] Shikhar Bahl, Mustafa Mukadam, Abhinav Gupta, and Deepak Pathak. In *Advances in Neural Information Processing Systems*.

[9] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15, Intelligent Agents [St. Catherine's College, Oxford, July 1995]*, page 103–129, GBR, 1999. Oxford University.

[10] Chayan Banerjee, Zhiyong Chen, and Nasimul Noman. Improved soft actor-critic: Mixing prioritized off-policy samples with on-policy experiences. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–9, 2022.

[11] Gabriel Barth-Maron, Matthew W. Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva TB, Alistair Muldal, Nicolas Heess, and Timothy P. Lillicrap. Distributed distributional deterministic policy gradients. In *6th Int. Conf. on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conf. Track Proc.* OpenReview.net, 2018.

[12] Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13(1):41–77, 2003.

[13] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[14] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[15] Aude G Billard, Sylvain Calinon, and Rüdiger Dillmann. Learning from humans. *Springer handbook of robotics*, pages 1995–2014, 2016.

[16] Miroslav Bogdanovic, Majid Khadiv, and Ludovic Righetti. Learning variable impedance control for contact sensitive tasks. *IEEE Robotics and Automation Letters*, 5(4):6129–6136, 2020.

[17] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, Sergey Levine, and Vincent Vanhoucke. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4243–4250, 2018.

[18] Serkan Cabi, Sergio Gómez Colmenarejo, Alexander Novikov, Ksenia Konyushkova, Scott Reed, Rae Jeong, Konrad Zolna, Yusuf Aytar, David Budden, Mel Vecerik, et al. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *arXiv preprint arXiv:1909.12200*, 2019.

[19] Carlos Celemin, Guilherme Maeda, Javier Ruiz del Solar, Jan Peters, and Jens Kober. Reinforcement learning of motor skills using policy search and human corrective advice. *The International Journal of Robotics Research*, 38(14):1560–1580, 2019.

[20] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. In *Advances in Neural Information Processing Systems*.

[21] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Proceedings of the 5th Conference on Robot Learning*, pages 297–307, 2022.

[22] Ziyang Chen, Xixi Hu, and Andrew Owens. Structure from silence: Learning scene structure from ambient sound. In *Proceedings of the 5th Conference on Robot Learning*, pages 760–772, 2022.

[23] Hao-Tien Lewis Chiang, Aleksandra Faust, Marek Fiser, and Anthony Francis. Learning navigation behaviors end-to-end with autorl. *IEEE Robotics and Automation Letters*, 4(2):2007–2014, 2019.

[24] Alex Church, John Lloyd, raia hadsell, and Nathan F. Lepora. Tactile sim-to-real policy transfer via real-to-sim image translation. In *Proceedings of the 5th Conference on Robot Learning*, pages 1645–1654, 2022.

[25] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021.

[26] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *Proceedings of the Conference on Robot Learning*, pages 885–897, 2020.

[27] Helei Duan, Jeremy Dao, Kevin Green, Taylor Apgar, Alan Fern, and Jonathan Hurst. Learning task space actions for bipedal locomotion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1276–1282, 2021.

[28] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

[29] Gyuho Eoh and Tae-Hyoung Park. Cooperative object transportation using curriculum-based deep reinforcement learning. *Sensors*, 21(14):4780, Jul 2021.

[30] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4397–4404, 2015.

[31] Alejandro Escontrela, Xue Bin Peng, Wenhao Yu, Tingnan Zhang, Atil Iscen, Ken Goldberg, and Pieter Abbeel. Adversarial motion priors make good substitutes for complex reward functions. *arXiv preprint arXiv:2203.15103*, 2022.

[32] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1407–1416, 2018.

[33] Cem Eteke, Doğancan Kebüde, and Barış Akgün. Reward learning

from very few demonstrations. *IEEE Transactions on Robotics*, 37(3):893–904, 2021.

[34] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135, 2017.

[35] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 158–168. PMLR, 08–11 Nov 2022.

[36] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 482–495, 2017.

[37] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

[38] Zipeng Fu, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. In *Proceedings of the 5th Conference on Robot Learning*, pages 928–937, 2022.

[39] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2052–2062, 2019.

[40] Florian Golemo, Adrien Ali Taiga, Aaron Courville, and Pierre-Yves Oudeyer. Sim-to-real transfer with neural-augmented robot simulation. In *Conf. on Robot Learning*, pages 817–828. PMLR, 2018.

[41] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018.

[42] Eric Heiden, David Millard, Erwin Coumans, Yizhou Sheng, and Gaurav S. Sukhatme. Neuralsim: Augmenting differentiable simulators with neural networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9474–9481, 2021.

[43] David Hoeller, Lorenz Wellhausen, Farbod Farshidian, and Marco Hutter. Learning a state representation and navigation in cluttered and dynamic environments. *IEEE Robotics and Automation Letters*, 6(3):5081–5088, 2021.

[44] Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. Distributed prioritized experience replay. In *International Conference on Learning Representations*, 2018.

[45] Jiancong Huang, Juan Rojas, Matthieu Zimmer, Hongmin Wu, Yisheng Guan, and Paul Weng. Hyperparameter auto-tuning in self-supervised robotic learning. *IEEE Robotics and Automation Letters*, 6(2):3537–3544, 2021.

[46] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[47] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.

[48] Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

[49] Stephen James and Pieter Abbeel. Coarse-to-fine q-attention with learned path ranking, 2022.

[50] Stephen James and Pieter Abbeel. Coarse-to-fine q-attention with tree expansion, 2022.

[51] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13739–13748, June 2022.

[52] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[53] Rishabh Jangir, Nicklas Hansen, Sambaran Ghosal, Mohit Jain, and Xiaolong Wang. Look closer: Bridging egocentric and third-person views with transformers for robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):3046–3053, 2022.

[54] Christian Jestel, Harmtmut Surmann, Jonas Stenzel, Oliver Urbann, and Marius Brehler. Obtaining robust control and navigation policies for multi-robot navigation via deep reinforcement learning. In *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*, pages 48–54, 2021.

[55] Gwanghyeon Ji, Juhyeok Mun, Hyeongjun Kim, and Jemin Hwangbo. Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robotics and Automation Letters*, 7(2):4630–4637, 2022.

[56] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *CoRR*, abs/2104.08212, 2021.

[57] Alexander Kanwischer and Oliver Urbann. A machine learning approach to minimization of the sim-to-real gap via precise dynamics modeling of a fast moving robot. In *2022 17th International Conference on Control Automation Robotics Vision (ICARCV)*, to appear.

[58] Steven Kapturowski, Georg Ostrovski, Will Dabney, John Quan, and Remi Munos. Recurrent experience replay in distributed reinforcement learning. In *International Conference on Learning Representations*, 2019.

[59] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[60] Pascal Klink, Hany Abdulsamad, Boris Belousov, and Jan Peters. Self-paced contextual reinforcement learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 513–529. PMLR, 30 Oct–01 Nov 2020.

[61] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[62] Svetoslav Kolev and Emanuel Todorov. Physically consistent state estimation and system identification for contacts. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 1036–1043, 2015.

[63] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

[64] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: rapid motor adaptation for legged robots. In Dylan A. Shell, Marc Toussaint, and M. Ani Hsieh, editors, *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021.

[65] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 143–156, 2017.

[66] Hoang Le, Nan Jiang, Alekh Agarwal, Miroslav Dudik, Yisong Yue, and Hal Daumé, III. Hierarchical imitation and reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2917–2926, 2018.

[67] Yann Lecun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. *A tutorial on energy-based learning*. 2006.

[68] Alex X. Lee, Coline Manon Devin, Yuxiang Zhou, Thomas Lampe, Konstantinos Bousmalis, Jost Tobias Springenberg, Arunkumar Byravan, Abbas Abdolmaleki, Nimrod Gileadi, David Khosid, Claudio Fantacci, Jose Enrique Chen, Akhil Raju, Rae Jeong, Michael Neunert, Antoine Laurens, Stefano Saliceti, Federico Casarini, Martin Riedmiller, raia hadsell, and Francesco Nori. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. In *Proceedings of the 5th Conference on Robot Learning*, pages 1089–1131, 2022.

[69] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline

reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[70] Lars Leyendecker, Markus Schmitz, Hans Aoyang Zhou, Vladimir Samsonov, Marius Rittstieg, and Daniel Lütticke. Deep reinforcement learning for robotic control in high-dexterity assembly tasks - a reward curriculum approach. In *2021 Fifth IEEE International Conference on Robotic Computing (IRC)*, pages 35–42, 2021.

[71] Chengshu Li, Fei Xia, Roberto Martín-Martín, and Silvio Savarese. Hrl4in: Hierarchical reinforcement learning for interactive navigation with mobile manipulators. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 603–616. PMLR, 30 Oct–01 Nov 2020.

[72] Tianyu Li, Nathan Lambert, Roberto Calandra, Franziska Meier, and Akshara Rai. Learning generalizable locomotion skills with hierarchical reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 413–419, 2020.

[73] Jacky Liang, Viktor Makoviychuk, Ankur Handa, Nuttapong Chentanez, Miles Macklin, and Dieter Fox. Gpu-accelerated robotic simulation for distributed reinforcement learning. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 270–282. PMLR, 29–31 Oct 2018.

[74] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *4th Int. Conf. on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conf. Track Proc.*, 2016.

[75] Naijun Liu, Yinghao Cai, Tao Lu, Rui Wang, and Shuo Wang. Real–sim–real transfer for real-world robot control policy learning with deep reinforcement learning. *Applied Sciences*, 10(5), 2020.

[76] Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

[77] Kendall Lowrey, Svetoslav Kolev, Jeremy Dao, Aravind Rajeswaran, and Emanuel Todorov. Reinforcement learning for non-prehensile manipulation: Transfer from simulation to physical system. In *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 35–42, 2018.

[78] Sha Luo, Hamidreza Kasaei, and Lambert Schomaker. Accelerating reinforcement learning for reaching using continuous curriculum learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

[79] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[80] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[81] Gabriel B Margolis, Tao Chen, Kartik Paigwar, Xiang Fu, Donghyun Kim, Sang bae Kim, and Pulkit Agrawal. Learning to jump from pixels. In *Proceedings of the 5th Conference on Robot Learning*, pages 1025–1034, 2022.

[82] Roberto Martín-Martín, Michelle A. Lee, Rachel Gardner, Silvio Savarese, Jeannette Bohg, and Animesh Garg. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1010–1017, 2019.

[83] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740, 2020.

[84] Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J. Pal, and Liam Paull. Active domain randomization. In *Proceedings of the Conference on Robot Learning*, pages 1162–1176, 2020.

[85] Bhairav Mehta, Ankur Handa, Dieter Fox, and Fabio Ramos. A user's guide to calibrating robotic simulators. In *Proceedings of the 2020 Conference on Robot Learning*, pages 1326–1340, 2021.

[86] Andrew Melnik, Luca Lach, Matthias Plappert, Timo Korthals, Robert Haschke, and Helge Ritter. Using tactile sensing to improve the sample efficiency and performance of deep deterministic policy gradients for simulated in-hand manipulation tasks. *Frontiers in Robotics and AI*, 8, 2021.

[87] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[88] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1928–1937, 2016.

[89] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[90] Fabio Muratore, Michael Gienger, and Jan Peters. Assessing transferability from simulation to reality for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4):1172–1183, 2021.

[91] Fabio Muratore, Theo Gruner, Florian Wiese, Boris Belousov, Michael Gienger, and Jan Peters. Neural posterior domain randomization. In *Proceedings of the 5th Conference on Robot Learning*, pages 1532–1542, 2022.

[92] Fabio Muratore, Fabio Ramos, Greg Turk, Wenhao Yu, Michael Gienger, and Jan Peters. Robot learning from randomized simulations: A review. *Frontiers in Robotics and AI*, 9, 2022.

[93] Vivek Myers, Erdem Biyik, Nima Anari, and Dorsa Sadigh. Learning multimodal rewards from rankings. In *Proceedings of the 5th Conference on Robot Learning*, pages 342–352, 2022.

[94] Ofir Nachum, Michael Ahn, Hugo Ponte, Shixiang (Shane) Gu, and Vikash Kumar. Multi-agent manipulation via locomotion using hierarchical sim2real. In *Proceedings of the Conference on Robot Learning*, pages 110–121, 2020.

[95] Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine.

[96] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

[97] Guochen Ning, Xinran Zhang, and Hongen Liao. Autonomic robotic ultrasound imaging system based on reinforcement learning. *IEEE Transactions on Biomedical Engineering*, 68(9):2787–2797, 2021.

[98] OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *CoRR*, abs/1910.07113, 2019.

[99] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[100] Jack Parker-Holder, Raghu Rajan, Xingyou Song, André Biedenkapp, Yingjie Miao, Theresa Eimer, Baohe Zhang, Vu Nguyen, Roberto Calandra, Aleksandra Faust, Frank Hutter, and Marius Lindauer. Automated reinforcement learning (autorl): A survey and open problems. *J. Artif. Int. Res.*, 74, aug 2022.

[101] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Comput. Surv.*, 54(5), jun 2021.

[102] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3803–3810, 2018.

[103] Xue Bin Peng, Glen Berseth, Kangkang Yin, and Michiel Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. 36(4), jul 2017.

[104] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Edward Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. In *Robotics: Science and Systems*, 07 2020.

[105] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo

Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Trans. Graph.*, 40(4), jul 2021.

[106] Karl Pertsch, Youngwoon Lee, and Joseph Lim. Accelerating reinforcement learning with learned skill priors. In *Proceedings of the 2020 Conference on Robot Learning*, pages 188–204, 2021.

[107] Karl Pertsch, Youngwoon Lee, Yue Wu, and Joseph J. Lim. Demonstration-guided reinforcement learning with learned skills. *5th Conf. on Robot Learning*, 2021.

[108] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7249–7255, 2019.

[109] Fabio Ramos, Rafael Possas, and Dieter Fox. Bayessim: Adaptive domain randomization via probabilistic inference for robotics simulators. In Antonio Bicchi, Hadas Kress-Gazit, and Seth Hutchinson, editors, *Robotics: Science and Systems*, 2019.

[110] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[111] Diego Rodriguez and Sven Behnke. Deepwalk: Omnidirectional bipedal gait by deep reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3033–3039, 2021.

[112] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

[113] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, pages 91–100, 2022.

[114] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[115] Stefan Schaal. In *Advances in Neural Information Processing Systems*.

[116] Julian Schrittwieser, Thomas Hubert, Amol Mandhane, Mohammadamin Barekatain, Ioannis Antonoglou, and David Silver. In *Advances in Neural Information Processing Systems*.

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

[118] Archit Sharma, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. In *Advances in Neural Information Processing Systems*.

[119] Jonah Siekmann, Yesh Godse, Alan Fern, and Jonathan Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7309–7315, 2021.

[120] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, pages 1–40, 2022.

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

[122] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999.

[123] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 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)*, pages 23–30. IEEE, 2017.

[124] Ya-Yen Tsai, Hui Xu, Zihan Ding, Chong Zhang, Edward Johns, and Bidan Huang. Droid: Minimizing the reality gap using single-shot human demonstration. *IEEE Robotics and Automation Letters*, 6(2):3168–3175, 2021.

[125] Patrick Varin, Lev Grossman, and Scott Kuindersma. A comparison of action spaces for learning manipulation tasks. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6015–6021, 2019.

[126] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Michal Walczak, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021.

[127] Laura von Rueden, Sebastian Mayer, Rafet Sifa, Christian Bauckhage, and Jochen Garcke. Combining machine learning and simulation to a hybrid modelling approach: Current and future directions. In *Int. Symp. on Intelligent Data Analysis*, pages 548–560. Springer, 2020.

[128] Jingkang Wang, Yang Liu, and Bo Li. Reinforcement learning with perturbed rewards. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6202–6209, Apr. 2020.

[129] Lirui Wang, Xiangyun Meng, Yu Xiang, and Dieter Fox. Hierarchical policies for cluttered-scene grasping with latent plans. *IEEE Robotics and Automation Letters*, 7(2):2883–2890, 2022.

[130] Lirui Wang, Yu Xiang, and Dieter Fox. Manipulation trajectory optimization with online grasp synthesis and selection. In *Robotics: Science and Systems (RSS)*, 2020.

[131] Lirui Wang, Yu Xiang, Wei Yang, Arsalan Mousavian, and Dieter Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *Proceedings of the 5th Conference on Robot Learning*, pages 70–80, 2022.

[132] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[133] Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, and Nando de Freitas. In *Advances in Neural Information Processing Systems*.

[134] William Whitney, Rajat Agarwal, Kyunghyun Cho, and Abhinav Gupta. Dynamics-aware embeddings. In *Int. Conf. on Learning Representations*, 2020.

[135] Nils Wilde, Erdem Biyik, Dorsa Sadigh, and Stephen L. Smith. Learning reward functions from scale feedback. In *Proceedings of the 5th Conference on Robot Learning*, pages 353–362, 2022.

[136] Josiah Wong, Viktor Makoviychuk, Anima Anandkumar, and Yuke Zhu. Oscar: Data-driven operational space control for adaptive and robust robot manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10519–10526, 2022.

[137] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[138] Haoping Xu, Yi Ru Wang, Sagi Eppel, Alan Aspuru-Guzik, Florian Shkurti, and Animesh Garg. Seeing glass: Joint point-cloud and depth completion for transparent objects. In *Proceedings of the 5th Conference on Robot Learning*, pages 827–838, 2022.

[139] Zhaoyang Yang, Kathryn Merrick, Lianwen Jin, and Hussein A. Abbass. Hierarchical deep reinforcement learning for continuous action control. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5174–5184, 2018.

[140] Denis Yarats, David Brandfonbrener, Hao Liu, Michael Laskin, Pieter Abbeel, Alessandro Lazaric, and Lerrel Pinto. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. In *ICLR 2022 Workshop on Generalizable Policy Learning in Physical World*, 2022.

[141] Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017.

[142] Yang Yu. Towards sample efficient reinforcement learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 5739–5743. AAAI Press, 2018.

[143] Jingwei Zhang, Lei Tai, Peng Yun, Yufeng Xiong, Ming Liu, Joschka Boedecker, and Wolfram Burgard. Vr-goggles for robots: Real-to-sim domain adaptation for visual control. *IEEE Robotics and Automation Letters*, 4(2):1148–1155, 2019.

[144] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744, 2020.