The Impact of Task Underspecification in Evaluating Deep Reinforcement Learning

Vindula Jayawardana
MIT
vindula@mit.edu

Catherine Tang
MIT
cattang@mit.edu

Sirui Li
MIT
siruil@mit.edu

Dajiang Suo
MIT
djsuo@mit.edu

Cathy Wu
MIT
cathywu@mit.edu

Abstract

Evaluations of Deep Reinforcement Learning (DRL) methods are an integral part of scientific progress of the field. Beyond designing DRL methods for general intelligence, designing task-specific methods is becoming increasingly prominent for real-world applications. In these settings, the standard evaluation practice involves using a few instances of Markov Decision Processes (MDPs) to represent the task. However, many tasks induce a large family of MDPs owing to variations in the underlying environment, particularly in real-world contexts. For example, in traffic signal control, variations may stem from intersection geometries and traffic flow levels. The select MDP instances may thus inadvertently cause overfitting, lacking the statistical power to draw conclusions about the method's true performance across the family. In this article, we augment DRL evaluations to consider parameterized families of MDPs. We show that in comparison to evaluating DRL methods on select MDP instances, evaluating the MDP family often yields a substantially different relative ranking of methods, casting doubt on what methods should be considered state-of-the-art. We validate this phenomenon in standard control benchmarks and the real-world application of traffic signal control. At the same time, we show that accurately evaluating on an MDP family is nontrivial. Overall, this work identifies new challenges for empirical rigor in reinforcement learning, especially as the outcomes of DRL trickle into downstream decision-making.

1 Introduction

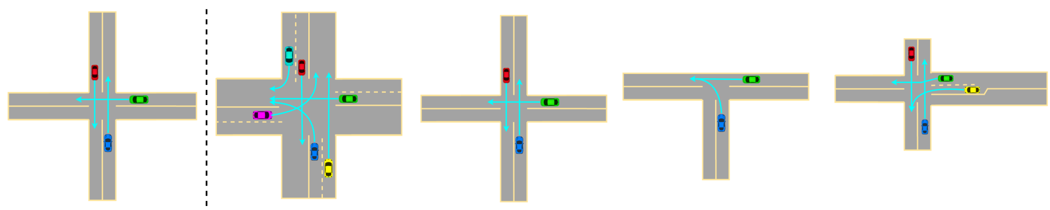
Deep reinforcement learning research has progressed rapidly in recent years, achieving super-human level performance in many applications. At the core of DRL research lies the need to engage in rigorous experimental design for conducting evaluations. The lack of rigorous experimental design could lead to profound implications. Researchers may inadvertently mislead themselves and draw incorrect conclusions about DRL including what factors contribute to the success of a method [17, 22], what factors make the results of a method reproducible [21], or whether a method successfully solves one task [45] or multiple tasks [1, 26]. As evidenced by these findings, strong empirical rigor is crucial for research progress as it allows the research community to confidently assess the current state of the field.

The real world induces many complexities for control tasks, and one major complexity is the existence of multiple instances of the same task. Consider the case of a traffic signal control task where the goal is to design a signal control strategy for an intersection. To reliably claim a DRL method solves the traffic signal control problem, one needs to show that the proposed method sufficiently works

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

for a considerable majority of the signalized intersection instances [34]. We see this requirement of evaluating on a family of instances as an emerging requirement in general, not just limited to traffic signal control, specifically as DRL trickles into real-world applications. We refer to such evaluations as assessing the *algorithmic generalization* of DRL methods *within a task*.

However, many studies that evaluate algorithmic generalization of DRL methods within a task ignore this requirement. In traffic signal control, the use of a few select intersection instances is popular for evaluations [4, 43, 3, 24, 44]. Similar discrepancies can also be seen in other application areas such as in autonomous driving [42, 25], healthcare [46], and in chemistry [48]. Such practices may fail to guard against evaluation overfitting [45], in which the DRL method under evaluation records misleadingly high performance by overspecializing to the evaluation instance. As DRL trickles into real-world implementations and critical use cases, not having such practices can have serious implications, ranging from the opportunity cost of employing lower quality methods to hurting confidence in the field.



(a) Default MDP

(b) MDP Family

Figure 1: An illustrative example of a). a default point MDP vs b). a family of point MDPs specified using some environment parameters ϕ for evaluations of traffic signal control.

To formalize, consider the traffic signal control task, which is typically represented as a single MDP (Figure 1a). At the same time, the task is *underspecified* and could easily be represented by a family of MDPs (Figure 1b), each specified using some environment parameters ϕ such as the number of lanes, turn configurations, and traffic inflow levels. Standard evaluation practice for assessing algorithmic generalization of DRL methods within the task of traffic signal control is to train and evaluate DRL methods on a *point MDP* (or a subset of MDPs) chosen from an implicit family of MDPs (e.g., selecting a specific ϕ). We refer to such evaluation approaches as *point MDP-based evaluations*[1]. Unfortunately, such point MDPs often appear to be selected arbitrarily or in a way that inadvertently simplifies the task. It is well known that the performance of a DRL method is sensitive to the underlying MDP [36, 21, 2]. In this work, **we hypothesize that the coupling of an arbitrary selection of point MDPs for evaluation and the general sensitivity of DRL methods can result in a substantial error in the true performance of DRL methods** over the implicit MDP family of a task. This, in turn, harms empirical rigor in DRL. We refer this phenomenon as the *task underspecification problem* in DRL evaluations.

In this work, we take a deep dive into this matter. Our main contribution is identifying an emerging problem motivated by the real-world application of RL: when a point MDP is used in evaluating DRL methods within a task, the point MDP does not exhibit adequate statistical power to draw conclusions about the corresponding MDP family. We demonstrate that this phenomenon exists not only in real-world applications such as traffic signal control but also in standard control benchmarks, which indicates that it may be prevalent across the field. Our methodology consists of augmenting DRL evaluations with appropriately parameterized families of MDPs. We show that in comparison to applying DRL methods to the point MDP, evaluating the MDP family often yields a substantially different relative ranking of methods, which could lead researchers to draw incorrect conclusions about the performance of some methods over others. We demonstrate this phenomenon experimentally in a case study of traffic signal control. We show that DRL methods which were originally reported as outperforming traditional traffic signal control methods, significantly underperform when a family of MDPs is used for evaluation, causing a substantial change in the ranking of methods.

Unlike standard DRL benchmark suites, in which the suite designer has full control over how many MDP instances are included in the suite [27, 39], in real-world applications, the domain dictates how

[1]We note that *point MDP-based evaluations* are not just limited to MDPs but applicable to all variants of MDP like partially observable MDPs. We use *point MDP* as a general terminology to refer to all such cases.

many point MDPs are in a task. Whereas benchmark suites often have dozens of MDP instances, we estimate that a task can easily have hundreds or thousands of point MDPs. This thus illuminates new computational and reporting obstacles for evaluating DRL methods within a task. We provide some initial studies on the statistical power of evaluating DRL methods by sampling from the MDP family and use of performance profiles for reporting and conclude that it is a nontrivial problem owing to the sensitivity of today's DRL methods.

Important distinctions: To put our contributions into the context of standard DRL practices, in this work we neither address the evaluations of algorithmic generalization across task families (standard DRL evaluations using task suites) nor evaluations of policy generalization within a task family (as addressed in multi-task, robust RL). Second, we focus on problems where a separate individual DRL *model* can be trained for each MDP (*i.e.,* to achieve overall better performance). Another parallel line is in which a single agent is designed to perform well on all MDPs of the family (*e.g.,* multi-task learning in robotics). However, in the scope of this work, we do not consider such problems.

In the following sections, unless otherwise stated, by *evaluation*, we mean evaluations in DRL when evaluating algorithmic generalization of DRL methods within a task.

2 Shortcomings of Point MDP-based Evaluations

We illustrate the shortcomings of point MDP-based evaluation by considering an experiment derived from popular DRL benchmarks. We use three popular control tasks (Pendulum, Quad [23], and Swimmer) as example underspecified tasks. For each task, we augment the nominal task to induce a family of MDPs. A summary of the tasks is given in Table 1, along with their augmentations. For each task family, we then specify five point MDPs. More details of the tasks and their corresponding modifications are given in the Appendix B.1. We consider three popular DRL algorithms (PPO [38], TRPO [37] and TD3 [19]) for evaluation on the tasks. We have verified that each modified point MDP is not under-actuated and that given actuator limits, all are solvable.

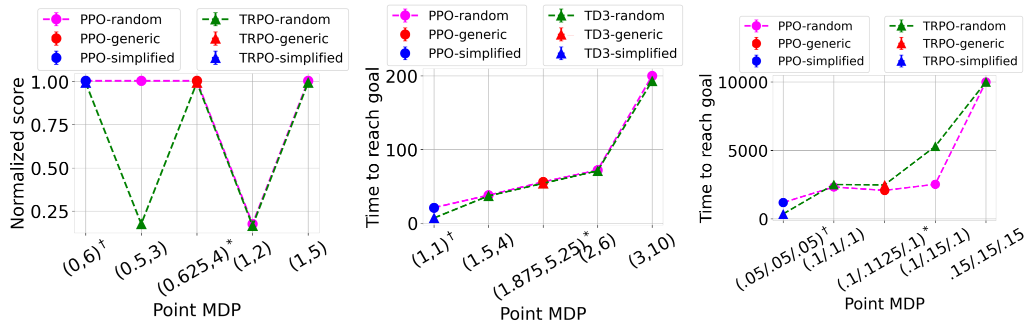
Table 1: Control task and the MDP families.

Task	Task description	MDP family
Quad	• 2D quadcopter in an obstacle course. • **Goal**: maneuver a 2D quadcopter through an obstacle course using its vertical acceleration as the control action.	• MDPs with varying obstacle course lengths (upper obstacle length and lower obstacle length).
Pendulum	• A pendulum that can swing. • **Goal**: swing the pendulum upright.	• MDPs with varying masses and lengths of the pendulum.
Swimmer	• MuJoCo 3-link swimming robot in a viscous fluid. • **Goal**: make the robot swim forward as fast as possible by actuating the two joints.	• MDPs with varying capsule sizes for the segments comprising the robot swimmer.

Observation 1: Reporting evaluations based on point MDPs can be misleading.

Previous works commonly use three criteria for selecting a point MDPs for evaluations. 1) *random MDP*: a random MDP is used to model a somewhat arbitrary selection of an MDPs from the family (*e.g.,* a random selection of a signalized intersection for training a traffic signal control agent [4]) 2) *generic MDP*: an MDP that represents the key characteristics of the MDPs in the family (*e.g.,* the use of a generic cancer progression model for training a chemotherapy designing agent [46]) and 3) *simplified MDP*: an MDP that simplifies the modeling (*e.g.,* zero-error instrument modeling when training an agent for chemical reaction optimization [48]).

In Figure 2, we report the evaluations of the DRL methods on each point MDP described in Table 1 for the three control tasks. That is, each method is trained and evaluated on each individual point MDP. Given an MDP family, all child MDPs are considered possible random MDPs. For each task, $\star$ denotes the generic MDP, and † represents the simplified MDP. The simplified MDP is chosen as the "easiest" MDP, i.e., simplifies the transition dynamics. The generic MDP is the parametric mean MDP of the other four MDPs in the family.

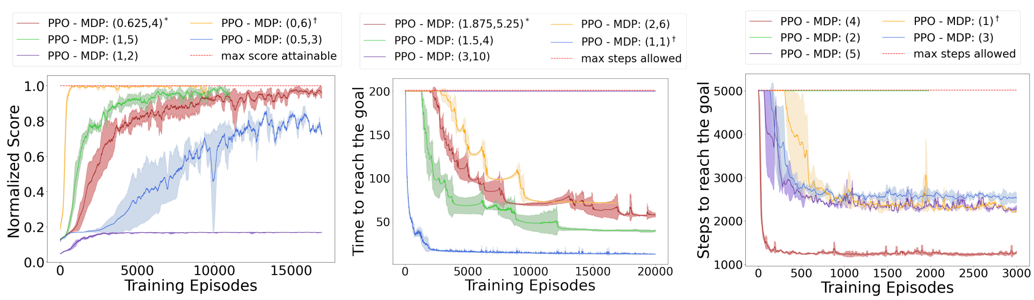


(a) Quad (b) Pendulum (c) Swimmer

Figure 2: Performance comparison of point MDP-based evaluations of the three control tasks. The x-axis represents the five point MDPs. For Quad, performance is a normalized score of the distance the quadcopter traveled before crashing or reaching the goal with respect to the total distance required to travel (higher the better). For the pendulum, it is the time to swing the pendulum upright (lower the better), and for the swimmer, it is the time to reach the goal (lower the better).

We observe a few interesting phenomena (1) the same method yield significant differences in performance under different point MDPs for all three tasks, (2) simplified MDPs generally achieve better performance than other point MDPs in the family, and (3) comparing DRL methods based on point MDPs can provide conflicting conclusions (e.g., conflicting relative performance benefits) based on which point MDP is used for evaluations. For example, for Quad, point MDP (0,6) indicates both PPO and TRPO are equally well-performing. However, under point MDP (0.5,3), we see that PPO outperforms TRPO with approximately 0.75 point difference. These observations highlight the variability of point MDP-based evaluations and the uncertainties involved. Such evaluations could mislead the community to incorrectly conclude that one method is better than another and thereby hinder the scientific progress of the field.

Observation 2: DRL training can be sensitive to the selected point MDP properties. It is generally known that DRL training is sensitive to the underlying MDP. Therefore, selecting a point MDP from a family can demonstrate a training impact that does not generalize to other MDPs in the family. Subsequently, the performance of the DRL method could be overestimated or underestimated.



(a) Quad (b) Pendulum (c) Swimmer

Figure 3: Training progress of each task for different point MDPs. For Quad, performance is a normalized score that indicates the distance the quadcopter traveled before crashing or reaching the goal with respect to the total distance required to travel (higher the better). In the pendulum, the training performance is measured as the time to swing the pendulum upright (lower the better). For the swimmer, the training performance is measured as the time to reach the goal (lower the better).

We demonstrate this phenomenon using our three control tasks in Figure 3 when using the PPO algorithm for training[2]. First, in Quad (Figure 3a), reaching the goal means achieving a normalized

[2]Under some point MDPs, not all parallel runs succeed. For point MDPs where some runs succeed, we only plot the runs that succeed. For point MDPs where all runs fail, we plot them as it is. We fix the number of total training steps to 5M for swimmer and 2M for quad and pendulum. Curves are truncated for better visibility.

score of 1. However, only under a few of the point MDPs, the agent reaches the goal during training. Specifically, 2 out of 5 settings converge to a local minimum (crashing on the same obstacle as training progress). Similarly, in Figure 3b for the pendulum, under one of the MDPs the agent does not achieve the goal (stuck at the 200 steps mark) while under the other four, the agent achieves the goal. Finally, we see similar behavior in Swimmer task in Figure 3c where under one point MDP the agent fails to achieve the goal (stuck at the 5000 steps) during training and succeeds under others. Further examples of training complications under other DRL algorithms are given in Appendix B.3.

We hypothesize one of the root causes of this phenomenon is the complexity differences in point MDPs. For example, in Quad, making the obstacle-free course narrower requires the DRL agent to explore some specific actions different from what it would otherwise need to explore when the course is wider. This effectively makes the training process harder for point MDPs with narrower paths. Another potential cause is MDP designers' over-fitting MDP design to point MDPs. For example, in the pendulum, the default reward function penalizes higher torque values while encouraging reaching the goal with a weighted composite reward. MDP designers may over-fit the weighting values to a selected point MDP which does not generalize to the entire family. This can result in point MDPs with higher pendulum mass failing if the weight on the torque limits is higher.

3 Notation and Formalism

Given a sequential decision-making task T and a reinforcement learning method R to be evaluated, we consider the setting where there are M possible point MDPs in the family of MDPs. In general, despite the illustrative examples in the previous section, M can be quite large. Let N denote a computational budget in terms of the number of models that can be trained[3]. We assume that $M \gg N$; that is, we cannot evaluate the given RL method on all MDPs in the family.

As we demonstrated in Section 2, the performance of a DRL method R can significantly depend on the choice of the point MDP. As suggested by previous work [1], it is therefore warranted to model the performance of R as a real-valued random variable X_R. This means a normalized *performance score* $s_{R,i}$ for a given R and a point MDP i is a realization of the random variable X_R. We normalize point MDP performance scores by linearly rescaling scores based on a given baseline. For example, scores in Atari games are typically normalized with respect to an average human [32, 1].

Given a family of MDPs, a point MDP i may be more important or common than another point MDP j. This is a common requirement in the real world where practitioners have predefined performance requirements, such as the performance of a signalized intersection in an urban area may be more important than a signalized intersection in a sub urban area in traffic signal control [11]. This *importance score* $p_{T,i}$ of a point MDP i on task T can be considered as a realization of a real-valued random variable Y_T. This means depending on the importance scores of each point MDP, a distribution can be generated for the MDP family. For $\tau \in \mathbb{R}^n$ where τ represents the point MDP context, we therefore define the point MDP distribution as $F(\tau) = P(Y_T)$.

Assumption: For a given task T, we assume $p_{T,i}$ is given for each MDP i.

Definition 1 *The overall performance of a DRL method R on task T is defined as $E_R^T = \mathbb{E}[X_R] = \sum_{i=1}^{|U|} s_{R,i} p_{T,i}$ where U is the set of point MDPs.*

However, obtaining E_R^T is not always possible because of the budget constraint $M \gg N$. Therefore, a potential solution is to select a subset V of point MDPs from the MDP family to perform an evaluation. Accordingly, the estimated performance of the method R on task T is $\hat{E}_R^T = \sum_{i=1}^{|V|} s_{R,i} p_{T,i}$. Clearly, if not careful, selecting different subsets can greatly affect the accuracy of the evaluations.

An intuitive approach to selecting a subset of point MDPs is to assess the *contribution* c_i of each point MDP i to overall evaluation. Contribution depends both on the importance score, which is given, and performance score, which incurs a computational cost to evaluate, and can be defined as $c_i = s_{R,i} p_{T,i}$. Thus, we seek to find the set of point MDPs that has the highest contributions to the overall evaluation. Given that we wish to estimate the contribution of individual point MDPs without

[3]Most cloud service providers charge users based on the time they use the services. Therefore, approximating the average number of models that can be trained given a pricing budget can be done with rough estimates of how long it takes to train one model on a single point MDP.

assessing the overall performance E_R^T, this poses a chicken-and-egg problem. In the subsequent sections, we propose approximation techniques that one can employ to identify a subset V based on the approximated contributions.

4 Case Study: Traffic Signal Control

To validate the shortcomings of point MDP-based evaluations, we consider an established benchmark that exhibits a large implicit family of MDPs describing a single task and wherein there could be significant real-world implications. In particular, we consider the evaluations of DRL methods on the traffic signal control task, leveraging the *RESCO* benchmark [4]. We use six algorithms from RESCO in our case study, namely: *IDQN*, *IPPO* [3], *MPLight* [10], *MPLight**, *Fine-tuned Fixed time* and *Max pressure* [41]. Further details can be found in Appendix C.1.

Naturally, traffic signal control should be considered on multiple intersection geometries and vehicle flow levels, hence a family of MDPs. We base our importance score $p_{T,i}$ of each intersection on the frequency of occurrence within a geographic region and the performance score $s_{R,i}$ as the normalized per vehicle average delay. Scores are normalized based on an untuned yet sufficiently performant fixed time controller baseline. The importance scores and the intersections used to build the point MDP distribution of the intersections are taken from Salt Lake City in Utah. We use 164 unique intersections and refer the reader to Appendix C.2 for more details on building this distribution.

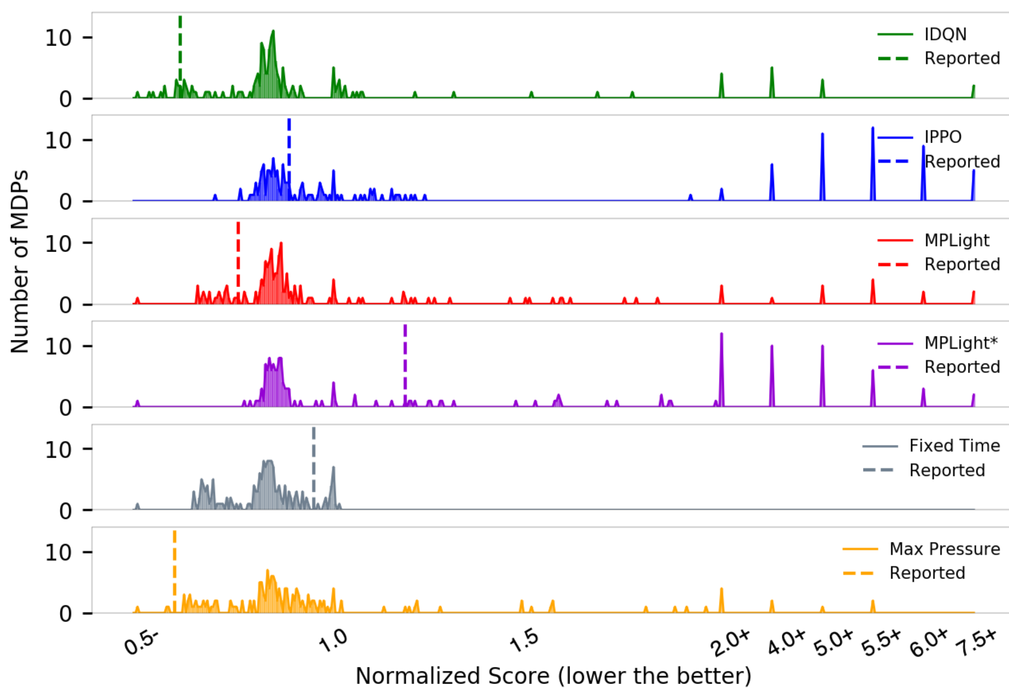


Figure 4: Performance vs the number of point MDPs that demonstrate the performance for all traffic signal control methods in RESCO. 164 unique point MDPs were considered for each method.

In Figure 4, we report the performance of each of six methods on 164 unique point MDPs and the reported performance as per [4][4]. First, we observe significant variations in the performance based on the point MDP used for evaluations. Specifically, all DRL methods demonstrate a significant variation while non-DRL methods demonstrate comparatively low yet considerable variations. Second, we observe that the reported performances in related literature can be significantly biased. As an example, IDQN, IPPO and MPLight performances are clearly overestimated. To quantitatively analyze the potential shortcomings, we denote the performance of each method over the entire MDP family in Figure 5.

[4]Reported performances are based on re-evaluations of the methods on Ingolstadt single intersection.

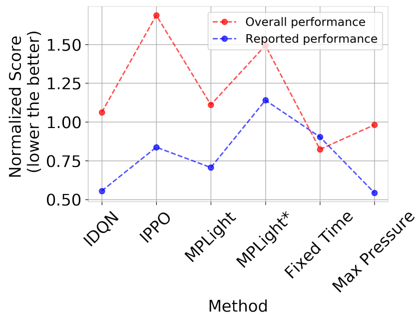


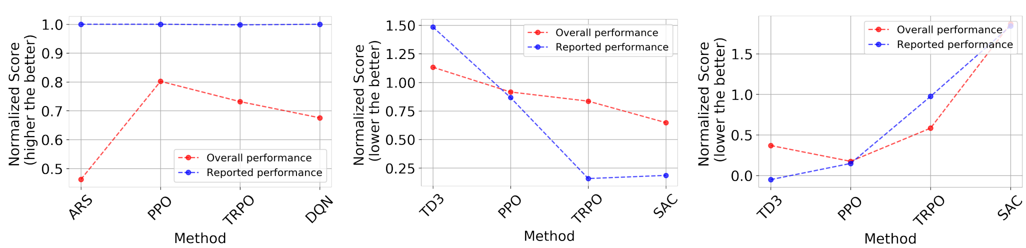
Figure 5: Overall performance (lower the better)

Interestingly, we see a significant result change from previously reported results. Although the Fixed time controller is regarded as an underperforming method as per reported performance in [4], we find that under the MDP family-based evaluations, the non-RL Fixed time controller and Max pressure controller perform significantly better than all the four DRL controllers. We note that fine-tuning a non-RL Fixed time controller is simple enough that it does not pose a computational burden and can be done easily, even on a regular computer. While reported performances ranked IPPO as a well-performing model with a normalized score of 0.84, we see that, in fact, it is the lowest-performing method and that the revised normalized score is as high as 1.7 (even cannot outperform an untuned fixed time controller). It is thus clear that point MDP-based evaluations can be misleading and may pose performance benefits that do not generalize to the MDP family. [5]

5 Further Evidence on Shortcomings of Point MDP-based Evaluations

To further validate the shortcomings of point MDP-based evaluations and the impact, we look at three popular DRL control tasks: cartpole (discrete actions), pendulum (continuous actions), and half-cheetah (continuous actions). For each task, we devise a family of MDPs as described in Table 5 in Appendix D.1. Our MDP families consist of 576, 180, and 120 point MDPs for cartpole, pendulum, and half cheetah, respectively. Due to space limitations, we provide an in-depth analysis of performance variations in Appendix D.2 and only provide a summary of the analysis in this section. In Figure 6, we show the significant discrepancies in overall performance and the reported performances. The reported performance of each task is measured by training and evaluating DRL methods on commonly used single point-MDP given in common benchmark suites.

We see significant result changes when evaluated on the point MDP family. For example, in cartpole (Figure 6a), all reported performances achieve the best score of 1.0 while we see a significantly different overall performance. Although under reported performance, all four DRL methods for cartpole are ranked as equally well-performing, we see an interesting rank change as some methods underperform when considering their overall performance. Similar insights can be seen in Pendulum (Figure 6b) and in half cheetah (Figure 6c).



(a) Cartpole (576 point MDPs) (b) Pendulum (180 point MDPs) (c) Half Cheetah (120 point MDPs)

Figure 6: Discrepancies between reported vs. overall performance of popular DRL methods in cartpole, pendulum and half cheetah when evaluated for algorithmic generalization within the task.

6 Reliable Evaluations Within a Task

In Section 2, 4 and 5, we demonstrate the shortcomings of point MDP-based evaluations. In this section, we discuss the challenges that arise as a result of conducting MDP family-based evaluations. We present three main challenges and an initial set of recommendations to the research community as summarized in Table 2.

[5]Results reported in this work should not be illustrated as evidence against using DRL for traffic signal control and should only be used as evidence of shortcomings in point MDP-based evaluations. Further studies are encouraged to study the benefits of DRL for traffic signal control without the point MDP-based assumptions.

Table 2: Summary of challenges and initial recommendations

Challenge	Our recommendation
Lack of benchmarks	• Create benchmarks that depict MDP families. • Publish datasets of MDP families of control tasks including point-MDP distributions. • Incentivize publication of such datasets and control task at leading conferences.
Large families of MDPs with limited computational budgets	• Adopt performance approximations using clustering and random sampling under a computational budget. • Standardize the evaluations by making the selected point-MDPs public.
Lack of emphasize on all point-MDP performances	• Use performance profiles to show a detailed view of how overall performance changes with point MDPs

6.1 Create data sets of MDP families to benchmark RL methods in evaluations

The use of benchmark datasets for evaluations is well-established in supervised learning. From computer vision [12] and biological data [29] to natural language processing [35], standard datasets are being used for evaluations of deep learning methods. In DRL, the practice is different. In assessing algorithmic generalization of DRL methods within a task, instead of published datasets, researchers use standardized benchmark suites like RESCO for traffic signal control [4], and Vinitsky et al. [42] for mixed autonomy traffic.

However, we recognize that current state-of-the-art benchmarks alone are not sufficient to standardize the evaluations in DRL. The main limitation is that they only provide select point MDPs and do not consider the family of possible MDPs. As many real-world applications inherently demonstrate a family of MDPs, the current DRL standardization of evaluations may steer the research community in a vacuum, while realistically useful DRL methods may get rejected or not even developed.

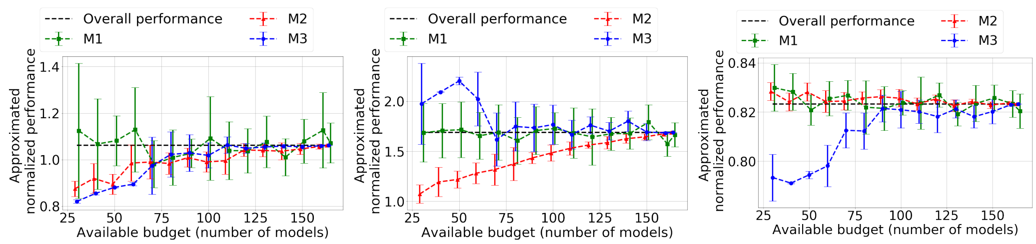
Therefore, our first recommendation is to create DRL benchmark suites which inherently demonstrate a requirement to incorporate a family of MDPs. However, which MDPs to include in a given family is task-dependent, and the expertise of domain experts may be needed. Second, we also encourage the research community to actively create MDP families for existing control tasks (e.g., different signalized intersections as traffic signal control MDP family) and publish them publically. Finally, we also encourage main artificial intelligence conferences with datasets and benchmark tracks like NeurIPS to encourage the community to publish such datasets and tasks and to include necessary check-ins in the paper submission checklists.

6.2 Evaluate DRL methods on a family of MDPs instead of point MDPs

In Section 2, 4 and 5, we demonstrated the shortcomings of point MDP-based evaluations. Therefore, our next recommendation is to encourage researchers to use families of MDPs instead of point MDPs in DRL evaluations. However, evaluating performance over an entire family of MDPs can often be computationally expensive to carry out in practice due to the large cardinality of the family. A solution is to conduct performance approximations. While more sophisticated methods like active learning approaches are possible, we resort to effective yet straightforward techniques to bolster the adoptability of the techniques within a wider community.

We present three techniques: (1) **M1**: random sampling with replacements from the point MDP distribution, (2) **M2**: random sampling without replacements, and (3) **M3**: clustering point MDPs using k-means and assigning probability mass of all point MDPs that belong to same cluster to its centroid. More details of each method and more analysis can be found in Appendix E.1.

In Figure 7, we denote the approximate evaluations of IDQN, IPPO and Fixed time methods in traffic signal control in comparison to the ground-truth performance while varying the budget size. In general, k-means clustering-based approximation produces better estimates than the other two techniques with smaller standard deviations. Specifically, if the budget size is half the total MDP family size, k-means clustering demonstrates reasonably accurate performance estimates. Also,



(a) IDQN (b) IPPO (c) Fixed Time

Figure 7: Approximated performance evaluations over a family of MDPs in traffic signal control with varying budget limits. Closer to the overall performance the better.

publishing the selection of point MDPs as data sets can enable reproducibility and standardization of the evaluations of the tasks. A sensitivity analysis of the proposed techniques to the underlying point MDP distribution is also given in Appendix E.2.

Remarks. We acknowledge that there are other factors to consider for a computational budget, including hyperparameter tuning [26] and training using multiple random seeds within each point MDP [1]. Here, we focus on the computational budget allotted for the task underspecification issue.

6.3 Use performance profile of the MDP family instead of point MDP performance

Inspired by the idea of performance profiling for DRL evaluations [1] and in optimization software [13], our final recommendation is to report the performance of a DRL method over a family of MDPs as a performance profile. Although the overall performance of a DRL method over a family of MDPs can yield more reliable evaluations than evaluating on a point MDP, it may encapsulate further insights into the method's performance. By representing the performance of a method as a performance profile, a more detailed visualization of the performance over the point MDPs can be illustrated.

In Figure 8, we present an example performance profile for k-means clustering-based performance approximation with a budget size of 80 models in the traffic signal control task.

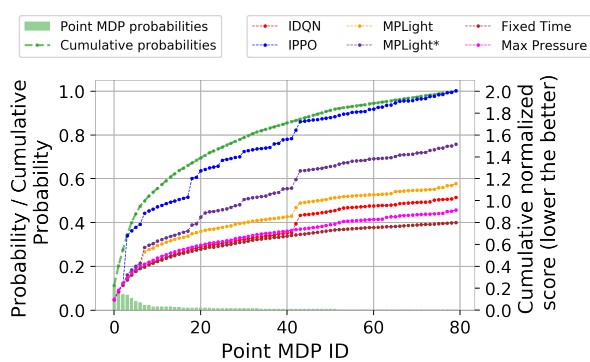


Figure 8: Performance profile of k-means clustering based performance approximation with budget size of 80 models in traffic signal control task.

A performance profile illustrates what point MDPs are most probable in the distribution and how each of the point MDP contributes to the final estimate of the performance. It enables direct comparison of methods. In particular, if the cumulative performance curve of method A is strictly below method B, method A is said to *stochastically dominate* method B (lower the score the better) [1]. Furthermore, point MDPs in which a method seemingly underperforms is easily visible, giving a clear overview of where the limitations and strengths are originating from.

7 Related Work

Deep reinforcement learning algorithms have notoriously high variance, resulting in reliability and reproducibility issues when applying them to real-world applications [1, 26, 9, 8]. Designing sound methodologies for conducting performance evaluations is, therefore, critical. To our knowledge, Falkenauer [18] first identified overfitting a design to a selected problem instance as a cause for concern. Whiteson et al. [45] later used the same motivation to argue similar overfitting can happen, particularly in reinforcement learning. We look at the algorithmic generalization of DRL methods within a task, which poses different requirements and properties compared to previous works.

The use of a family of problem instances is not new. It has been used in combinatorial auctions [30] and in reinforcement learning [7, 28]. Recently, MDP families have been used to achieve better generalization in DRL. Benjamins et al. [6] argue that generalization in DRL is held back by factors stemming in part from a lack of problem formalization. Benchmarks suites such as CARL [5] are proposed to use MDP families to study generalization. Eimer et al. [16] further show that such DRL problems demonstrate challenges that the DRL community has not looked at carefully.

The family of MDPs has been formally modeled in multiple ways in the literature. The most recent and general method is to model the problem as a Contextual Markov Decision Processes (cMDP) [6] where all MDPs in the family share the same MDP configuration except for the transition function and the reward function. In Hidden Parameter MDPs [14], only the transition function changes but keep the reward function fixed over the family of MDPs. Epistemic POMDPs introduced by Ghosh et al. [20] are a special case of a cMDP where the context is assumed to be unobservable. They show that there is implicit partial observability under generalization to unseen test conditions from a limited number of training conditions. This phenomenon translates even a fully observed MDP into POMDP. In comparison to these formulations, we do not restrict what components of an MDP can or should change in evaluations. We let that be defined by the domain of the task.

8 Conclusion and Future Work

In this work, we identify an important yet overlooked issue of task underspecification in DRL evaluations—the reliance of reporting outcomes on select *point* MDPs. We experimentally demonstrate that evaluating the MDP family often yields a substantially different relative ranking of methods compared to evaluating on select MDPs. Moreover, evaluating on a family of MDPs is not trivial and is faced with multiple challenges. One exciting avenue for future work is to explore if similar shortcomings occur when evaluating the algorithmic generalization of DRL methods across tasks. Furthermore, our recommendations for the related challenges when conducting reliable evaluations with a family of MDPs are only a starting point for more focused research. Future research can shed light on these directions, including designing efficient yet effective methods that can produce good approximate performance estimates. Overall, we intend for our findings to raise awareness of task underspecification that impacts the empirical rigor of DRL and aim to help move the needle toward a more disciplined science overall.

9 Broader Impact

Although there is not a single definition of responsible machine learning, the Institute for Ethical AI & Machine Learning has developed a series of eight principles to guide the responsible development of machine learning systems. *Practical accuracy* is one of the principles, which emphasizes that *accuracy and cost metric functions are aligned to the domain-specific applications*. This article contributes to bolstering the practical accuracy of RL, when employed for a complex downstream decision (e.g. whether to adopt an RL method for a societal system), by highlighting some limitations of current standard RL research practices and proposing to explicitly consider a family of MDPs that constitutes the complex decision. On the other hand, even with more empirically rigorous ML practices, there are still subjective aspects of the task distribution (analogous to the family of MDPs in RL), so it remains important to not over-index on RL-based evaluations.

10 Acknowledgments

This work was supported by the MIT Amazon Science Hub, the US DOT's Federal Highway Administration and Utah Department of Transportation under project number F-ST99(783), the MIT-IBM Watson AI Lab, a gift from Mathworks, and the National Science Foundation (NSF) under grant number 2149548. The authors acknowledge MIT SuperCloud and the Lincoln Laboratory Supercomputing Center for providing computational resources supporting the research results in this paper. The authors are also grateful for insightful discussions with Jiaqi Zhang and the constructive suggestions from the anonymous reviewers.

References

[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

[2] Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, et al. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*, 2020.

[3] James Ault, Josiah P. Hanna, and Guni Sharon. Learning an interpretable traffic signal control policy. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, page 88–96, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems.

[4] James Ault and Guni Sharon. Reinforcement learning benchmarks for traffic signal control. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[5] Carolin Benjamins, Theresa Eimer, Frederik Schubert, André Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Lindauer. Carl: A benchmark for contextual and adaptive reinforcement learning. In *NeurIPS 2021 Workshop on Ecological Theory of Reinforcement Learning*, December 2021.

[6] Caroline Benjamins, Theresa Eimer, Frederik Schubert, Aditya Mohan, Andr'e Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Thomas Lindauer. Contextualize me - the case for context in reinforcement learning. *ArXiv*, abs/2202.04500, 2022.

[7] Shalabh Bhatnagar, Mohammad Ghavamzadeh, Mark Lee, and Richard S Sutton. Incremental natural actor-critic algorithms. *Advances in neural information processing systems*, 20, 2007.

[8] Johan Bjorck, Carla P Gomes, and Kilian Q Weinberger. Is high variance unavoidable in rl? a case study in continuous control. *International Conference on Learning Representations*, 2022.

[9] Stephanie CY Chan, Samuel Fishman, John Canny, Anoop Korattikara, and Sergio Guadarrama. Measuring the reliability of reinforcement learning algorithms. *International Conference on Learning Representations*, 2020.

[10] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3414–3421, Apr. 2020.

[11] Janusz Chodur, Krzysztof Ostrowski, and Marian Tracz. Variability of capacity and traffic performance at urban and rural signalised intersections. *Transportation Research Procedia*, 15:87–99, 2016. International Symposium on Enhancing Highway Performance (ISEHP), June 14-16, 2016, Berlin.

[12] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020.

[13] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.

[14] Finale Doshi-Velez and George Dimitri Konidaris. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. *IJCAI : proceedings of the conference*, 2016:1432–1440, 2016.

[15] Utah DOT. Udot data portal, 2022.

[16] Theresa Eimer, Carolin Benjamins, and Marius Lindauer. Hyperparameters in contextual rl are highly situational. In *NeurIPS 2021 Workshop on Ecological Theory of Reinforcement Learning*, 2021.

[17] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on ppo and trpo. *International Conference on Learning Representations*, 2020.

[18] Emanuel Falkenauer. On method overfitting. *Journal of Heuristics*, 4(3):281–287, 1998.

[19] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[20] Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P. Adams, and Sergey Levine. Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability. In *Neural Information Processing Systems*, 2021.

[21] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[22] Andrew Ilyas, Logan Engstrom, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. A closer look at deep policy gradients. *International Conference on Learning Representations*, 2020.

[23] Jeevana Priya Inala, Osbert Bastani, Zenna Tavares, and Armando Solar-Lezama. Synthesizing programmatic policies that inductively generalize. In *International Conference on Learning Representations*, 2020.

[24] Vindula Jayawardana, Anna Landler, and Cathy Wu. Mixed autonomous supervision in traffic signal control. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1767–1773, 2021.

[25] Vindula Jayawardana and Cathy Wu. Learning eco-driving strategies at signalized intersections. *European Control Conference (ECC)*, 2022.

[26] Scott Jordan, Yash Chandak, Daniel Cohen, Mengxue Zhang, and Philip Thomas. Evaluating the performance of reinforcement learning algorithms. In *International Conference on Machine Learning*, pages 4962–4973. PMLR, 2020.

[27] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, K. Czechowski, D. Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Ryan Sepassi, G. Tucker, and Henryk Michalewski. Model-based reinforcement learning for atari. *International Conference on Learning Representations*, 2020.

[28] Shivaram Kalyanakrishnan and Peter Stone. An empirical analysis of value function-based and policy search reinforcement learning. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 749–756, 2009.

[29] Michael K. K. Leung, Andrew Delong, Babak Alipanahi, and Brendan J. Frey. Machine learning in genomic medicine: A review of computational problems and data sets. *Proceedings of the IEEE*, 104(1):176–197, 2016.

[30] Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the 2nd ACM conference on Electronic commerce*, pages 66–76, 2000.

[31] Open Street Map. Open street map, 2022.

[32] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[33] Utah Department of Transportation. Automated traffic signal performance measures component details. 2020.

[34] Afshin Oroojlooy, Mohammadreza Nazari, Davood Hajinezhad, and Jorge Silva. Attendlight: Universal attention-based reinforcement learning model for traffic signal control. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[35] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets. *Proceedings of the 18th BioNLP Workshop and Shared Task*, 2019.

[36] Daniele Reda, Tianxin Tao, and Michiel van de Panne. Learning to locomote: Understanding how environment design matters for deep reinforcement learning. In *Motion, Interaction and Games*, pages 1–10. 2020.

[37] John Schulman, Sergey Levine, P. Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. *ArXiv*, abs/1502.05477, 2015.

[38] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

[39] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *ArXiv*, abs/1801.00690, 2018.

[40] UDOT. Aadt unrounded, 2019.

[41] Pravin Varaiya. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177–195, 2013.

[42] Eugene Vinitsky, Aboudy Kreidieh, Luc Le Flem, Nishant Kheterpal, Kathy Jang, Cathy Wu, Fangyu Wu, Richard Liaw, Eric Liang, and Alexandre M Bayen. Benchmarks for reinforcement learning in mixed-autonomy traffic. In *Conference on robot learning*, pages 399–409. PMLR, 2018.

[43] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. CoLight. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, nov 2019.

[44] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2496–2505. ACM, 2018.

[45] Shimon Whiteson, Brian Tanner, Matthew E. Taylor, and Peter Stone. Protecting against evaluation overfitting in empirical reinforcement learning. In *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 120–127, 2011.

[46] Yufan Zhao, Michael R. Kosorok, and Donglin Zeng. Reinforcement learning design for cancer clinical trials. *Statistics in Medicine*, 28, 2009.

[47] Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, J. Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Jessie Li. Learning phase competition for traffic signal control. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[48] Zhenpeng Zhou, Xiaocheng Li, and Richard N. Zare. Optimizing chemical reactions with deep reinforcement learning. *ACS Central Science*, 3:1337 – 1344, 2017.