Continual World: A Robotic Benchmark For Continual Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Continual learning (CL) — the ability to continuously learn, building on previously acquired knowledge — is a natural requirement for long-lived autonomous reinforcement learning (RL) agents. While building such agents, one needs to balance opposing desiderata, such as constraints on capacity and compute, the ability to not catastrophically forget, and to exhibit positive transfer on new tasks. Understanding the right trade-off is conceptually and computationally challenging, which we argue has led the community to overly focus on *catastrophic forgetting*. In response to these issues, we advocate for the need to prioritize forward transfer and propose *Continual World*, a benchmark consisting of realistic and meaningfully diverse robotic tasks built on top of Meta-World [52] as a testbed. Following an in-depth empirical evaluation of existing CL methods, we pinpoint their limitations and highlight unique algorithmic challenges in the RL setting. Our benchmark aims to provide a meaningful and computationally inexpensive challenge for the community and thus help better understand existing and future solutions.

1 Introduction

Change is ubiquitous. Unsurprisingly, due to evolutionary pressure, humans can quickly adapt and creatively reuse their previous experience. In contrast, although biologically inspired, deep learning (DL) models excel mostly in static domains that satisfy the i.i.d. assumption, as for example in image processing [27, 47, 9, 38], language modelling [50, 10] or biological applications [45]. As the systems are scaled up and deployed in open-ended settings, such assumptions are increasingly questionable; imagine, for example, a robot that needs to adapt to the changing environment and the wear-and-tear of its hardware. *Continual learning* (CL), an area that explicitly focuses on such problems, has been gaining more attention recently. The progress in this area could offer enormous advantages for deep neural networks [18] and move the community closer to the long-term goal of building intelligent machines [19].

Evaluation of CL methods is challenging. Due to the sequential nature of the problem that disallows parallel computation, evaluation tends to be expensive, which has biased the community to focus on toy tasks. These are mostly in the domain of supervised learning, often relying on MNIST. In this work we expand on previous discussions on the topic [43, 15, 29, 44] and introduce a new benchmark, *Continual World*. The benchmark is built on realistic robotic manipulation tasks from Meta-World [52], benefiting from its diversity but also being computationally cheap. Moreover, we provide shorter auxiliary sequences, all of which enable a quick research cycle. On the conceptual level, a fundamental difficulty for evaluating CL algorithms comes from the different desiderata for a CL solution. These objectives are often opposing each other, forcing practitioners to explicitly or implicitly make trade-offs in their algorithmic design that are data-dependent. *Continual World* provides more meaningful relationships between tasks, answering recent calls [18] to increase attention on forward transfer.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Additionally, we provide an extensive evaluation of a spectrum of commonly used CL methods. It highlights that many approaches can deal relatively well with *catastrophic forgetting* at the expense of other desiderata, in particular forward transfer. This emphasizes our call for focusing on *forward transfer* and the need of more benchmarks that allow for common structure among the tasks.

This main contribution of this work is a CL benchmark that poses optimizing forward transfer as the central goal and shows that existing methods struggle to outperform simple baselines in terms of the forward transfer capability. We release the code both for the benchmark and 7 CL methods, which aims to provide the community helpful tools to better understand existing and future solutions.

2 Related work

The field of continual learning has grown considerably in the past years, with numerous works forming new subfields [22] and finding novel applications [46]. For brevity, we focus only on the papers proposing RL-based benchmarks and point to selected surveys of the entire field. [18] provide a high-level overview of CL and argue that learning in a non-stationary setting is a fundamental problem for the development of AI, highlighting the frequent connections to neuroscience. On the other hand, [12, 36] focus on describing, evaluating, and relating CL methods to each other, providing a taxonomy of CL solutions that we use in this work.

The possibility of applying CL methods in reinforcement learning scenarios has been explored for a long time, see [24] for a recent review. However, no benchmark has been widely accepted by the community so far, which is the aim of this work. Below we discuss various benchmarks and environments considered in the literature.

Supervised settings MNIST has been widely used to benchmark CL algorithms in two forms [26]. In the permuted MNIST, the pixels of images are randomly permuted to form new tasks. In the split MNIST, tasks are defined by classifying non-overlapping subsets of classes, e.g. 0 vs. 1 followed by 2 vs. 3. A similar procedure has been applied to various image classification tasks like CIFAR-10, CIFAR-100, Omniglot or mini-ImageNet [2, 44, 4]. Another benchmark is CORe50 [30], a dataset for continuous object recognition. Recent work [28] proposes a benchmark based on language modeling. We find that many of these benchmarks are challenging and allow to measure forgetting. However, we argue they are not geared towards measuring forward transfer or for highlighting important RL-specific characteristics of the CL problem.

Atari The Atari 2600 suite [7] is a widely accepted RL benchmark. Sequences of different Atari games have been used for evaluating continual learning approaches [41, 26]. Using Atari can be computationally expensive, though, e.g., training a sequence of ten games typically requires 100M steps or more. More importantly, as [41] notes, these games lack a meaningful overlap, limiting their relevance for studying transfers. **Continuous control** [31, 23] use continuous control tasks such as Humanoid or Walker2D. However, the considered sequences are short, and the range of experiments is limited. [33] use Meta-World tasks, similarly to us, for evaluations of their continual learning method, but the work is not aimed at building a benchmark. As such, it uses the Meta-World's MT10 preset and does not provide an in-depth analysis of the tasks or other CL methods. **Maze navigation** A set of 3D maze environments is used in [41]. The map structure and objects that the agent needs to collect change between tasks. It is not clear, though, if the tasks provide enough diversity. [29] propose CRLMaze, 3D navigation scenarios for continual learning, which solely concentrate on changes of the visual aspects. **StarCraft** [43] present a StarCraft campaign (11 tasks) to evaluate a high-level transfer of skills. The main drawback of this benchmark is excessive computational demand (often more than 1B frames). **Minecraft** [48] propose simple scenarios within the Minecraft domain along with a hierarchical learning method. The authors phrase the problem as lifelong learning and do not use typical CL methods. **Jelly Bean World** [37] provide interesting procedurally generated grid world environments. The suite is configurable and can host a non-stationary setting. It is unclear, however, if such environments reflect the characteristics of real-world challenges.

3 Continual learning background

Continual learning (CL) is an area of research which focuses on building algorithms capable of handling non-stationarity. They should be able to sequentially acquire new skills and solve novel tasks without forgetting the previous ones. Such systems are desired to accommodate over extended

periods swiftly, which is often compared to human capabilities and alternatively dubbed as lifelong learning. CL is intimately related to multi-task learning, curriculum learning, meta-learning, with some key differences. Multi-task assumes constant access to all tasks, thus ignoring non-stationarity. Curriculum learning focuses on controlling the task ordering and often the learning time-span. Meta-learning, a large field of its own, sets the objective to develop procedures that allow fast adaptation within a task distribution and usually ignores the issue of non-stationarity.

The CL objective is operationalized by the training and evaluation protocols. The former typically consists of a sequence of tasks (their boundaries might be implicit and smooth). The latter usually involves measuring *catastrophic forgetting*, *forward transfer*, and *backward transfer*. The learning system might also have constrained resources: *computations*, *memory*, *size of neural networks*, and the *volume of data samples*. A fundamental observation is that the above aspects and desiderata are conflicting. For example, given unlimited resources, one might mitigate forgetting simply by storing everything in memory and paying a high computational cost of rehearsing all samples from the past.

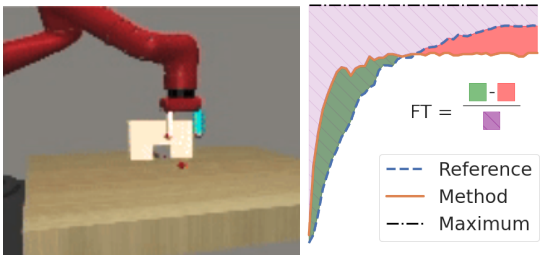

Figure 1: Left graph shows task PEG-UNPLUG-SIDE-V1 and the right graph presents forward transfer from SHELF-PLACE-V1 to PEG-UNPLUG-SIDE-V1. In this case $FT = 0.10$.

Another pair of objectives that are problematic for current methods are forgetting and forward transfer. For neural networks, existing methods propose to limit network plasticity. These alleviate the problem of forgetting, however, at the cost of choking the further learning process. We advocate for more nuanced approaches. Importantly, to make the transfer possible, our benchmark is composed of related tasks. We also put modest bounds on resources. This requirement is in line with realistic scenarios, demanding computationally efficient adaptation and inference. In a broader sense, we hope to address a data efficiency challenge, one of the most significant limitations of the current deep (reinforcement) learning methods. We conjecture that forward transfer might greatly improve the situation and possibly one day enable us to create systems with human-level cognition capabilities, in line with similar thoughts expressed in [18].

4 Continual World benchmark

Continual World is a new benchmark designed to be a testbed for evaluating RL agents on the challenges advocated by the CL paradigm, described in Section 3, as well as highlighting the RL specific algorithmic challenges for CL (see Section 6.1). As such it is aimed at being valuable to both the CL and RL communities. Continual World consists of realistic robotic manipulation tasks, aligned in a sequence to enable the study of forward transfer. It is designed to be challenging while computationally accessible.[1] The benchmark is based on Meta-World, a suite of robotic tasks already established in the community. This enables easy comparisons with the related fields of multi-task and meta-learning reinforcement learning, potentially highlighting one benefit of CL framing, namely that of dealing with different reward scales as we discuss more in detail in Appendix H. Continual World comes with open-source code that allows for easy development and testing of new algorithms, and provides implementations of 7 existing algorithms. Finally, it allows highlighting RL specific challenges for the CL setting. We believe that our work is a step in the right direction towards reliable benchmarks of CL. We realize, however, that it will need to evolve as the field progresses. We leave a discussion on future directions and limitations to Section 4.4.

4.1 Metrics

We start with defining metrics. These are rather standard in the CL setting [40]. Assume $p_i(t) \in [0, 1]$ to be the performance of task i at time t. As a measure of performance, we take the average success rate of achieving a goal specified by a given task when using randomized initial conditions and

[1] We use 8 core machines without GPU. Training the CW20 sequence of twenty tasks takes about 100 hours. We also provide shorter 10 and 3 task sequences to speed up the experimental loop further.

stochastic policies (see also Section 4.3).[2] Each task is trained for $\Delta = 1M$ steps. The main sequence has $N = 20$ tasks and the total sample budget is $T = N \cdot \Delta = 20M$. The i-th task is trained during the interval $t \in [(i-1) \cdot \Delta, i \cdot \Delta]$. We report the following metrics:

Average performance. The average performance at time t is (see Figure 3)

$$\mathrm{P}(t) := \frac{1}{N} \sum_{i=1}^{N} p_i(t). \tag{1}$$

Its final value, $\mathrm{P}(T)$, is a traditional metric in the CL research. We use it for tuning hyperparameters.

Forward transfer. We measure the forward transfer of a method as the normalized area between its training curve and the training curve of the reference, single-task, experiment, see Figure 1. Let $p_i^b \in [0, 1]$ be the reference performance[3] then the forward transfer for the task i, denoted by FT_i, is

$$\mathrm{FT}_i := \frac{\mathrm{AUC}_i - \mathrm{AUC}_i^b}{1 - \mathrm{AUC}_i^b}, \quad \mathrm{AUC}_i := \frac{1}{\Delta} \int_{(i-1)\cdot\Delta}^{i\cdot\Delta} p_i(t)\mathrm{d}t, \quad \mathrm{AUC}_i^b := \frac{1}{\Delta} \int_{0}^{\Delta} p_i^b(t)\mathrm{d}t,$$

The average forward transfer for all tasks, FT, is defined as

$$\mathrm{FT} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{FT}_i. \tag{2}$$

In our experiments, we also measure backward transfer. As it is negligible, see Appendix E.1.

Forgetting. For task i, we measure the decrease of performance after ending its training, i.e.

$$F_i = p_i(T) - p_i(i \cdot \Delta). \tag{3}$$

Similarly to FT, we report $F = \frac{1}{N} \sum_{i=1}^{N} F_i$.

4.2 Continual World tasks

This section describes the composition of Continual World benchmark and the rationale behind its design. We decided to base on Meta-World [52], a fairly new but already established robotic benchmark for multi-task and meta reinforcement learning. From a practical standpoint, Meta-World utilizes the MuJoCo physics engine [49], prized for speed and accuracy. Meta-World provides 50 distinct manipulation tasks with everyday objects using a simulated robotic Sawyer arm. Although the tasks vary significantly, the structure and semantics of observation and action spaces remain the same, allowing for transfer between tasks. Each observation is a 12-dimensional vector containing (x, y, z) coordinates of the robot's gripper and objects of interest in the scene. The 4-dimensional action space describes the direction of the arm's movement in the next step and the gripper actuator delta. Reward functions are shaped to make each task solvable. In evaluations, we use a binary *success metric* based on the distance of the task-relevant object to its goal position. This metric is interpretable and enables comparisons between tasks. For more details about the rewards and evaluation metrics, see [52, Section 4.2, Section 4.3].

CW20, CW10, triplets sequences The core of our benchmark is CW20 sequence. Out of 50 tasks defined in Meta-World, we picked those that are not too easy or too hard in the assumed sample budget $\Delta = 1M$. Aiming to strike a balance between the difficulty of the benchmark and computational requirements, we selected 10 tasks. The tasks and their ordering was based on the transfer matrix (see the next paragraph), so that there is a high variation of forward transfers (both in the whole list and locally). We refer to these ordered tasks as CW10, and CW20 is CW10 repeated twice. We recommend using CW20 for final evaluation; however, CW10 is already very informative in most cases. Due to brevity constraints, we present an ablation with an alternative ordering of the tasks and a longer sequence of 30 tasks in Appendix G, however these experiments do not alter our findings. Additionally, to facilitate a fast development cycle, we propose a set of triplets, sequences of three tasks which exhibit interesting learning dynamics.

The CW10 sequence is: HAMMER-V1, PUSH-WALL-V1, FAUCET-CLOSE-V1, PUSH-BACK-V1, STICK-PULL-V1, HANDLE-PRESS-SIDE-V1, PUSH-V1, SHELF-PLACE-V1, WINDOW-CLOSE-V1, PEG-UNPLUG-SIDE-V1.

[2] Stochastic evaluations are slightly more smooth and have little difference to the deterministic ones.

[3] Note that we avoid trivial tasks, for which $\mathrm{AUC}_i^b = 1$ is making the metric ill-defined. Additionally, we acknowledge the dependency on the hyperparameters of the learning algorithm and that there are alternative quantities of interest like relative improvement in performance rather than faster learning.

Transfer matrix Generally, the relationship between tasks and its impact on learning dynamics of neural networks is hard to quantify, where semantic similarity does not typically lead to transfer [13]. To this end, we consider a minimal setting, in which we finetune on task t_2 a model pretrained on t_1, using the same protocol as the benchmark (e.g., different output heads, see Section 4.3). This provides neural network-centric insight into the relationship between tasks summarized in Figure 2, and allows us to measure *low-level transfer* between tasks, i.e., the ability of the model to reuse previously acquired features. See Appendix D for more results and extended disucssion.

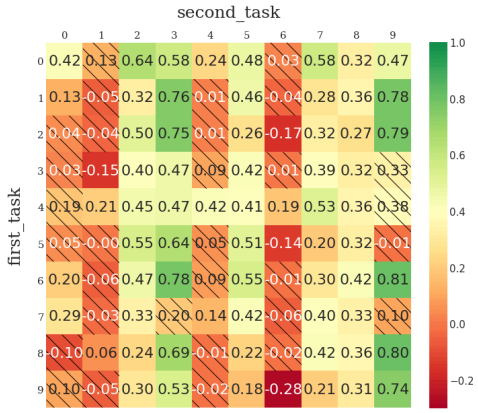

Figure 2: Transfer matrix, see Section 4.2. Each cell represents the forward transfer from the first task to the second one. We shaded the cells for which 0 belongs to their 90% confidence interval.

Notice that there are only a few negative forward transfer cases, and those are of a rather small magnitude (perhaps unsurprisingly, as the tasks are related). There are also visible patterns in the matrix. For instance, some tasks such as PEG-UNPLUG-SIDE-V1 or PUSH-BACK-V1 benefit from a relatively large forward transfer, (almost) irrespective of the first task. Furthermore, the average forward transfer given the second task (columns) is more variable than the corresponding quantity for the first task (rows).

Note that some transfers on the diagonal (i.e., between the same tasks) are relatively small. We made a detailed analysis of possible reasons, which revealed that the biggest negative impact is due to the replay buffer resets, which seems, however, unavoidable for off-diagonal cases, see Section 6.1 for details.

Importantly, we use this matrix to estimate what level of forward transfer a good CL method should be able to achieve. We expect that a model which is able to remember all meaningful aspects of previously seen tasks would transfer at least as well as if one were just fine-tuning after learning the best choice between the previous tasks. For a sequence $t_1, \ldots, t_N$ we set the reference forward transfer, RT, to be

$$\text{RT} := \frac{1}{N} \sum_{i=2}^{N} \max_{j<i} \text{FT}(t_j, t_i), \tag{4}$$

where $\text{FT}(t_j, t_i)$ is the transfer matrix value for t_j, t_i. For the CW20 sequence, the value is $\text{RT} = 0.46$. Note that a *model can do better* that this by composing knowledge from multiple previous tasks.

4.3 Training and evaluation details

We adapt the standard Meta-World setting to CL needs. First, we use separate policy heads for each task, instead of the original one-hot task ID inputs (we provide ablation experiments for this choice in Appendix G). Second, in each episode we randomize the positions of objects in the scene to encourage learning more robust policies. We use an MLP network with 4 layers of 256 neurons.

For training, we use soft actor-critic (SAC) [16], a popular and efficient RL method for continuous domains. SAC is an off-policy algorithm using replay buffer, which is an important aspect for CL, particularly for methods relying on rehearsing old trajectories. SAC is based on the so-called maximum entropy principle; this results in policies which explore better and are more robust to changes in the environment dynamics. Both of these qualities might be beneficial in CL.

We note that the size of the neural network and optimization details of the SAC algorithm (like batch size) put constraints on "the amount of compute". Intentionally, these are rather modest, which is in line with CL desiderata, see Section 3. Similarly, we limit the number of timesteps to $1M$, which is a humble amount for modern-day deep reinforcement learning. We picked tasks to be challenging but not impossible within this budget. We note that training in the RL setting tends to be less stable than in the supervised one. We recommend using multiple seeds, in our experiments, we typically used 20 and calculate confidence intervals; we used the bootstrap method. We choose hyperparameters that maximize average performance (1). In our experiments, we tune common parameters for SAC and

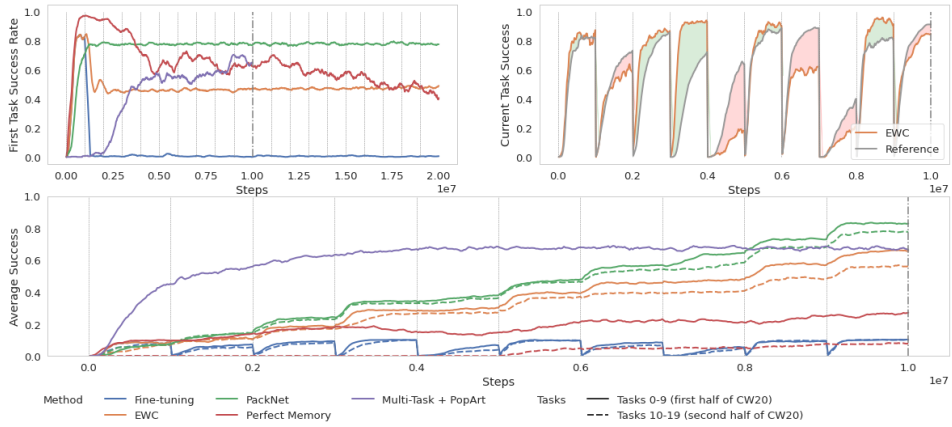

Figure 3: Training curves for selected CL methods and multi-task. The upper left panel shows the performance on the first task for a subset of methods throughout the whole training. Note that due to the use of different output heads, we do not see a second bump when revisiting this task at time $10M$. The upper right panel shows the performance on the current task being learn for EWC compared to a reference (a model learning only that task from scratch). The bottom plot shows the average performance. Solid lines show the performance of the model training on the first 10 tasks (where 1 means being able to solve all of them). Dashed lines show the performance of learning the same tasks in the second half of the benchmark. Note that dashed lower are below solid ones, indicating lower performance on the second pass, even if the agent has already previously learned the tasks and have access to relevant features.

the method-specific hyperparameters separately. All details of the training and evaluation setup is presented in Appendix A.

4.4 Limitations of Continual World

As any benchmark, we are fully aware that ours will not cover the entire spectrum of problems that one might be interested in. Here we summarize a few limitations that we hope to overcome in future instantiation of this benchmark:

Input space We use a small 12 dimensional observation space. This is key to achieve modest computational demand. However, richer inputs could allow for potentially more interesting forms of transfer (e.g., based on visual similarity of objects) and would allow to infer the task from the observation, which is currently impossible.

Reliance on SAC We use the SAC algorithm [16], which is considered a standard choice for continuous robotic tasks. However, there is a potential risk of overfitting to the particularities of this algorithm and exploring alternative RL algorithms is important.

Task boundaries We rely on task boundaries. One can rely on task inference mechanisms (e.g. [34, 39]) to resolve this limitation, though we acknowledge the importance to extend the benchmark towards allowing and testing for task inference capabilities. Also testing for algorithms dealing with continuous distributional drift is not possible in the current format.

Output heads We rely on using a separate head for each new task, similar to many works on continual learning. We opt for this variant based on its simplicity and better performance than using one-hot encoding to indicate a task. We believed that the lack of semantics of the one-hot encoding would further impede transfer, as the relationship between tasks can not be inferred. We carry ablation studies with using one-hot encoding as an input and a single head architecture, setting that is already compatible with our benchmark. We regard this aspect as an important future work, and in particular, we are exploring alternative encoding of input to make this choice more natural. A coherent domain, like Continual World, provides a unique opportunity to exploit a consistent output layer as its semantics does not change between tasks.

The difficulty and number of tasks The number of tasks is relatively small. CW20, the main sequence we use, consists of only 10 different tasks, which are then repeated. We believe the repetition of tasks is important for a CL benchmark, leading to interesting observations. We check

also that results are qunatitatively similar on a sequen of 30 tasks, see Appendix G. However, longer sequences, potentially unbounded, are needed to understand various limitations of existing algorithms. For example, the importance of *graceful forgetting* or dealing with systems that run out of capacity, a scenario where *there is no multi-task solution for the sequence of observed tasks*. This is particularly of interest for methods such as PackNet [32]. Additionally, we provide the number of tasks in advance. Dealing with an unknown number of tasks might raise additional interesting questions. Finally, in future iterations of the benchmark, it is important to consider more complex tasks or more complex relationship between tasks to remain a challenge to existing methods. Our goal was to provide a benchmark that approachable by existing methods, as not to stifle progress.

Low-level transfer We focus on low-level transfers via neural network features/weights. This might not allow to explore the ability of the learning process to exploit the compositionality of behavior or to rely on a more interesting semantic level. While we believe such research is crucial, we argue that solving low-level transfer is equally important and might be a prerequisite. So, for now, it is beyond the scope of this work, though future iterations of the benchmark could contain such scenarios.

5 Methods

We now sketch 7 CL methods evaluated on our benchmark. Some of them were developed for RL, while others were meant for the SL context and required non-trivial adaptation. We aimed to cover different families of methods; following [12], we consider three classes: regularization-based, parameter isolation and replay methods. See Appendix B for an extended description and discussion.

Regularization-based Methods This family builds on the observation that one can reduce forgetting by protecting parameters that are important for the previous tasks. The most basic approach often dubbed **L2** [26] simply adds a L_2 penalty, which regularizes the network not to stray away from the previously learned weights. In this approach, each parameter is equally important. **Elastic Weight Consolidation (EWC)** [26] uses the Fisher information matrix to approximate the importance of each weight. **Memory-Aware Synapses (MAS)** [3] also utilizes a weighted penalty, but the importance is obtained by approximating the impact each parameter has on the output of the network. **Variational Continual Learning (VCL)**, follows a similar path, but uses variational inference to minimize the Kullback-Leibler divergence between the current distribution of parameters (posterior) and the distribution for the previous tasks (prior).

Parameter Isolation Methods This family (also called modularity-based) forbid any changes to parameters that are important for the previous tasks. It may be considered as a "hard" equivalent of regularization-based methods. **PackNet** [32] "packs" multiple tasks into a single network by iteratively pruning, freezing, and retraining parts of the network at task change. PackNet is an extension of ProgressiveNet [41], developed in the RL context.

Replay Methods Methods of this family keep some samples from the previous tasks and use them for training or as constraints to reduce forgetting. We use a **Perfect Memory** baseline, a modification of our setting which remembers all the samples from the past (i.e., without resetting the buffer at the task change). We also implemented **Averaged Gradient Episodic Memory (A-GEM)** [11], which projects gradients from new samples such to not interfere with previous tasks. We find that A-GEM does not perform well on our benchmark.

Multi-task learning In multi-task learning, a field closely related to CL, tasks are trained simultaneously. By its design, it does not suffer from forgetting, however it is considered to be hard as multiple task "compete for attention of a single learning system", see [20, 42]. We find that using reward normalization as in PopArt [20] is essential to achieve good performance. See Appendix H.

6 Experiments

Now we present empirical results; these are evaluations of a set of 7 representative CL methods (as described in Section 5) on our Continual World benchmark. We focus on *forgetting and transfers* while keeping fixed constraints on computation, memory, number of samples, and neural network architecture. Our main empirical contributions are experiments on the long CW20 sequence and following high-level conclusions. For summary see Table 1, Figure 3 and for an extensive discussion,

we refer to Appendix E (including results for the shorter sequence, CW10). In Appendix G we provide various ablations and detailed analysis of sensitivity to the CL specific hyperparameters.

Performance The performance averaged over tasks (eq. (1)) is a typical metric for the CL setting. PackNet seems to outperform other methods, approaching 0.8 from the maximum of 1.0, outperforming multi-task solutions which might struggle with different reward scale, a problem elegantly avoided in the CL framing. Other methods perform considerably worse. A-GEM and Perfect Memory struggle. We further discuss possible reasons in Section 6.1.

method	performance	forgetting	f. transfer
Fine-tuning	0.05 [0.05, 0.06]	0.73 [0.72, 0.75]	**0.20 [0.17, 0.23]**
L2	0.43 [0.39, 0.47]	0.02 [0.00, 0.03]	-0.75 [-0.87, -0.65]
EWC	0.60 [0.56, 0.64]	0.02 [-0.00, 0.05]	-0.19 [-0.25, -0.14]
MAS	0.51 [0.49, 0.53]	0.00 [-0.01, 0.02]	-0.52 [-0.58, -0.48]
VCL	0.48 [0.46, 0.50]	0.01 [-0.01, 0.02]	-0.48 [-0.56, -0.42]
PackNet	**0.80 [0.79, 0.82]**	0.00 [-0.01, 0.01]	**0.18 [0.14, 0.21]**
Perfect Memory	0.14 [0.13, 0.16]	0.05 [0.04, 0.06]	-1.37 [-1.46, -1.30]
A-GEM	0.07 [0.07, 0.08]	0.72 [0.71, 0.74]	0.17 [0.13, 0.20]
MT	0.50 [0.47, 0.54]	—	—
MT (PopArt)	0.66 [0.62, 0.70]	—	—
RT	—	—	**0.46**

Table 1: Results on CW20, for CL methods and multi-task training. Metrics are defined in Section 4.1, RT is eq. (4). We used 20 seeds and provide 90% confidence intervals.

Forgetting We observe that most CL methods are usually efficient in mitigating forgetting. However, we did not notice any boost when revisiting a task (see Figure 3). Even if a different output head was employed, relearning the internal representation should have had an impact unless it changed considerably when revisiting the task. We found A-GEM difficult to tune; consequently with the best hyperparameter settings, it is relatively similar to the baseline fine-tuning method.

Transfers For all methods, forward transfer for the second ten tasks (and same tasks are revisited) drops compared to the first ten tasks. This is in stark contrast to forgetting, which seems to be well under control. Among all methods, only fine-tuning and PackNet are able to achieve positive forward transfer (0.20 and 0.18, resp.) as well as on the first (0.32 and 0.22, resp.) and the second (0.08 and .14, resp.) half of tasks. However, these are considerably smaller than RT $= 0.46$, which in principle can even be exceeded, and which should be reached by a model that remembers all meaningful aspects of previously seen tasks, see (4). These results paint a pretty grim picture: we would expect improvement, rather than deterioration in performance, when revisiting previously seen tasks. There could be multiple reasons for this state of affairs. It could be attributed to the loss of plasticity, similar to the effect observed in [5]. Another reason could be related to the interference between CL mechanisms or setting and RL, for instance, hindering exploration. We did not observe any substantial cases of **backward transfer**; see Appendix E.

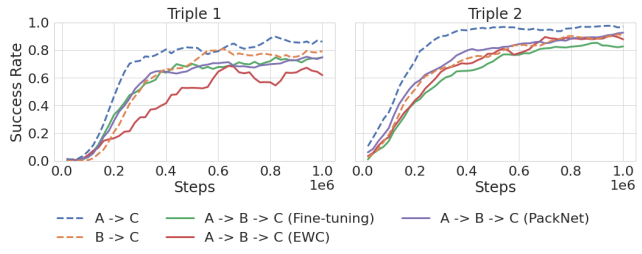

Figure 4: **How forgetting impacts the forward transfer.** An ideal agent learning on a sequence $A \to B \to C$ should have at least as good performance on task C as an agent which just learns $A \to C$. In reality, an interfering task B reduces this transfer, even when continual learning approaches are used.

Triplets experiments We illustrate how forgetting and forward transfer interact with each other in a simpler setting of three task sequences, see Figure 4 and Appendix F. We focus on sequences of tasks $A \to B \to C$, where $A \to C$ has significant positive forward transfer and $B \to C$ has a smaller or even negative transfer. An efficient CL agent should be able to use information from A to get good performance on C. However, interference introduced by B reduces the final forward transfer . The drive for reducing forgetting in CL agents have been primarily to perform well on previous tasks when we revisit them. With this example, we argue that an equally important reason to improve memory of CL agents is to efficiently use past experiences to learn faster on new tasks.

PackNet PackNet stands out in our evaluations. We conjecture that developing related methods might be a promising research direction. Besides further increasing performance, one could mitigate the limitations of PackNet. PackNet relies on knowing task identity during evaluation. While this assumption is met in our benchmark, it is an interesting topic for future research to develop methods that cope without task identity. Another nuisance is that PackNet assigns some fixed fraction of parameters to a task. This necessitates knowledge of the length of the sequence in advance. Additionally, when the second ten tasks of CW20 start, PackNet performance degrades, showing its potentially inefficient use of capacity and past knowledge, given that the second ten tasks are identical with the first ten and hence no additional capacity is needed. In a broader context, we speculate that parameter isolation methods might be a promising direction towards better CL methods.

Other observations In stark contrast with the supervised learning setting, we found that replay based methods (Perfect Memory and A-GEM) suffer from poor performance. This is even though we allow for a generous replay, which could store the whole experience. Explaining and amending this situation is, in our view, an important research question. We conjecture that this happens due to the regularization of the critic network (which was unavoidable for these methods). We found multi-task learning attaining lower scores than PackNet, the best CL method and comparable to the second one, EWC. We think this suggests interesting research directions for multi-task learning.

6.1 RL-Related Challenges

Reinforcement learning brings a set of issues not present in the SL setting, e.g., exploration, varying reward scales, and stochasticity of environments. We argue that it is imperative to have a reliable benchmark to assess the efficiency of CL algorithms with respect to these problems. We find that some current methods are not well adjusted to the RL setting and require non-trivial conceptual considerations and careful tuning of hyperparameters, see details in Appendix C.

An important design choice is whether or not to regularize the critic in the actor-critic framework (e.g. in SAC). We find it beneficial to focus on reducing forgetting in the actor, while allowing the critic to freely adapt to the current task (note that critic is used only in training of the current task), similar to [44]. This is much different to the supervised setting, in which the multi-task learning is often an upper bound to CL. On the other hand, a forgetful critic is controversial. This can be sharply seen when the same task is repeated and the critic needs to learn from scratch. Additionally, not all methods can be trivially adapted to the 'actor-only regularization' setting, as for example replay based methods. In Appendix C we present experiments with critic regularization for EWC.

Another aspect is exploration and its non-trivial impact on transfers. As it was observed, transfers from a given task to the same one are sometimes poor. This results from the fact that at the task change the replay buffer is emptied and SAC collects new samples, usually by using the uniform policy. Learning on these random samples reduces performance on the current task and thus the forward transfer. Experimentally, we find that not resetting the buffer or using the current policy for exploration improves the transfer on the diagonal, but at the same time it harms off-diagonal transfers.

7 Conclusions and Future Work

We present Continual World, a continual reinforcement learning benchmark, and an in-depth analysis of how existing methods perform on it. The benchmark is aimed at facilitating and standardizing the CL system evaluation, and as such, is released with code, including implementation of 7 representative CL algorithms. We argue for more attention to *forward transfer* and the interaction between forgetting and transfer, as many existing methods seem to sacrifice transfer to alleviate forgetting. In our opinion, this should not be the aim of CL, and we need to strike a different balance between these objectives.

We made several observations, both conceptual and empirical, which open future research directions. In particular, we conjecture that parameter isolation methods are promising. Further, we identified a set of critical issues at the intersection of RL and CL. Resolving critic regularization and efficient use of multi-task replays seem to be the most pressing. Our benchmark highlights some challenges, which in our view are relevant and tangible now. In the long horizon, achieving high-level transfers, removing task boundaries, and scaling up are significant goals for future editions of Continual World. Our work is foundational research and does not lead to any direct negative applications.

References

[1] Joshua Achiam. Spinning Up in Deep Reinforcement Learning. 2018.

[2] Tameem Adel, Han Zhao, and Richard E. Turner. Continual learning with adaptive weights (CLAW). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[3] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11207 of *Lecture Notes in Computer Science*, pages 144–161. Springer, 2018.

[4] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11849–11860, 2019.

[5] Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[6] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[7] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, jun 2013.

[8] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1613–1622. JMLR.org, 2015.

[9] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[10] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[11] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[12] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*, 2019.

[13] Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *CoRR*, abs/1812.02224, 2018.

[14] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[15] Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *CoRR*, abs/1805.09733, 2018.

[16] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018.

[17] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *CoRR*, abs/1812.05905, 2018.

[18] Raia Hadsell, Dushyant Rao, Andrei A. Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 24(12):1028 – 1040, 2020.

[19] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245 – 258, 2017.

[20] Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with popart. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3796–3803. AAAI Press, 2019.

[21] Ferenc Huszár. Note on the quadratic penalties in elastic weight consolidation. *Proceedings of the National Academy of Sciences*, page 201717042, 2018.

[22] Khurram Javed and Martha White. Meta-learning representations for continual learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1818–1828, 2019.

[23] Christos Kaplanis, Murray Shanahan, and Claudia Clopath. Policy consolidation for continual reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3242–3251. PMLR, 09–15 Jun 2019.

[24] Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives, 2020.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[26] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.

[28] Germán Kruszewski, Ionut-Teodor Sorodoc, and Tomas Mikolov. Evaluating online continual learning with calm, 2021.

[29] Vincenzo Lomonaco, Karan Desai, Eugenio Culurciello, and Davide Maltoni. Continual reinforcement learning in 3d non-stationary environments. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 999–1008. IEEE, 2020.

[30] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, volume 78 of *Proceedings of Machine Learning Research*, pages 17–26. PMLR, 2017.

[31] Kevin Lu, Igor Mordatch, and Pieter Abbeel. Adaptive online planning for continual lifelong learning. *CoRR*, abs/1912.01188, 2019.

[32] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7765–7773. IEEE Computer Society, 2018.

[33] Jorge A. Mendez, Boyu Wang, and Eric Eaton. Lifelong policy gradient learning of factored policies for faster training without forgetting. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[34] Kieran Milan, Joel Veness, James Kirkpatrick, Michael H. Bowling, Anna Koop, and Demis Hassabis. The forget-me-not process. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3702–3710, 2016.

[35] Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[36] German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[37] Emmanouil Antonios Platanios, Abulhair Saparov, and Tom M. Mitchell. Jelly bean world: A testbed for never-ending learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[38] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021.

[39] Dushyant Rao, Francesco Visin, Andrei A. Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7645–7655, 2019.

[40] Natalia Díaz Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there is more than forgetting: new metrics for continual learning. *CoRR*, abs/1810.13166, 2018.

[41] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

[42] Tom Schaul, Diana Borsa, Joseph Modayil, and Razvan Pascanu. Ray interference: a source of plateaus in deep reinforcement learning. *CoRR*, abs/1904.11455, 2019.

[43] Jonathan Schwarz, Daniel Altman, Andrew Dudzik, Oriol Vinyals, Yee Whye Teh, and Razvan Pascanu. Towards a natural benchmark for continual learning. In *Continual learning Workshop, Neurips 2018*, 2018.

[44] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4535–4544. PMLR, 2018.

[45] Andrew W. Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Zídek, Alexander W. R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T. Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nat.*, 577(7792):706–710, 2020.

[46] Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. LAMOL: language modeling for lifelong language learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[47] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019.

[48] Chen Tessler, Shahar Givony, Tom Zahavy, Daniel J. Mankowitz, and Shie Mannor. A deep hierarchical approach to lifelong learning in minecraft. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1553–1561. AAAI Press, 2017.

[49] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 5026–5033. IEEE, 2012.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[51] Jeffrey Scott Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985.

[52] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 1094–1100. PMLR, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] see Section 4.4.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 7
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [N/A]
(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The codes is included in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] see details in Appendix A.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] we used 20 random seeds, see also details in Appendix A.6.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] see Appendix A.7.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We base on the MetaWorld benchmark [52], which we clearly indicate a few time, including the abstract.
   (b) Did you mention the license of the assets? [Yes] We use MIT licence; see Appendix A.1.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]