Towards a natural benchmark for continual learning

Jonathan Schwarz [1] **Daniel Altman** [2] **Andrew Dudzik** [1] **Oriol Vinyals** [1] **Yee Whye Teh** [1] **Razvan Pascanu** [1]

Abstract

Continual Learning (CL) has recently seen a surge of interest, highlighting the importance of the topic for machine learning and artificial intelligence. While most recent work has been dedicated towards algorithmic improvements, robust evaluation remains unsolved - a shortcoming we explore in this work. The CL problem is usually described as a list of desiderata when facing a non-stationary stream of data, in most cases structured as a known sequence of different tasks. The goal of a CL scheme is then to leverage previously acquired skills when facing new problems, but also to retain skills and perform well on previously encountered tasks. Ideally, a continual learning algorithm does so with constant memory and computational footprint in order to scale, and can learn without being aware of task boundaries. These different desiderata are in tension with each other, making both problem specification and evaluation difficult. In this work, we explore the relationship between data and evaluation and argue that one needs to test the system in natural scenarios. Thus, we propose a new benchmark based on the popular video game StarCraft II in the hope to better understand existing approaches to continual learning. A video of human play on the proposed benchmark can be found here: `https://goo.gl/vdzkut`.

1. Introduction

As a problem, CL is (usually) defined as learning from a non-stationary stream of data, structured as a sequence of tasks[1] to better highlight and control the non-stationarity aspect. This setting contradicts the basic assumption of statistical learning, namely that data is independently and identically distributed, hence making learning difficult.

The reasons for why data is non-IID can be diverse. For sequential modelling, one might want to preserve order in an attempt to discover long and very long-term structure. Unfortunately, this can lead to overfitting the most recently seen examples while globally underfitting the dataset [1]. In other cases, parts of the dataset might need to be deleted periodically for legal reasons. When facing real time distributional shifts, it might be infeasible to retrain past data or train new models from scratch [2]. In active or reinforcement learning (RL), non-stationarity is imposed by how the model observes data. In the RL case, observations are the outcome of agent interactions with the environment and hence the distribution of observations changes as the agent learns. Because of the richness of existing reinforcement learning environments, this change need not be a gradual drift, but could be sudden.

One usually expects the system, exposed to the sequence of fairly distinct tasks forming the CL problem, to achieve several goals: (i) It must not forget previously learnt skills (i.e. avoid *catastrophic forgetting*). (ii) We expect forward transfer, i.e. the ability to accelerate learning on new tasks by exploiting previously acquired skills. In addition, the system should be capable of backward transfer, possibly improving performance on past tasks. (iii) The CL scheme ought to operate with a fixed resource footprint (in terms of memory, model size, computation) – which has to be a prerequisite in order to be able to scale to a large number of tasks. This is required in life-long learning scenarios, when the system has to continuously interact and adapt to its environment. (iv) Ideally, the system should be capable of doing so without knowledge of task boundaries (or task identities) or when no boundaries exist at all (i.e. one task slowly morphs into another).

It is important to note that some of these constraints are in direct contradiction. Perfectly remembering previously learnt skills is impossible (provided fixed capacity) when facing a potentially unbounded number of problems. Moreover, being able to perfectly remember previous tasks while expecting transfer or generalisation, are also goals in tension each other. Therefore, certain trade-offs need to be made implicitly, meaning different desiderata cannot be targeted in isolation[2]. We

[1] While what *a task* is, or what it means to have *sufficiently diverse tasks* might not be well defined concepts, this framework does provide a clean and constructive way to analyse the problem.

[2] As this bears the risk of progressing on a goal while sacrificing another

Level	Combat	Economy	Map exploration	Build orders
① MoveToBeacon				
② FindAndDefeatZerglings	•		•	
③ GatherMinerals		•		
④ GatherGas		•	•	•
⑤ BuildMarines		•	•	•
⑥ BuildMarauders		•	•	•
⑦ GatherMineralsAndGas		•	•	•
⑧ BuildArmy		•		•
⑨ DefendAttackWaves	•			
⑩ FindEnemyBase			•	
⑪ DestroyEnemyBase	•	•	•	•

Figure 1 & Table 1: Left: Example in-game screenshots on levels ①, ④, ⑨ and ⑪. Best viewed on a computer. **Right:** A categorisation of high-level skills required to solve the problems in each level.

argue that meaningful trade-offs are conditioned on the problem instance (i.e. the data used), which is the central theme of this work. Motivated by a recent surge in interest in the topic [e.g 2–17] we stress the importance of meaningful benchmarks.

Thus, we propose a new challenge in the form of a large scale problem going beyond the settings considered in recent work. The challenge was developed by a professional map designer, thus appearing natural to humans, a characteristic lacking in current benchmarks. We further argue that the proposed benchmark is a suitable tool to improve the understanding of the state of the field. Finally, we hope it will guide researchers in developing approaches consistent with above desiderata.

2. Related work

We divide approaches to CL in three categories, similar to [2]:

A first set of approaches address the problem by means of regularisation [e.g 7, 14, 18, 19]. In their purest form, these methods assume no data from previous tasks and can be seen as approximating the missing terms of the multi-task objective (e.g. by a quadratic regulariser). From a Bayesian perspective, they gradually build an informative prior that restricts learning such that previous tasks are not forgotten. The regularisation slows down learning for certain important-for-previous-tasks weights which is similar to the reduction in plasticity for neurons during memory consolidation [7, 14].

A second family of approaches address the problem in a structural manner [e.g. 11, 20]. They rely on localising learning to a subset of weights specific to the current task, and protecting previously learnt tasks by freezing the rest of the weights. These approaches do not follow the popular end-to-end learning paradigm, and the focus of adaptation is often heuristically defined rather than learnt. The resulting models end up reflecting the structure of the learnt sequence of tasks, being fundamentally modular, where different tasks are learnt in separate interconnected modules.

A third family of approaches rely on replay to allow the minimisation of the multi-task objective rather than the loss on the current task. There are usually two subset of approaches, based on how the previous seen data is stored, i.e. directly or compressed via a generative model [e.g 21]. Experience replay can be an effective means to reduce forgetting, but it remains unclear how to scale such methods to a very large number of tasks [6].

Of course, many approaches are in-between these rough categories. For example Progress&Compress [13] relies on both a regularisation term and structural changes of the model. Elastic Weight Consolidation [14] and Learning Without Forgetting [22] use different output layers per task, which is a structural change. Recently, small replay buffers (known as core sets) have also been introduced to Bayesian methods [18]. Furthermore, there exist approaches that are not captured at all by this categorisation, or that might not directly target continual learning, but have been shown to be important building blocks. The Forget-me-not process [23], for instance, is an approach for automatically inferring the identity of the current task from observation. Knowledge distillation [24] has also been proven to be an important tool for transfer.

Finally, [10] provide a recent comprehensive review of the field. [2] and [3], similar to this work, acknowledge and discuss the important role the data and metrics plays in defining the potential trade-offs between the different desiderata, and hence the performance of a given approach.

Note that while most modern algorithmic approaches have been developed in recent years, the problem itself has been

identified long before [e.g. 25–28].

3. Existing benchmarks and evaluations

The de-facto benchmarks, used by most published works, rely on MNIST [29] as the underlying source of data. For example, Permuted MNIST applies different fixed permutations of the pixels to generate new tasks [15]. Other variants of the benchmark rely on splitting the set of possible labels into several groups (known as Split MNIST), each group representing a different task.

Fundamentally, one issue with these benchmarks is the use of MNIST itself as the main source of data. The resulting tasks lack complexity, making it difficult to measure or observe either positive or negative forward transfer. Given the large capacity of neural networks, the relative simplicity makes it hard to understand how the system deals with capacity constraints. In the case of Permuted MNIST, a further issue is the lack of resemblance between examples from different tasks.

As a result, evaluation efforts usually focus on avoiding catastrophic forgetting, ignoring other important aspects of CL. Unfortunately, besides these common benchmarks, most works do not agree on additional experiments, which further increases the difficulty of a proper comparison between existing approaches.

Lastly, the analysis of the approach itself can sometimes be problematic. We argue that the default analysis for continual learning ought to measure both forward and backward transfer. In addition, other important aspects, such as the computational footprint, both in terms of memory and computation, should be reported as a function of the number of tasks. The bound on computation acknowledges the asynchronous nature of the world, e.g. in a reinforcement or online learning setting, the environment does not wait on the agent to act. Hence it is important for the system to be able to react and learn quickly, regardless of the number of tasks it had been exposed to.

However, reducing these aspects to a single number is difficult. It implies the definition of a weighting between the different aspects of CL,which might be hard to infer. The issue is further hidden by the considered data distribution. For example, if tasks are completely unrelated (e.g. random sequence of Atari games, as used in [14]), there is little to no potential of positive forward transfer beyond low-level visual features, again shifting the focus on negative backward transfer (forgetting), regardless of how performance is measured. Hence, if the tasks are similar, the role of forward transfer might increase. There are also several kinds of transfer or interference possible (e.g. composing knowledge instead of just reusing it, preserving a policy for different visuals, or different policies for the same visuals). This leads to most works focusing on one particular aspect, ignoring many of the others. However, due to the trade-off discussed in Section 1, they cannot be treated in isolation.

Therefore we think we need a richer set of benchmarks accepted by the community and a richer set of evaluation metrics that can make this trade-off more explicit and provide a rounded perspective of the pros and cons of proposed approaches. Moving in this direction, [3] proposes to look at interplay between forgetting and positive transfer which we consider a step in the right direction.

4. New challenge

4.1. Description

We propose a new benchmark for continual learning based on the video game StarCraft II. StarCraft provides an ideal environment for building a CL benchmark as it can act as a persistent complex environment in which an agent can sequentially learn multiple skills that are later composed and reused. The challenge is designed to feel natural to a human player, hence providing opportunity for skill transfer and composition that is natural to humans. To this end, the sequence of proposed tasks is structured as a campaign designed to teach an agent basic game play (similar to how a tutorial for a human player might be designed). As the complexity of the tasks grows gradually, performing well on future levels rests on remembering and composing skills learnt along the way. To also test resilience against forgetting, not all skills are present in all levels. In Figure 1, we show an intuitive (yet not complete) list of skills required per level.

The campaign is structured into 11 different levels, each with a descriptive name. Note that the first 6 levels are meant to teach basic skills required to play the game (using a dense reward structure). Example skills are economy (collect minerals or gas), combat, build orders needed to construct certain buildings and units, map exploration to find the enemy, among others. In contrast, the last 5 tasks are more complex, and the reward signal is sparser. The skills previously learnt somewhat

in isolation now need to be deployed and combined in order to succeed.

In each level, agents are allowed a fixed budget of episodes to learn from before advancing to the next problem. Episodes have a fixed time limit, meaning that an agent may fail by running out of time or in case of defeat. Crucially, we disallow re-visits to previous levels (although past data may be stored) and force agents to proceed to the next level regardless of whether currently taught skills have been mastered. Note also that recent work has stressed the difference between *Single Head* and *Multi Head* evaluation. A head refers to an output layer of a neural network. As all levels share a common action space, we enforce the more difficult *Single Head* case.

Within the campaign the environment stays persistent, offering common structure between levels. However, as is, the campaign is not targeting smoothly evolving tasks. For example, the starting point of any level is independent of the state at the end of the previous level. Note that a future version of the benchmark might introduce such tasks.

To make the benchmark more amenable to researchers with limited computational resources, we consider three tracks of the challenge. In its most difficult form, the campaign has to be solved as a pure reinforcement learning (RL) problem. For the second and third track, we provide expert agents that have been pre-trained on each level. In-between RL and supervised learning, the track two objective is joint minimisation of the RL loss while learning from a potentially sub-optimal teacher by distillation. This significantly accelerates learning as thoroughly discussed in [31]. The third and simplest track relies on distillation from provided experts alone.

The different tracks thus offer a somewhat gradual shift between a pure RL task to a supervised one, removing some of the complexity of the underlying RL system (e.g. exploration).

Level	Random	Experts	Multi-Task	Finetuning	Dropout [30]	SER [6]
①	187	2000	2000	39	85	1998
②	283	1869	1790	256	246	1009
③	18	2000	2000	3	22	70
④	3	1903	1791	6	1	358
⑤	73	1937	1977	23	44	393
⑥	0	2000	1937	0	3	983
⑦	7	2000	1987	0	1	1193
⑧	0	1973	1963	0	0	390
⑨	930	1998	1997	1311	1308	1808
⑩	58	1152	1114	40	150	1340
⑪	34	1990	1728	1500	1361	1764

Figure 2: Performance on each task at the end of training. Shown is the mean over the last 50 episodes. Note that only methods on the right side of the table should be considered approaches to continual learning. SER: Selective experience replay. More results in the Appendix.

4.2. Baseline results

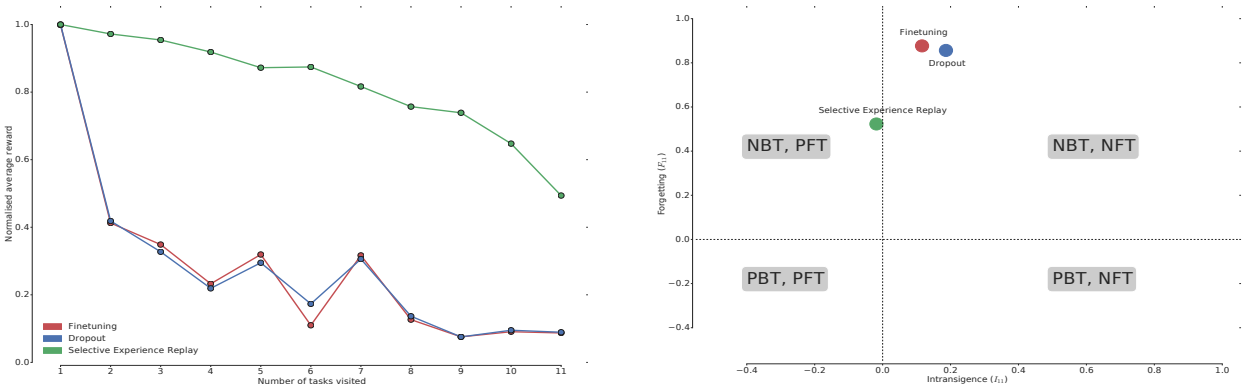
We provide baseline results for the distillation-only track in Figure 2, with a computational budget of $150m$ steps per level. As baseline methods, we compare sequential training without protection (Finetuning), Dropout [15] and SER, a method based on replay of past data [6]. Note that we provide a fairly large replay buffer for SER, giving the algorithm a significant advantage. This is mainly to demonstrate that continual learning is indeed possible.

Further insights into the behaviour of the algorithms is given in Figure 3. We observe that none of the baseline methods achieve sufficient positive forward or backward transfer. Full learning curves are shown in the appendix.

Note that, when seen as a pure RL task, the campaign is extremely difficult[3], where even in the multi-task regime it is difficult to learn within a reasonable budget of frames. We see this as a potential challenge for future approaches. However, given independently trained experts on each level, the challenge becomes significantly more accessible.

In future work, our intent is to explore all aspects of CL for many more approaches in a full variant of the manuscript. We also intend to collate different measures used in practice and apply them across the different baselines. Additionally, we believe there is potential to draw inspiration from related fields, both in terms of methods and evaluation. One particular proposal comes meta-learning, where algorithms are often evaluated in few-shot settings. In contrast, current performance on a past task is typically measured w.r.t to the current policy. A new metric might record the amount of data and learning required to recover performance. Retraining on a previously seen tasks usually results in considerably faster learning. A similar idea is used in [9]. We believe the proposed campaign offers an ideal environment to explore all these evaluations and provide several ways of understanding the shortcomings and strengths of existing approaches.

[3] In worst case scenario up to $12b$ frames were required to train the experts



(a) Average reward over all tasks encountered thus far. Normalised by maximum achievable performance.

(b) Interplay of forgetting and intransigence. P: Positive, N:Negative, F: Forward, B: Backward, T: Transfer

Figure 3: Results highlighting model behaviour over the course of training on the full campaign. A definition of the metrics is provided in the Appendix.

5. Discussion

In this work we propose a new benchmark for continual learning, designed as a campaign for StarCraft II, which offers a natural way to explore the ability of an agent to remember and combine previously seen skills. We offer the problem in multiple tracks, varying in scope and difficulty.

Finally we regard this abstract as being work in progress. We intend to use the provided challenge as a tool to better understand proposed evaluation methods and plan on writing a full report describing our findings. We plan on releasing all environment code necessary to test algorithms on the challenge and aim to also provide implementations for considered baselines.

References

[1] A. Graves, "Generating sequences with recurrent neural networks," 2013.

[2] S. Farquhar and Y. Gal, "Towards robust evaluations of continual learning," CoRR, vol. abs/1805.09733, 2018.

[3] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in ECCV (11), vol. 11215 of Lecture Notes in Computer Science, pp. 556–572, Springer, 2018.

[4] H. Ritter, A. Botev, and D. Barber, "Online structured laplace approximations for overcoming catastrophic forgetting," CoRR, vol. abs/1805.07810, 2018.

[5] X. He and H. Jaeger, "Overcoming catastrophic interference using conceptor-aided backpropagation," in International Conference on Learning Representations, 2018.

[6] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," arXiv preprint arXiv:1802.10269, 2018.

[7] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in Proceedings of the 34th International Conference on Machine Learning (D. Precup and Y. W. Teh, eds.), vol. 70 of Proceedings of Machine Learning Research, (International Convention Centre, Sydney, Australia), pp. 3987–3995, PMLR, 06–11 Aug 2017.

[8] D. Lopez-Paz and M. A. Ranzato, "Gradient episodic memory for continual learning," in Advances in Neural Information Processing Systems 30 (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 6467–6476, Curran Associates, Inc., 2017.

[9] C. Kaplanis, M. Shanahan, and C. Clopath, "Continual reinforcement learning with complex synapses," in Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, pp. 2502–2511, 2018.

[10] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," 2018.

[11] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016.

[12] A. A. Rusu, M. Vecerik, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," CoRL, 2016.

[13] J. Schwarz, J. Luketina, W. M. Czarnecki, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in ICML 2018, 2018.

[14] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," Proceedings of the National Academy of Sciences, 2017.

[15] I. J. Goodfellow, M. Mirza, X. Da, A. C. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgeting in gradient-based neural networks," ICLR, 2014.

[16] P. Ruvolo and E. Eaton, "ELLA: An efficient lifelong learning algorithm," in Proceedings of the 30th International Conference on Machine Learning (S. Dasgupta and D. McAllester, eds.), vol. 28 of Proceedings of Machine Learning Research, (Atlanta, Georgia, USA), pp. 507–515, PMLR, 17–19 Jun 2013.

[17] J. Schmidhuber, "Powerplay: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem," Frontiers in psychology, vol. 4, p. 313, 2013.

[18] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," arXiv preprint arXiv:1710.10628, 2017.

[19] H. Ritter, A. Botev, and D. Barber, "Online structured laplace approximations for overcoming catastrophic forgetting," arXiv preprint arXiv:1805.07810, 2018.

[20] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 806–813, 2014.

[21] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in Advances in Neural Information Processing Systems, pp. 2990–2999, 2017.

[22] Z. Li and D. Hoiem, "Learning without forgetting," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[23] K. Milan, J. Veness, J. Kirkpatrick, M. Bowling, A. Koop, and D. Hassabis, "The forget-me-not process," in Advances in Neural Information Processing Systems, pp. 3702–3710, 2016.

[24] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015.

[25] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in Psychology of learning and motivation, vol. 24, pp. 109–165, Elsevier, 1989.

[26] R. M. French, "Catastrophic forgetting in connectionist networks," Trends in cognitive sciences, vol. 3, no. 4, pp. 128–135, 1999.

[27] J.-P. Pfister, P. Dayan, and M. Lengyel, "Synapses with short-term plasticity are optimal estimators of presynaptic membrane potentials," Nature neuroscience, vol. 13, no. 10, p. 1271, 2010.

[28] R. M. French and N. Chater, "Using noise to compute error surfaces in connectionist networks: A novel means of reducing catastrophic forgetting," Neural computation, vol. 14, no. 7, pp. 1755–1769, 2002.

[29] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist, vol. 2, 2010.

[30] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," arXiv preprint arXiv:1312.6211, 2013.

[31] S. Schmitt, J. J. Hudson, A. Zidek, S. Osindero, C. Doersch, W. M. Czarnecki, J. Z. Leibo, H. Kuttler, A. Zisserman, K. Simonyan, et al., "Kickstarting deep reinforcement learning," arXiv preprint arXiv:1803.03835, 2018.

[32] V. Zambaldi, D. Raposo, A. Santoro, V. Bapst, Y. Li, I. Babuschkin, K. Tuyls, D. Reichert, T. Lillicrap, E. Lockhart, et al., "Relational deep reinforcement learning," arXiv preprint arXiv:1806.01830, 2018.

[33] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al., "Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures," arXiv preprint arXiv:1802.01561, 2018.

[34] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, et al., "Starcraft ii: A new challenge for reinforcement learning," arXiv preprint arXiv:1708.04782, 2017.

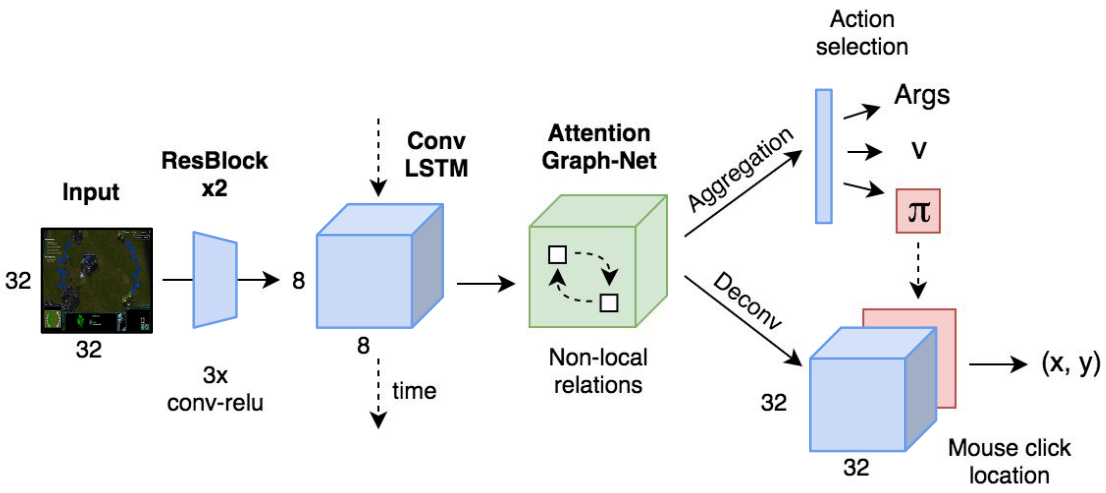


Figure 4: Model architecture.

A. Baseline details

A.1. Reinforcement learning

As an agent architecture, we make use of the model in [32] which we found to be suitable for the targeted difficulty of each level. We made use of identical hyperparameters to Tables 2 & 3 in [32], except for the trajectory unroll length, which we set to 40. Agents were trained using advantage-actor-critic with off-policy correction [33]. We performed a search of moderate size and report hyperparameters for task-specific experts in Table 2. All results are collected by actors dedicated to acting on specific tasks over the course of training. Trajectories used for evaluation did not influence the agent's learning algorithm (i.e. no gradients were computed based on these observations). We approximate an agent's performance on a specific task by taking the mean over the last 50 episode rewards.

In order to ease the exploration problem, we restricted the action set to disallow invalid actions according to the game. As an example, in case a certain building can only be placed nearby a specific resource, we mask actions corresponding to attempts of creating the building on an invalid part of the map. Note that humans are given this information through the user interface. All code necessary to use this restricted action set will be open-sourced.

To further reduce difficulty of the reinforcement learning problem, the campaign can be played from features (semantic layers telling what type of object are at any possible position on the map) rather than pixels. For a detailed explanation, see [34].

A.2. Continual learning algorithms

Selective experience replay [6] was used with the *Global Distribution Matching* selection strategy, aiming to maintain a replay buffer matching the multi-task data distribution of tasks encountered thus far. As this method was mainly used to showcase that progress on the challenge can be made with continual learning algorithms, we allowed for a relatively large replay buffer, storing up to 150m frames ($\approx$ 117,000 training examples assuming a batch size of 32 and unroll length of 40). Each training batch was formed by mixing 50% of the the required size as examples from the buffer.

For Dropout [15], we used a dropout keep probability of p=0.8, only applied after linear layers and disabled during evaluation.

No further design choices were introduced for Finetuning. For the distillation-only track, we train on each level for 150m environment frames, acting according to the student policy.

Level	Learning rate	Baseline cost	Entropy cost	Env. frames at checkpoint/ until convergence
1	$5 \cdot 10^{-5}$	0.01	5.0	500m/83m
2	$5 \cdot 10^{-5}$	0.01	5.0	15.7b/245m
3	$5 \cdot 10^{-5}$	0.01	0.5	2.6b/800m
4	$5 \cdot 10^{-5}$	0.01	5.0	5.5b/2b
5	$1 \cdot 10^{-4}$	0.01	0.1	6b/3.7b
6	$5 \cdot 10^{-5}$	0.01	0.5	1.3b/500m
7	$5 \cdot 10^{-5}$	0.01	5.0	10b/1.8b
8	$5 \cdot 10^{-5}$	0.1	0.5	16.5b/12b
9	$5 \cdot 10^{-5}$	0.01	0.5	15b/920m
10	$1 \cdot 10^{-4}$	0.01	0.5	-/6.5b
11	$5 \cdot 10^{-5}$	0.01	5.0	12b/3.8b

Table 2: Hyper-parameters for task-specific experts.

B. Metrics

In terms of metrics used to evaluate continual learning systems, we rely on the evaluation proposed in [3], who attempt to capture the interplay between forgetting and forward transfer. Specifically, let $r_{k,j} \in [0, 1]$ denote the normalised (by maximum performance) mean episode reward on task j at the end of training on task k.

A metric for the overall performance after training on tasks $1, \ldots, k$ is then:

$$R_k = \frac{1}{k} \sum_{j=1}^{k} r_{k,j} \tag{1}$$

As the authors of [3] note, this metric makes reasoning about forgetting and transfer difficult. Thus, additional metrics that better capture positive and negative transfer on both and the current (forward transfer) and past tasks (backward transfer) should be reported.

A measure of forward transfer is the difference between the reward achieved by a reference model (proposed to be given by the Multi-Task performance) on task k and the current performance of the evaluated method:

$$I_k = r_k^* - r_{k,k} \tag{2}$$

Note $I_k \in [-1, 1]$. This metric is refereed to as intransigence, defined to the observed inability of continual learning systems to learn new tasks due to capacity issues. Thus, in the optimal setting have $I_k = -1$, meaning a method performs optimally on a task that cannot be solved by a Multi-Task method. For a discussion on observed intransigence, see also [13].

Forgetting is defined as the difference between the maximum performance on a task obtained throughout the learning process and its current performance:

$$f_j^k = \max_{l \in 1,\ldots,k-1} r_{l,j} - r_{k,j}, \forall j < k \tag{3}$$

$$F_k = \frac{1}{k-1} \sum_{j=1}^{k-1} f_j^k \tag{4}$$

As with intransigence, we have $F_k \in [-1, 1]$. Negative F_k corresponds to positive backward transfer, positive values indicate forgetting.

C. Campaign description

All details of individual levels in the campaign are provided in Table 3.

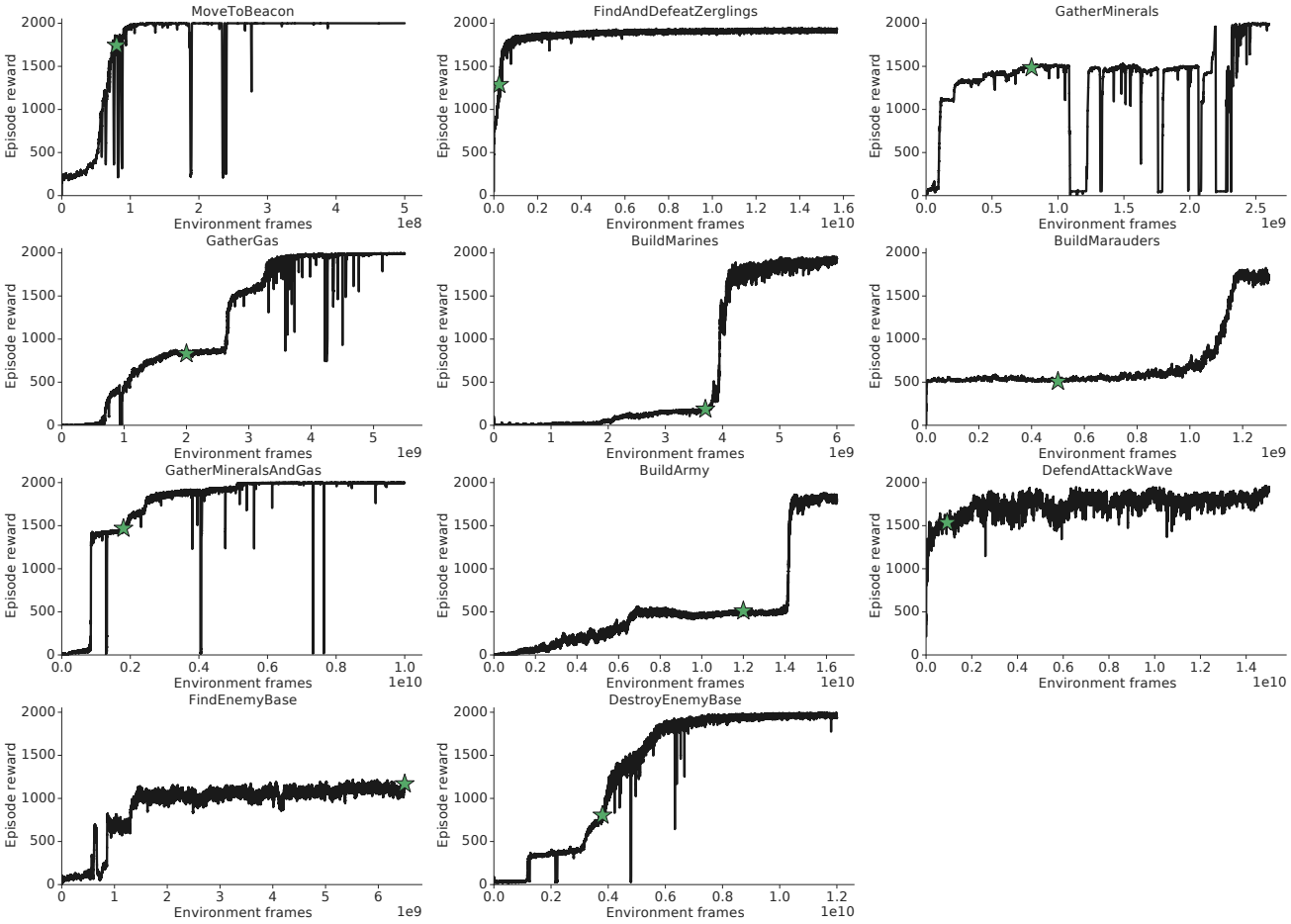


Figure 5: Reward curves for task-specific experts. The green star indicates when a checkpoint was taken for the tracks of the challenge involving distillation from an expert. Note that a varying number of environment frames (up to 16b) were required to train each expert. An appropriate level of smoothing was applied independently for each task to improve clarity of the results.

Level	Time limit	Minerals	Gas	units	Buildings	Enemy Units	Rewards
1: MoveToBeacon	2m			1 Marine			200 per Beacon
2: Find and defeat Zerglings	3m20s			3 Marines		25 Zerglings (respawning)	50 per Zergling
3: Gather Minerals	1m50s	50		12 workers	2 Supply depots		2 per mineral 50 per worker
4: Gather Gas	3m	150		12 workers			4 per unit Gas 100 per refinery
5: Build Marines	7m30s	300		12 workers			180 per Marine 100 for barracks 100 for a supply depot
6: Build Marauders	8m	150	50	12 workers	3 Supply depots Barracks Refinery		300 per Marauder 200 for Tech lab
7: Gather Minerals and Gas	3m10s	75		12 workers			1 per mineral 4 per unit Gas
8: Build Army	6m	600	75	12 workers	2 Supply depots Refinery		100 per Marine 400 per Marauder
9: Defend Attack Waves	5m	800		12 workers 24 Marines 5 Marauders	8 Supply depots Barracks	4 Attack waves	Proportional to damage dealt
10: Find enemy Base	6m	800		24 workers 6 Marauders		64 Zerglings	Proportional to time remaining
11: Destroy enemy Base	14m	400	50	24 workers 12 Marines 4 Marauders	2 Supply depots Barracks Refinery	8 Roaches 8 Mutalisk	Proportional to damage dealt

Table 3: Full campaign description.

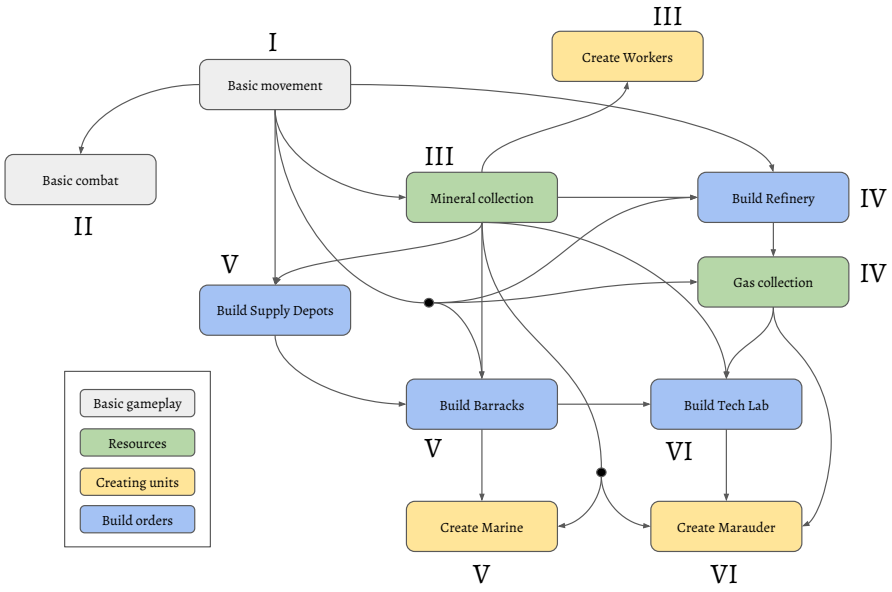


Figure 6: Skills graph.

C.1. Additional results

We provide additional results for the distillation-only in Figure **??**. The figure shows the performance on each task over the course of training. An interesting observation in the results is the presence of significant backward transfer. For instance, it can be seen that the performance on "GatherMinerals" increases when the agent is learning task "GatherMineralsAndGas". The same phenomenon is observed when acting on "BuildMarines", which requires mineral collection to afford the creation of new units. Note that no task re-visits are allowed.

C.2. In-game screenshots

In-game screenshots are shown in figure 9

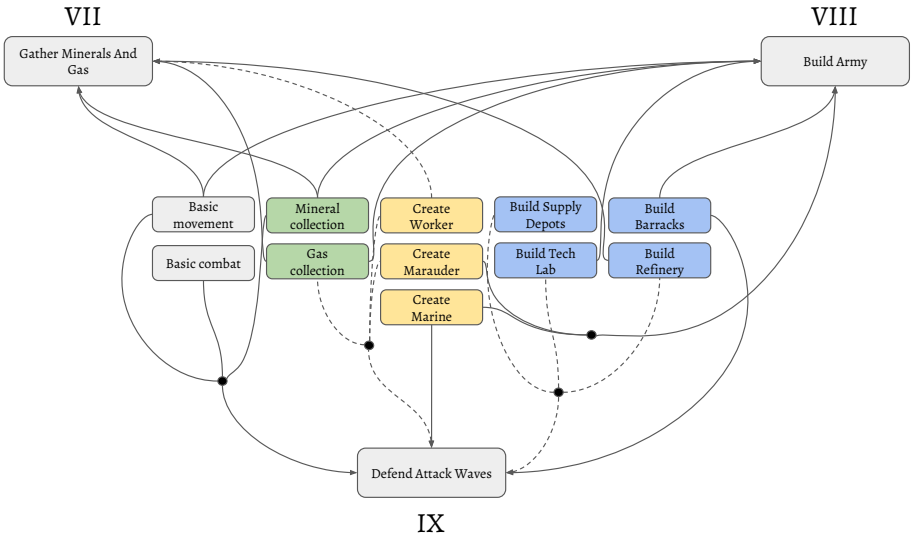


Figure 7: Skills graph. Dashed arrows indicate skills that might be useful both are not necessarily required.

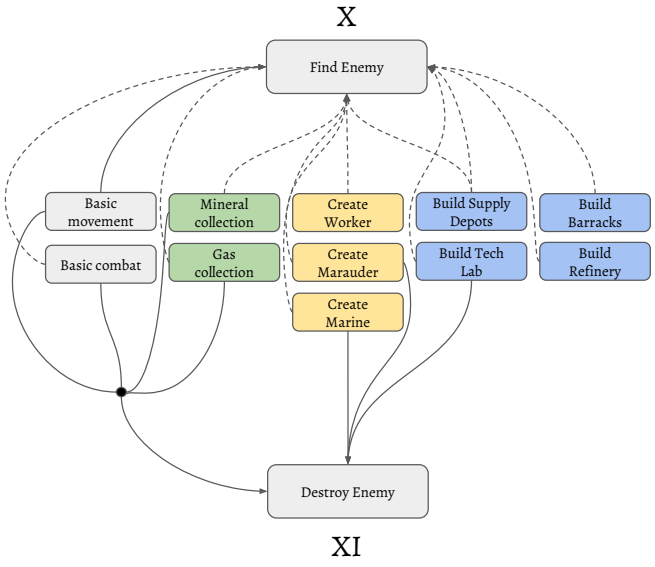


Figure 8: Skills graph. Dashed arrows indicate skills that might be useful both are not necessarily required.

Figure 9: Example in-game screenshots on levels 1, 4, 9 and 11. Best viewed on a computer..