Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

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

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Carpenter, G. A. and Grossberg, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer vision, graphics, and image processing*, 37(1):54–115, 1987.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018a.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018b.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, J. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 1282–1289. PMLR, 2019.

Crandall, J. W., Oudah, M., Ishowo-Oloko, F., Abdallah, S., Bonnefon, J.-F., Cebrian, M., Shariff, A., Goodrich, M. A., Rahwan, I., et al. Cooperating with machines. *Nature communications*, 9(1):1–12, 2018.

Foerster, J., Song, F., Hughes, E., Burch, N., Dunning, I., Whiteson, S., Botvinick, M., and Bowling, M. Bayesian action decoder for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 1942–1951, 2019.

Foerster, J. N., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

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

Henderson, P., Chang, W.-D., Shkurti, F., Hansen, J., Meger, D., and Dudek, G. Benchmark environments for multitask learning in continuous domains. *arXiv preprint arXiv:1708.04352*, 2017.

Hong, Z.-W., Su, S.-Y., Shann, T.-Y., Chang, Y.-H., and Lee, C.-Y. A deep policy inference q-network for multi-agent systems. *arXiv preprint arXiv:1712.07893*, 2017.

Hu, H. and Foerster, J. N. Simplified action decoder for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1912.02288*, 2019.

Hu, H., Lerer, A., Peysakhovich, A., and Foerster, J. " other-play" for zero-shot coordination. *arXiv preprint arXiv:2003.02979*, 2020.

Isele, D. and Cosgun, A. Selective experience replay for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Kaplanis, C., Shanahan, M., and Clopath, C. Continual reinforcement learning with complex synapses. In *International Conference on Machine Learning*, pp. 2497–2506. PMLR, 2018.

Kaplanis, C., Shanahan, M., and Clopath, C. Policy consolidation for continual reinforcement learning. *arXiv preprint arXiv:1902.00255*, 2019.

Kapturowski, S., Ostrovski, G., Quan, J., Munos, R., and Dabney, W. Recurrent experience replay in distributed reinforcement learning. In *International conference on learning representations*, 2018.

Kempka, M., Wydmuch, M., Runc, G., Toczek, J., and Jaśkowski, W. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8. IEEE, 2016.

Khetarpal, K., Riemer, M., Rish, I., and Precup, D. Towards continual reinforcement learning: A review and perspectives. *arXiv preprint arXiv:2012.13490*, 2020.

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

Lomonaco, V. and Maltoni, D. Core50: a new dataset and benchmark for continuous object recognition. *arXiv preprint arXiv:1705.03550*, 2017.

Lomonaco, V., Desai, K., Culurciello, E., and Maltoni, D. Continual reinforcement learning in 3d non-stationary environments. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition Workshops, pp. 248–249, 2020.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*, pp. 6467–6476, 2017.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Mirzadeh, S. I., Farajtabar, M., Pascanu, R., and Ghasemzadeh, H. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

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

Platanios, E. A., Saparov, A., and Mitchell, T. Jelly bean world: A testbed for never-ending learning. *arXiv preprint arXiv:2002.06306*, 2020.

Prabhu, A., Torr, P. H., and Dokania, P. K. Gdumb: A simple approach that questions our progress in continual learning. In *Proceedings of the European Conference on Computer Vision*, 2020.

Premack, D. and Woodruff, G. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4): 515–526, 1978.

Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., and Botvinick, M. Machine theory of mind. In *International conference on machine learning*, pp. 4218–4227. PMLR, 2018.

Ring, M. B. Child: A first step towards continual learning. In *Learning to learn*, pp. 261–292. Springer, 1998.

Roady, R., Hayes, T. L., Vaidya, H., and Kanan, C. Stream-51: Streaming classification and novelty detection from videos. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pp. 4528–4537. PMLR, 2018.

Shi, X., Li, D., Zhao, P., Tian, Q., Tian, Y., Long, Q., Zhu, C., Song, J., Qiao, F., Song, L., Guo, Y., Wang, Z., Zhang, Y., Qin, B., Yang, W., Wang, F., Chan, R. H. M., and She, Q. Are we ready for service robots? the openloris-scene datasets for lifelong slam, 2020.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

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

Stojanov, S., Mishra, S., Thai, N. A., Dhanda, N., Humayun, A., Yu, C., Smith, L. B., and Rehg, J. M. Incremental object learning from contiguous views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8777–8786, 2019.

Stone, P., Kaminka, G. A., Kraus, S., Rosenschein, J. S., et al. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI*, pp. 6, 2010.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

Tesauro, G. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Thrun, S. Lifelong learning algorithms. In *Learning to learn*, pp. 181–209. Springer, 1998.

Xu, M., Ding, W., Zhu, J., Liu, Z., Chen, B., and Zhao, D. Task-agnostic online reinforcement learning with an infinite mixture of gaussian processes, 2020.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 70:3987, 2017.

Zhang, A., Ballas, N., and Pineau, J. A dissection of overfitting and generalization in continuous reinforcement learning. *arXiv preprint arXiv:1806.07937*, 2018.

Zhang, K., Yang, Z., and Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pp. 7693–7702. PMLR, 2019.

Appendices

A. Pool of Agents

Here is the cross-play matrix of all the 100 agents trained with different MARL algorithms. There are five types of architectures (Table 3) with two different seeds per MARL algorithm.

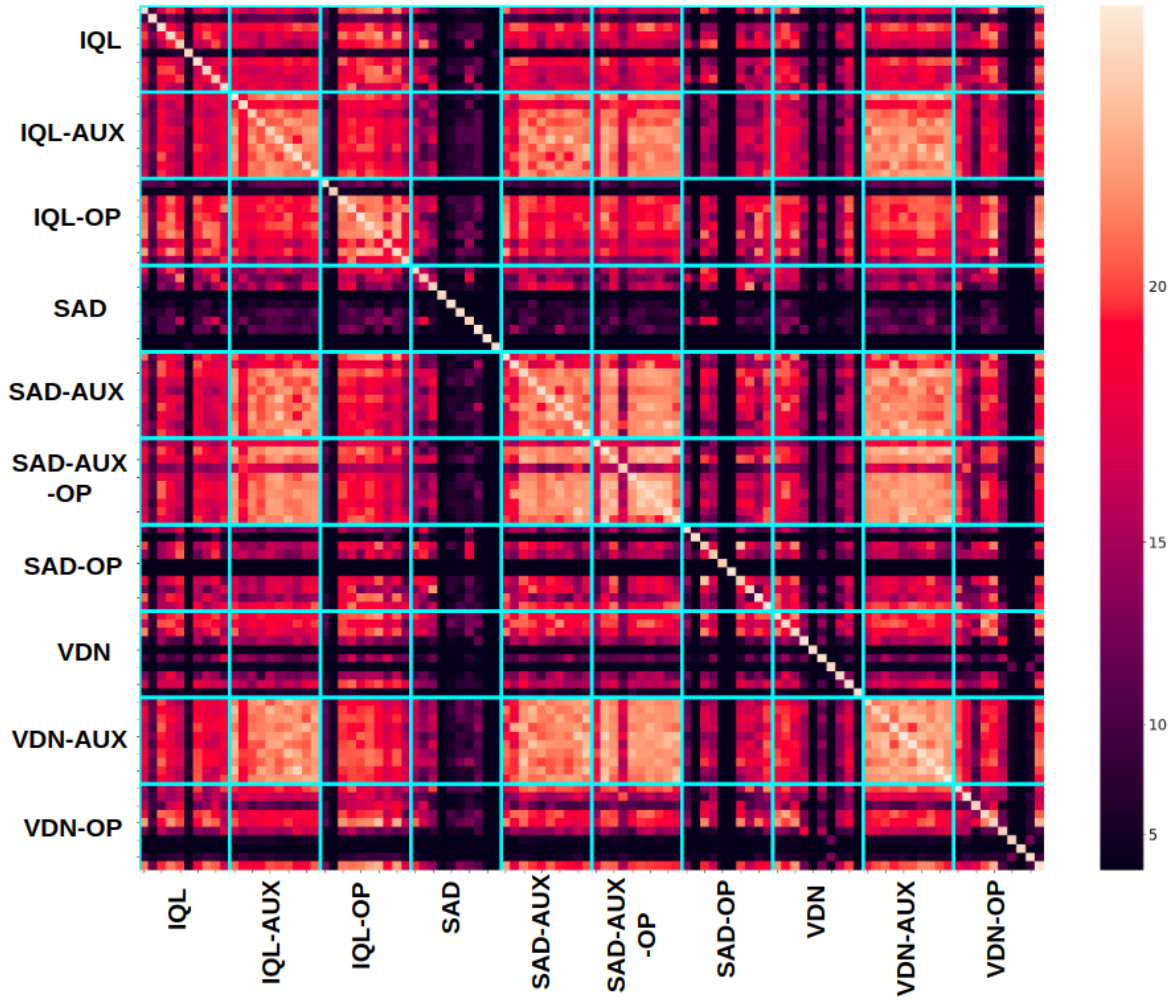

Figure 6. The pool of 100 agents pre-trained through Self-Play using different MARL methods (IQL/VDN/OP/AUX/SAD, and their combinations). 10 agents having 5 different architectures with 2 seeds are generated with each of these MARL methods. $(i, j)_{th}$ element is the average score of agent i paired with j over 5k games. The diagonal entries indicate SP scores.

B. List of agents

In this section, we present the exact type of agents that we use as the *learner* and its *partners* in both *easy* and *hard* settings, as well as the set of 10 partners used in section 5.3 and in Appendix C. All these settings have an IQL agent of Type-2 as the *learner* and a sequence of 5/10 agents (can be extended to any number of agents) as its partners. Table 3 has details of the exact architectures corresponding to these *Types*.

- **Easy** : *Learner* — $\left\{ \text{IQL (Type-2)} \right\}$
  Partners — $\left\{ \text{IQL (Type-1), VDN (Type-3), VDN (Type-5), IQL+OP (Type-2), VDN+OP (Type-5)} \right\}$.
- **Hard** : *Learner* — $\left\{ \text{IQL (Type-2)} \right\}$
  Partners — $\left\{ \text{VDN+OP (Type-3), VDN (Type-4), VDN (Type-5), IQL+OP (Type-3), VDN (Type-3)} \right\}$.
  The partner agents in the Figure 2 are these *hard* agents.
- **10 agents**: *Learner* — $\left\{ \text{IQL (Type-2)} \right\}$
  Partners — $\left\{ \text{VDN (Type-2), VDN (Type-3), IQL+OP (Type-2), VDN+OP (Type-5), IQL (Type-4), VDN+OP (Type-1), VDN (Type-4), IQL+OP (Type-3), VDN+OP (Type-1), VDN (Type-5)} \right\}$.

The below are the set of 20 held-out agents that we use for across method evaluation in Tables 6 and 7.

Inter-CP : $\left\{ \text{IQL (Type-1), IQL (Type-3), IQL+OP (Type-4), IQL+OP (Type-5), VDN+AUX (Type-2), VDN+AUX (Type-3), SAD+OP (Type-3), SAD+OP (Type-1), SAD+OP+AUX (Type-3), SAD+OP+AUX (Type-1), SAD+AUX (Type-3), SAD+AUX (Type-1), SAD (Type-3), SAD (Type-2), IQL+AUX (Type-3), IQL+AUX (Type-1), VDN (Type-4), VDN (Type-2), VDN+OP (Type-5), VDN+OP(Type-4)} \right\}$.

Table 3. Exact architectures used in the pool.

AGENT	RNN TYPE	NUM OF FEED-FORWARD LAYERS	NUM OF RNN LAYERS	RNN HID DIM
TYPE-1	LSTM	1	1	256
TYPE-2	LSTM	2	2	256
TYPE-3	LSTM	1	2	512
TYPE-4	GRU	1	2	256
TYPE-5	GRU	2	1	512

C. LLL algorithms benchmarks

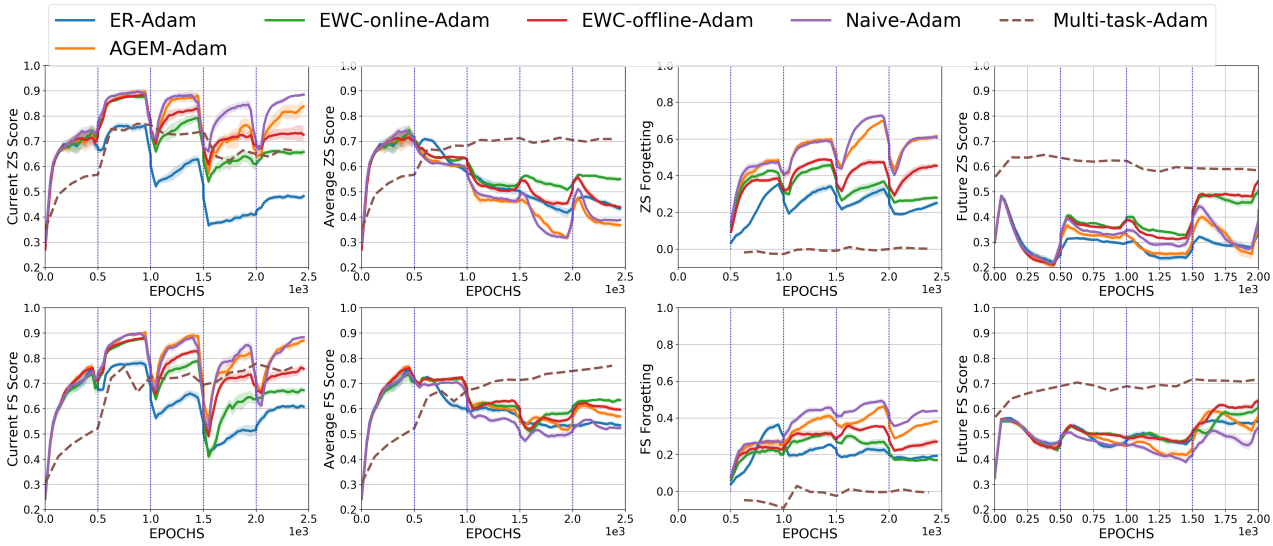

Figure 7. Zero-shot (top row) and Few-shot (bottom row) performance of different LLL algorithms with Adam optimizer on *hard* task. From left to right: current score, average score, forgetting and average future score respectively.

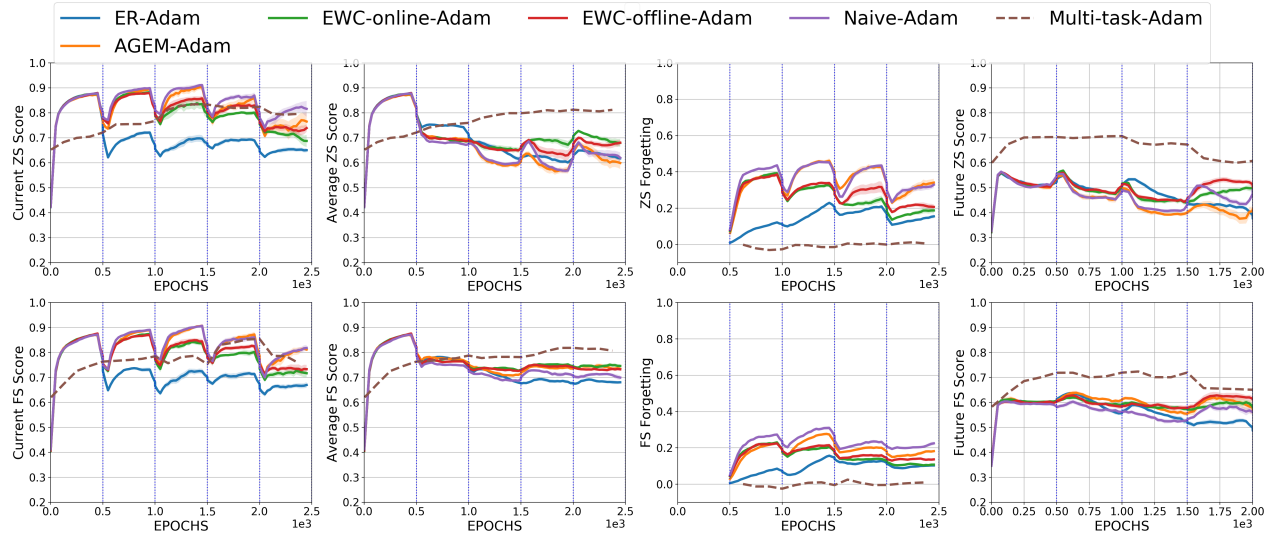

Figure 8. Zero-shot (top row) and Few-shot (bottom row) performance of different LLL algorithms with Adam optimizer on *easy* task. From left to right: current score, average score, forgetting and average future score respectively.

For Figures 7-10, the *learner* is pre-trained with IQL method and is continually trained with either *hard* or *easy* agents mentioned in section B.

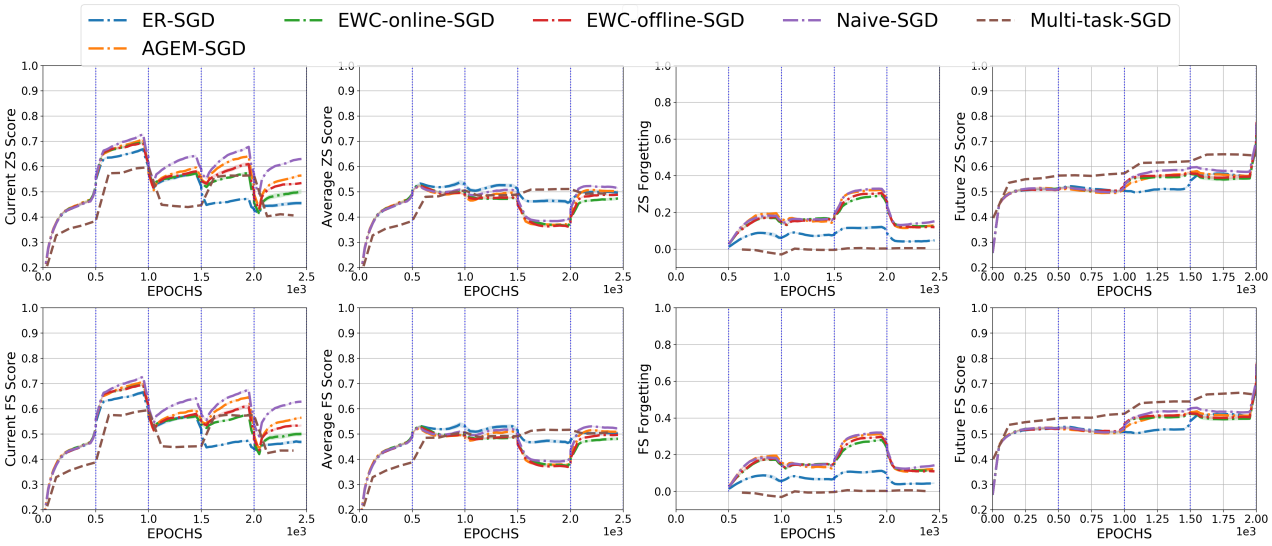

Figure 9. Zero-shot (top row) and Few-shot (bottom row) performance of different LLL algorithms with SGD optimizer on *hard* task. From left to right: current score, average score, forgetting and average future score respectively.

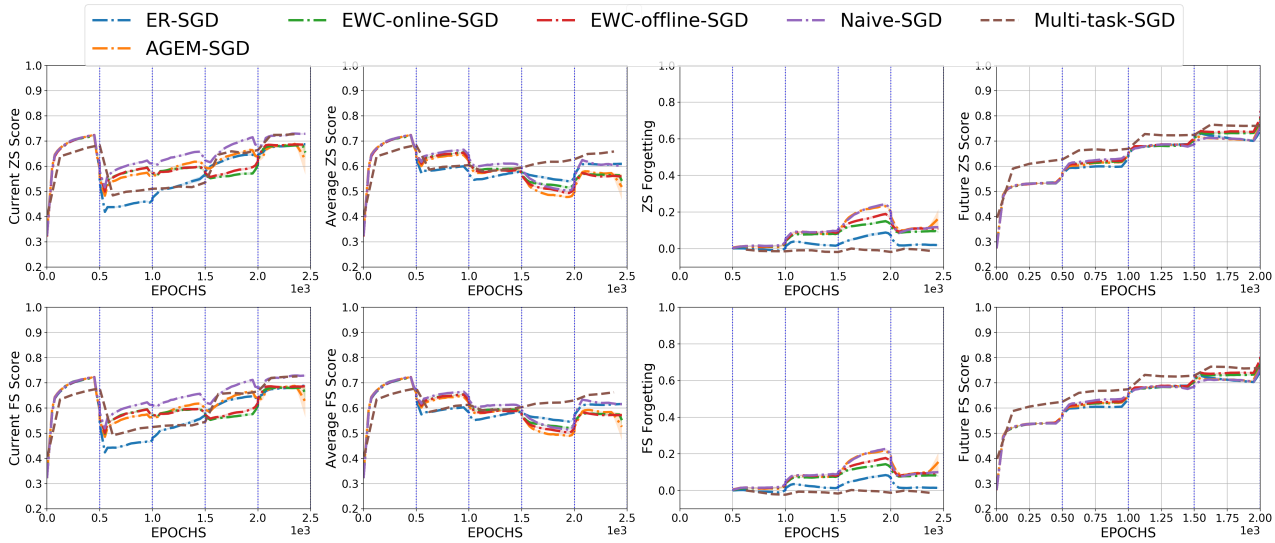

Figure 10. Zero-shot (top row) and Few-shot (bottom row) performance of different LLL algorithms with SGD optimizer on *easy* task. From left to right: current score, average score, forgetting and average future score respectively.

The sequential order of partners were chosen at random from the pre-trained pool in both *easy* and *hard* setting. Careful curation of the partner ordering and its effect on lifelong learning is left as future work.

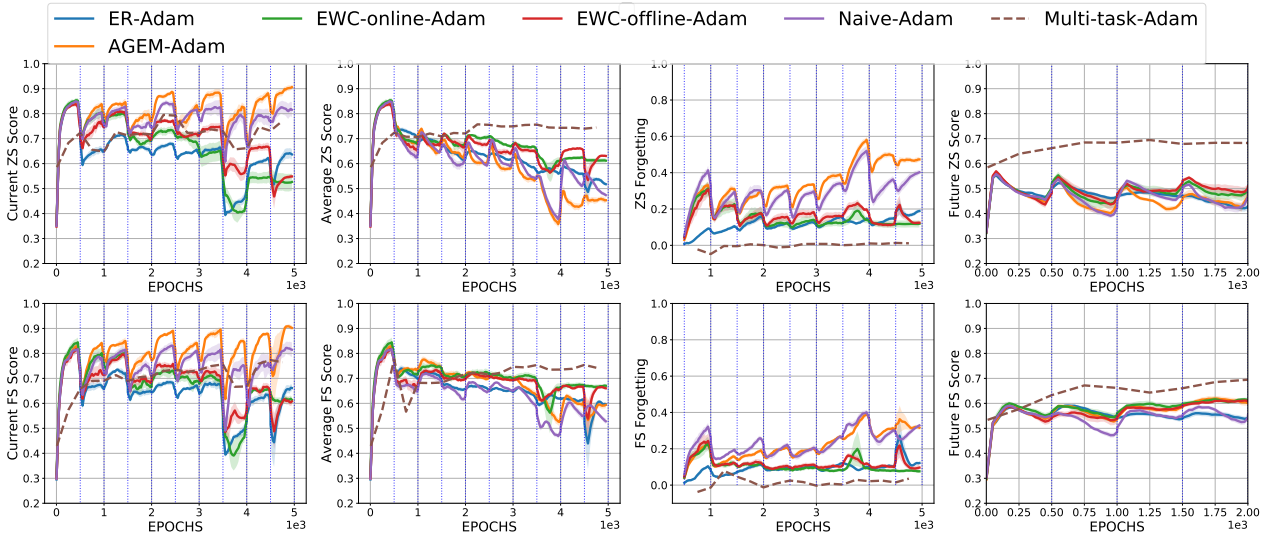

Figure 11. Zero-shot (top row) and Few-shot (bottom row) performance of different LLL algorithms with Adam optimizer on 10 tasks. From left to right: current score, average score, forgetting and average future score respectively.

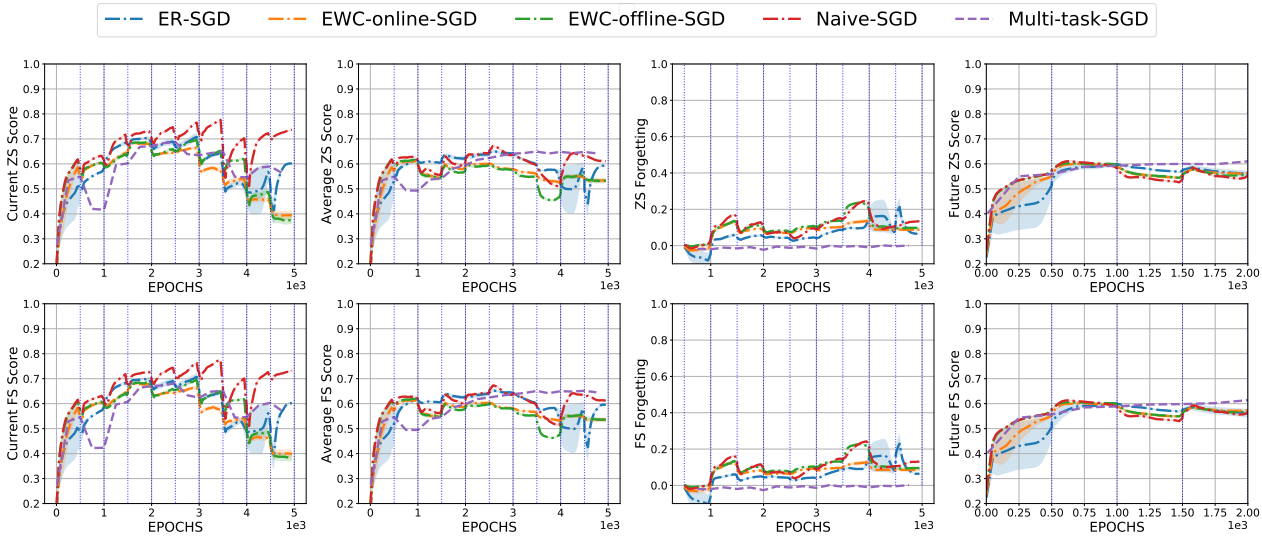

Figure 12. Zero-shot (top row) and Few-shot (bottom row) performance of different LLL algorithms with SGD optimizer on 10 tasks. From left to right: current score, average score, forgetting and average future score respectively.

For Figures 11-12, the *learner* is pre-trained with IQL method and is continually trained with 10 partners mentioned in section B.

D. All hyperparameters and experiment details

Table 4. All common hyperparameters and their description.

HYPERPARAMETERS	VALUE	DESCRIPTION
$batchsize$	32	BATCHSIZE USED FOR BOTH TRAINING AND FEW-SHOT EVALUATION
max_train_steps	200M	MAXIMUM NUMBER OF TRAINING STEPS PER TASK
max_eval_steps	500K	MAXIMUM NUMBER OF TRAINING STEPS DURING FEW-SHOT EVALUATION
$burn\_in\_frames$	10K	NUMBER OF SAMPLES USED TO WARM-UP REPLAY BUFFER
$eval\_burn\_in\_frames$	1K	NUMBER OF SAMPLES USED TO WARM-UP EVALUATION REPLAY BUFFER
$replay\_buffer\_size$	32768	REPLAY BUFFER SIZE DURING CONTINUAL TRAINING
$eval\_replay\_buffer\_size$	10000	REPLAY BUFFER SIZE FOR FEW-SHOT EVALUATION
$epoch\_len\_size$	200	NUMBER OF GRADIENT UPDATES PER EPOCH
$eval\_epoch\_len\_size$	50	NUMBER OF GRADIENT UPDATES FOR FEW-SHOT EVALUATION
$eval\_freq$	25	LEARNER IS EVALUATED AFTER EACH 25 EPOCHS
num_thread	10	NUMBER OF THREADS USED FOR R2D2 ACTORS
$num\_game\_per\_thread$	80	NUMBER OF GAME PER THREADS USED FOR R2D2 ACTORS
$eval\_num\_thread$	10	NUMBER OF THREADS USED FOR R2D2 ACTORS DURING FEW-SHOT EVALUATION
$eval\_num\_game\_per\_thread$	10	NUMBER OF GAMES PER THREADS USED FOR R2D2 ACTORS DURING FEW-SHOT EVALUATION
$sgd\_momentum$	0.8	MOMENTUM FOR SGD OPTIMIZER

Table 5. Specific hyperparameters to each algorithm and their description

HYPERPARAMETERS	VALUE	DESCRIPTION
ewc_lambda	50000	EWC
ewc_gamma	1	EWC
$replay\_buffer\_size$	163840	MULTI-TASK

E. MARL algorithms benchmarks

In order to obtain the *Intra-CP* scores for the existing MARL methods in the Table 2 and Table 7 (referenced as BEST in caption), we take the agent from each training method that performs best with the **Inter-CP** agents listed above in section B and evaluate them with the other 9 agents of the same method from the pretrained pool (Figure 6). However, in order to obtain the *Intra-CP* scores for each MARL method in the Table 6 (referenced as AVG in caption), we pick one agent, evaluate it with the rest (barring itself) and repeat the same for all other agents. The average of these scores are reported. A similar process is followed for reporting *Inter-CP* scores. The method of evaluating our LLL methods remains consistent in all the Tables(2, 6, 7). For IQL+ER, we start with the IQL agent that has the least cross-play score and train it with *Hard* agents sequentially using ER algorithm. In the case of IQL+AUX+ER, we start with an IQL agent that is pre-trained with AUX and is continually trained with the *Hard* agents using ER algorithm. This continually trained agent is then evaluated with 9 other agents in either IQL or IQL+AUX respectively in order to obtain *Intra-CP* scores. However, please note that the auxiliary task is used only during pre-training and is not used during continual training. Note that the middle row in the Table 7 is generated using the latest models released by (Hu et al., 2020).

Table 6. AVG : Comparison with other MARL algorithms on self-play (SP), cross-play evaluation scores within method (*Intra-CP*), and across different methods (*Inter-CP*). C: centralized training, GA: agents share their greedy action along with their standard action, L: true labels of cards needed, SYM: symmetries of the game needed upfront, P: require access to some pre-trained agents in sequence, UP: Having access to all the fixed pre-trained agents at the same time. (↑ / ↓ = Difference in score after continual training red: pre-trained with MARL method, blue: trained continually with LLL method)

TRAINING METHOD	SP	INTRA-CP	INTER-CP	LIMITATIONS
SAD	23.78 ± 0.03	4.38 ± 0.66	8.40 ± 0.23	C+GA
SAD+AUX	23.82 ± 0.02	21.15 ± 0.26	17.01 ± 0.22	C+GA+L
SAD+OP	23.67 ± 0.03	12.00 ± 0.86	12.79 ± 0.24	C+SYM+GA
SAD+AUX+OP	23.88 ± 0.03	22.01 ± 0.03	17.08 ± 0.22	C+SYM+L+GA
IQL + ER	20.91 ± 0.05 (↓ 2.98)	15.73±0.39 (↑ **7.06**)	16.32±0.21 (↑ **8.09**)	P
IQL+AUX + ER	22.34± 0.06 (↓ 1.46)	20.90± 0.06 (↓0.15)	**19.17±0.22**(↑**1.33**)	L+P
IQL + MULTI-TASK	20.93±0.09 (↓ 2.96)	16.05± 0.30(↑ **7.38**)	**17.88**±0.17 (↑ **9.65**)	UP

Table 7. BEST : Comparison with other MARL algorithms on self-play (SP), cross-play evaluation scores within method (*Intra-CP*), and across different methods (*Inter-CP*). C: centralized training, GA: agents share their greedy action along with their standard action, L: true labels of cards needed, SYM: symmetries of the game needed upfront, P: require access to some pre-trained agents in sequence, UP: Having access to all the fixed pre-trained agents at the same time. (↑ / ↓ = Difference in score after continual training, red: pre-trained with MARL method, blue: trained continually with LLL method, * : results obtained using models released by (Hu et al., 2020))

Training Method	SP	Intra-CP	Inter-CP	Limitations
SAD	23.85 ± 0.03	7.70 ± 0.69	14.60 ± 0.24	C + GA
SAD + AUX	23.57 ± 0.03	20.97 ± 0.80	18.51 ± 0.23	C + GA + L
SAD + OP	24.14 ± 0.03	10.10 ± 0.87	16.09 ± 0.25	C + Sym + GA
SAD + AUX + OP	23.40 ± 0.04	21.23 ± 0.25	17.77 ± 0.23	C + Sym + L + GA
SAD*	23.97 ± 0.04	2.52 ± 0.0.34	11.46 ± 0.35	C + GA
SAD + AUX*	24.09 ± 0.03	17.65 ± 0.69	17.60 ± 0.42	C + GA + L
SAD + OP*	23.93 ± 0.02	15.32 ± 0.65	17.50 ± 0.34	C + Sym + GA
SAD + AUX + OP*	24.06 ± 0.02	22.07 ± 0.11	17.45 ± 0.38	C + Sym + L + GA
IQL + ER	20.91 ± 0.05 (↓ 2.98)	15.73 ± 0.39 (↑ **7.06**)	16.32 ± 0.21 (↑ **8.09**)	P
IQL + AUX + ER	22.34 ± 0.06 (↓ 1.46)	20.90 ± 0.06 (↓ 0.15)	**19.17 ± 0.22** (↑ **1.33**)	L + P
IQL + Multi-task	20.93 ± 0.09 (↓ 2.96)	16.05 ± 0.30 (↑ **7.38**)	**17.88 ± 0.17** (↑ **9.65**)	UP