A Technical details

A.1 Hyperparameters

parameter	value
learning rate	1×10^{-3}
batch size	128
discount factor γ	0.99
target output std σ_t	0.089
replay buffer size	1M

Table 5: Hyperparameters used for the underlying SAC algorithm.

We use hyperparameters for the underlying SAC algorithm as in [47]; for completeness we present them in Table 5. We use the Adam optimizer with the default values of $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

Below we list tested hyperparameter ranges and final values for individual CL methods; for baselines included in Continual World, ranges were based on the optimal value reported in [47]:

- L2: we search regularization parameter in $\{1000, 10000, 100000, 1000000\}$. Selected value is 100000.
- EWC: we search regularization parameter in $\{1000, 10000, 100000, 1000000\}$. Selected value is 10000.
- MAS: we search regularization parameter in $\{1000, 10000, 100000, 1000000\}$. Selected value is 1000.
- VCL: we search regularization parameter in $\{0.01, 0.1, 1, 10\}$. Selected value is 1.
- PackNet: we use the number of retraining steps $= 100000$, and we use global gradient norm clipping 2×10^{-5}, as in [47].
- Perfect Memory: we search over batch size in $\{128, 256, 512\}$. Selected value is 256.
- A-GEM: we search over episodic batch size in $\{128, 256\}$. Selected value is 128.
- Behavioral cloning: we search over actor's regularization coefficient in $\{10, 100, 1000\}$, episodic batch size in $\{128, 256\}$; selected values are $(100, 128)$. Episodic memory per task is set to $M = 10000$, and we use global gradient norm clipping 0.1 for stability. After we tuned these parameters, we searched regularization coefficient for the critic from values $\{0, 1 \times 10^{-4}, 1 \times 10^{-3}, 0.01, 0.1, 1, 10, 100\}$; the selected value is 0 (no regularization for critic).

A.2 SAC

In this work, we use SAC [15] as an underlying RL algorithm. It is an off-policy actor-critic algorithm, based on the maximum entropy principle. The critic approximates the entropy-adjusted Q-function under the current policy and is trained with the following loss [1]:

$$\mathbb{E}_{(s,a,r,s',d)}\left[(Q_{\phi_i}(s,a) - \hat{Q})^2\right],$$

where

$$\hat{Q} := r + \gamma(1-d)\left(\min_{j=1,2} Q_{\phi_{target,j}}(s',a') - \alpha \log(\pi_\theta(a'|s')\right), \quad a' \sim \pi_\theta(\cdot|s').$$

The actor searches for actions that maximize the Q-function, i.e. it maximizes the following objective:

$$\mathbb{E}_{s,a\sim\pi}\left[Q^\pi(s,a) - \alpha \log \pi(a|s)\right].$$

Algorithm 1 Behavioral cloning

input: number of tasks N, SAC actor π, SAC critics q_1, q_2, expert buffer $\mathcal{D}_{ex} := \emptyset$, expert batch size B_{ex}, regularization coefficients for actor and critic r_{actor}, r_{critic}
Train SAC on task t_1.
Gather actor and critic outputs as targets to populate $\mathcal{D}_{ex}$.
for task t_i, $i := 2, \ldots, N$ **do**
    Train SAC on task t_i, with the following modified update rule:
        Compute the SAC loss l_{SAC}
        Sample $(o^{1..B_{ex}}, \hat{\pi}^{1..B_{ex}}, \hat{q}_1^{1..B_{ex}}, \hat{q}_2^{1..B_{ex}}) \sim \mathcal{D}_{ex}$ ▷ state, tgt policy dist, tgt preds
        $l_{actor} := \frac{1}{B_{ex}} \sum_{j=1}^{B_{ex}} D_{KL}(\pi(o^j) \parallel \hat{\pi}^j)$
        $l_{critic} := \frac{1}{B_{ex}} \sum_{j=1}^{B_{ex}} \left((q_1(o^j) - \hat{q}_1^j)^2 + (q_2(o^j) - \hat{q}_2^j)^2 \right)$.
        Minimize $l_{\text{SAC}} + r_{actor} \cdot l_{actor} + r_{critic} \cdot l_{critic}$
    Gather actor and critic outputs as targets to extend $\mathcal{D}_{ex}$.
end for

B Details on the methods

B.1 Behavioral cloning

In Behavioral cloning (see Algorithm 1), at the end of each task, we randomly sample a subset from the SAC buffer, label it using the outputs of the current (trained) networks and add it to a separate buffer as "expert" data. In the subsequent tasks, we add auxiliary losses to the SAC's objective to imitate these expert data; for the actor, we use the KL divergence, and for the critics, we use the L2 loss (which can be derived as KL divergence between mean-parameterized Gaussian distributions).

B.2 Baselines from Continual World benchmark

Here, we briefly describe the baselines from Continual World that we are using, and refer the reader to Appendix B of [47] for more details. The simplest continual learning baseline is **Fine-tuning** where the model is simply trained on the sequence of tasks without applying any kind of mechanism for avoiding forgetting or encouraging forward transfer. Then, we consider three standard methods from the family of regularization methods, **L2** [37], **EWC** [37] and **MAS** [2] which apply quadratic regularization to network weights while using different mechanisms for establishing per-parameter regularization coefficients. **VCL** [31] applies variational inference to Bayesian neural networks to facilitate continual learning. Using a different approach, **A-GEM** [6] projects the gradients according to constraints obtained from data from the buffer. **Perfect memory** is a simple method that keeps all of the data from the past tasks in the SAC's buffer. Finally, **PackNet** [26] freezes a fraction of the parameters of the network after each task so that the performance does not deteriorate on the previous tasks.

C Results for all methods

Results for all methods, including the baselines from Continual World, as well as `ClonEx-SAC` and its ablations, are presented in Table 6 (CW10) and Table 7 (CW20).

D Additional experiments on transfer in isolation

D.1 Additional matrices

In the main text (see Section 4) we summarized the performance of different combinations of carried over components by calculating the statistics over the whole transfer matrix and presented them in Table 2. In this section, we provide the full matrices to give a broader picture of the experiments. Figure 3 contains the result of transferring combinations of components as described previously in the main text. As mentioned earlier in Section 4.1, transferring the actor, the critic, and the exploration policy simultaneously leads to the best results.

Table 6: Results of all the methods on CW10 sequence. Average performance, forgetting, and forward transfer are shown in columns. 90% bootstrap confidence intervals are shown.

method	performance	forgetting	f. transfer
Fine-tuning	0.10 [0.10, 0.10]	0.75 [0.73, 0.76]	0.31 [0.27, 0.34]
Fine-tuning, best-return exploration	0.10 [0.10, 0.11]	0.73 [0.71, 0.75]	0.30 [0.25, 0.34]
A-GEM	0.13 [0.12, 0.14]	0.68 [0.66, 0.70]	0.26 [0.22, 0.29]
ClonEx-SAC	0.86 [0.84, 0.87]	0.02 [0.01, 0.04]	0.44 [0.42, 0.46]
Behavioral cloning	0.84 [0.81, 0.86]	0.02 [0.01, 0.03]	0.41 [0.38, 0.43]
EWC	0.64 [0.60, 0.68]	0.06 [0.03, 0.09]	0.04 [-0.04, 0.12]
L2	0.53 [0.49, 0.58]	0.02 [-0.00, 0.04]	-0.34 [-0.47, -0.21]
MAS	0.53 [0.50, 0.57]	0.11 [0.09, 0.13]	-0.06 [-0.14, -0.00]
PackNet	0.84 [0.81, 0.86]	-0.01 [-0.02, 0.00]	0.26 [0.22, 0.29]
Perfect memory	0.27 [0.24, 0.30]	0.03 [0.00, 0.05]	-1.13 [-1.23, -1.04]
VCL	0.55 [0.51, 0.59]	-0.03 [-0.05, 0.01]	-0.37 [-0.47, -0.28]

Table 7: Results of all the methods on CW20 sequence. Average performance, forgetting, and forward transfer are shown in columns. 90% bootstrap confidence intervals are shown.

method	performance	forgetting	f. transfer
Fine-tuning	0.05 [0.05, 0.05]	0.74 [0.72, 0.76]	0.19 [0.15, 0.23]
Fine-tuning, best-return exploration	0.05 [0.05, 0.06]	0.74 [0.72, 0.76]	0.21 [0.18, 0.24]
A-GEM	0.07 [0.06, 0.08]	0.69 [0.68, 0.71]	0.13 [0.10, 0.17]
ClonEx-SAC	0.87 [0.86, 0.88]	0.02 [0.01, 0.03]	0.54 [0.52, 0.55]
Behavioral cloning	0.83 [0.81, 0.85]	0.02 [0.01, 0.03]	0.36 [0.34, 0.38]
EWC	0.61 [0.59, 0.63]	0.03 [0.01, 0.05]	-0.15 [-0.21, -0.09]
L2	0.50 [0.46, 0.53]	0.00 [-0.01, 0.01]	-0.48 [-0.59, -0.37]
MAS	0.53 [0.49, 0.57]	0.07 [0.05, 0.09]	-0.16 [-0.22, -0.10]
PackNet	0.80 [0.79, 0.82]	0.00 [-0.01, 0.01]	0.18 [0.14, 0.22]
Perfect memory	0.09 [0.06, 0.12]	0.10 [0.08, 0.12]	-1.32 [-1.41, -1.24]
VCL	0.50 [0.48, 0.53]	-0.01 [-0.03, 0.01]	-0.45 [-0.53, -0.37]

D.2 Transferring the optimizer

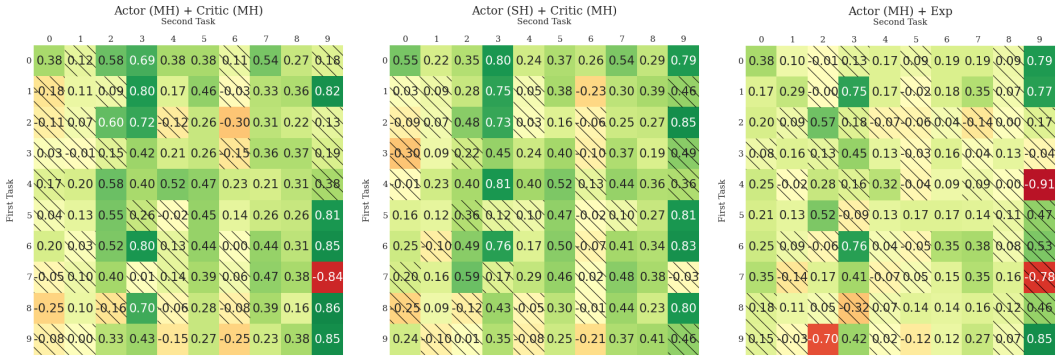
Since we are using the Adam optimizer (Appendix A), we can also try to transfer the running statistics of the gradients used by the optimizer. However, as we show in Table 8, this effect is negligible.

Setting	FT	FT (no diag)	# pos.	# neg.	# neutral
Transfer optimizer	0.02 [-0.02, 0.06]	0.02 [-0.02, 0.06]	19	5	76

Table 8: Results when carrying over the optimizer. Its impact is negligible.

D.3 Freezing the critic's parameters

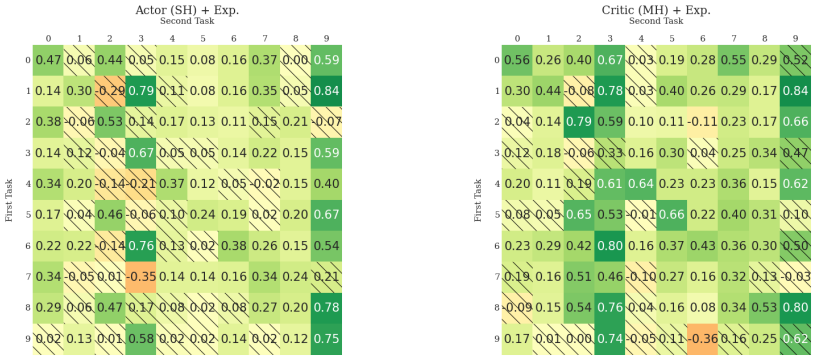
The *Critic (train only head)* row in Table 1 shows that simply reusing the features obtained by the penultimate layer of the critic is not enough to achieve good transfer. In fact, the results suggest that in this setting the model is barely able to learn anything at all. In order to investigate this issue, we conduct a series of experiments where we freeze different parts of the critic's backbone. As it consists of 4 layers (which we number $L1, L2, L3, L4$), we test multiple possible freezing combinations and present the results in Table 9. Note that in all experiments the output head is not frozen and can be freely adapted. Surprisingly, the results show that freezing the early layers degrades the performance more than freezing the further layers, contradicting the feature reuse theory of transfer learning. We hypothesize that the model instead learns more abstract skills which require rewiring in the early layers rather than the later layers.



(a) Actor (MH) and Critic (MH) carried over.

(b) Actor (SH) and Critic (MH) carried over.

(c) Actor (MH) and exploration carried over.



(d) Actor (SH) and exploration carried over.

(e) Critic (MH) and exploration carried over

(f) Actor (SH), critic (MH), and exploration carried over.

(g) Actor (MH), critic (MH), and exploration carried over.

Figure 3: The effect of carrying over different combinations of components on the performance on pairs of tasks from CW10. We shade an entry if the 90% confidence interval contains 0, indicating that we cannot be sure whether the component which was carried over makes a difference.

Table 9: The impact of freezing.

name	FT	FT (no diag)	# pos.	# neg.	# neutral
Freeze $L1, L2$	-0.36 [-0.41, -0.31]	-0.42 [-0.47, -0.37]	18	63	19
Freeze $L1, L2, L3$	-0.82 [-0.86, -0.78]	-0.87 [-0.91, -0.82]	0	92	8
Freeze $L1, L2, L3, L4$	-1.29 [-1.32, -1.25]	-1.30 [-1.34, -1.27]	0	100	0
Freeze $L2, L3, L4$	-0.28 [-0.32, -0.24]	-0.29 [-0.34, -0.25]	14	65	21
Freeze $L3, L4$	0.07 [0.03, 0.10]	0.06 [0.02, 0.10]	33	27	40

D.4 Exploration

By default, the exploration phase of SAC in our experiments takes 10k steps (see Section 3.5). In this subsection, we present the results verifying how the length of the exploration phase impacts the outcome. In Table 10 we show the results for different exploration phase lengths when only the exploration policy is carried over, and Table 11 shows corresponding results when we carry over both the exploration phase and the actor. As shown, shorter exploration phases usually perform better.

Table 10: Effect of the exploration length when carrying over only the exploration policy.

Exp. Len.	FT	FT (no diag)	# pos.	# neg.	# neutral
500	0.06 [0.02, 0.09]	0.03 [-0.01, 0.07]	20	5	75
1000	0.06 [0.02, 0.09]	0.03 [-0.01, 0.07]	24	3	73
5000	0.13 [0.10, 0.17]	0.10 [0.07, 0.14]	36	5	59
10000	0.10 [0.06, 0.13]	0.07 [0.03, 0.10]	27	7	66
20000	0.10 [0.07, 0.14]	0.08 [0.04, 0.11]	30	17	53
50000	-0.00 [-0.04, 0.03]	-0.04 [-0.08, -0.01]	22	44	34

Table 11: Effect of exploration length when carrying over the actor's parameters and the exploration policy.

Exp. len.	FT	FT (no diag)	# pos.	# neg.	# neutral
500	0.20 [0.16, 0.23]	0.17 [0.13, 0.21]	54	1	45
1000	0.18 [0.15, 0.22]	0.16 [0.13, 0.20]	51	0	49
5000	0.22 [0.19, 0.25]	0.20 [0.17, 0.24]	57	1	42
10000	0.16 [0.13, 0.20]	0.14 [0.10, 0.17]	49	3	48
20000	0.15 [0.12, 0.19]	0.14 [0.10, 0.17]	43	3	54
50000	0.13 [0.10, 0.16]	0.11 [0.08, 0.15]	32	12	56

D.5 Transfer independence

We hint in Section 4.1 the effects of carrying over the actor, critic and exploration add up with regard to the induced transfer. In this section we study this further by performing a linear regression analysis. Consider $s_a \in \{0, 1\}$ denoting whether the actor's parameters are carried over ($s_a = 1$) or not ($s_a = 0$); analogously, we define s_c, s_e for the critic's parameters and the exploration. The transfer $t(s_a, s_c, s_e)$ reported in Table 2 fulfills:

$$t(s_a, s_c, s_e) = s_a \cdot w_a + s_c \cdot w_c + s_e \cdot w_e + r(s_a, s_c, s_e),$$

where $w_a = 0.062, w_c = 0.202, w_e = 0.093$, with the residual error r being small; namely $|r(s_a, s_c, s_e)/t(s_a, s_c, s_e)| \leq 0.05$.

We consider this effect somewhat surprising: there is no synergy nor interference for transferring different components. An explanation of this phenomenon requires further studies. It has an interesting practical implication that one can hope to be able to improve the transfer of any component and assemble it into a 'full' solution.

E Additional experiments on transfer in continual learning.

E.1 Exploration

Results for different exploration strategies (see Section 5.1) on top of 5 CL methods: Fine-tuning, Behavioral cloning, EWC, L2, PackNet, are presented in Table 12 (CW10) and Table 13 (CW20). We can see that informed exploration strategies help for all considered methods except for PackNet, especially in the forward transfer metric. The effect is more pronounced for the CW20 sequence.

Table 12: Results of different exploration strategies added on top of 5 different CL methods, for CW10 sequence. For the description of exploration strategies, see Section 5.1.

Method, exploration	performance	forgetting	f. transfer
Fine-tuning, random	0.10 [0.10, 0.10]	0.74 [0.73, 0.76]	0.29 [0.25, 0.33]
Fine-tuning, best-return	0.10 [0.10, 0.11]	0.73 [0.71, 0.75]	0.30 [0.25, 0.34]
Fine-tuning, preceding	0.10 [0.10, 0.11]	0.75 [0.72, 0.77]	0.32 [0.27, 0.36]
Fine-tuning, uniform-previous	0.10 [0.10, 0.10]	0.75 [0.74, 0.77]	0.35 [0.31, 0.38]
Behavioral cloning, random	0.84 [0.81, 0.86]	0.02 [0.01, 0.03]	0.41 [0.38, 0.43]
Behavioral cloning, best-return	0.86 [0.84, 0.87]	0.02 [0.01, 0.04]	0.44 [0.42, 0.46]
Behavioral cloning, preceding	0.84 [0.82, 0.86]	0.02 [0.01, 0.03]	0.39 [0.36, 0.41]
Behavioral cloning, uniform-previous	0.86 [0.85, 0.88]	0.01 [0.00, 0.03]	0.45 [0.41, 0.48]
EWC, random	0.63 [0.60, 0.66]	0.05 [0.03, 0.08]	0.03 [-0.04, 0.09]
EWC, best-return	0.70 [0.68, 0.73]	0.04 [0.01, 0.06]	0.25 [0.21, 0.28]
EWC, preceding	0.70 [0.67, 0.73]	0.02 [-0.00, 0.04]	0.09 [0.03, 0.15]
EWC, uniform-previous	0.72 [0.69, 0.75]	0.04 [0.02, 0.06]	0.24 [0.19, 0.28]
L2, random	0.52 [0.48, 0.56]	0.01 [-0.01, 0.03]	-0.47 [-0.61, -0.33]
L2, best-return	0.63 [0.60, 0.65]	-0.01 [-0.02, 0.00]	-0.13 [-0.21, -0.05]
L2, preceding	0.59 [0.55, 0.62]	0.01 [-0.02, 0.05]	-0.25 [-0.34, -0.16]
L2, uniform-previous	0.63 [0.59, 0.67]	-0.01 [-0.02, 0.01]	-0.16 [-0.28, -0.06]
PackNet, random	0.84 [0.81, 0.86]	-0.01 [-0.02, 0.00]	0.26 [0.22, 0.29]
PackNet, best-return	0.85 [0.83, 0.86]	-0.01 [-0.02, 0.00]	0.23 [0.20, 0.26]
PackNet, preceding	0.84 [0.82, 0.85]	-0.01 [-0.03, -0.00]	0.24 [0.20, 0.27]
PackNet, uniform-previous	0.84 [0.81, 0.86]	-0.00 [-0.02, 0.01]	0.21 [0.15, 0.26]

Table 13: Results of different exploration strategies added on top of 5 different CL methods, for CW20 sequence. For the description of exploration strategies, see Section 5.1.

Method, exploration	performance	forgetting	f. transfer
Fine-tuning, random	0.05 [0.05, 0.05]	0.74 [0.73, 0.76]	0.20 [0.16, 0.23]
Fine-tuning, best-return	0.05 [0.05, 0.06]	0.74 [0.72, 0.76]	0.21 [0.18, 0.24]
Fine-tuning, preceding	0.05 [0.05, 0.05]	0.76 [0.74, 0.78]	0.24 [0.21, 0.28]
Fine-tuning, uniform-previous	0.06 [0.05, 0.06]	0.75 [0.74, 0.77]	0.24 [0.20, 0.27]
Behavioral cloning, random	0.83 [0.81, 0.85]	0.02 [0.01, 0.03]	0.36 [0.34, 0.38]
Behavioral cloning, best-return	0.87 [0.86, 0.88]	0.02 [0.01, 0.03]	0.54 [0.52, 0.55]
Behavioral cloning, preceding	0.84 [0.83, 0.86]	0.02 [0.01, 0.03]	0.40 [0.38, 0.42]
Behavioral cloning, uniform-previous	0.86 [0.85, 0.87]	0.03 [0.02, 0.04]	0.51 [0.48, 0.53]
EWC, random	0.60 [0.59, 0.62]	0.03 [0.01, 0.04]	-0.14 [-0.19, -0.09]
EWC, best-return	0.71 [0.69, 0.73]	0.01 [-0.00, 0.03]	0.28 [0.25, 0.31]
EWC, preceding	0.61 [0.59, 0.64]	0.02 [0.00, 0.03]	-0.14 [-0.19, -0.09]
EWC, uniform-previous	0.70 [0.68, 0.73]	0.02 [0.01, 0.03]	0.21 [0.17, 0.25]
L2, random	0.45 [0.41, 0.49]	0.02 [0.00, 0.05]	-0.56 [-0.68, -0.45]
L2, best-return	0.62 [0.59, 0.65]	-0.00 [-0.01, 0.01]	-0.02 [-0.09, 0.05]
L2, preceding	0.48 [0.44, 0.52]	0.04 [0.01, 0.07]	-0.47 [-0.57, -0.39]
L2, uniform-previous	0.59 [0.56, 0.62]	0.00 [-0.01, 0.01]	-0.15 [-0.25, -0.06]
PackNet, random	0.80 [0.79, 0.82]	-0.00 [-0.01, 0.01]	0.18 [0.14, 0.22]
PackNet, best-return	0.82 [0.81, 0.83]	-0.00 [-0.01, 0.01]	0.23 [0.21, 0.25]
PackNet, preceding	0.81 [0.80, 0.83]	-0.01 [-0.02, -0.00]	0.20 [0.16, 0.24]
PackNet, uniform-previous	0.80 [0.78, 0.82]	0.00 [-0.01, 0.01]	0.23 [0.18, 0.27]

Table 14: Results for different ways of transferring previous data – behavioral cloning (applied to only the actor, or both the actor and the critic), and directly retaining past tuples in SAC buffer (Transfer RL buffer), on 100 pairs of tasks from CW10 (averaged). Base transfer denotes that the exploration policy and the parameters of the actor and the critic are carried over. FT and FT (no diag) represent average forward transfer across all pairs with and without considering the diagonal (transfer from a task to the same task), respectively. Subsequent columns denote the number of pairs with the positive, negative, and neutral transfer.

name	FT	FT (no diag)	# pos.	# neg.	# neutral
Base transfer	0.35 [0.31, 0.38]	0.32 [0.29, 0.36]	70	1	29
BC (actor)	0.01 [-0.03, 0.04]	0.00 [-0.03, 0.04]	28	9	63
BC (actor) + base transfer	0.37 [0.34, 0.40]	0.35 [0.32, 0.38]	77	2	21
BC (actor+critic)	-0.56 [-0.61, -0.52]	-0.62 [-0.67, -0.58]	3	77	20
BC (actor+critic) + base transfer	-0.04 [-0.08, 0.01]	-0.12 [-0.16, -0.07]	35	34	31
Transfer RL buffer	-0.82 [-0.86, -0.78]	-0.90 [-0.94, -0.85]	4	88	8
Transfer RL buffer + base transfer	-0.59 [-0.63, -0.54]	-0.71 [-0.76, -0.66]	12	75	13

Table 15: Average performance and forward transfer metrics on CW10 for Behavioral cloning, for different values of the critic regularization coefficient.

critic's regularization coef.	performance	f. transfer
0	0.82 [0.80, 0.83]	0.34 [0.30, 0.37]
1e-10	0.83 [0.81, 0.85]	0.36 [0.33, 0.39]
1e-09	0.83 [0.81, 0.84]	0.36 [0.32, 0.39]
1e-08	0.79 [0.78, 0.81]	0.32 [0.28, 0.35]
1e-07	0.83 [0.81, 0.84]	0.35 [0.32, 0.38]
1e-06	0.81 [0.80, 0.83]	0.37 [0.36, 0.39]
1e-05	0.81 [0.79, 0.82]	0.37 [0.35, 0.39]
0.0001	0.82 [0.81, 0.84]	0.39 [0.36, 0.41]
0.001	0.77 [0.75, 0.78]	0.33 [0.30, 0.35]
0.01	0.71 [0.69, 0.73]	0.23 [0.20, 0.26]
0.1	0.68 [0.66, 0.69]	0.11 [0.07, 0.14]
1	0.65 [0.63, 0.66]	-0.02 [-0.06, 0.02]
10	0.62 [0.60, 0.63]	-0.18 [-0.22, -0.14]
100	0.48 [0.46, 0.50]	-0.58 [-0.64, -0.53]

E.2 Data rehearsal

Table 14 complements the results of Section 5.2 and presents the impact of transferring data on the transfer. The data transfer is done either by behavioral cloning or by directly carrying over the SAC buffer. The metrics are computed over 100 pairs of tasks from CW10. We can see that Behavioral cloning (actor only) does not have an impact on the transfer when training on pairs; the effect becomes visible only for longer sequences (see Table 4). Other considered methods harm the transfer.

E.3 Regularizing the critic

We vary the coefficient for critic regularization and measure its impact on final metrics for three different methods: Behavioral cloning, EWC, and L2. We set the actor's regularization weight to the optimal value and run a sweep over the regularization coefficient of the critic. We run experiments on the CW10 sequence. The results are presented in Tables 15, 16, 17, indicating that direct regularization of the critic does not significantly improve the performance.

Table 16: Average performance and forward transfer metrics on CW10 for EWC, for different values of the critic regularization coefficient.

critic's regularization coef.	performance	f. transfer
0	0.66 [0.64, 0.67]	0.08 [0.04, 0.11]
1e-10	0.64 [0.62, 0.66]	0.05 [0.01, 0.09]
1e-09	0.64 [0.62, 0.66]	0.06 [0.02, 0.10]
1e-08	0.62 [0.60, 0.64]	0.06 [0.02, 0.10]
1e-07	0.62 [0.59, 0.64]	0.06 [0.01, 0.10]
1e-06	0.63 [0.61, 0.65]	0.01 [-0.04, 0.06]
1e-05	0.63 [0.60, 0.65]	0.02 [-0.02, 0.05]
0.0001	0.61 [0.59, 0.63]	-0.03 [-0.07, 0.01]
0.001	0.55 [0.53, 0.57]	-0.10 [-0.15, -0.05]
0.01	0.50 [0.47, 0.52]	-0.25 [-0.31, -0.19]
0.1	0.40 [0.37, 0.42]	-0.50 [-0.56, -0.44]
1	0.27 [0.25, 0.29]	-0.85 [-0.92, -0.79]
10	0.18 [0.16, 0.19]	-1.24 [-1.32, -1.18]
100	0.12 [0.10, 0.13]	-1.45 [-1.52, -1.39]
10000	0.11 [0.10, 0.13]	-1.45 [-1.53, -1.38]

Table 17: Average performance and forward transfer metrics on CW10 for L2, for different values of the critic regularization coefficient.

critic's regularization coef.	performance	f. transfer
0	0.53 [0.50, 0.56]	-0.40 [-0.48, -0.33]
1e-10	0.53 [0.51, 0.55]	-0.37 [-0.44, -0.32]
1e-09	0.55 [0.53, 0.58]	-0.35 [-0.42, -0.28]
1e-08	0.53 [0.50, 0.56]	-0.34 [-0.41, -0.28]
1e-07	0.52 [0.50, 0.55]	-0.38 [-0.45, -0.32]
1e-06	0.54 [0.51, 0.56]	-0.41 [-0.49, -0.35]
1e-05	0.55 [0.52, 0.57]	-0.36 [-0.43, -0.29]
0.0001	0.52 [0.49, 0.56]	-0.47 [-0.56, -0.39]
0.001	0.53 [0.50, 0.55]	-0.44 [-0.52, -0.37]
0.01	0.53 [0.50, 0.55]	-0.41 [-0.50, -0.34]
0.1	0.49 [0.46, 0.52]	-0.45 [-0.54, -0.36]
1	0.49 [0.46, 0.52]	-0.44 [-0.53, -0.35]
10	0.50 [0.46, 0.53]	-0.40 [-0.50, -0.31]
100	0.45 [0.43, 0.48]	-0.49 [-0.57, -0.41]
10000	0.25 [0.23, 0.27]	-1.08 [-1.17, -1.01]
100000	0.13 [0.12, 0.15]	-1.41 [-1.48, -1.35]

F Infrastructure

In our experiments, we use CPU servers, provided through a cloud service. Throughout the whole project, we conducted over 100.000 runs with 12 cores per run and an average of 10 hours per run, which in the end sums up to over 12M CPU hours.

G Continual World benchmark

We briefly present the Continual World benchmark in Figure 4.

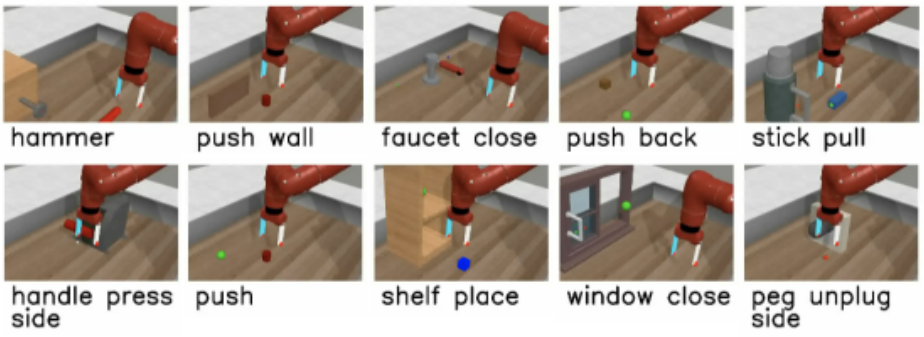


Figure 4: Continual World benchmark adopts robotic tasks from Meta-World benchmark. Depicted above is the CW10 sequence. The CW20 sequence contains tasks from CW10 repeated twice. Tasks are trained sequentially, each one for 1M steps.