Task-Free Continual Learning
Supplementary Materials

Rahaf Aljundi * Klaas Kelchtermans* Tinne Tuytelaars
KU Leuven, ESAT-PSI, Belgium
firstname.lastname@esat.kuleuven.be

These supplementary materials contain the following extra information:

- Hyperparameters and architectural details for the experiments from section 4.
- Example images of the soap series.
- Results on collision avoidance in an extra lengthy simulated corridor.
- Details and extra results on the real-world collision avoidance with the Turtlebot.
- Closing discussion and guidelines on the application of continual learning in an online setting.

Small corrections: Please note that in the ablation study the formula for decaying importance weights (l.645) should have been $\Omega_t = (\Omega_{t-1} + \Omega^*)/2$.

Also, for the real-world experiment the Turtlebot was not pretrained in simulation (as mentioned in l.741) but three times randomly initialized.

*Rahaf Aljundi and Klaas Kelchtermans contributed equally to this work and listed in alphabetical order.

	Exp 1	Exp 2	Exp 3
Architecture	Alexnet	Tiny v2	Tiny v2
Initialization	imagenet	random	random
Learning rate	0.0001	0.01	0.01
Optimizer	SGD	SGD	SGD
Hard Buffer Size	100	40	30
Regularization Weight	100	0.5	0.5
Threshold Mean Loss	0.3	0.5	0.5
Threshold Variance Loss	0.1	0.1	0.02
Length Window Loss	5	5	5

Table 1: Hyperparameters for different experiments: exp 1 $\sim$ Soap Series (4.1), exp 2 $\sim$ Simulated Corridor (4.2) and exp 3 $\sim$ Real Turtlebot (4.3).

1. Hyperparameters and architectural details

As to be able to reproduce the results, we provide the reader with the used hyperparameters and networks, see table 1. Regularization weight corresponds to λ, the continual learning weight in Equation 5.

The Tiny v2 network for the collision avoidance task is a network build especially small in order to allow faster training. The details of the network can be found in figure 1.

2. Examples of the soap series data (Sec. 4.1)

In figure 2, 4 example frames are shown for each of the different soap series: Big Bang Theory, Breaking Bad and Mad Men. These examples demonstrate the scene diversity and large variance in imaging conditions. As mentioned in the paper, Breaking Bad is more actor-centric with a majority of the frames showing only the main character, making it less suited for the self-supervised setup.

3. Larger experiment on collision avoidance in simulation (Sec. 4.2)

To demonstrate both the strengths and weaknesses of our continual learning method, we expanded the corridor experiment of Sec. 4.2 to a sequence of 10 corridors, equal to about 20 minutes flying time or around 10.000 frames. The sequence of different corridors is depicted in figure 5, exhibiting a large variety in textures and obstacles. The length of the sequence allows us to see the longer trend of continual learning.

While training the models online, we evaluate the accuracy on different corridors separately. Due to an imbalance over actions within one corridor, we perform an evaluation based on the total accuracy averaged over the different actions, referred as 'Weighted Accuracy'. When a model becomes degenerated, thus only predicting the most common action in a corridor, an unnormalized accuracy would remain high.

We observe that this data imbalance also affects the online learning as often multiple gradient steps are taken in

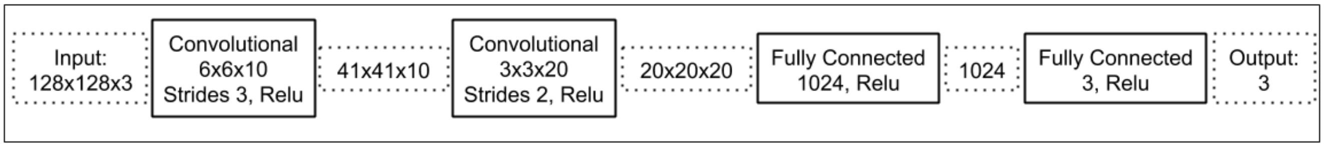

Figure 1: Architecture of the Tiny v2 network used in the monocular collision avoidance experiments (4.2 and 4.3).

Figure 2: Four example images for each soap series, from left to right: Big Bang Theory, Breaking Bad and Mad Men.

favor of only some actions. To bypass this impediment, we experiment here with an additional normalization constraint on the hard buffer forcing an equal distribution over all actions.

Results

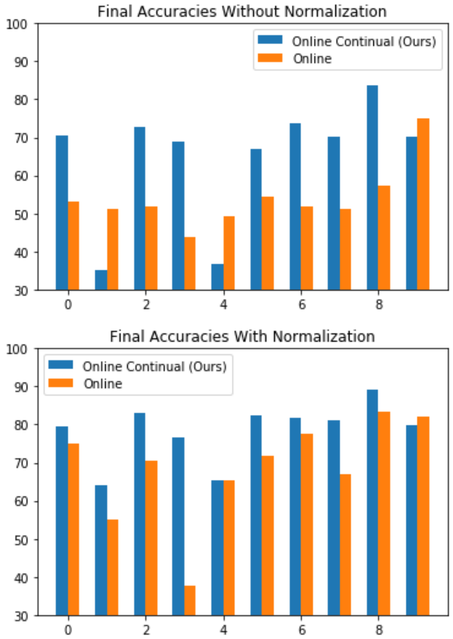

Figure 3: Accuracy's for all 10 corridors at the end of training on the corridor sequence without (top) and with (bottom) the normalization constraint on the hard buffer.

In fig. 3, we show the improvement obtained by our proposed online continual learning method over the online baseline, both with and without the normalization constraint on the buffer with hard examples. The bars express the final accuracy of each corridor as a mean over three models trained with different seeds. The normalization constraint has a positive effect on both continual and normal online learning. Our online continual learning process clearly outperforms the online baseline for most of the corridors. Without the action normalization, the models fail to learn certain corridors, like 1 and 4, resulting in no knowledge that can be preserved by our continual learning method. However if the model grasps information while going through a corridor, it succeeds at preserving it with continual learning, outperforming the online baseline with 15 to 20% accuracy. Moreover with the action normalization, continual learning succeeds at acquiring knowledge in each corridor, outperforming the baseline in all but last corridors.

Figure 6 provides a more in-depth analysis. Here, we show the evolution over time of the cross-entropy loss and the total accuracy over all corridors. We also report the evolution over time of the weighted accuracy, for each corridor separately. From these plots, one can conclude that the buffer normalization clearly has beneficial effects for online learning, especially in the green areas (corresponding to learning taking place on imbalanced corridors). However, the constraint leaves less room in the hard buffer for recent samples causing a slower adaptation of the model during training, as can be observed in the red areas, allowing the models without normalization to improve faster.

In multiple examples, highlighted with blue, the continual learning allows a preservation of knowledge seen before, demonstrating the success of our method. The trend is most clear for the early corridors as the forgetting tends to be worse over time. This phenomenon is also responsible for the total accuracy reaching 80% for continual learning instead of only 70% for normal online learning. This positive trend can be expected to increase when learning over even longer sequences.

In some cases, highlighted in orange, the baseline performance of an old corridor improves while training in a new corridor, reaching a similar accuracy as our continual learning method. In other words, the impact of forgetting seems less as the baseline is able to learn the same knowledge again.

This lengthy experiment demonstrates the strengths of continual learning, including the expected positive trend when applying it to longer sequences of data.

4. Collision avoidance on real Turtlebot (Sec. 4.3)

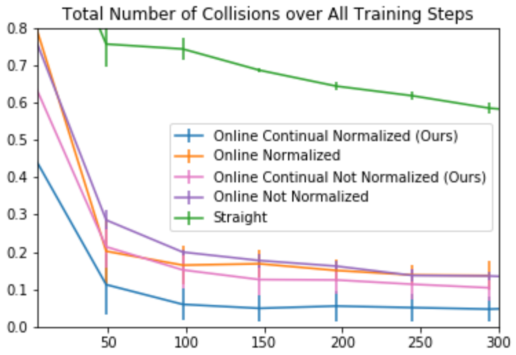

Figure 4: Performance expressed as the average number of collisions - i.e. the total number of collisions divided by the total number of gradient steps.

In this proof-of-concept, an neural network steers a turtlebot around one big yellow object (see figure 7 in the paper). Each frame is kept in a buffer containing the 40 most recent frames combined with expert labels. Every 10 frames a gradient step is taken. When a collision is detected by the Lazer Range Finder, the training is paused and the Turtlebot turns automatically such that the closest obstacle is at its back. The hyperparameters can be found in table 1. Each model is trained three times and takes about 20 minutes, or 300 gradient steps, till convergence.

Extra results and baselines are shown in figure 4, plotting the total number of collisions divided by the total number of gradient steps. Driving straight leads to an average of 0.6 collisions per gradient step. Adding action normalization in the hard buffer and applying continual learning both have a clear positive influence. The action normalization allows an even larger improvement of our continual learning method over the baseline.

This real-world experiment differs in two significant ways from the previous experiments. First, the agent stays in one domain that does not vary over time. Second, the agent acts within the environment to create new data making the setup on-policy and online. Although there are no domain or task changes over time, our continual learning method has a clear positive effect. This result fully supports our claim of ”Task-Free” continual learning, namely that it is not required to have significant changes in your data in order to do better than a normal online learner. The continual learning method inherently stabilizes the online learning in an on-policy setup.

However, a major challenge in online/on-policy learning is dealing with uninformative states. These states lead to less information in a batch and thus slower training. Samples that do contain relevant information, are better preserved in the online-continual setting, resulting in faster learning. Unfortunately, the exact moment of large information gain varies over different runs resulting in a higher variance. This explains the larger variance in figure 4 and figure 7 in the main paper.

5. Closing discussion / General guidelines

When considering applying continual learning to a specific problem, it is best to keep two guidelines in mind:

The *mean and variance threshold* of the loss window should be carefully chosen. If both thresholds are too low, the model will not use the MAS regularization; conversely, if too high, the model will slow down the learning by preserving irrelevant information as the importance weights are updated too frequently. The latter case in combination with global averaging usually deteriorates the final performance. Therefore, it is recommended to place the threshold low enough while still allowing importance weight updates. As relaxing the threshold, results in more updates, a decaying update rule allows the model to forget irrelevant previous knowledge. In practice, we discovered that the mean threshold could remain quite high, as long as the variance threshold is low. Moreover, meta-learning techniques, such as learning-to-learn, could automate these settings.

A *trainable task* is a necessary condition: In order to have the MAS regularization exceed in performance over the baseline, the task must be actually trainable. Although this seems obvious, it is far from trivial in an online learning setting, due to the non-i.i.d. nature. Predicting whether continual learning will perform better than typical online learning depends on stable training. For instance, in the collision avoidance task, including an action normalization constraint in the hard buffer, clearly improves the stability of online learning.

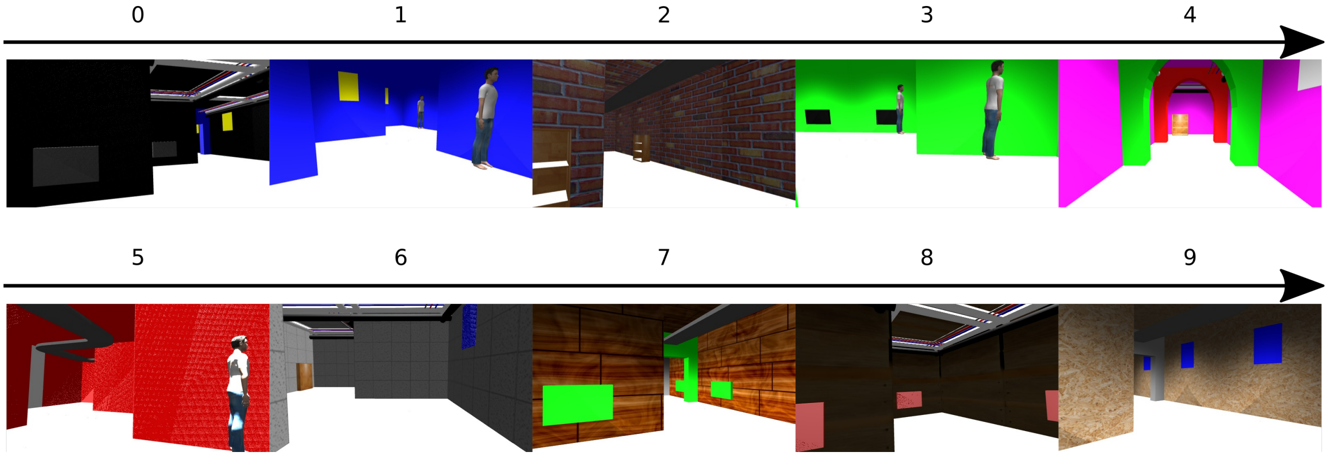

Figure 5: Example views in the longer corridor sequence, corresponding to 10 environments depicted in lexicographic order.

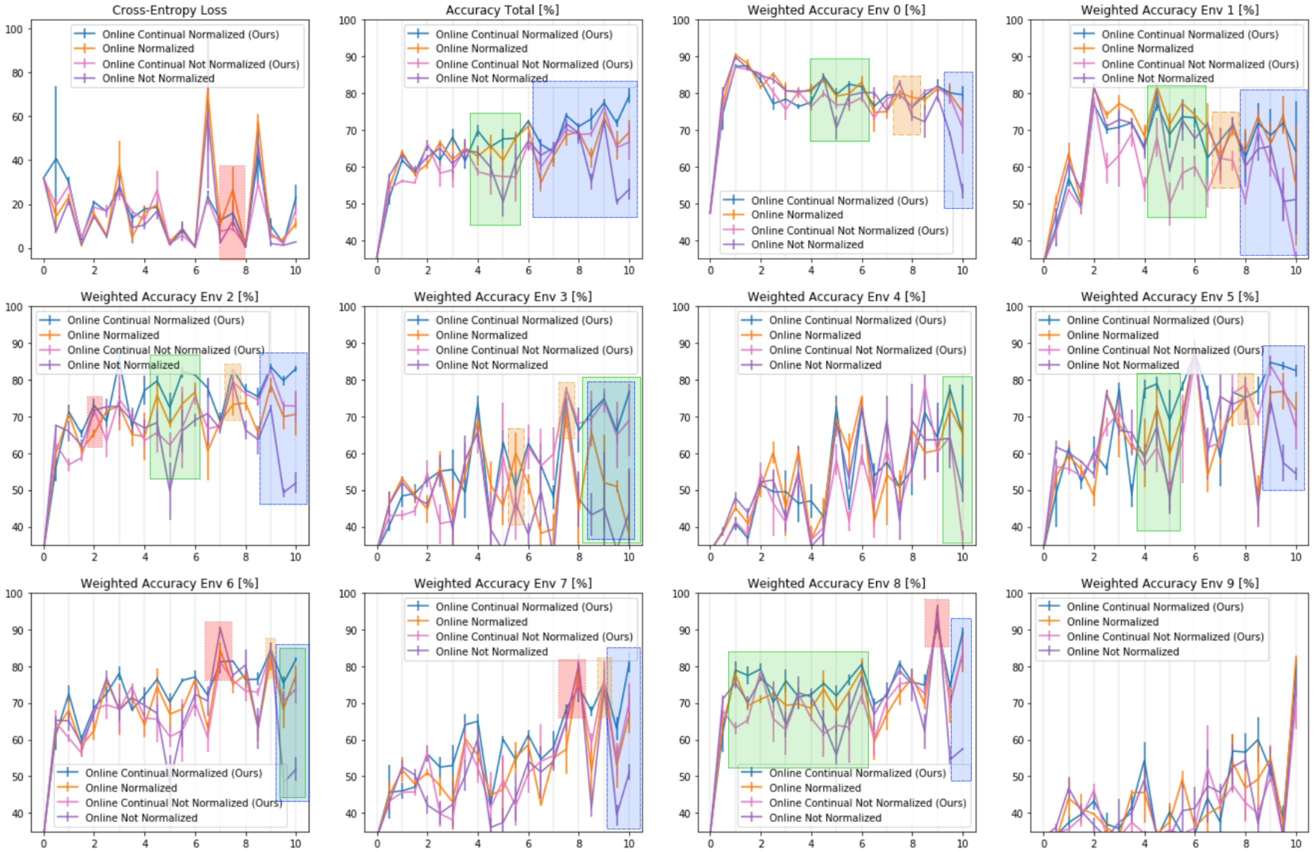

Figure 6: Cross-Entropy loss and accuracy's on total and separate corridors while training online on the sequence of 10 corridors. Blue squares indicate continual learning outperforming baseline models. Green squares indicate positive normalization effects for both continual and baseline models. Red squares indicate slower learning due to normalization constraint. Orange squares indicate learning forgotten knowledge by the baseline model.

In conclusion, we successfully extended continual learning to a task-free online learning algorithm and demonstrated its advantage in following applications: face recognition in soap series, and monocular collision avoidance both on a drone in simulation and on a Turtlebot in the real-world.