Continual Learning for Image-Based Camera Localization – Supplementary material –

Algorithm 1 Reservoir

N ← number of all instances observed
$s \sim Random(0, 1) \times N$
if $s < |B|$ **then**
  Replace a random instance in B with $(x, \mathbf{y})$
else
  Ignore
end if

Algorithm 2 Class-balance

if c is not largest **then**
  Select a random instance of largest class
  Replace it with $(x, \mathbf{y})$
else
  $m_c \leftarrow$ number of currently stored instances of c
  $n_c \leftarrow$ number of total instances observed of c
  $u \sim Random(0, 1)$
  if $u < m_c/n_c$ **then**
    Replace an instance of *class*(x) with $(x, \mathbf{y})$
  else
    Ignore
  end if
end if

1. Baseline Algorithms

As mentioned in our main paper, the baseline algorithms *Reservoir* [4] and *Class-balance* [1] are presented in Algorithm 1 and 2 respectively.

2. Training Details

For continual learning training purposes, we adopt two *dataloaders*, namely *trainloader* and *bufferloader*, in our training procedure except the first scene training. *Trainloader* randomly loads data from the current scene and generates ground truth labels to feed to the model. While in *bufferloader*, the training frames are loaded from our buffer and ground truth labels are either generated the same as in *trainloader* or extracted from intermediate representations. The loaded image pairs $< t_i, b_i >$ are fed to the training network separately and their losses are summed together as the final loss term as shown in Eq 3 (*c.f.* Sec.3.1) in the main paper, where i indicates the i^{th} training step.

Rep-buff. Since we have two classification layers and one regression layer in our training model, the loss terms in Eq 4 (*c.f.* Sec.3.2) can be written as:

$$L_t = \alpha_1 \cdot e_{l1} + \alpha_2 \cdot e_{l2} + \beta \cdot e_{3D} \tag{1}$$

Where e_l is the classification loss and e_{3D} is the summed regression loss for each pixel. The weighting coefficients are set to (1,1,100000) during the training. However, in our *Rep-buff*, the real-value prediction for regression layer $\tilde{y}_{3D}$ might be unbounded, and could give highly erroneous guidance to the loss term e_{3D} [3]. Hence, instead of directly minimizing $\hat{\tilde{y}}_{3D}$ w.r.t. $\tilde{y}_{3D}$, as with [3], the $\tilde{y}_{3D}$ is applied as a upper bound. That is, the prediction $\hat{\tilde{y}}_{3D}$ is made as close as possible to the ground truth and not penalized if its performance surpasses $\tilde{y}_{3D}$, we have the β in Eq 1 for *Rep-buff*:

$$\beta = \begin{cases} 100000, & \text{if } \left\| \hat{\tilde{y}}_{3D} - y_{3D} \right\|^2 > \| \tilde{y}_{3D} - y_{3D} \|^2. \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Benefits of y_1 over y_{3D}. As mentioned in our main paper, dense 3D points y_{3D} provide an accurate estimate of coverage score. However, it is computationally intensive when considering all the pixels in a candidate image. Thus, only the hierarchical cluster labels are selected to represent the dense 3D points. We have 25 labels for each level and in total 625 labels for an individual scene. In our combined scenes **i7S**, **i12S**, and **i19S**, we combine the labels tree at the first level as in [2]. That is, for **i19S**, the first level contains 475 branches. 625 ÷ Average number of 3D points per scene is the reduced computations in such a setting.

3. Disorder Scenes

At this point, we provide the results for **i7S** where models are trained with three different permutations of scenes, with buffer size $B = 256$ and *Img-buff* replay. Note that we preserve the order of the last scene. Table 1 present the accuracy on the test images of individual scenes and overall

Scene	Accuracy (%)		
	Reservoir	Class-balance	Buff-CS (ours)
Heads	78.60	89.20	90.90
Kitchen	47.52	46.84	37.18
Pumpkin	43.65	38.60	55.50
Chess	93.40	96.10	93.45
Office	76.14	69.65	69.48
Fire	86.30	93.50	92.95
Stairs	73.30	78.60	76.10
Average	71.27	73.21	73.65

Scene	Accuracy (%)		
	Reservoir	Class-balance	Buff-CS (ours)
Office	58.85	56.78	60.43
Heads	73.60	89.40	89.60
Fire	79.10	81.05	84.00
Chess	88.40	85.80	93.65
Kitchen	48.82	50.34	42.86
Pumpkin	51.30	59.00	54.25
Stairs	80.60	76.40	80.10
Average	68.69	71.25	72.12

Scene	Accuracy (%)		
	Reservoir	Class-balance	Buff-CS (ours)
Heads	70.00	90.90	93.30
Office	61.10	60.93	63.58
Chess	88.40	86.40	93.85
Kitchen	47.08	42.56	40.25
Fire	83.45	88.00	87.95
Pumpkin	51.65	47.95	53.55
Stairs	81.50	82.10	81.30
Average	69.03	71.26	73.40

Table 1: The accuracy on individual scenes of **i7S** with different training order after the training is complete. The order for the last scene is preserved. All methods employs an *Img-buff* buffer of size 256.The best results are highlighted in red.

accuracy on **i7S**. The results show that our method is not affected by the order of scenes and performs comparably to or exceeds the baselines as mentioned in the main paper.

4. Rep-buff

In all our experiments the number of cluster levels used is $L = 2$. The corresponding network predictions at each cluster level, $\hat{y_1}, \hat{y_2}$ and the final 3D coordinates $\hat{y_{3D}}$ are stored in *Rep-buff*. We analyze the influence of experience-replay using intermediate predictions $\hat{y_1}, \hat{y_2}$ and the final layer predictions $\hat{y_{3D}}$ in this section.

The different layers are assigned weights, $\alpha_1, \alpha_2, \beta$. For

Scenes	Accuracy (%)		
	Weights = 0,0,1	Weights = 0,0,1e5	Weights = 1,1,0
apt1/kitchen	84.59	77.87	82.91
apt1/living	77.28	74.65	79.11
apt2/bed	85.29	78.92	80.88
apt2/kitchen	96.19	95.71	97.62
apt2/living	79.08	80.23	77.65
apt2/luke	46.79	42.15	45.83
office1/gates362	67.88	64.25	72.02
office1/gates381	50.43	51.57	56.41
office1/lounge	87.46	86.54	86.54
office1/manolis	81.18	79.42	74.65
office2/5a	94.16	94.77	93.96
office2/5b	97.28	96.26	97.04
Average	78.97	76.86	78.72

Table 2: The accuracy on individual scenes of **i12S** with buffer size $B = 128$ and different assigned weights on *Rep-buff*.

$(\alpha_1, \alpha_2, \beta)$ = (1,1,0) and (0,0,1) respectively, results in table 2 show that both these conditions obtain comparable performance. For *Buff-CS*, the results are better than the weight distribution (1,1,100000) originally proposed by [2].

5. Additional Results on i12S

Similar to the **i7S** results presented in the main paper, we show the results on individual scenes of **i12S** after training is completed in Table 3, and the average accuracy over different stages of the training process on each scene of **i12S** in Table 4. In addition, the accuracy on individual scenes of **i12S** at each stage of the training is provided in Fig. 1. The results show for majority of scenes *Buff-CS* outperforms *Class-balance* and *Reservoir* across different evaluation settings.

References

[1] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *ICML*, 2020.

[2] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020.

[3] Muhamad Risqi U Saputra, Pedro PB de Gusmao, Yasin Almalioglu, Andrew Markham, and Niki Trigoni. Distilling knowledge from a deep pose regressor network. In *ICCV*, 2019.

[4] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

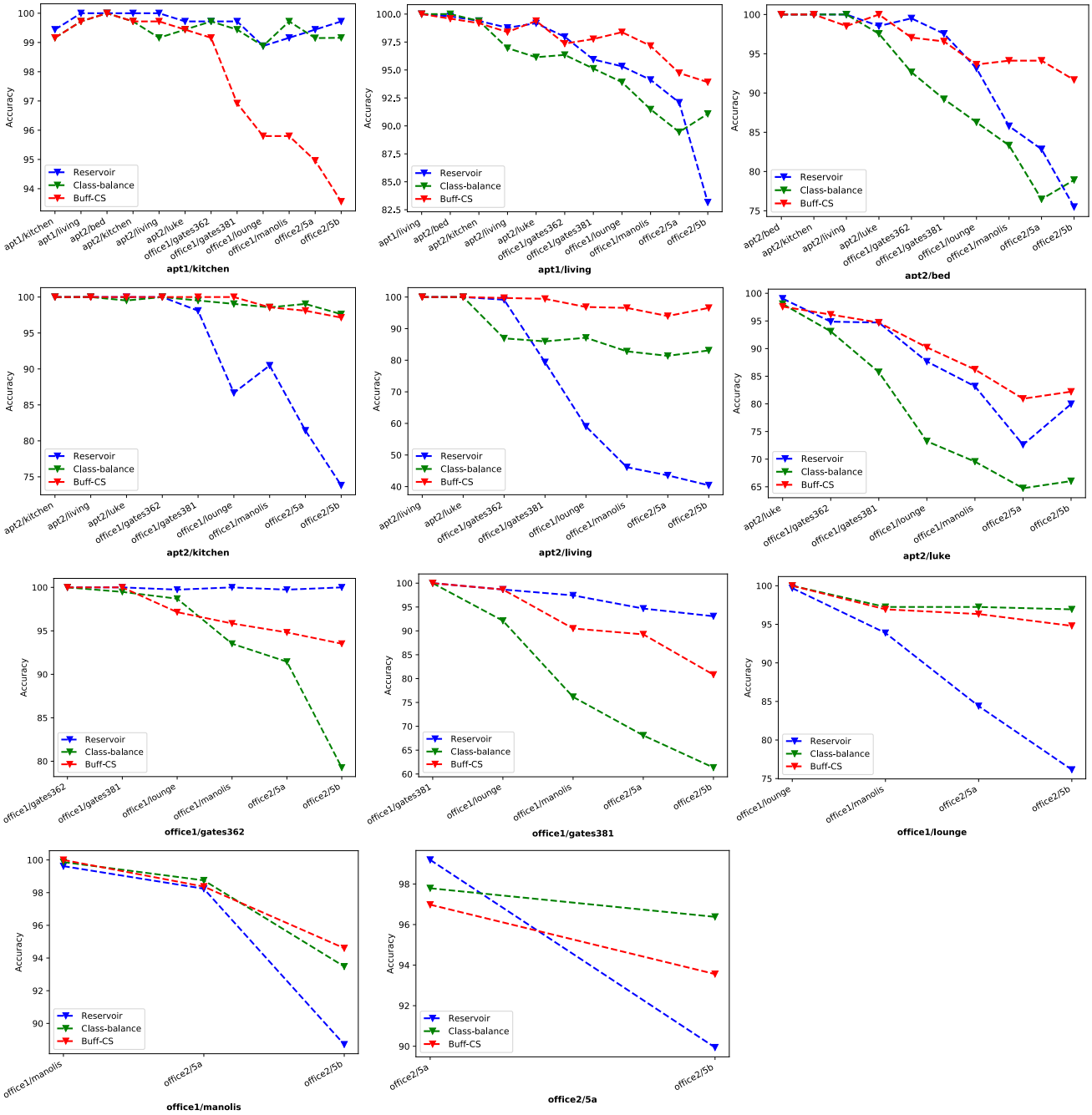

Figure 1: The accuracy (error < 5 cm, 5°) on individual scenes of **i12S** (except for the last scene) at each stage of the training. The x axis indicates the training progress. All methods employs an *Img-buff* buffer of size 256. The results show for majority of scenes *Buff-CS* outperforms *Class-balance* and *Reservoir*.

Scenes	Accuracy (%)		
	Reservoir	Class-balance	Buff-CS(ours)
apt1/kitchen	97.68	98.60	93.84
apt1/living	81.95	89.05	95.74
apt2/bed	74.02	73.53	87.25
apt2/kitchen	71.90	98.09	98.57
apt2/living	41.54	81.38	94.27
apt2/luke	80.93	60.10	84.54
office1/gates362	100	82.38	91.45
office1/gates381	93.44	62.77	76.26
office1/lounge	71.86	96.02	93.88
office1/manolis	87.58	92.72	94.60
office2/5a	92.35	98.19	94.97
office2/5b	97.28	96.05	96.79
Average	82.54	85.72	91.85

Table 3: The percentage of accurately localized test images (error < 5 cm, 5°) on **i12S** with the buffer size $B = 256$, after the training is complete. Here we use *Img-buff* for replay. The best results are highlighted in red.

Scenes	Average Accuracy (%)		
	Reservoir	Class-balance	Buff-CS(ours)
apt1/kitchen	99.65	99.44	97.83
apt1/living	95.98	95.44	97.81
apt2/bed	93.28	90.44	96.57
apt2/kitchen	92.28	99.26	99.31
apt2/living	70.95	88.41	97.89
apt2/luke	87.43	78.64	89.72
office1/gates362	99.91	93.74	96.89
office1/gates381	96.77	79.54	91.85
office1/lounge	88.53	97.86	97.02
office1/manolis	95.52	97.37	97.66
office2/5a	94.57	97.09	95.27
office2/5b	99.51	97.04	94.07
Total Average	92.83	92.86	95.99

Table 4: The average accuracy over different stages of the training process on each scene of **i12S** with the buffer size $B = 256$ and *Img-buff* replay. Our method has overall better performance compared to the other two methods.