A Proofs

Proof of Theorem 1. Let $g : [1, m_T] \times \mathbb{Z}^{T+} \rightarrow [1, T]$ be a function that maps the batch index to its corresponding task index. We abbreviate $g(i, (m_t)_{t=1}^{T})$ as $g(i)$. Then we have

$$\begin{aligned}
\hat{\mathbb{E}}[S_{m_T}] &= (1-\eta_{m_T})\hat{\mathbb{E}}[S_{m_T-1}] + \eta_{m_T} S_{m_T} && (17)\\
&= (1-\eta_{m_T})\hat{\mathbb{E}}[S_{m_T-1}] + \eta_{m_T} r_{m_T} S_{m_T}^{T} + \frac{\eta_{m_T}(1-r_{m_T})}{T-1}\sum_{t=1}^{T-1} S_{m_T}^{t} && (18)\\
&= \sum_{i=1}^{m_T} \eta_i r_i \prod_{j=i+1}^{m_T} (1-\eta_j) S_i^{g(i)} + \sum_{j=m_{g(i)}+1}^{m_T} \frac{\eta_j(1-r_j)}{T-1} \prod_{k=j+1}^{m_T} (1-\eta_k) S_j^{g(j)} && (19)
\end{aligned}$$

where the first part of Eq. (19) can be considered as the contribution of the "current" tasks of all batches to the running statistics, and the second part can be considered as the contribution of the "past" tasks. Now we can conclude the proof by noting that the contribution of task t as the "current" tasks to the running statistics starts from $m_{t-1}+1$ to m_t and the contribution as the "past" task starts from $m_t + 1$ to m_T. $\square$

Proof of Corollary 2. The first result can be obtained by noting that η is a constant and directly applying the sum formula of geometric series. When $m_T - m_{T-1} = m_{T-1} - m_{T-2} = \cdots = m_1$, we have that

$$w_t = \frac{\bar{\eta}^{m_{T-t}} - \bar{\eta}^{m_{T-t+1}}}{(1-\bar{\eta}^{m_T})Z} = \frac{\bar{\eta}^{(T-t)m_1} - \bar{\eta}^{(T-t+1)m_1}}{(1-\bar{\eta}^{Tm_1})Z} = \frac{\bar{\eta}^{(T-t)m_1}(1-\bar{\eta}^{m_1})}{(1-\bar{\eta}^{Tm_1})Z} \approx \frac{\bar{\eta}^{(T-t)m_1}}{Z}$$

The approximation error is upper bounded by

$$|\epsilon| = \frac{\bar{\eta}^{(T-t)m_1}}{Z}\left(\frac{1-\bar{\eta}^{m_1}}{1-\bar{\eta}^{m_1 T}} - 1\right) < \frac{\bar{\eta}^{m_1}}{(1-\bar{\eta}^{m_1 T})Z} \leq \mathcal{O}\left(\frac{\bar{\eta}^{m_1}}{1-\bar{\eta}^{m_1 T}}\right)$$

$\square$

Proof of Corollary 3. Necessity:

Let us start by assuming that $w_t = w_{t+1}$. By applying Eq. (6) we have the following

$$\sum_{i=m_{t-1}+1}^{m_t} \eta_i r_i \prod_{j=i+1}^{m_T} (1-\eta_j) = \sum_{i=m_t+1}^{m_{t+1}} \frac{\eta_i(r_i T - 1)}{T-1} \prod_{j=i+1}^{m_T} (1-\eta_j)$$

Simplifying the equation we obtain that

$$\begin{aligned}
\sum_{i=m_{t-1}+1}^{m_t} r(1-\eta)^{m_T-i} &= \sum_{i=m_t+1}^{m_{t+1}} \frac{rT-1}{T-1}(1-\eta)^{m_T-i}\\
\frac{r}{(1-\eta)^{m_{t-1}+1}} \sum_{i=0}^{m_t-m_{t-1}} \frac{1}{(1-\eta)^i} &= \frac{rT-1}{(T-1)(1-\eta)^{m_t+1}} \sum_{i=0}^{m_{t+1}-m_t} \frac{1}{(1-\eta)^i}\\
\frac{r(\eta-1)(1-\eta)^{m_t-m_{t-1}} + r}{\eta(1-\eta)^{m_t+1}} &= \frac{(rT-1)[(\eta-1)(1-\eta)^{m_{t+1}-m_t} + 1]}{\eta(T-1)(1-\eta)^{m_{t+1}+1}}\\
r(1-\bar{\eta}^{m_t-m_{t-1}+1}) &= \frac{(rT-1)(1-\bar{\eta}^{m_{t+1}-m_t+1})}{(T-1)\bar{\eta}^{m_{t+1}-m_t}}\\
r - \frac{r(T-T\bar{\eta}^{m_{t+1}-m_t+1})}{(T-1)\bar{\eta}^{m_{t+1}-m_t}(1-\bar{\eta}^{m_t-m_{t-1}+1})} &= \frac{\bar{\eta}^{m_{t+1}-m_t+1} - 1}{(T-1)\bar{\eta}^{m_{t+1}-m_t}(1-\bar{\eta}^{m_t-m_{t-1}+1})}
\end{aligned}$$

$$
\begin{aligned}
r &= \frac{\bar{\eta}^{m_{t+1}-m_t+1}-1}{\bar{\eta}^{m_{t+1}-m_t}(1-T)(\bar{\eta}^{m_t-m_{t-1}+1}-1)+T(\bar{\eta}^{m_{t+1}-m_t+1}-1)} \\
&= \frac{\bar{\eta}^{m_1+1}-1}{\bar{\eta}^{m_1}(1-T)(\bar{\eta}^{m_1+1}-1)+T(\bar{\eta}^{m_1+1}-1)} \\
&= \frac{1}{T-\bar{\eta}^{m_1}(T-1)} \\
&\approx \frac{1}{T}
\end{aligned}
$$

The approximation error is upper bounded by

$$
|\epsilon| = \frac{\bar{\eta}^{m_1}(T-1)}{T(T-\bar{\eta}^{m_1}(T-1))} < \frac{\bar{\eta}^{m_1}}{T-\bar{\eta}^{m_1}(T-1)} \leq \mathcal{O}(\bar{\eta}^{m_1})
$$

Sufficiency:

$$
w_{t+1}-w_t = \sum_{i=m_t+1}^{m_{t+1}} \frac{\eta_i(r_iT-1)}{T-1} \prod_{j=i+1}^{m_T}(1-\eta_j) - \sum_{i=m_{t-1}+1}^{m_t} \eta_i r_i \prod_{j=i+1}^{m_T}(1-\eta_j) \tag{20}
$$

Substituting $r = 1/T$ into Eq. (20) yields

$$
\begin{aligned}
w_{t+1}-w_t &= -\sum_{i=m_{t-1}+1}^{m_t} \frac{\eta_i}{T} \prod_{j=i+1}^{m_T}(1-\eta_j) \\
&= -\frac{\eta}{T} \sum_{i=m_{t-1}+1}^{m_t} (1-\eta)^{m_T-i} \\
&= -\frac{\bar{\eta}^{m_T-m_t-1}(1-\bar{\eta}^{m_t-m_{t-1}+1})}{T} \\
&\approx 0
\end{aligned}
$$

The approximation error is upper bounded by

$$
|\epsilon| = \frac{\bar{\eta}^{m_T-m_t-1}(1-\bar{\eta}^{m_t-m_{t-1}+1})}{T} < \frac{\bar{\eta}^{m_T-m_t-1}}{T} \leq \mathcal{O}(\bar{\eta}^{m_1})
$$

□

Proof of Corollary 4. Substituting $\eta(i) = 1/(1+i)$ into Eq. (6)

$$
\begin{aligned}
w_t &= \left[\sum_{i=m_{t-1}+1}^{m_t} \eta_i r_i \prod_{j=i+1}^{m_T}(1-\eta_j) + \sum_{i=m_t+1}^{m_T} \frac{\eta_i(1-r_i)}{T-1} \prod_{j=i+1}^{m_T}(1-\eta_j)\right] / Z \\
&= \left[\sum_{i=m_{t-1}+1}^{m_t} \frac{r_i}{1+m_T} + \sum_{i=m_t+1}^{m_T} \frac{1}{T-1}\frac{1-r_i}{1+m_T}\right] / Z \\
&= \left[\frac{r+\frac{1-r}{T-1}}{1+m_T}\right] / Z
\end{aligned}
$$

We can now conclude by noting that w_t is independent of the choice of t. □

B Implementation Details

The hyperparameter $\tilde{\eta}$ is fixed to 0.1, κ is selected from $\{0.1, 0.4, 0.7, 1.0\}$, and λ is selected from $\{0.01, 0.1, 1.0, 10.0\}$ for Split CIFAR-10 and Split CIFAR-100 and $\{0.00001, 0.0001, 0.001, 0.01\}$ for Split Mini-ImageNet. Code is available in the supplementary material and will be released upon acceptance. All experiments are performed on eight NVIDIA RTX A4000 GPUs. The amount of compute is easily affordable, which can be inferred from the running times given in Table 3.

C Extended Results

C.1 Time Complexity Analysis

We provide in Table 3 the floating point operations per step (FLOPs/step) of different normalization methods and the total running times (in seconds) under different implementations (the exact calculation of FLOPs can be found in the attached code). It can be seen that compared to BN, our proposed method only slightly increases the computation, while CN almost doubles the computation because it combines both GN (without affine transformation) and BN. Note that the FLOPs only give the theoretical computation; the actual running time depends on the implementation. We measure the actual running time of two normalization implementations: 1) using the plain `torch` primitive and 2) using the `torch.nn` with cuDNN as the backend. Due to engineering difficulties, we do not currently implement a cuDNN-optimized version of our method. However, it can be seen from Table 3 that on the plain implementation our method only slightly increases the running time, which is consistent with the FLOPs analysis. This suggests that our method has the potential to achieve comparable time complexity as BN through a well-engineered cuDNN implementation.

Table 3: Comparison of FLOPs per step of different normalization layers and total running times (in seconds) under different implementations on Split CIFAR-100 using ER-ACE with $|B| = 10$ and $|M| = 2000$ as the baseline approach. - to be exploited.

	BN	GN	CN	Ours
FLOPs/step	49.20M	49.18M	86.09M	49.32M
	1×	0.99×	1.75×	1.01×
Time (Plain)	430	440	535	489
	1×	1.02×	1.24×	1.14×
Time (cuDNN)	324	322	376	-
	1×	0.99×	1.16×	-

C.2 Impacts of Memory Buffer Selection Strategies

Fig. 6 shows that our method obtains substantial improvements over both reservoir sampling and ring buffer, which demonstrates the robustness of AdaB^{2}N to the memory buffer selection strategy.

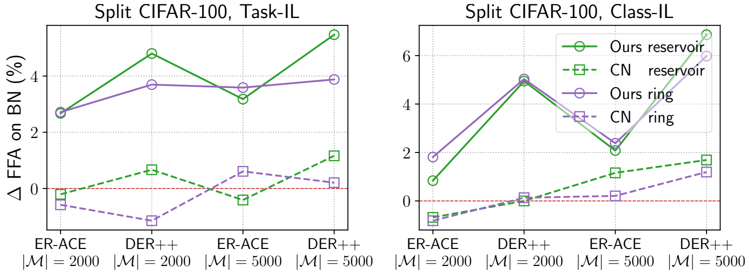

Figure 6: Performance improvements (i.e., Δ FAA) for online continual learning over different selection strategies of memory buffer and different memory sizes $|\mathcal{M}|$ (with batch size $|B| = 10$).

C.3 Forgetting Measure

Table 4 shows that our method is generally the lowest in terms of forgetting measure.

C.4 Full Results for Dynamics of Normalization Statistics

We provide the norm dynamics of the batch statistics and population statistics for all normalization layer in Figs. 7 and 8. For better illustration, each curve in the figures indicates the mean after Gaussian smoothing with a kernel size of 20 and the shaded area indicates the 0.05×variance. It can be seen that our method roughly tracks the joint training (JT) on many layers (e.g., 2, 3, 5, 7, 8, 12, 15, 17, 18, 19).

Table 4: Forgetting measure (↓) of **online task-incremental learning** with batch size $|B| = 10$. We use **bold**, underline, and *italic* to indicate the first, second, and third best results respectively.

Method	Split CIFAR-10		Split CIFAR-100		Split Mini-ImageNet	
	$\|\mathcal{M}\|$=500	$\|\mathcal{M}\|$=2000	$\|\mathcal{M}\|$=2000	$\|\mathcal{M}\|$=5000	$\|\mathcal{M}\|$=2000	$\|\mathcal{M}\|$=5000
ER-ACE w/ BN	3.31±0.95	1.76±1.84	2.15±1.11	1.71±0.98	*3.14±1.42*	*2.80±1.07*
ER-ACE w/ LN	4.03±3.25	*0.99±1.43*	2.92±0.81	1.46±0.44	2.93±0.08	2.53±0.93
ER-ACE w/ IN	1.74±1.04	1.76±0.96	*2.11±0.61*	**1.41±0.63**	3.86±0.59	5.07±1.20
ER-ACE w/ GN	1.30±0.36	1.73±1.61	**0.86±0.26**	3.21±0.96	4.13±1.98	3.07±1.44
ER-ACE w/ CN	*1.67±0.25*	0.90±0.51	2.53±0.53	2.03±0.33	3.72±1.26	4.21±1.37
ER-ACE w/ Ours	**1.22±0.60**	**0.84±0.78**	1.81±0.66	*1.58±1.35*	**2.21±2.38**	**1.94±0.87**
DER++ w/ BN	2.35±0.73	**0.22±0.03**	1.26±0.42	*1.31±1.07*	**1.97±0.94**	2.60±0.32
DER++ w/ LN	1.61±1.05	*0.65±0.22*	2.02±1.28	1.88±1.41	3.16±0.70	*2.95±0.61*
DER++ w/ IN	1.90±0.47	0.68±0.62	2.91±1.89	1.78±1.03	*3.46±0.56*	3.64±0.40
DER++ w/ GN	**1.42±1.03**	0.89±1.05	2.32±1.45	1.41±0.63	3.88±1.52	3.70±1.29
DER++ w/ CN	3.89±0.52	2.05±0.40	*1.63±0.77*	1.07±0.28	4.04±1.17	3.85±1.08
DER++ w/ Ours	*1.81±1.74*	0.38±0.17	**0.92±0.58**	**0.73±0.41**	3.65±1.49	**1.36±0.32**

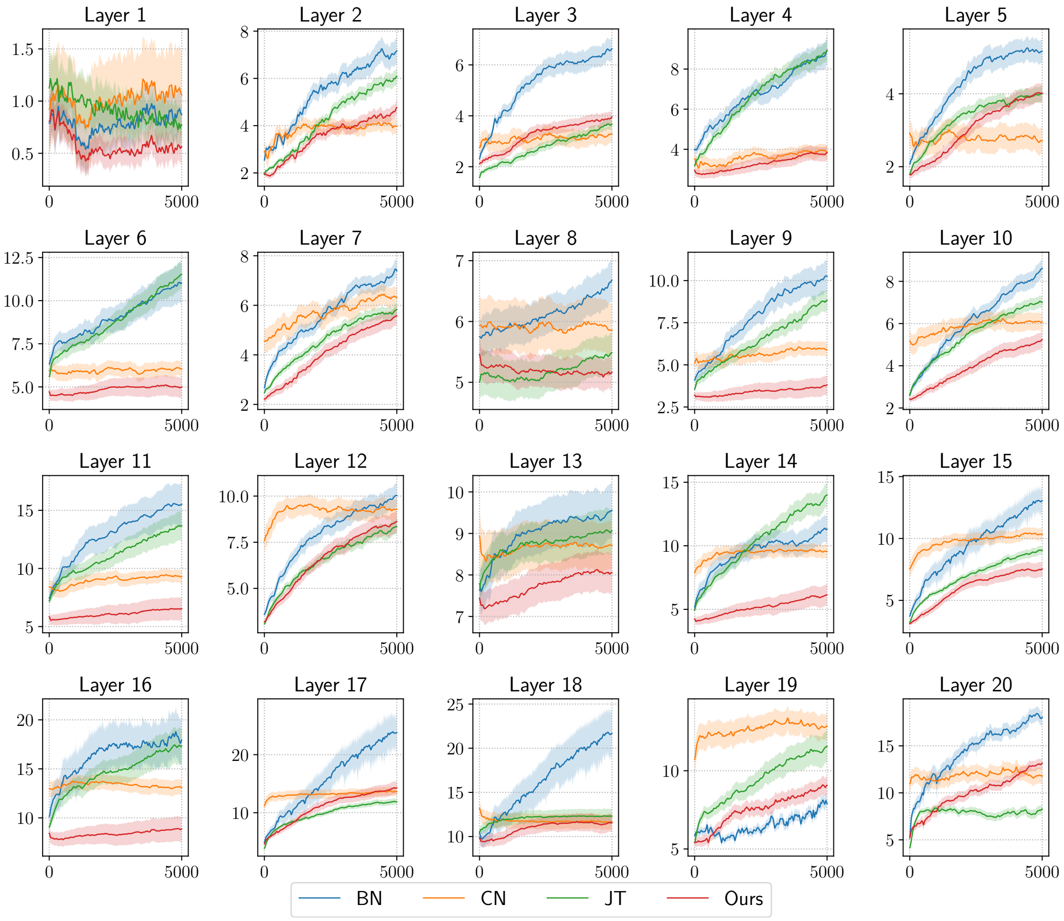

Figure 7: Norm dynamics of the batch statistics of all normalization layer.

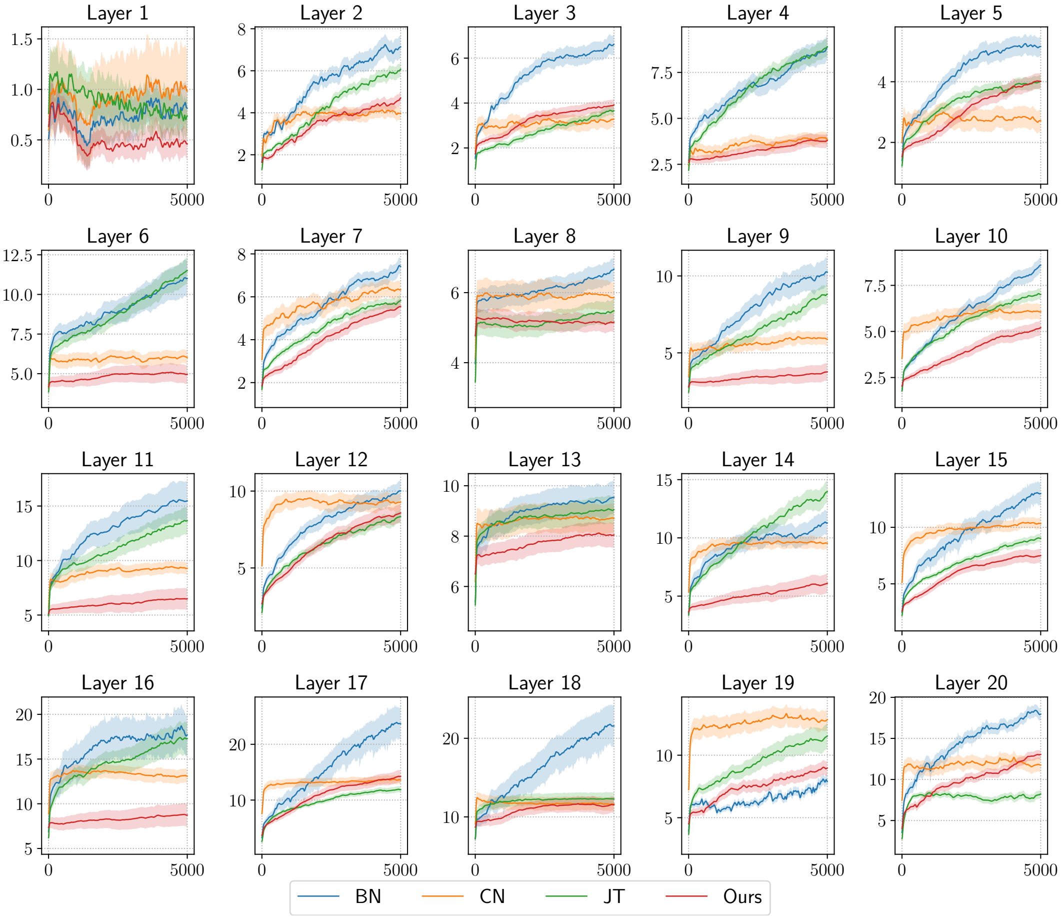

Figure 8: Norm dynamics of the population statistics of all normalization layer.