AUC Maximization in Imbalanced Lifelong Learning (Supplementary Material)

Xiangyu Zhu[1,2] **Jie Hao**[2] **Yunhui Guo**[3] **Mingrui Liu**[2]

[1]Meituan, China
[2]Department of Computer Science, George Mason University, USA
[3]Department of Computer Science, University of Texas at Dallas, USA

A DIANA OUTPERFORMS OTHER METHODS WITH CLASS BALANCED SAMPLING

Table 1: The results of memory-based methods with Class Balanced Reservoir Sampling on Split CIFAR100, Split CUB200, and Split AWA2.

Method	Split-CIFAR			Split-CUB			Split-AWA2		
	AUC (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%
A-GEM	58.3 ± 3.5	60.3 ± 0.3	19.5 ± 3.6	50.8 ± 4.0	50.2 ± 0.3	17.6 ± 2.5	55.4 ± 1.8	49.9 ± 4.0	30.6 ± 2.6
Gdumb	70.6 ± 1.4	63.1 ± 0.9	6.3 ± 0.5	66.8 ± 5.5	**55.3 ± 5.2**	4.3 ± 2.7	98.4 ± 0.8	92.5 ± 2.6	0.2 ± 0.6
DER	64.9 ± 3.2	58.3 ± 1.4	13.4 ± 3.3	65.0 ± 5.8	51.7 ± 2.0	10.3 ± 3.2	86.1 ± 14.1	71.6 ± 8.0	10.6 ± 11.4
MEGA	66.1 ± 5.7	59.4 ± 2.6	3.2 ± 1.5	50.8 ± 1.1	50.0 ± 0.4	13.0 ± 2.1	57.1 ± 6.3	50.5 ± 5.9	17.3 ± 1.4
RM	**77.4 ± 1.9**	**69.8 ± 1.7**	4.3 ± 1.9	58.8 ± 2.8	54.4 ± 1.1	11.3 ± 1.2	80.5 ± 6.0	71.1 ± 8.8	5.6 ± 2.0
DIANA	76.7 ± 1.0	69.3 ± 1.7	**3.1 ± 0.7**	**73.4 ± 2.1**	51.0 ± 0.5	**0.2 ± 0.3**	**99.5 ± 0.2**	**87.4 ± 2.9**	**0.1 ± 0.1**

Table 2: The results of memory-based methods with Class Balanced Reservoir Sampling on medical images ISIC2019 and satellite images EuroSat.

Method	ISIC2019			EuroSat		
	AUC (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%
A-GEM	66.5 ± 10.3	**90.6 ± 0.1**	12.8 ± 11.9	79.4 ± 4.4	55.6 ± 4.0	14.4 ± 5.0
Gdumb	57.2 ± 9.0	71.1 ± 15.9	22.9 ± 9.1	81.6 ± 8.4	73.4 ± 8.9	2.4 ± 6.6
DER	61.8 ± 13.7	84.5 ± 12.6	17.7 ± 17.1	81.2 ± 4.5	65.3 ± 6.2	7.4 ± 4.2
MEGA	54.9 ± 9.6	62.3 ± 15.9	18.8 ± 10.7	77.8 ± 7.3	69.7 ± 10.3	2.0 ± 3.6
RM	72.8 ± 5.0	71.5 ± 3.7	10.3 ± 2.5	86.3 ± 2.6	79.3 ± 1.5	0.7 ± 4.2
DIANA	**77.7 ± 4.2**	78.2 ± 4.7	**1.9 ± 3.2**	**90.9 ± 1.7**	**83.8 ± 2.7**	**-1.0 ± 1.8**

For imbalanced lifelong learning, class balanced sampling is an efficient way to alleviate the harm of imbalanced data stream in some cases. Class-Balancing Reservoir Sampling (CBRS) is an enhanced Reservoir Sampling strategy for memory-based methods under imbalanced lifelong learning settings. It maintains a class-balanced memory by replacing instances of the major class. Thus, in experience replay, training batches from episodic memory are collected in a balanced form. However, since memory-based methods require data from both the current task and episodic memory, they still need to handle an imbalanced data stream from the current task even though the memory is balanced. Moreover, the distributions of balanced and imbalanced data streams differ significantly, which may lead to server gradient interference between these two streams.

We examine memory-based approaches with CBRS instead of vanilla Reservoir Sampling, the learning rate and batch size are consistent with Section 5.1. We implement CBRS according to the pseudo-code described in Chrysakis and Moens [2020]. When the memory is not full, all samples are stored in memory. After the memory is full, all classes are categorized as the full class and the not-full class. Once a class has been the largest class in memory, it's marked as the full class and would be ignored by the population. On the contrary, the instances of the full class would be replaced by the not-full class.

Table 1 and 2 present the results with CBRS. The evolution of average AUC during the lifelong learning process is shown in

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Figure 1. GEM is not included because this algorithm is not compatible with CBRS, it uses Ring Buffer instead of Reservoir Sampling and each task has individual memory space. Compared with vanilla Reservoir sampling in Table 3 and 4, using CBRS significantly improves the performance of most algorithms, except A-GEM. On Split-CIFAR, A-GEM, GDumb, DER, MEGA, and DIANA increase 1.5%, 5.1%, 2.5%, 3.0%, and 8.3% in terms of AUC score respectively. We also compare DIANA with another class-balanced method Rainbow Memory (RM), it uses a different class-balanced sampling strategy. Our proposed DIANA beats RM except Split-CIFAR.

We give a detailed analysis of each method below.

- A-GEM obtains no benefits from CBRS. Since the current data stream is still imbalanced, we analyze that it's not suitable for A-GEM to rectify the current gradient based on the reference gradient which is computed on a balanced replay buffer.
- GDumb has stable and significant increments on all 5 datasets, because it's only trained on balanced memory data, and doesn't suffer from the gradient interference problem.
- DER increases on Split-CIFAR, Split-AWA2, and EuroSat, while dropping performance on ISIC2019 and Split-CUB.
- MEGA gets boosted on Split-CIFAR and Split-CUB, but drops on the other three datasets.
- DIANA has much better AUC than other methods as shown in Table 1 and Table 2. There are two interesting observations. First, on ISIC2019 dataset, DIANA has worse accuracy than A-GEM (78.2% versus 90.6%) but better AUC value (77.7% versus 66.5%). The reason is that the dataset ISIC2019 is very imbalanced (See Section 5.1 Datasets). Second, on EuroSat dataset, DIANA has a negative forgetting measure, which means that DIANA with CBRS can help learn previous tasks when learning new tasks.

In summary, DIANA achieves higher AUC than other memory-based approaches when applying Class-Balancing Reservoir Sampling. DIANA has consistent improvements on all 5 datasets. We have a conjecture that the two-model framework alleviates the gradient interference between imbalanced data stream and balanced replay buffer, according to Section 4.2.

B RESULT DETAIL

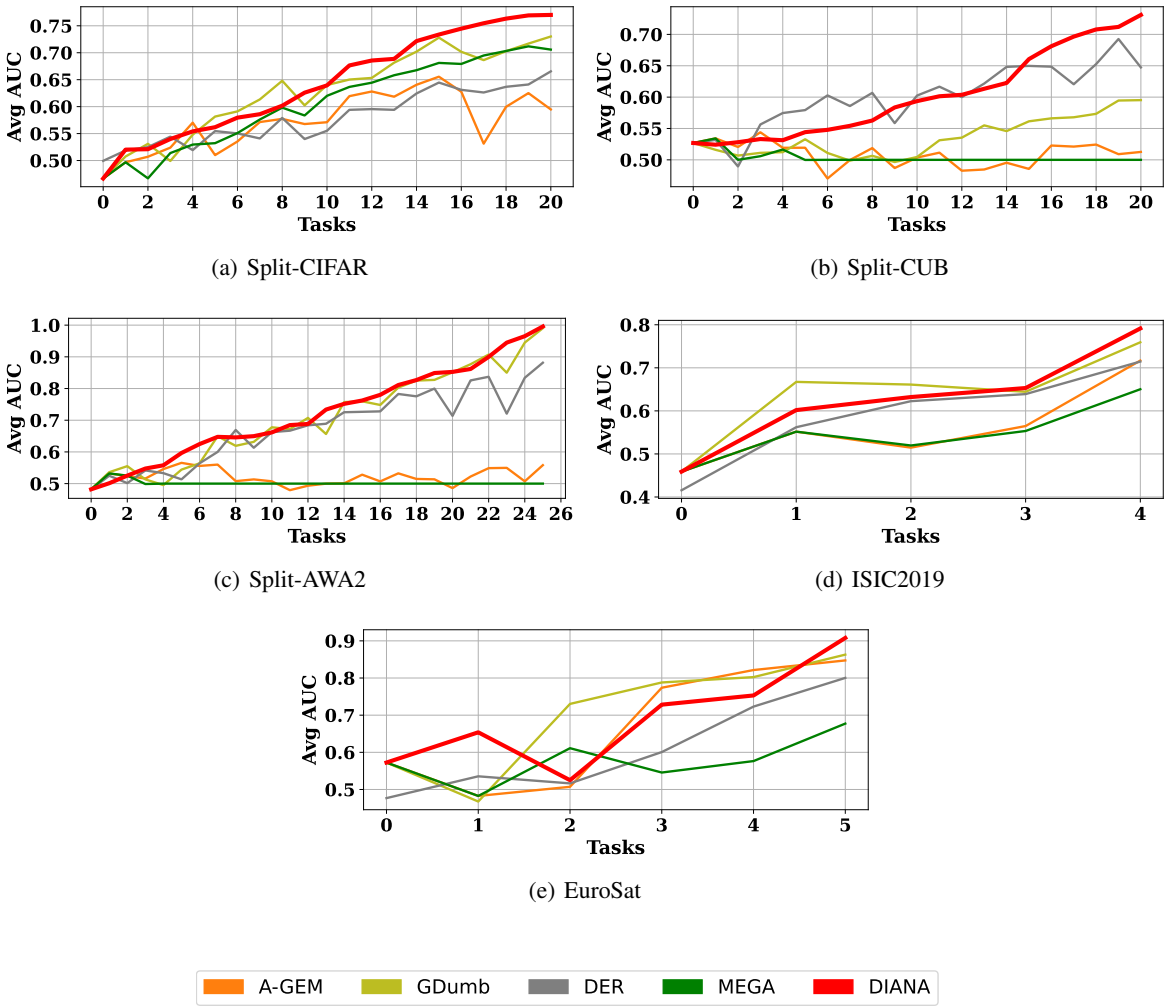
We show the detailed results of all the methods on imbalanced benchmarks and practical data in Figure 2, Table 3 and Table 4.

C IMBALANCED RATIO

In Figure 4, we vary the imbalanced ratio to evaluate the robustness of different methods. We consider three imratio settings: $\{0.01, 0.05, 0.1\}$. As expected, when the imratio increases, all the algorithms generally achieve better performance. Across all three settings, DIANA achieves the highest performance except when $imratio = 0.01$ on ISIC2019. It is worth mentioning that when imratio is 0.01, i.e., only 1% of samples are positive, all the methods drop drastically including DIANA. The possible reason is that when imratio is 0.01, the positive samples are very rare so it is difficult for all the methods to learn efficiently.

D BALANCED VERSUS IMBALANCED

In this section, it's studied how imbalance affects lifelong learning. In hyperplane, if the direction of gradient descent changes a lot, it's probably unstable and hard to optimize. Thus, we can exploit the deviation of direction to indicate whether a task is getting harder or not. We first store all gradients in a task and get an averaged gradient as an anchor. Then calculate the angle between the anchor and each mini-batch gradient. Drawing the histograms of angle deviation, we can see the distribution of gradient direction before and after making it imbalanced. It's plotted in Figure 5. After making imbalanced, the histograms have a wider range and larger std, which means the directions of gradient vary more dramatically.



(a) Split-CIFAR

(b) Split-CUB

(c) Split-AWA2

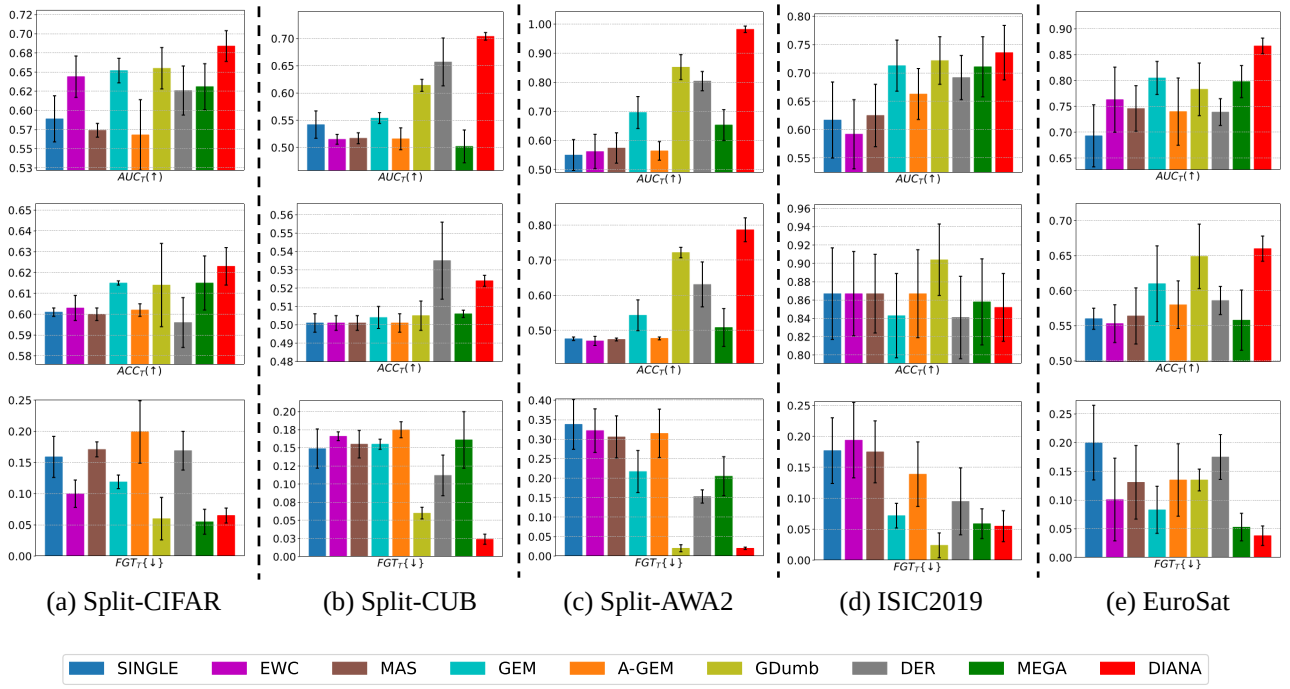
(d) ISIC2019

(e) EuroSat

Figure 1: Evolution of average AUC of memory-based methods with Class Balanced Reservoir Sampling.

References

Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pages 1952–1961. PMLR, 2020.



(a) Split-CIFAR (b) Split-CUB (c) Split-AWA2 (d) ISIC2019 (e) EuroSat

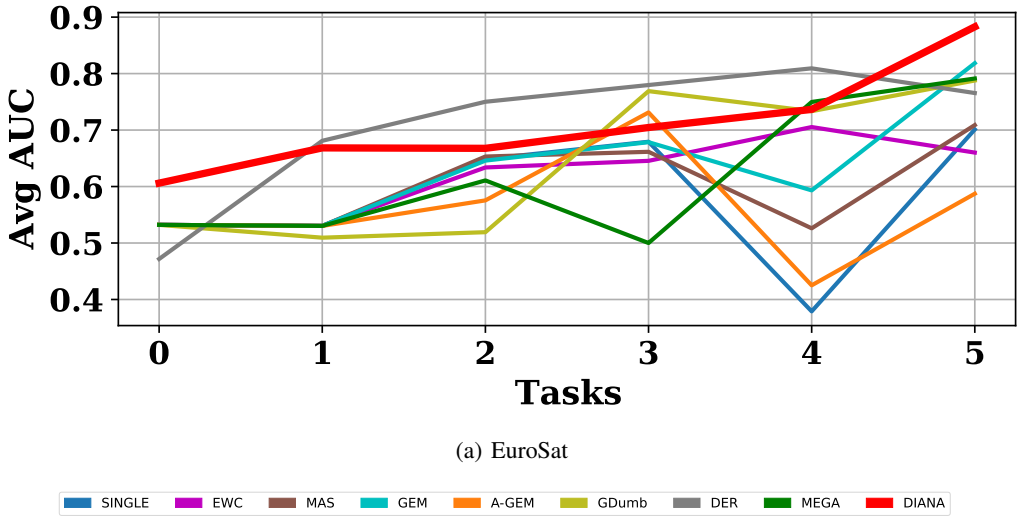
Figure 2: Performance of DIANA and other baselines on Split-CIFAR, Split-CUB, Split-AWA2, ISIC, and EuroSAT, each result is averaged after 5 runs with different random seeds. The Standard Deviation (std) is represented by the black line.

Table 3: The results of average AUC, average ACC, and average Forgetting (FGT) of different methods on Split CIFAR100, Split CUB200, and Split AWA2.

Method	Split-CIFAR			Split-CUB			Split-AWA2		
	AUC (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%
SINGLE	58.9 ± 3.0	60.1 ± 0.2	15.9 ± 3.3	54.2 ± 2.5	50.1 ± 0.5	14.9 ± 2.7	55.0 ± 5.3	47.6 ± 0.5	33.8 ± 6.4
EWC	64.4 ± 2.7	60.3 ± 0.6	10.0 ± 2.2	51.5 ± 0.9	50.1 ± 0.4	16.6 ± 0.6	56.2 ± 5.9	47.0 ± 1.3	32.2 ± 5.6
MAS	57.4 ± 0.9	60.0 ± 0.3	17.1 ± 1.2	51.7 ± 1.0	50.1 ± 0.4	15.5 ± 1.9	57.4 ± 5.2	47.4 ± 1.3	30.6 ± 5.4
GEM	65.2 ± 1.6	61.5 ± 0.1	11.9 ± 1.1	55.4 ± 1.0	50.4 ± 0.6	15.5 ± 0.7	69.6 ± 5.5	54.3 ± 4.4	21.7 ± 5.4
A-GEM	56.8 ± 4.6	60.2 ± 0.3	19.9 ± 5.0	51.6 ± 2.0	50.1 ± 0.5	17.5 ± 1.1	56.4 ± 3.2	47.7 ± 0.4	31.5 ± 6.2
GDumb	65.5 ± 2.7	61.4 ± 2.0	6.0 ± 3.4	61.4 ± 1.1	50.5 ± 0.8	6.0 ± 0.8	85.2 ± 4.3	72.2 ± 1.5	**1.9 ± 0.9**
DER	62.6 ± 3.2	59.6 ± 1.2	16.9 ± 3.1	65.7 ± 4.4	**53.5 ± 2.1**	11.2 ± 3.8	80.4 ± 3.3	63.1 ± 6.4	15.3 ± 1.7
MEGA	63.1 ± 3.0	61.5 ± 1.3	**5.5 ± 2.0**	50.1 ± 3.0	50.6 ± 0.3	16.1 ± 3.9	65.3 ± 5.3	50.8 ± 5.4	20.5 ± 5.0
DIANA	**68.4 ± 2.0**	**62.3 ± 0.9**	6.6 ± 1.2	**70.4 ± 0.7**	52.3 ± 0.3	**2.4 ± 0.7**	**98.2 ± 1.1**	**78.7 ± 3.4**	2.0 ± 0.3

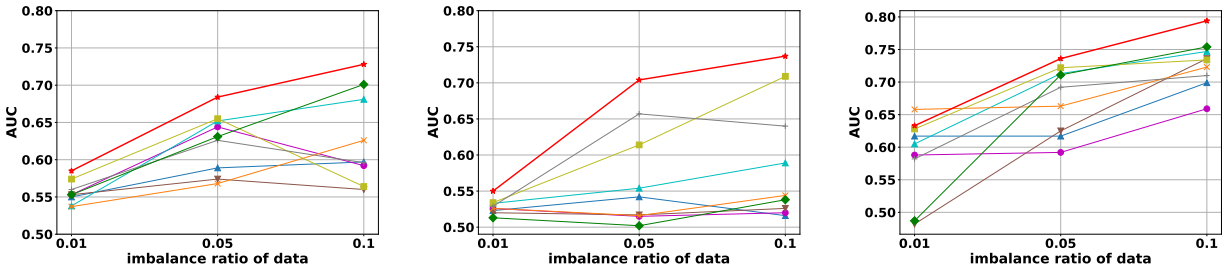
Table 4: The results of average AUC, average ACC, and average Forgetting (FGT) of different methods on medical images ISIC2019 and satellite images EuroSat.

Method	ISIC2019			EuroSat		
	AUC (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%
SINGLE	61.7 ± 7.7	86.7 ± 5.0	17.7 ± 5.3	69.3 ± 6.0	56.0 ± 1.5	20.0 ± 6.5
EWC	59.2 ± 8.1	86.7 ± 5.0	19.4 ± 6.1	76.3 ± 6.3	55.3 ± 2.7	10.1 ± 7.2
MAS	62.5 ± 6.5	86.7 ± 5.0	17.5 ± 6.0	74.6 ± 4.4	56.4 ± 4.0	13.1 ± 6.4
GEM	71.3 ± 6.5	84.3 ± 4.6	7.2 ± 2.0	80.5 ± 3.2	61.0 ± 5.4	8.3 ± 4.1
A-GEM	66.3 ± 6.5	86.7 ± 4.8	13.9 ± 5.2	74.0 ± 6.5	58.0 ± 3.4	13.5 ± 6.3
GDumb	72.2 ± 4.2	**90.4 ± 0.5**	**4.4 ± 2.0**	78.3 ± 5.5	64.9 ± 4.6	4.8 ± 1.9
DER	69.2 ± 3.9	84.1 ± 4.5	9.5 ± 5.4	73.9 ± 2.6	58.6 ± 2.0	17.5 ± 3.9
MEGA	71.1 ± 5.3	85.8 ± 4.7	5.9 ± 2.4	79.8 ± 3.1	55.8 ± 4.3	5.3 ± 2.4
DIANA	**73.6 ± 4.8**	85.2 ± 3.7	5.5 ± 2.5	**86.7 ± 1.5**	**66.0 ± 1.8**	**3.8 ± 1.7**



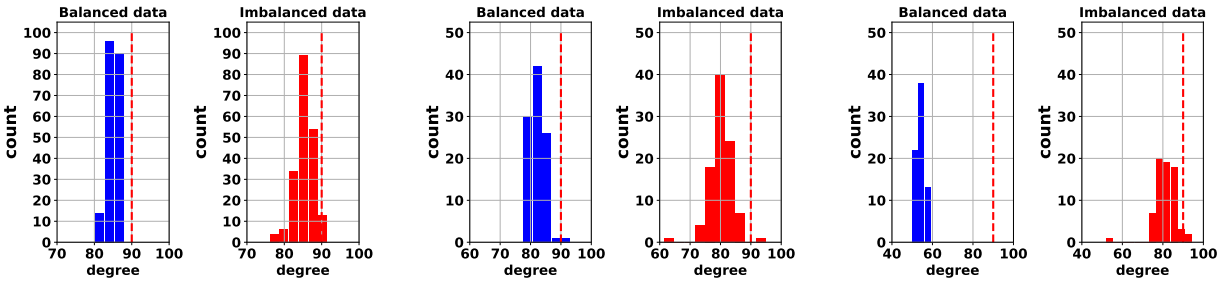
(a) EuroSat

Figure 3: Evolution of average AUC on EuroSat.



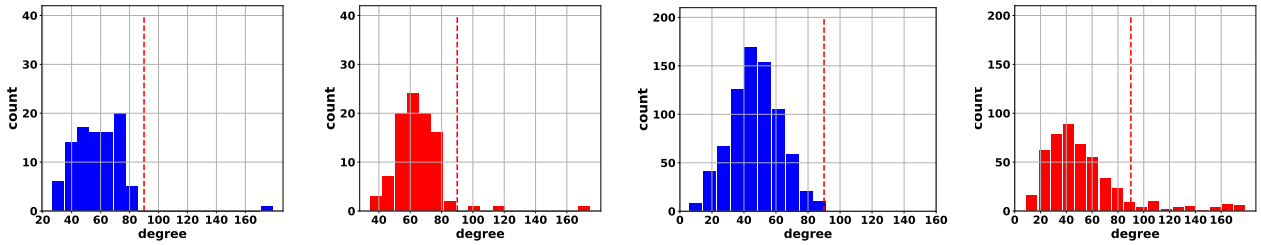
(a) Split-CIFAR (b) Split-CUB (c) ISIC2019

Figure 4: Average AUC on the data with different imbalanced ratio.



(a) Split-CIFAR (b) Split-CUB (c) ISIC2019

Figure 5: The distributions of the directions of the gradients on the current tasks. (a) balance std: 1.59, imbalance std: 3.50; (b) balance std: 2.58; imbalance std: 3.85; (c)balance std: 2.24, imbalance std: 5.52.



(a) Balanced Split-CUB (b) Imbalanced Split-CUB (c) Balanced Split-CIFAR (d) Imbalanced Split-CIFAR

Figure 6: The distributions of the angles between the current gradient and reference gradient, where the proportion of angles greater than 90 degrees is: (a) 1.05%, (b) 3.16%, (c) 0.13%, (d) 9.68%