Exploiting Inter-Sample Information for Long-tailed Out-of-Distribution Detection

Supplementary Material

A. Hyper-parameter tuning

We used a validation set of 10% and 17% of training data for CIFAR10/100-LT and ImageNet-LT respectively. For experiments on CIFAR10/100-LT, we empirically set $k = 7$ to create the k-NN graphs with self-loops using 512-dimensional pre-trained embeddings. The pre-trained model is trained for 100 epochs using SGD with Nesterov momentum and a cosine learning rate, with batch size 256 following [13]. To update the BN statistics, BN layers are fine-tuned for another 20 epochs with a learning rate of 0.001 using all the training data. GCN is trained for 200 epochs using the Adam optimizer with an initial learning rate of 0.001 and a cosine annealing learning rate scheduler. For ImageNet-LT, k is set to 2 and pre-trained embeddings are 2048-dimensional. BN layers are fine-tuned for 20 epochs with a learning rate of 0.001. GCN is trained for 250 epochs using the Adam optimizer with an initial learning rate of 0.01 which is decayed by a factor of 10 at epoch 15 and 100. On all datasets, we set $\lambda = 0.5$ following [39].

B. Additional ablation studies

B.1. Sensitivity analysis on k and λ

As seen, given a meaningful feature space, our graph combination can achieve a remarkable improvement, especially in terms of FPR95 and ACC_{tail}. However, the improvement in tail-classes compensates for a slight decline in ACC_{head}. This compensation can be attributed to the interplay between samples from tail-classes and the representative samples from head-classes. To provide a better explanation, we observe the behaviour of head and tail classes concerning the number of neighbor connections available for each sample. By increasing the k in the k-NN graph, we increase the number of edges in the overall graph. As we can see in Fig. 4, the increase in k causes ACC_{head} to drop, while ACC_{tail} continues to grow along with the overall ACC. This suggests that too much neighborhood information around head-samples is undesirable, leading to over-smoothing of representative samples. In contrast, this additional information benefits the tail-samples, resulting in a much-improved ACC_{tail} and a higher overall ACC.

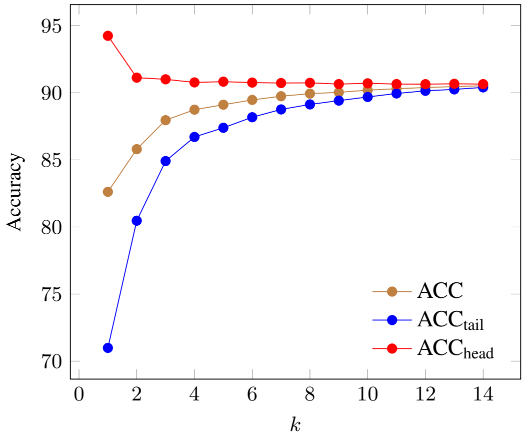


Figure 4. Variation of CIFAR10-LT ID classification accuracy over selection of k in the k-NN graph.

Table 5. Ablation study on λ for CIFAR10-LT as ID. Average values on all six $\mathcal{D}_{\text{out}}^{\text{test}}$ are reported.

λ	AUROC	AUPR	FPR95	ACC	ACC_{head}	ACC_{tail}
0.00	96.95	96.58	12.10	89.77	90.84	88.70
0.25	97.33	97.03	10.70	89.77	90.78	88.76
0.50	97.42	97.15	10.58	89.73	90.74	88.73
1.00	97.46	97.22	10.70	89.64	90.70	88.58

We also conduct an ablation study on λ that controls the importance of the OE loss term as summarized in Tab. 5. The performance of our method is stable with respect to different λ values, showing a notable decrease only when λ is set to 0.

B.2. Model architecture

In Tab. 3, we employ ResNet18 as the pre-trained backbone model, aligning with prior work [3, 39] that addresses OOD detection in long-tailed recognition. In this section, we demonstrate the versatility of our method across various model architectures. Specifically, Fort *et al.* [7] suggested that utilizing a pre-trained ViT can enhance OOD detection, although they did not evaluate their method on long-tailed IDs. We therefore replace pre-trained ResNet18 model with a ViT-L_16 model pre-trained on ImageNet-1k [33] dataset. We observed that for ViTs, it would be more helpful to fine-tune the entire network before feature extraction. This results in even better performance (ViT-L_16+GCN in Tab. 6) than the ResNet18 (ResNet18+GCN) underscoring the effectiveness of our method across diverse model architectures.

Table 6. Ablation study on model architecture for CIFAR10-LT and CIFAR100-LT as ID. Average values on all six $\mathcal{D}_{\text{out}}^{\text{test}}$ are reported. The best results are shown in bold. Mean over six random runs are reported.

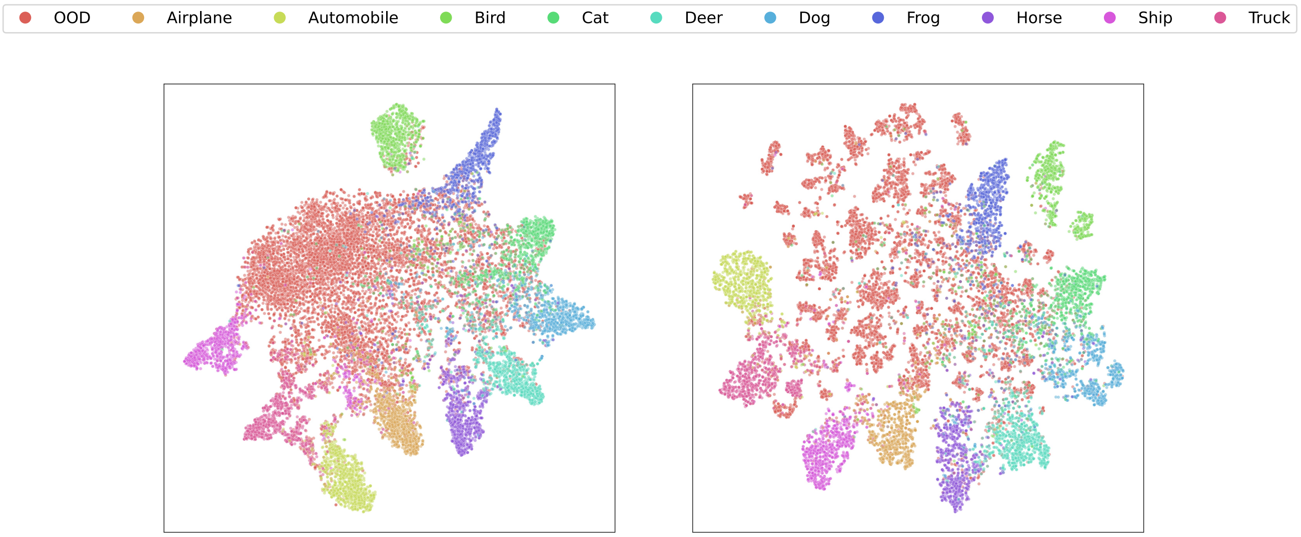
$\mathcal{D}_{\text{in}}$	Method	AUROC	AUPR	FPR95	ACC	ACC$_{\text{head}}$	ACC$_{\text{tail}}$
CIFAR10-LT	ResNet18	92.59	93.15	34.06	80.23	94.18	66.28
	ResNet18+GCN	97.43	97.17	10.60	89.74	90.71	88.76
	ViT-L_16	97.81	96.55	7.66	94.97	97.30	92.64
	ViT-L_16+GCN	98.34	97.10	4.36	95.80	96.18	95.42
CIFAR100-LT	ResNet18	78.01	74.64	64.11	55.66	77.94	33.38
	ResNet18+GCN	85.09	82.19	50.34	62.97	74.76	51.17
	ViT-L_16	86.81	77.51	37.51	69.74	88.62	50.86
	ViT-L_16+GCN	89.26	79.33	28.31	74.90	86.20	63.61

Table 7. Ablation study on batch-wise inference for CIFAR10-LT as ID. Average values on all six $\mathcal{D}_{\text{out}}^{\text{test}}$ are reported. The best results are shown in bold. Mean over six random runs are reported.

Batch Size	AUROC	AUPR	FPR95	ACC	ACC$_{\text{head}}$	ACC$_{\text{tail}}$
512	95.96	95.84	16.43	83.19	87.87	78.50
1024	96.57	96.41	14.13	86.35	88.81	83.90
2048	96.86	96.66	13.01	87.58	89.37	85.80
4096	97.11	96.90	11.98	88.63	90.14	87.11
8192	97.25	96.97	11.29	89.15	90.40	87.90
All Samples	**97.43**	**97.17**	**10.60**	**89.74**	**90.71**	**88.76**

Table 8. Ablation study on batch-wise inference for CIFAR100-LT as ID. Average values on all six $\mathcal{D}_{\text{out}}^{\text{test}}$ are reported. The best results are shown in bold. Mean over six random runs are reported.

Batch Size	AUROC	AUPR	FPR95	ACC	ACC$_{\text{head}}$	ACC$_{\text{tail}}$
512	69.15	67.49	77.66	32.90	48.13	17.68
1024	75.22	73.56	69.90	42.17	57.75	26.59
2048	79.81	77.79	62.14	50.45	65.33	35.56
4096	82.51	80.08	56.43	56.09	69.73	42.46
8192	83.84	81.21	53.30	59.42	71.94	46.90
All Samples	**85.09**	**82.19**	**50.34**	**62.97**	**74.76**	**51.17**



(a) MLP.

(b) GCN.

Figure 5. Feature space representations obtained from the (a) MLP and (b) GCN models for CIFAR10-LT as ID test set and CIFAR-100 as OOD test set using t-SNE. Equal number of test samples from each ID class are visualized. The GCN demonstrates distinct gaps between the decision boundaries of ID and OOD samples compared to the MLP.

Table 9. Performance on CIFAR10-LT with all CIFAR10-related classes removed from the pre-training dataset. Average values on all six $\mathcal{D}_{\text{out}}^{\text{test}}$ are reported.

Method	AUROC	AUPR	FPR95	ACC	ACC$_{\text{head}}$	ACC$_{\text{tail}}$
cifar_excluded	96.55	96.33	14.66	87.78	88.95	86.60
All	97.43	97.17	10.60	89.74	90.71	88.76

B.3. An alternative inference method

In our method, during inference, we construct a test graph using all ID and OOD test samples. However, in practical scenarios, inference is typically performed on batches of test samples which may belong to either ID or OOD. To ensure compatibility with such practical scenarios, we conduct experiments by processing batches of test samples instead of the full test data. In particular, for each randomly selected batch of data, we generate the k-NN graph over these data points alone to make the inference on that specific batch. Results for CIFAR10-LT and CIFAR100-LT as ID are indicated in Tab. 7 and Tab. 8, respectively. Results indicate that our method exhibits stability in batch inference, delivering nearly comparable performance to processing all samples, particularly noticeable with larger batch sizes.

B.4. Effect of the pre-training dataset

To further examine the effect of the pre-training dataset, similar to an ablation study of [13], we removed 153 CIFAR-10 related classes from Downsampled ImageNet and reran the experiments for CIFAR10-LT. This further ensures the pre-trained model has not seen any tail-class or super-class related instances. Results shown in Tab. 9 indicate similar performance gains. As we can see, our method still achieves SOTA (w.r.t. other baselines in Tab. 3) with only a small degradation compared to the model that uses all samples. This demonstrates that the pre-trained model do not rely on seeing CIFAR-10 related samples, and that simply training on more natural images increases the overall performance [13].

B.5. Graph structure

To further validate the importance of the graph structure, we replace the shallow linear classifier in the "Pre-train+Gau." baseline with a deeper Multi-Layer Perceptron (MLP) classifier. This MLP has an equivalent number of layers and trainable parameters as the GCN. Results for CIFAR10-LT and CIFAR100-LT as ID are shown in Tab. 10 and Tab. 11, respectively. As the MLP does not consider any inter-sample relationships, the outcome demonstrates lower performance for the "Pre-train+Gau.+MLP" baseline compared to the combination with GCN (Pre-train+Gau.+GCN). This observation is particularly evident in terms of FPR95 and ACC$_{\text{tail}}$.

Table 10. Ablation study on importance of GRL using a MLP for CIFAR10-LT as ID. The best results are shown in bold. Mean over six random runs are reported.

(a) OOD detection results.

$\mathcal{D}_{\text{out}}^{\text{test}}$	Method	AUROC	AUPR	FPR95
Texture	Pre-train+Gau.+MLP	99.12	98.54	3.21
	Pre-train+Gau.+GCN	**99.50**	**98.92**	**1.10**
SVHN	Pre-train+Gau.+MLP	99.35	**99.73**	2.61
	Pre-train+Gau.+GCN	**99.68**	99.71	**0.67**
CIFAR100	Pre-train+Gau.+MLP	91.11	91.70	41.16
	Pre-train+Gau.+GCN	**92.08**	**92.21**	**35.28**
TinyImageNet	Pre-train+Gau.+MLP	94.40	92.93	28.66
	Pre-train+Gau.+GCN	**95.61**	**93.64**	**19.93**
LSUN	Pre-train+Gau.+MLP	98.85	98.84	4.87
	Pre-train+Gau.+GCN	**99.57**	**99.43**	**0.74**
Places365	Pre-train+Gau.+MLP	97.55	**99.17**	14.86
	Pre-train+Gau.+GCN	**98.12**	99.12	**5.89**
Average	Pre-train+Gau.+MLP	96.73	96.82	15.90
	Pre-train+Gau.+GCN	**97.43**	**97.17**	**10.60**

(b) ID classification results.

Method	ACC	ACC$_{\text{head}}$	ACC$_{\text{tail}}$
Pre-train+Gau.+MLP	87.63	**94.64**	80.61
Pre-train+Gau.+GCN	**89.74**	90.71	**88.76**

Table 11. Ablation study on importance of GRL using a MLP for CIFAR100-LT as ID. The best results are shown in bold. Mean over six random runs are reported.

(a) OOD detection results.

$\mathcal{D}_{\text{out}}^{\text{test}}$	Method	AUROC	AUPR	FPR95
Texture	Pre-train+Gau.+MLP	**90.95**	**84.76**	39.54
	Pre-train+Gau.+GCN	90.88	84.10	**39.32**
SVHN	Pre-train+Gau.+MLP	95.20	97.51	18.89
	Pre-train+Gau.+GCN	**97.02**	**98.19**	**10.19**
CIFAR10	Pre-train+Gau.+MLP	73.39	68.21	70.79
	Pre-train+Gau.+GCN	**75.08**	**74.17**	**70.62**
TinyImageNet	Pre-train+Gau.+MLP	79.19	67.42	65.46
	Pre-train+Gau.+GCN	**80.06**	**68.96**	**63.48**
LSUN	Pre-train+Gau.+MLP	81.03	72.32	68.22
	Pre-train+Gau.+GCN	**83.29**	**75.09**	**59.57**
Places365	Pre-train+Gau.+MLP	83.21	92.05	60.41
	Pre-train+Gau.+GCN	**84.22**	**92.63**	**58.87**
Average	Pre-train+Gau.+MLP	83.83	80.38	53.88
	Pre-train+Gau.+GCN	**85.09**	**82.19**	**50.34**

(b) ID classification results.

Method	ACC	ACC$_{\text{head}}$	ACC$_{\text{tail}}$
Pre-train+Gau.+MLP	61.62	**79.12**	44.13
Pre-train+Gau.+GCN	**62.97**	74.76	**51.17**

This fact can be further clarified by examining the feature space representations from these two models, as depicted in Fig. 5. As seen, there are distinct gaps between the decision boundaries of ID and OOD samples in the representations obtained from the GCN (Fig. 5b) compared to the MLP (Fig. 5a).

Table 12. Comparison of the training and inference speed (averaged per-image) for CIFAR100-LT *vs.* CIFAR10 experiments on an NVIDIA-A100 GPU. For our method, different k values (indicated in brackets) are compared.

Method	**Train. time(s)**	**Infer. time(ms)**
OECC	715.57	0.27
EnergyOE	687.40	0.24
OE	690.09	0.11
PASCL	1265.63	0.13
BE-OE	706.87	0.24
EAT	3526.58	0.14
COCL	659.74	0.11
Pre-train+KNN	603.96	1.67
Ours	478.51 (k=3) 533.13 (k=7) 620.10 (k=10)	2.09 (k=3) 2.10 (k=7) 2.11 (k=10)

B.6. Computational cost

Tab. 12 compares the training and inference times of our method with other baselines. For our method, the total training time includes the time taken for Gaussianization, graph creation, and GCN training. Despite these components, the overall training time remains shorter because we employed a lightweight GCN model with only 578,860 parameters. This is in contrast to most of the other methods, which require 11,599,752 parameters to be trained from scratch. However, similar to KNN method, inference takes slightly longer than other methods due to the overhead introduced by k nearest neighbour search and the graph creation process. Notably, varying k values does not significantly impact the inference speed.

B.7. Performance on balanced ID datasets

We also compare the performance of our method on balanced ID datasets in Tab. 13. Note that some LTR baselines, including PASCL, OS, EAT, and COCL (which we did not include in the table), cannot be directly evaluated or reduced to other baselines when there are zero tail-classes. For our method, we used the same experimental settings as in the LT experiments, except that the imbalance ratio ρ was set to 1. As shown, our method outperforms most baselines for OOD detection across all metrics while demonstrating comparable performance in ID classification accuracy. There is a small effect of reduction of ID accuracy, when using higher k values, which result in larger number of neighbors around ID nodes in the graph. However, the inter-sample relationships between nodes greatly benefit the OOD detection. This is part of the trade-off between leveraging useful relational information and the risk of over-smoothing, a common issue inherent to many graph-based approaches.

Table 13. OOD detection results and ID classification results for balanced CIFAR-10 and CIFAR-100 as ID. Average values on all six $\mathcal{D}_{\text{out}}^{\text{test}}$ are reported. The best results are bolded. Mean over six random runs are reported.

$\mathcal{D}_{\text{in}}$	**Method**	**AUROC**	**AUPR**	**FPR95**	**ACC**
CIFAR-10	OECC	96.33	95.38	14.36	91.57
	EnergyOE	96.77	96.72	14.82	93.30
	OE	96.67	95.97	13.80	93.64
	BE-OE	96.83	96.70	14.51	93.00
	Pre-train+KNN	94.82	94.03	26.36	**95.91**
	Ours	**97.84**	**97.66**	**8.86**	91.40
CIFAR-100	OECC	84.03	77.94	45.26	69.55
	EnergyOE	85.84	80.99	43.02	74.95
	OE	83.92	78.07	49.49	71.36
	BE-OE	85.85	80.91	**42.93**	74.83
	Pre-train+KNN	85.52	80.71	55.56	**79.82**
	Ours	**87.17**	**83.91**	43.62	70.37

B.8. Detailed ablation results

In Table 4, we presented an ablation study over various components of our proposed approach. Due to space limitations, we omitted the detailed results over separate OOD data. Detailed ablation study results over separate $\mathcal{D}_{\text{out}}^{\text{test}}$ for CIFAR10-LT and CIFAR100-LT are shown in Tab. 14 and Tab. 15, respectively.

C. More details on Gaussianization

Regularization and normalization techniques are commonly applied to neural networks to ensure stable training and improved generalization performance [17, 19]. Among various normalization methods for controlling hidden activations, Gaussianization ensures that the activation layer representations follow a standard normal distribution. This approach relies on batch normalization (BN) layers, which take hidden activations as input to the next layer and normalize them to have zero mean and unit variance. During inference, the network uses 'running statistics' (running mean-μ_{running} and running variance-σ_{running}) that are computed and updated during training to normalize activations, i.e.,

$$\hat{x}_i = \frac{x_i - \mu_{\text{running}}}{\sqrt{\sigma^2_{\text{running}} + \epsilon}},$$

where x_i denotes the current data input and ϵ is a small constant added for numerical stability. In our pipeline, we used a backbone model f pre-trained on a different dataset ($\mathcal{D}_{\text{pretrain}}$) to extract initial feature representations for each training input. However, when calculating these feature representations, the running statistics from $\mathcal{D}_{\text{pretrain}}$ are used, which differ from those of the current training data ($\mathcal{D}_{\text{train}}$). This results in a misalignment of the activations from the intended standard normal distribution during inference on $\mathcal{D}_{\text{train}}$. To correct this, we need to fine-tune the BN layers of the backbone f on our training data $\mathcal{D}_{\text{train}}$, while keeping all other layers fixed. This fine-tuning will update the μ_{running} and σ_{running} to reflect the distribution of $\mathcal{D}_{\text{train}}$, aligning the activations with the intended standard normalization.

D. Limitations

A common limitation of our method can be that computational overhead introduced by graph creation (Appendix B.6). Future work could explore other graph-based approaches, such as GraphSAGE [10], which use neighborhood sampling to reduce computational costs. On the other hand, there is a trade-off between leveraging useful relational information by means of message passing and the risk of over-smoothing, which is a common issue in many graph-based approaches. We leave it to future work to introduce edge-weighted mechanisms and graph sampling techniques to address these potential limitations.

E. Code

Code to reproduce the results of our method is available at https://github.com/hdnugit/Graph_OOD_LTR.

Table 14. Ablation study for separate $\mathcal{D}_{\text{out}}^{\text{test}}$ for CIFAR10-LT as ID. The best and second-best results are bolded and underlined, respectively. Mean over six random runs are reported.

$\mathcal{D}_{\text{out}}^{\text{test}}$	**Pre-train**	**Gau.**	**GRL**	**Method**	**AUROC**	**AUPR**	**FPR95**
Texture	✗	✗	✗	Scratch (OE)	92.59	83.32	25.10
	✗	✗	✓	Scratch+GCN	86.34	69.51	32.64
	✓	✗	✗	Pre-train	94.42	90.58	26.77
	✓	✗	✓	Pre-train+GCN	99.16	98.53	2.10
	✓	✓	✗	Pre-train+Gau.	96.58	94.49	19.23
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**99.50**	**98.92**	**1.10**
SVHN	✗	✗	✗	Scratch (OE)	95.10	97.14	16.15
	✗	✗	✓	Scratch+GCN	87.67	91.66	32.16
	✓	✗	✗	Pre-train	97.00	98.71	15.64
	✓	✗	✓	Pre-train+GCN	99.44	99.63	1.72
	✓	✓	✗	Pre-train+Gau.	96.17	98.30	18.54
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**99.68**	**99.71**	**0.67**
CIFAR100	✗	✗	✗	Scratch (OE)	83.40	80.93	56.96
	✗	✗	✓	Scratch+GCN	79.55	73.12	56.27
	✓	✗	✗	Pre-train	84.13	83.90	57.24
	✓	✗	✓	Pre-train+GCN	88.87	88.38	44.18
	✓	✓	✗	Pre-train+Gau.	85.92	86.48	55.13
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**92.08**	**92.21**	**35.28**
TinyImageNet	✗	✗	✗	Scratch (OE)	86.14	79.33	47.78
	✗	✗	✓	Scratch+GCN	81.75	70.13	44.80
	✓	✗	✗	Pre-train	88.21	85.00	46.27
	✓	✗	✓	Pre-train+GCN	93.38	90.34	26.22
	✓	✓	✗	Pre-train+Gau.	90.20	88.02	42.69
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**95.61**	**93.64**	**19.93**
LSUN	✗	✗	✗	Scratch (OE)	91.35	87.62	27.86
	✗	✗	✓	Scratch+GCN	86.08	79.82	33.54
	✓	✗	✗	Pre-train	93.26	92.88	31.92
	✓	✗	✓	Pre-train+GCN	99.38	99.37	1.88
	✓	✓	✗	Pre-train+Gau.	94.30	94.40	30.40
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**99.57**	**99.43**	**0.74**
Places365	✗	✗	✗	Scratch (OE)	90.07	95.15	34.04
	✗	✗	✓	Scratch+GCN	84.09	90.98	35.92
	✓	✗	✗	Pre-train	91.18	96.66	40.32
	✓	✗	✓	Pre-train+GCN	97.76	99.09	9.55
	✓	✓	✗	Pre-train+Gau.	92.37	97.25	38.36
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**98.12**	**99.12**	**5.89**
Average	✗	✗	✗	Scratch (OE)	89.77	87.25	34.65
	✗	✗	✓	Scratch+GCN	84.25	79.20	39.22
	✓	✗	✗	Pre-train	91.37	91.29	36.36
	✓	✗	✓	Pre-train+GCN	96.33	95.89	14.28
	✓	✓	✗	Pre-train+Gau.	92.59	93.15	34.06
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**97.43**	**97.17**	**10.60**

Table 15. Ablation study for separate $\mathcal{D}_{\text{out}}^{\text{test}}$ for CIFAR100-LT as ID. The best and second-best results are bolded and underlined, respectively. Mean over six random runs are reported.

$\mathcal{D}_{\text{out}}^{\text{test}}$	**Pre-train**	**Gau.**	**GRL**	**Method**	**AUROC**	**AUPR**	**FPR95**
Texture	✗	✗	✗	Scratch (OE)	76.71	58.79	68.28
	✗	✗	✓	Scratch+GCN	78.70	56.03	62.57
	✓	✗	✗	Pre-train	77.39	62.69	65.45
	✓	✗	✓	Pre-train+GCN	88.70	81.86	46.35
	✓	✓	✗	Pre-train+Gau.	83.79	72.87	54.47
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**90.88**	**84.10**	**39.32**
SVHN	✗	✗	✗	Scratch (OE)	77.61	86.82	58.04
	✗	✗	✓	Scratch+GCN	79.04	85.16	62.77
	✓	✗	✗	Pre-train	83.80	91.15	49.25
	✓	✗	✓	Pre-train+GCN	89.42	93.85	34.56
	✓	✓	✗	Pre-train+Gau.	91.65	95.90	33.24
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**97.02**	**98.19**	**10.19**
CIFAR10	✗	✗	✗	Scratch (OE)	62.23	57.57	80.64
	✗	✗	✓	Scratch+GCN	64.98	59.53	78.15
	✓	✗	✗	Pre-train	64.02	60.22	80.23
	✓	✗	✓	Pre-train+GCN	72.80	68.96	71.20
	✓	✓	✗	Pre-train+Gau.	68.99	66.00	77.65
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**75.08**	**74.17**	**70.62**
TinyImageNet	✗	✗	✗	Scratch (OE)	68.04	51.66	76.66
	✗	✗	✓	Scratch+GCN	68.37	51.73	77.60
	✓	✗	✗	Pre-train	71.47	56.95	74.29
	✓	✗	✓	Pre-train+GCN	75.73	62.25	69.33
	✓	✓	✗	Pre-train+Gau.	74.70	61.74	70.48
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**80.06**	**68.96**	**63.48**
LSUN	✗	✗	✗	Scratch (OE)	77.10	61.42	63.98
	✗	✗	✓	Scratch+GCN	76.26	57.55	65.52
	✓	✗	✗	Pre-train	65.18	53.89	81.17
	✓	✗	✓	Pre-train+GCN	79.14	69.66	69.21
	✓	✓	✗	Pre-train+Gau.	72.06	62.26	77.96
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**83.29**	**75.09**	**59.57**
Places365	✗	✗	✗	Scratch (OE)	75.80	86.68	65.72
	✗	✗	✓	Scratch+GCN	76.26	85.78	64.88
	✓	✗	✗	Pre-train	71.18	85.69	77.00
	✓	✗	✓	Pre-train+GCN	82.57	91.73	62.65
	✓	✓	✗	Pre-train+Gau.	76.89	89.04	70.86
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**84.22**	**92.63**	**58.87**
Average	✗	✗	✗	Scratch (OE)	72.91	67.16	68.89
	✗	✗	✓	Scratch+GCN	73.93	65.97	68.58
	✓	✗	✗	Pre-train	72.17	68.43	71.23
	✓	✗	✓	Pre-train+GCN	81.40	78.05	58.88
	✓	✓	✗	Pre-train+Gau.	78.01	74.64	64.11
	✓	✓	✓	Pre-train+Gau.+GCN (ours)	**85.09**	**82.19**	**50.34**