

Learning to Generate Novel Classes for Deep Metric Learning - Supplementary Material

Kyungmoon Lee¹

kyungmoon@postech.ac.kr

Sungyeon Kim¹

sungyeon.kim@postech.ac.kr

Seunghoon Hong²

seunghoon.hong@kaist.ac.kr

Suha Kwak¹

suha.kwak@postech.ac.kr

¹ POSTECH

Pohang, South Korea

² KAIST

Daejeon, South Korea

This supplementary material presents an overall training pipeline of L2A-NC and additional experimental results, which are omitted from the main paper due to the space limit. Section 1 shows the full training algorithm which summarizes how the proposed L2A-NC works. Section 2 presents additional ablation studies on the proposed L2A-NC. Section 3 also provides additional qualitative results of image retrieval on benchmark datasets [4, 5, 6]. Lastly, we present additional t -SNE visualization results to compare the learned embedding space before and after applying our L2A-NC.

1 Algorithm

We first present algorithm 1 below to help readers understand the proposed L2A-NC framework. In detail, we first pretrain the main embedding network f via a specific proxy-based loss J_{met} consisting of only proxies and data points of real classes. The driving rationale behind this procedure is that since the main embedding network f can not well represent how distributions of real classes form from scratch, the embedding vectors X from the embedding network f can not offer correct signals for the conditional generator g when the conditional generator g optimizes $J_{div}(X, \tilde{X})$.

Next, we pretrain the conditional generator g as well, utilizing the embedding network f which is just pretrained in advance. Since, from scratch, the conditional generator would have difficulty generating embedding vectors of novel classes as desired in the target joint training phase, we warm up the generator to minimize the divergence loss J_{div} and the proxy-based metric learning loss J_{met} . Since novel classes which are already well separated from real classes would limit the additional signals for the embedding network in the joint training phase, the generator, in this phase, optimizes J_{met} which addresses proxies and embedding vectors of novel classes only.

Finally, and most importantly, during the joint training phase, the embedding network and generator jointly optimize J_{met} containing proxies and embedding vectors of both real

and novel classes. Note that J_{div} is also minimized together only by the conditional generator. It keeps regularizing the generator to synthesize realistic novel classes in this phase, even though the embedding space constantly changes while J_{met} is optimized.

Algorithm 1 Learning to Augment Novel Classes (L2A-NC)

Input: Real embedding vectors X whose labels are denoted as Y , novel-class labels \tilde{Y} , and latent variable Z .

Parameter: Parameters of the embedding network θ_f , parameters of the generator θ_g , proxies for real classes P , and proxies for novel classes \tilde{P}

Output: Parameters of the embedding network θ_f .

- 1: Pretrain $\{\theta_f, P\}$ minimizing $J_{met}(X, Y, P)$ defined at Eq.(6) of our main paper.
 - 2: Initialize and pretrain $\{\theta_g, \tilde{P}\}$ minimizing $J_{met}(\tilde{X}, \tilde{Y}, \tilde{P}) + \lambda_{div}J_{div}(X, \tilde{X})$.
 - 3: **while** not reaching the max iterations **do**
 - 4: Embed real embedding vectors X from the embedding network f .
 - 5: Generate embedding vectors of novel classes $\tilde{X} = g(\tilde{Y}, Z)$ as defined at Eq.(1) of our main paper.
 - 6: Optimize $\{\theta_f, \theta_g, P, \tilde{P}\}$ minimizing $J_{met}(X \cup \tilde{X}, Y \cup \tilde{Y}, P \cup \tilde{P}) + \lambda_{div}J_{div}(X, \tilde{X})$ defined at Eq.(7) of our main paper.
 - 7: **end while**
 - 8: **return** θ_f
-

2 Additional Ablation Studies

In this section, we conduct additional ablation studies on the proposed L2A-NC, which are omitted from the main paper due to the space limit. First, we study the impact of how many embedding vectors of novel classes exist in a mini-batch. Next, we evaluate the impact of the hyperparameter λ_{div} which balances two losses: J_{met} and J_{div} . These ablation studies were conducted on the Cars-196 dataset with two proxy-based losses: Norm-softmax and Proxy-Anchor [9] which are chosen as an elementary version and a modern one, respectively.

Ratio	50%	100%	200%†	400%
Norm-softmax	85.4	85.4	86.0	86.2
Proxy-Anchor [9]	87.3	87.5	87.9	87.5

Table 1: Recall@1 versus the ratio of embedding vectors of novel classes per those of real classes in a mini-batch. †: The results with this ratio are reported as quantitative results in experiments.

Impact of the number of novel-class embedding vectors within a mini-batch. Thanks to the training strategy of the proposed framework, our conditional generator can augment novel-class embedding vectors of any size into a mini-batch. Therefore, we investigate the effect of how many embedding vectors of novel classes exist in a mini-batch on the generalization ability. We measure retrieval accuracy while varying the size of novel-class embedding vectors in a mini-batch. As shown in Table 1, the overall results demonstrate that the more embedding vectors of novel classes in a mini-batch, the better the model performance.

It is reasonable since a proxy-based loss calculated with more classes would offer richer semantic relations between training classes, which, in turn, results in a better generalized model. However, if there are too many embedding vectors of novel classes, the performance boosts may be limited as only a few of them would interact with those of real classes, which is somewhat aligned to the analysis shown in Table 2 (left) of our main paper. As Table 1 shows, we chose the ratio of embedding vectors of novel classes per those of real classes in a mini-batch as 200%. Considering that we fixed the number of novel classes as 200% of that of real classes, all classes including novel classes exist, on average, in equal proportions within a mini-batch.

λ_{div}	0.0	0.1	1.0†	10.0
Norm-softmax	84.3	85.8	86.0	86.2
Proxy-Anchor [9]	86.7	87.6	87.9	87.4

Table 2: Recall@1 versus the value of λ_{div} . †: The results with this value are reported as quantitative results in experiments.

Impact of the hyperparameter λ_{div} . Here we investigate the sensitivity of the generalization of the embedding network to the hyperparameter λ_{div} . λ_{div} is chosen among {0.0, 0.1, 1.0, 10.0}. The results are shown in Table 2. First of all, the largest performance gap was shown between the case of $\lambda_{div} = 0.0$ and that of $\lambda_{div} = 0.1$. This result supports the need for *realistic* novel classes. At the same time, it can also be interpreted to be aligned with a large body of literature on hard negative mining [14, 15, 16, 17]. On the other hand, we can see the performance boosts are somewhat trivial when increasing λ_{div} from 0.1 to 10.0. We speculate that this result is attributed to the warm-up phase of the conditional generator, which also confirms its motivation. Note that we fixed λ_{div} for the quantitative experimental results as 1.0 to equally balance two losses: J_{met} and J_{div} .

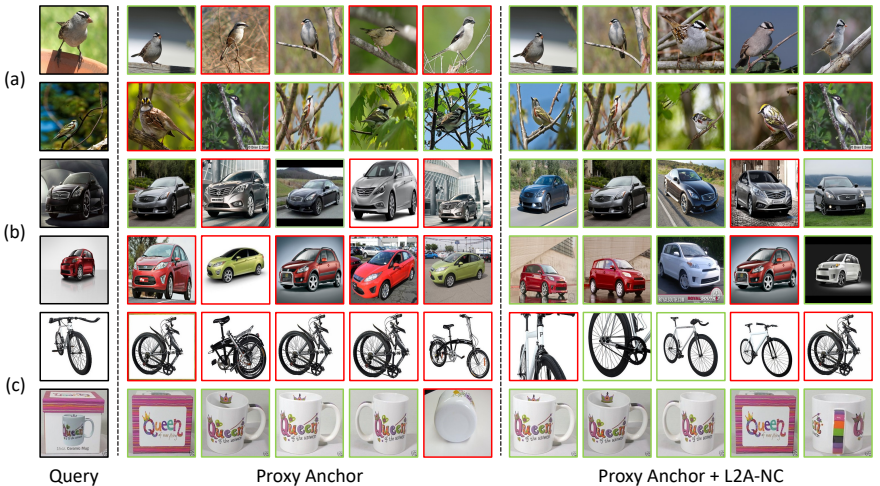


Figure 1: Qualitative results of the vanilla Proxy-Anchor [9] and that combined with L2A-NC on the (a) CUB-200-2011, (b) Cars-196, and (c) SOP datasets. Images with the green boundary are correct results while those with the red boundary are failure cases.

3 Additional Qualitative Results

More qualitative examples for image retrieval on the CUB-200-2011, Cars-196, and SOP datasets are shown in Figure 1. The results compare the image retrieval examples of the vanilla Proxy-Anchor [4] and that combined with our L2A-NC. The overall results show that Proxy-Anchor combined with L2A-NC records significantly higher retrieval accuracy than the vanilla Proxy-Anchor. Specifically, on the CUB-200-2011 (a), both models retrieve birds visually similar to the query, but the version with L2A-NC only successfully retrieves accurate results. Similarly, on the Cars-196 (b), L2A-NC helps the embedding network produce more accurate retrievals despite different object colors or viewpoints. Meanwhile, on the SOP (c), as shown in the 5th row of Figure 1, the vanilla Proxy-Anchor failed to retrieve even a single correct one. However, the version with L2A-NC produced accurate retrievals even in this extremely difficult case. To sum it up, we further prove the advantages of the proposed method by showing additional qualitative results before and after applying the proposed method in the image retrieval task.

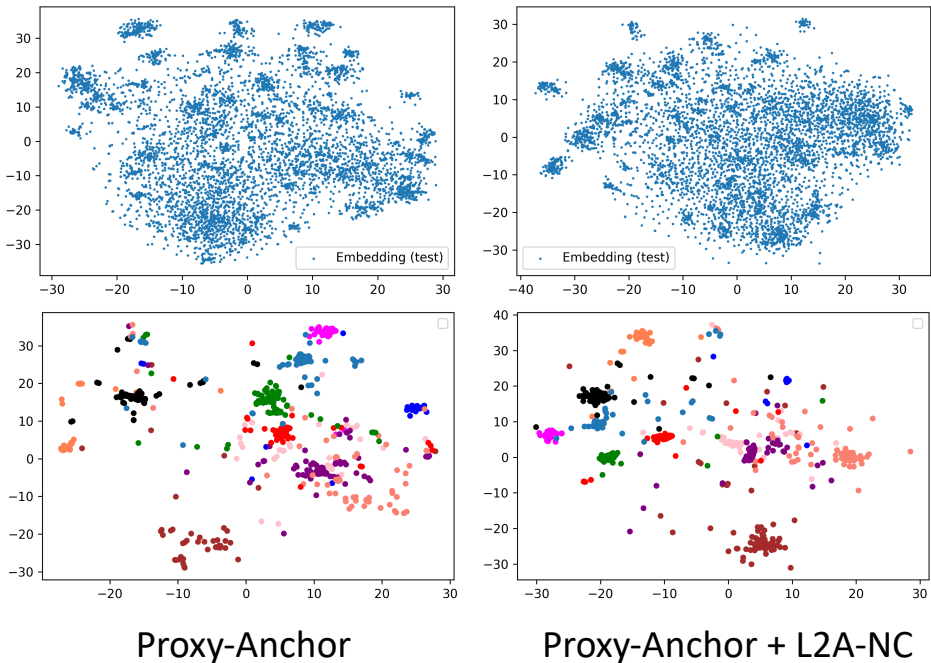


Figure 2: t -SNE visualization of the learned embedding space before and after applying our L2A-NC on the test split of the CUB dataset. **Top:** Visualization for all test classes. All classes are represented by a single color. **Bottom:** Visualization for a subset of test classes. Each color represents the same test class.

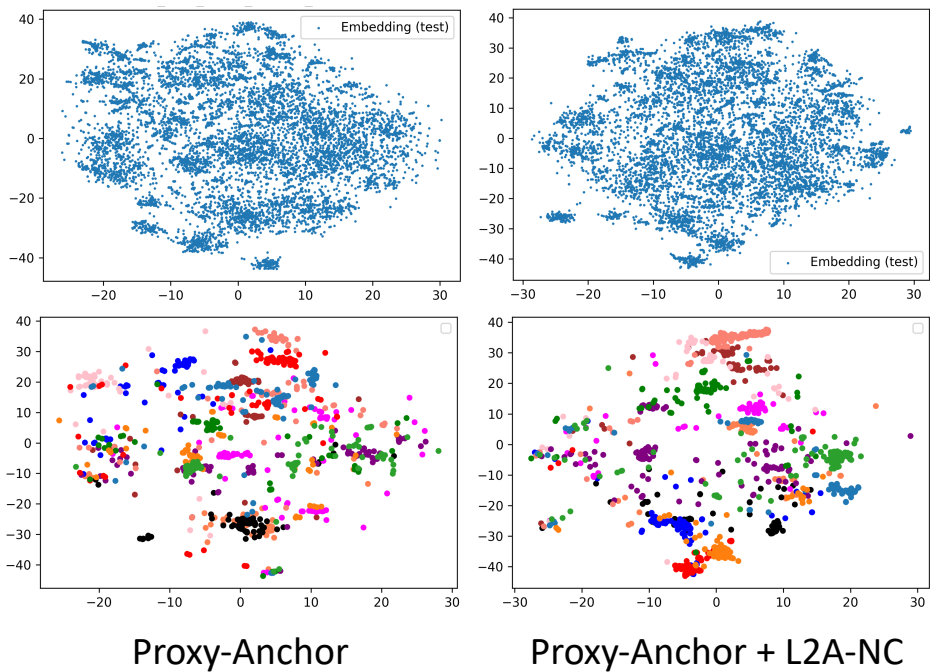


Figure 3: t -SNE visualization of the learned embedding space before and after applying our L2A-NC on the test split of the Cars-196 dataset. **Top:** Visualization for all test classes. All classes are represented by a single color. **Bottom:** Visualization for a subset of test classes. Each color represents the same test class.

References

- [1] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [2] Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [3] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [4] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proc. IEEE International Conference on Computer Vision (ICCV) Workshops*, 2013.
- [5] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [6] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [7] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, 2011.
- [8] Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.