RESCIENCE C

Replication / ML Reproducibility Challenge 2022

[Re] Hypergraph-Induced Semantic Tuplet Loss for Deep Metric Learning

Jicheng Yuan[1] and **Danh Le-Phuoc**[1,2]

[1]Open Distributed Systems, Technical University of Berlin, Berlin, Germany – [2]Fraunhofer Institute for Open Communication Systems, Berlin, Germany

Edited by
Koustuv Sinha,
Maurits Bleeker,
Samarth Bhargav

Received
04 February 2023

Published
20 July 2023

DOI
10.5281/zenodo.8173666

Reproducibility Summary

Scope of Reproducibility – Our work reproduced the paper *Hyper-graph-Induced Semantic Tuplet (HIST) Loss for Deep Metric Learning* [1] and investigate the HIST loss with the following five claims: (i) HIST loss performs consistently regardless of the batch size, (ii) its performance is irrespective of the number of HGNN layers, (iii) the positive scaling factor of semantic tuplets enhances semantic relation modeling, (iv) a large temperature parameter τ is effective; if τ >16, HIST loss is insensitive to the scaling parameter, (v) the HIST achieved SOTA performances under the standard evaluation settings [2, 3].

Methodology – To verify above claims, we extended the experiments proposed in [1]. Our study involves: (a) reproducing the HIST loss performances as configurations proposed in [1], (b) optimizing HIST loss using Bayesian optimization, and (c) investigating the impacts and robustnesses of key modules (prototypical distributions and semantic tuplets). These experiments were conducted on 2 NVIDIA V100 GPUs and took about 1,108 GPU hours.

Results – Our study confirms three (iii, iv and v) claims from [1]. However, it cannot fully support the other claims (i and ii). Using the configurations given in [1], we achieved comparable performances on the CARS196 dataset, but large deviances were observed on other datasets, which dropped by 1.5% and 1% R@1 using ResNet50. By implementing Bayesian optimization, the performances are improved by 0.7% and 0.4% R@1 on CARS196 and CUB-200-2011. We also close the performance gap for SOP dataset from -1% to -0.6% compared to the proposed results in [1] using ResNet50.

What was easy – [1] is a well-structured and -written work, that allows us to clearly comprehend the primary concepts.

What was difficult – [1] did not investigate joint contributions between various modules like hidden sizes and embedding sizes. Additionally, the performance of HIST cannot be fully reproduced with given configurations proposed in [1]. To address this, an additional hyperparameter search has to be performed, which is time-consuming.

Communication with original authors – We attempted to contact the authors of [1] for more details regarding the experimental configurations and hyperparameters. Unfortunately, before completing this report, we did not receive any responses.

Copyright © 2023 J. Yuan and D. Le-Phuoc, released under a Creative Commons Attribution 4.0 International license.
Correspondence should be addressed to Jicheng Yuan (jicheng.yuan@tu-berlin.de)
The authors have declared that no competing interests exist.
Code is available at https://github.com/jichengyuan/MLRC2022_HIST. – SWH swh:1:rev:aa9738a884343fb4798f8b06c7a8b9288eca6b14.
Open peer review is available at https://openreview.net/forum?id=JJQbk2hIQ5.

1 Introduction

Deep Metric Learning (DML) is a crucial area of research in the field of computer vision, which tries to learn a feature embedding that maps input data into a feature space where the distance between the embeddings corresponds to the similarity between the inputs [4, 5]. This is a key step for tasks such as image retrieval [6, 7, 8], face recognition [9, 10, 11], and person re-identification [12, 13, 14]. However, traditional pairwise [15] and triplet loss [16] functions have limitations, e.g. slow convergence and lack of relations between data samples [15] when dealing with samples with similar visual appearances among the same and different classes. The proposed method demonstrated in [1] addressed these limitations by introducing the HIST loss function, designed to exploit multilateral semantic relations and leverage in-batched semantic relations for each sample and class by hypergraph modeling. Moreover, HIST loss outperforms the state-of-the-art techniques [17, 18, 19] on CUB-200-2011, CARS196, and SOP datasets under the standard evaluation settings [2, 3, 20, 7]. Our work aims to verify the performances and effectiveness of the HIST loss, as well as to confirm the main claims/results proposed in [1] in the context of image retrieval.

2 Scope of reproducibility

Our scope of this work focuses on verifying the effectiveness of the proposed HIST loss in [1], which specifically addresses the problem of multilateral semantic relations for intra- and inter-classes samples in each mini-batched and utilizes a HGNN to model class-discriminative visual semantics. The main claims in [1] are:

- **Claim 1**: The proposed HIST loss performs consistently performance regardless of mini-batch size N_b.
- **Claim 2**: Regardless of the quantity of HGNN layers L, the HIST loss shows consistently performance.
- **Claim 3**: The scaling factor α in constructing the HGNN, which controls the reflection ratio of negative samples, reveals that the positive value contributes to reliable performance on HIST loss for modeling semantic relations of samples.
- **Claim 4**: Large temperature scaling parameter τ is effective in deep metric learning, and HIST loss is not sensitive to the scaling parameter λ_s if τ >16.
- **Claim 5**: The HIST loss achieves SOTA performances on CUB-200-2011, CARS196, and SOP datasets under the standard evaluation settings [2, 3, 20, 7].

3 Methodology

Initially, we attempted to reproduce experiments and verify claims by utilizing the provided code repository[1] and configurations given in [1], which were evaluated on three datasets, namely CUB-200-2011, CARS196 and SOP, with two backbones, ResNet-50[21] and BN-Inception[22], pretrained on ImageNet-1k [23]. However, during conducting experiments, we found that only the performances on the CARS196 dataset can be fully reproduced, while the other datasets yielded inconsistent results to those proposed in [1]. In addition, the existing code repository did not contain the implementation of the multi-layer HGNN. Hence, we partially reimplemented and extended the existing code and conducted a hyperparameter search utilizing Bayesian optimization. Specifically, we performed approximately 200 runs on each backbone for CUB-200-2011 and CARS196

[1] https://github.com/ljin0429/HIST

datasets, while for the SOP, due to its vast size and limited resources, we conducted 30 experiments on each backbone. In the end, we retrained the model based on the best configurations we found and compared our results to those proposed in [1]. All experiments were conducted using 2 Nvidia Tesla V100 16GB GPUs for approximately 1,108 GPU hours.

3.1 Model descriptions

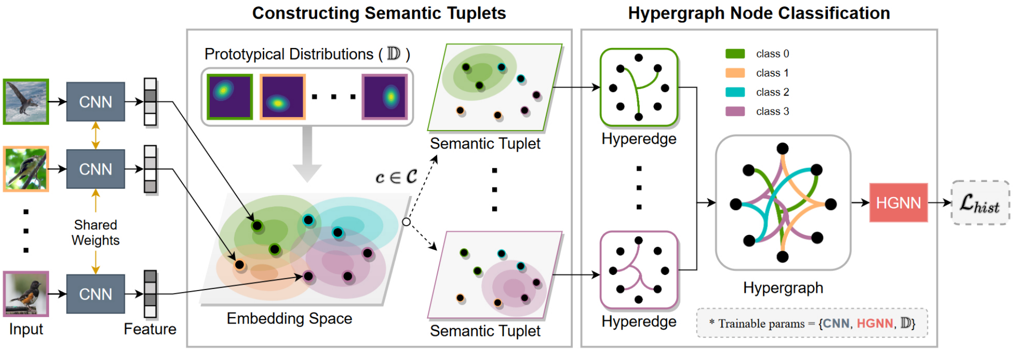


Figure 1. Overview of the pipeline for Hypergraph-Induced Semantic Tuplet (HIST) loss [1].

The proposed approach, as demonstrated in Figure 1, employs pretrained CNN as the backbone to extract features from an input image x_i and output a D-dimensional feature map $f_i \in \mathbb{R}^D$. Then, given the $\mathcal{C}$ classes in the training set, the learnable distributions are applied to model the feature distribution for each class, which are denoted by $\mathbb{D} = \{\mathcal{D}_1, ..., \mathcal{D}_\mathcal{C}\}$. Accordingly, the distribution loss is defined by the equation 1,

$$\mathcal{L}_D = \frac{1}{N_b} \sum_{i=1}^{N_b} -logP_i \tag{1}$$

where P_i is the probability for the i_{th} sample to measure its assigning quality according to its true distribution and is defined by the equation 2,

$$P_i = \frac{\exp(-\tau d_m^2(f_i, \mathcal{D}_+))}{\sum_{\mathcal{D}_\mathcal{C} \in \mathcal{D}} \exp(-\tau d_m^2(f_i, \mathcal{D}_\mathcal{C}))} \tag{2}$$

where τ is a hyperparameter [24] to scale the contribution between positive/negative samples, d_m^2 is the squared Mahalanobis distance, in which the $\mathcal{D}_+$ works to measure the distance between the modeled true categorical centroid and the positive samples, while $\mathcal{D}_\mathcal{C}$ is employed to calculate the distances among other classes. In this way, the distribution loss $\mathcal{L}_D$ can guide to assign samples in each mini-batch to their corresponding centroids and aim to model the true feature distributions and alleviate the intra-classes variations among samples from the same class, which are caused by different points of views, backgrounds or poses. Afterward, the semantic tuplets for each class c will be constructed based on the measured distances between the learned distributions and feature embedding to model multilateral semantic relations using the equation 3,

$$S_{ij} = \begin{cases} 1 & if\ y_i = \mathcal{C}_j, \\ e^{-\alpha d_m^2(\mathrm{f}_i, \mathcal{D}_{\mathcal{C}_j})} & otherwise, \end{cases} \tag{3}$$

where S_{ij} means the semantic relation between i-th sample in one mini-batch and j-th class in $\mathcal{C}$. For each semantic tuplet S_{ij} depicted in Figure 1, the weight of positive samples for class $\mathcal{C}_j$ will be assigned to 1; otherwise, the weight-relation will be regulated by the measured distance between learned distributions $\mathcal{D}_{\mathcal{C}_j}$ and extracted features $\mathbf{f}_i$ from the backbone. α is a hyperparameter affecting the reflection ratio of negative samples. Then, a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ will be constructed denoted by the equation 4,

$$\mathcal{H}_{ij} = \begin{cases} 1 & if\ v_i \in e_j, \\ 0 & otherwise, \end{cases} \tag{4}$$

where the nodes $v_i \in \mathcal{V}$ represent the embedding features and hyperedges $e_j \in \mathcal{E}$ denote the semantic tuplets, which reflect the semantic relation between positive and negative samples with incidence weights in $[0, 1]$.

In addition, according to the definition of semantic tuplets, if all the samples connected by one hyperedge belong to the same class $\mathcal{C}_j$, this hyperedge e_j is a positive hyperedge; otherwise, it will be defined as a negative hyperedge. In detail, the dimension of the output from the constructed hypergraph is set to be the same as the dimension of semantic tuples, *i.e.* $\mathrm{Z}_{out} \in \mathbb{R}^{N_b \times C}$, where N_b is the number of samples in each mini-batch and $\mathcal{C}$ is the number of classes. Afterward, output logits followed by a softmax function can then represent the final discriminative probabilities. During training, the CE loss between the ground truth and output of HGNN is defined by the equation 5.

$$\mathcal{L}_{CE} = -\frac{1}{N_b} \sum_{i=1}^{N_b} \sum_{j=1}^{C} Y_{ij} log \widehat{Y}_{ij} \tag{5}$$

In the end, the HIST loss, which is applied to optimize the whole network, will be denoted by the equation 6,

$$\mathcal{L}_{HIST} = \mathcal{L}_D + \lambda_s \mathcal{L}_{CE} \tag{6}$$

where λ_s is a hyperparameter to scale the CE loss and balance the contribution between the learned distributions and hypergraph.

3.2 Datasets

We conducted our experiments on CUB-200-2011, CARS196, and Stanford Online Products(SOP) datasets in the context of image retrieval. In detail, CARS-196 contains 16,185 car images with 196 different classes; CUB-200-2011 comprises 11,788 bird images from 200 distinct classes; SOP is the largest one of the three datasets, which consists of 120,053 product images and 22,634 classes. For each dataset, the experiments are conducted on two ImageNet pretrained backbones(ResNet-50 and BN-Inception). For the data-split, we followed the standard evaluation setting as proposed in [1], which designates half of the total number of classes for training and the remainder for evaluation. Table 1 provides a summary of the overall statistics for each dataset.

Dataset	URL	#Images		#Classes	
		Train	Test	Train	Test
CUB-200-2011 [25]	Link	5,864	5,924	100	100
CARS196 [26]	Link	8,054	8,131	98	98
SOP [3]	Link	59,551	60,502	11,318	11,316

Table 1. Dataset statistics for Train-Test Configuration.

3.3 Hyperparameters

Initially, we conducted experiments in accordance with the configurations and hyperparameters given in [1]. Unfortunately, we only obtained results consistent with those proposed in [1] on the CARS-196 dataset. For the SOP and CUB-200-2011 datasets, the performances based on the reported hyperparameters dropped significantly. More details are shown in Table 3. Hence, we performed an extended hyperparameter search utilizing Bayesian optimization for the following factors, i.e. the scaling factor of HIST loss λ_s, the scaling factor of semantic tuplets α, the mini-batch size N_b and the temperature factor of the prototypical distributions τ. More details about our parameter-search for each dataset and backbone are attached in the appendix, as Table 5 and Table 6. The proposed parameters in [1] and our best-searched parameters are shown in Table 2. Additionally, we also visualize the embedding features from CARS196 on ResNet-50, as shown in Figure 4 and Figure 5.

Backbone	Hyper-parameters	Datasets		
		CARS196	CUB-200-2011	SOP
ResNet-50[21]	τ	32(32)	32(24)	16(20)
	b_s	32(32)	32(32)	32(32)
	α	0.9(0.9)	1.1(1.15)	2.0(2.1)
	λ_s	1.0(0.8)	1.0(1.0)	1.0(0.5)
BN-Inception[22]	τ	24(22)	16(20)	16(12)
	b_s	32(32)	32(32)	32(32)
	α	0.9(0.9)	1.0(0.95)	1.6(1.55)
	λ_s	1.0(1.5)	1.0(1.2)	1.0(1.5)

Table 2. Hyperparameters proposed in [1] and our best-searched results(in bracket) when using ResNet-50/BN-Inception as the backbone. The values depicted in **red** are from our search, which are different from those in [1].

3.4 Experimental setup and code

The existing code did not include the configurations to investigate the effect of HIST loss under the varied number of HGNN layers and distance metrics. Hence, we completed the code for our experiments based on the existing repository, which is available at this repository. To provide a fair comparison, we evaluate our results by the metrics proposed in [1], namely *Recall@K*(*R@K*), and *Normalized Mutual Information*(*NMI*). Sec.4 and Sec.5 will provide more details about our reproduced results and a discussion of our results and findings.

3.5 Computational requirements

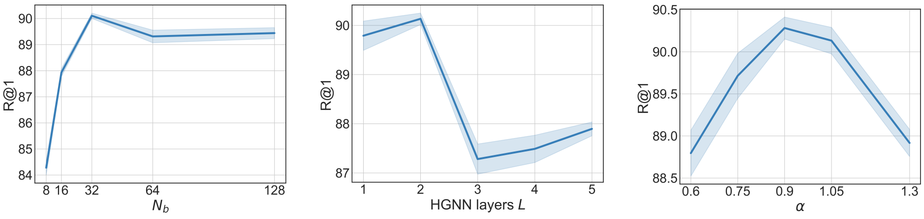
All experiments were performed on our server with two Nvidia Tesla V100-16GB GPUs. Hyperparameter search by Bayesian Optimization took approximately a total of 1,050 GPU hours. More details about running-time and configurations on each dataset and backbone are attached in Table 7, Table 9, and Table 8 of the appendix.

4 Results

In this section, we will report our results for the aforementioned experiments and verify claims 1 - 5 in Sec.2. Overall, not all the claims can be supported by our reproduced results. In addition, we conduct additional experiments to investigate the performance of HIST loss affected by other modules, e.g. the number of layers for HGNN, distinct distance metrics, and the embedding sizes, which were not included or not discussed in depth in [1].

4.1 Results reproducing original paper

Based on our reproduced results, shown in Figure 2, Figure 3(b), and Table 3, claims 3, 4, and 5 can be fully supported. Oppositely, claims 1 and 2 can not be completely approved due to the inconsistent performance of HIST loss under distinct parameter settings.



(a) Impact of N_b (b) Impact of L (c) Impact of α

Figure 2. Impact of distinct hyperparameters on CARS196 using ResNet-50 as Backbone.

Claim 1: Performance of HIST loss is consistent regardless of mini-batch size N_b – Figure 2(a) depicts the performances of HIST loss under various N_b. It indicates that when N_b is larger than 32, the performances for learning the multilateral semantic relations by semantic tuplets and HIST loss are almost consistent. Accordingly, with small bach-size, such as 8 or 16, due to the limited number of learned features per batch, discriminative features between positive/negative samples can not be learned efficiently. Therefore, claim 1 can only be supported within a limited range, i.e. $N_b \geq 32$.

Claim 2: Performance of HIST loss is consistent regardless of HGNN layers L – To verify Claim 2, we conduct experiments with various layers of HGNN, i.e. $L \in \{1, 2, 3, 4, 5\}$. Figure 2(b) indicates that when $L \geq 3$, the performances of HIST loss decrease significantly, by about -3% R@1 for CARS196 using ResNet-50 as backbone compared to $L = 2$. Hence, claim 2 can not be fully supported based on our reproduced results.

Claim 3: Positive scaling factor α contributes reliable and consistent performance – Figure 2(c) indicates that the performances of HIST loss differ slightly with diverse α. It suggests that the hist loss is insensitive to positive α, which can adjust the contributions of positive and negative samples while constructing semantic tuplets. This allows claim 3 to be supported.

Claim 4: Large temperature scaling parameter τ is effective; if τ >16, HIST loss is insensitive to the scaling parameter λ_s – To verify the effectiveness of τ and λ_s, we set distinct values, i.e. $\tau \in \{8, 16, 24, 32\}$ and $\lambda_s \in \{0.6, 1.0, 1.5, 2.0\}$. Figure 3(b) demonstrates that our reproduced results are consistent with the claim 4 proposed in [1], i.e. the performances are consistent if $\tau \geq 16$. Moreover, HIST loss is not sensitive to λ_s .

Claim 5: The SOTA performances of HIST loss under the standard evaluation settings [2, 3, 20, 7]. – Table 3 shows the original([OA]HIST) and our reproduced results([RE]HIST) under standard evaluation settings. For CARS196 and CUB-200-2011 using ResNet50 as backbone, our results after tuning([TUNE]HIST) are improved by 0.7% and 0.4% R@1 than the reported results. For the SOP dataset, the reproduced results under proposed hyperparameters and configurations are not well compared to those in [1]. The differences are -2.4% and -0.9% @R1 using the backbone BN-Inception and ResNet-50, respectively. By utilizing our settings in Table 2, the gap between the reported and our fine-tuned results

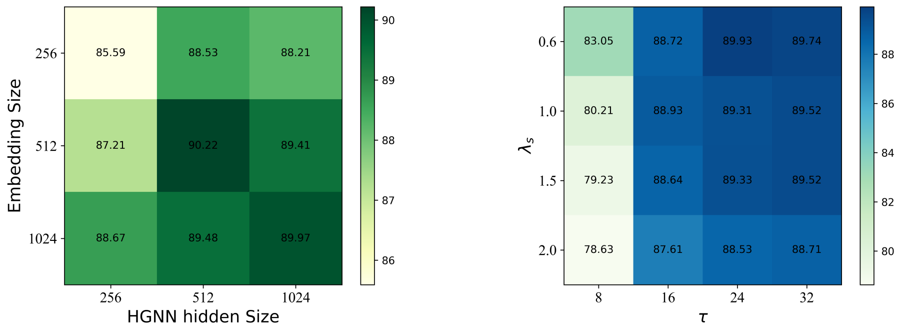
Backbone	Method	CUB-200-2011				CARS-196				SOP			
		R@1	R@2	R@4	NMI	R@1	R@2	R@4	NMI	R@1	R@10	R@100	NMI
BN-Inception	[OA]HIST	**70.0**	**80.2**	**87.5**	**71.0**	87.6	**92.8**	95.5	73.2	**79.8**	**91.2**	**96.4**	**92.5**
	[RE]HIST	68.7	78.3	86.2	70.3	87.6	**92.8**	**95.7**	73.1	77.4	89.6	95.4	89.9
	[TUNE]HIST	**70.0**	79.4	86.6	70.8	**88.1**	92.7	**95.7**	**73.4**	77.9	90.1	95.8	90.5
ResNet-50	[OA]HIST	71.6	81.4	88.3	74.3	89.8	94.0	96.5	75.5	**81.6**	**92.2**	**96.8**	**93.0**
	[RE]HIST	70.1	79.7	87.4	72.9	89.8	94.4	96.6	75.4	80.7	91.2	96.4	91.9
	[TUNE]HIST	**72.3**	**81.8**	**88.5**	**75.6**	**90.2**	**94.5**	**96.8**	**75.8**	81.0	91.6	96.3	92.2

Table 3. Reported, reproduced, and our fine-tuned Results under the standard evaluation settings as proposed in [1]. *OA, RE, TUNE* denote the results were the highest scores quoted from [1], reproduced based on the reported configurations in [1], reproduced by our best-searched settings by Bayesian optimization. The best results are marked in **bold**.

are close to 1.9% and 0.6% R@1 on BN-Inception and ResNet-50. Given that the vast majority of reproduced results are compatible with the reported ones. Hence, claim 5 can be almost supported.

4.2 Joint contribution from multi-parameters and Ablation study on distance metrics

In addition to the claimed impact of the reported hyperparameters in [1] above, other factors also play an important role in the robustness of HIST loss, e.g. the hidden-size of HGNN, the embedding size of the backbone, the various distance metrics, and their joint contribution to the HIST loss. Hence, we conduct additional experiments to investigate the robustness further.



(a) Impact of embedding size and HGNN hidden-size

(b) Impact of τ and λ_s

Figure 3. Joint contribution of hyperparameters on CARS196 using ResNet-50.

Distance Metrics	R@1	R@2	R@4
Euclidean	87.2	92.5	95.4
Cosine	88.7	93.6	96.1
Mahalanobis	**90.2**	**94.5**	**96.8**

Table 4. Robustness of HIST loss with various distance metrics on CARS196 using ResNet-50.

Joint contribution from the embedding sizes and HGNN hidden-size — Figure 3(a) indicates that HIST loss is consistent if the embedding or HGNN hidden-size is larger than 256. Moreover, both values should be compatible with each other, i.e the best performances are on the diagonal. Otherwise, if both are set to 256, the ability of the feature representation from the backbone and HGNN will be constrained.

Ablation study on distance metrices – In addition, we conduct experiments with distinct distance metrics to investigate the robustness of HIST loss further. Table 4 indicates that HIST loss performed insignificant deviations under various distance metrics for constructing semantic tuplets.

5 Discussion

Overall, by comparing our results shown in Sec.4 to those reported in [1], we can conclude that 3 out of 5 claims presented in Sec.2 can be supported. However, the other two claims can not be fully supported. For claim 1, the performance of HIST loss dropped significantly when $N_b < 16$. This indicates that the small N_b cannot contribute to effectively constructing positive/negative sample pairs. Hence, it will impact the quality of constructed semantic tuplets and reduce the effectiveness of HGNN; for claim 2, the impact of the number of HGNN layers L is not isolated. Accompany with other hyperparameters, e.g. the number of HGNN hidden-size. Their interactions will contribute a cumulative effect on the performance of HIST. Due to the constrained resources, we cannot conduct additional experiments to investigate deeper in this direction. Regarding claims 3 and 4, our experiments are consistent with the results from [1]. Therefore, for constructing semantic tuplets, the positive scaling parameter α can yield consistent results for HIST loss. Finally, for claim 5, based on the standard settings proposed in [1], we cannot reproduce comparable performances as those from [1], which may be due to the versions of CUDA, cuDNN, etc. In addition, by using Bayesian optimization to search for hyperparameters, we achieved improved performance on CARS196 and CUB-200-2011 datasets; for the SOP dataset, with our configuration in Table 2, the margin between our results after tuning and those from [1] has been decreased, but still are -0.6% and -1.9% R@1 on ResNet-50 and BN-Inception. Due to limited resources, we cannot perform larger-scale experiments to verify and improve the performance of the SOP dataset.

5.1 What was easy

The paper[1] is well-structured and contains explicit theoretical derivations, which provided us with a clear understanding of the concepts for semantic tuplets, HGNN, and HIST loss underlying the proposed method. Benefiting from these, the reimplementation of the multi-layered HGNN, distinct distance metrics, and the reproducing of experiments are highly efficient.

5.2 What was difficult

We tried many distinct experimental configurations to reproduce comparable results as those proposed in [1]. The performance of HIST loss might vary greatly according to the interaction of multiple parameters, such as the HGNN hidden size and HGNN layers. Due to the restricted computational capability, we cannot thoroughly investigate that. In addition, intuitively, a larger embedding size will bring more diverse learned feature representations; nevertheless, due to the HGNN as a downstream module, their effects should be carefully balanced for the final learned features. Moreover, HGNN can guide the backbone to find discriminative features between positive/negative samples; accordingly, the backbone should contribute well-learned features that contribute to the high quality of semantic tuplets, which HGNN can then utilize to learn.

5.3 Conclusion

In this study, we reproduce the effectiveness of HIST loss under various hyperparameters, e.g. the scaling parameter α. the number of HGNN layers L, the temperature parameter τ and distinct modules, e.g. HGNN, distance metrics, and semantic tuplets. From our reproduced results in Sec.4, we can conclude that the performances of HIST loss cannot be fully reproduced by the reported experimental settings in [1], but this point will not influence the effectiveness of HIST loss. Through hyperparameter search, we reproduce comparable and better performances than SOAT on CUB-200-2011 and CARS196 datasets. In addition, for the scalability of HIST loss to other large datasets, the constraints of the high demand of computing power from HGNN and similarity estimation need to be investigated and solved. Moreover, by introducing the HIST loss, multilateral semantic relations and the centroids across classes can be well constructed utilizing semantic tuplets and prototypical distributions. Finally, some factors, which are not investigated in depth in [1], e.g. combined contributions from HGNN hidden-size, the embedding size of backbone, and various distance metrics, will affect the performance of the HIST loss as well. Hence, these factors need to be further investigated and adjusted for other downstream tasks.

References

1. J. Lim, S. Yun, S. Park, and J. Y. Choi. “Hypergraph-induced semantic tuplet loss for deep metric learning.” In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. 2022, pp. 212–222.
2. S. Kim, D. Kim, M. Cho, and S. Kwak. “Proxy anchor loss for deep metric learning.” In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. 2020, pp. 3238–3247.
3. H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. “Deep metric learning via lifted structured feature embedding.” In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2016, pp. 4004–4012.
4. E. Xing, M. Jordan, S. J. Russell, and A. Ng. “Distance metric learning with application to clustering with side-information.” In: **Advances in neural information processing systems** 15 (2002).
5. K. Q. Weinberger and L. K. Saul. “Distance metric learning for large margin nearest neighbor classification.” In: **Journal of machine learning research** 10.2 (2009).
6. Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao. “Hypergraph Neural Networks.” In: **AAAI 2019** (2018).
7. X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott. “Multi-similarity loss with general pair weighting for deep metric learning.” In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. 2019, pp. 5022–5030.
8. Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. “No fuss distance metric learning using proxies.” In: **Proceedings of the IEEE international conference on computer vision**. 2017, pp. 360–368.
9. F. Schroff, D. Kalenichenko, and J. Philbin. “Facenet: A unified embedding for face recognition and clustering.” In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2015, pp. 815–823.
10. Y. Sun, Y. Chen, X. Wang, and X. Tang. “Deep learning face representation by joint identification-verification.” In: **Advances in neural information processing systems** 27 (2014).
11. W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. “Sphereface: Deep hypersphere embedding for face recognition.” In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2017, pp. 212–220.
12. W. Chen, X. Chen, J. Zhang, and K. Huang. “Beyond triplet loss: a deep quadruplet network for person re-identification.” In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2017, pp. 403–412.
13. T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang. “Joint detection and identification feature learning for person search.” In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2017, pp. 3415–3424.
14. I. Filković, Z. Kalafatić, and T. Hrkać. “Deep metric learning for person Re-identification and De-identification.” In: **2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)**. IEEE. 2016, pp. 1360–1364.
15. K. Sohn. “Improved deep metric learning with multi-class n-pair loss objective.” In: **Advances in neural information processing systems** 29 (2016).
16. E. Hoffer and N. Ailon. “Deep metric learning using triplet network.” In: **Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3**. Springer. 2015, pp. 84–92.

17. Y. Zhu, M. Yang, C. Deng, and W. Liu. "Fewer is more: A deep graph metric learning perspective using fewer proxies." In: **Advances in Neural Information Processing Systems** 33 (2020), pp. 17792–17803.
18. F. Xu, M. Wang, W. Zhang, Y. Cheng, and W. Chu. "Discrimination-aware mechanism for fine-grained representation learning." In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. 2021, pp. 813–822.
19. W. Zheng, B. Zhang, J. Lu, and J. Zhou. "Deep relational metric learning." In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. 2021, pp. 12065–12074.
20. Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin. "Softtriple loss: Deep metric learning without triplet sampling." In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. 2019, pp. 6450–6458.
21. K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2016, pp. 770–778.
22. S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In: **International conference on machine learning**. pmlr. 2015, pp. 448–456.
23. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database." In: **2009 IEEE conference on computer vision and pattern recognition**. Ieee. 2009, pp. 248–255.
24. G. Hinton, O. Vinyals, and J. Dean. "Distilling the knowledge in a neural network." In: **arXiv preprint arXiv:1503.02531** (2015).
25. C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. "The caltech-ucsd birds-200-2011 dataset." In: (2011).
26. J. Krause, M. Stark, J. Deng, and L. Fei-Fei. "3d object representations for fine-grained categorization." In: **Proceedings of the IEEE international conference on computer vision workshops**. 2013, pp. 554–561.

6 Appendix

Datasets	Hyper-parameters			
	λ_s	α	τ	N_b
CARS196	{0.5,0.8,1.0,1.2}	{0.9,0.95,1.0,1.1,1.2}	{16,20,24,28,32}	{8,16,32}
CUB-200-2011	{0.5,0.8,1.0,1.2,1.5}	{0.9,0.95,1.0,1.05,1.15}	{16,20,24,28,32,36}	{8,16,32}
SOP	{0.5,0.8,1.0,1.2,1.5}	{1.9,1.95,2.0,2.1,2.15}	{12,14,16,20,22,24}	{8,16,32}

Table 5. Hyper-parameters tuning on ResNet-50

Datasets	Hyper-parameters			
	λ_s	α	τ	N_b
CARS196	{0.5,0.8,1.0,1.2,1.5}	{0.9,0.95,1.0,1.1,1.15}	{16,20,22,28,32}	{8,16,32}
CUB-200-2011	{0.5,0.8,1.0,1.2,1.5}	{0.9,0.95,1.0,1.05,1.15}	{16,20,24,28,32,36}	{8,16,32}
SOP	{0.5,0.8,1.0,1.2,1.5}	{1.5,1.55,1.6,1.65,1.7,1.75}	{12,14,16,20,22,24}	{8,16,32}

Table 6. Hyper-parameters tuning on BN-Inception

Dataset	Running-time	Epochs	Batch-size	Embedding-size
CARS196	~0.9 Ghour	60	32	32
	~0.6 Ghour	50		
CUB-200-2011	~1.1 Ghours	30		
	~1.7 Ghours	40		
SOP	~9.3 Ghours	60		
	~7.2 Ghours	60		

Table 7. Running-time for each dataset based on the HIST configuration(Ghour is short for GPU-hour). Blue values represent the runnings on BN-Inception; Red values stand for the runnings on ResNet-50.

Module	Name	CUB-200-2011	CARS196	SOP
HGNN	#layers	2		
	Hidden size	512		
	lr-HGNN	5e-4	5e-4	1e-3
$\mathbb{D}$	Initialization	He-normal		
	lr-D	5e-2	1e-1	1e-2
Hyper-parameters	λ_s	1.2	1.5	1.5
	τ	20	22	12
	α	0.95	1.0	1.55
Training	Batch size	32		
	Learning rate	1e-4		
	Optimizer	AdamW		
	Warm-up	True		
	Epochs	30	60	60
	Weight decay	1e-4	5e-5	1e-4
	lr scheduler	Step(5/0.5)	Step(10/0.5)	Step(10/0.5)
	BN freeze	True	True	False

Table 8. Configurations for training and evaluation on the backbone BN-Inception.

Module	Name	CUB-200-2011	CARS196	SOP
HGNN	#layers	2		
	Hidden size	512		
	lr-HGNN	6e-4	1e-3	1e-3
$\mathbb{D}$	Initialization	He-normal		
	lr-D	1e-1	1e-1	1e-2
Hyper-parameters	λ_s	1	0.8	0.5
	τ	24	32	20
	α	1.15	0.9	2.1
Training	Batch size	32		
	Warm-up	True		
	Optimizer	Adam		
	Learning-rate	1.2e-4	1e-4	1e-4
	Epochs	40	50	60
	Weight decay	5e-5	1e-4	1e-4
	lr scheduler	Step(5/0.5)	Step(10/0.5)	Step(10/0.5)
	BN freeze	True	True	False

Table 9. Configurations for training and evaluation on the backbone ResNet-50.

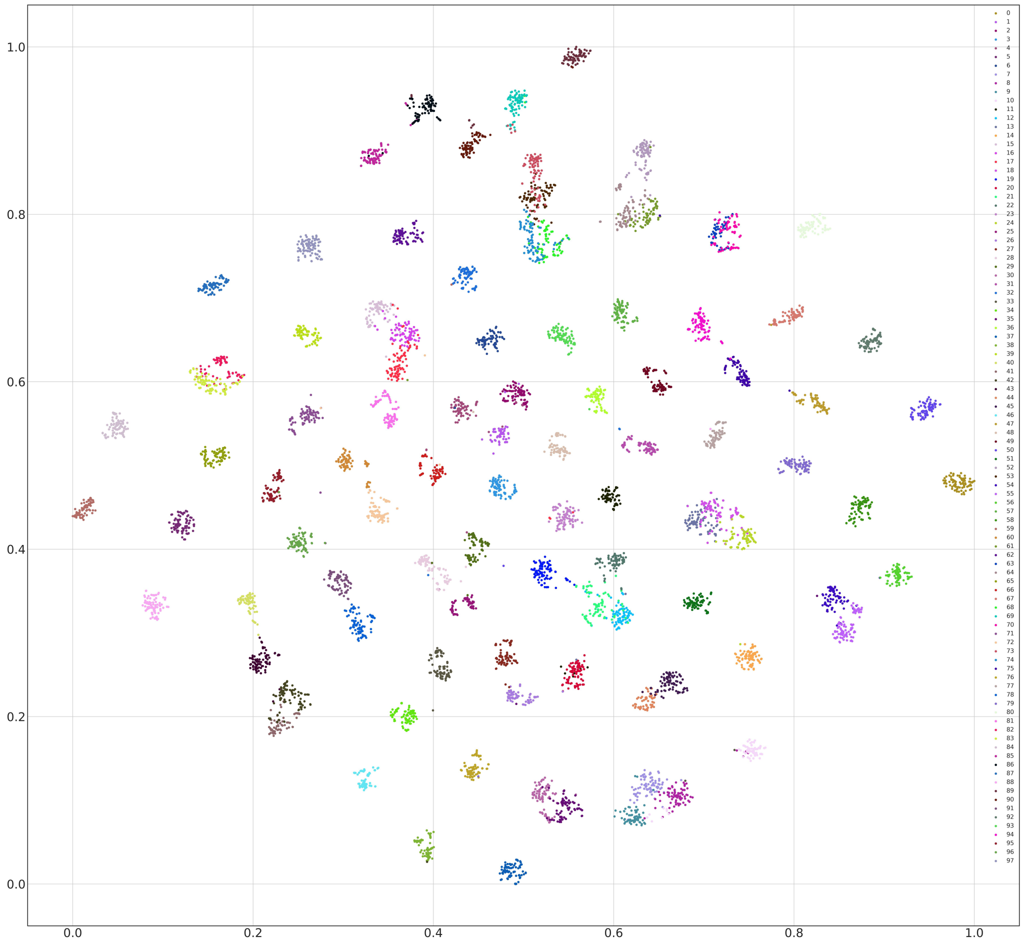


Figure 4. Embedding visualization on CARS196 training-set using ResNet50 as backbone.

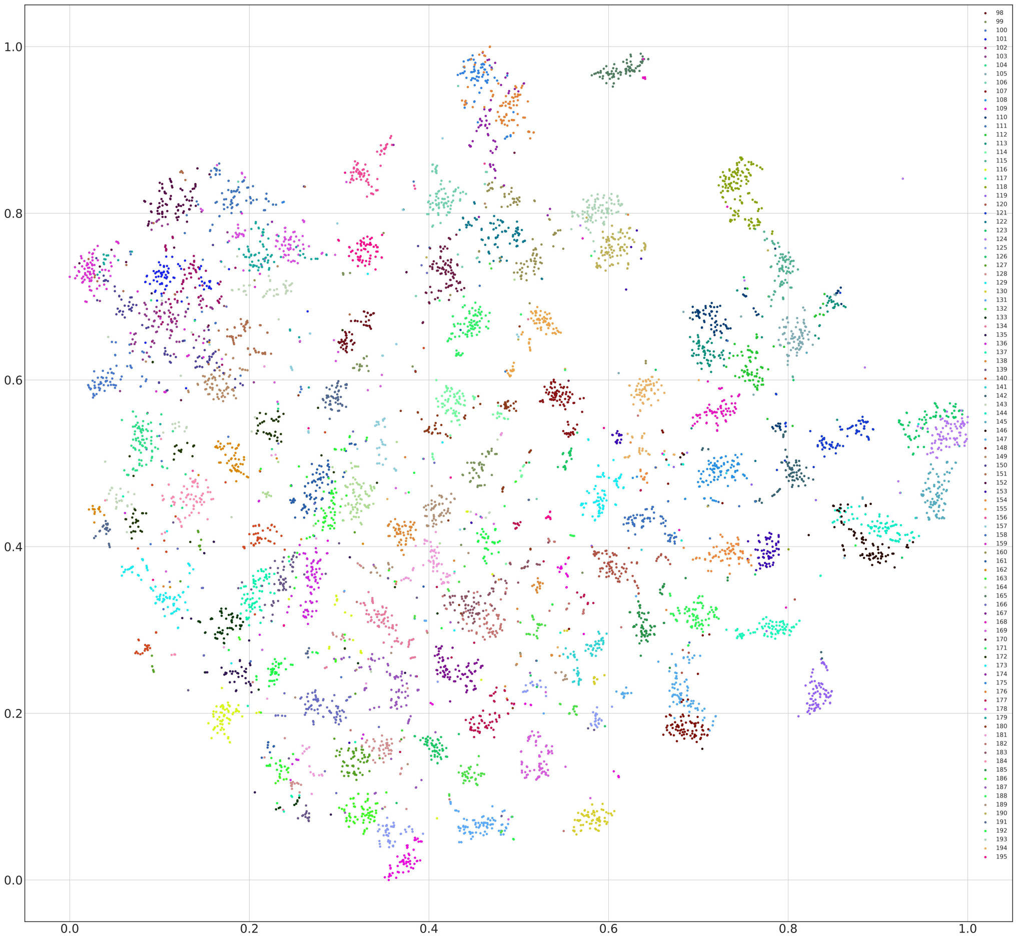


Figure 5. Embedding visualization on CARS196 test-set using ResNet50 as backbone.