A. Experimental Details

A.1. Datasets

In Table A.1, we provide a statistical summarization of the eight generic and fine-grained datasets. Among these benchmarks, the generic datasets including CIFAR-10, CIFAR-100, ImageNet-100 and ImageNet-1k consist of categories of open-world, whereas the fine-grained benchmarks including CUB, Stanford Cars, Oxford Pets, and Flowers102 are largely domain-specific. Following the GCD protocol [22], we select the first half of classes as the known categories for each benchmark dataset, except for CIFAR-100, in which we select the first 80 classes as the known categories.

Table A.1. Statistics of the benchmark datasets.

Dataset	Labelled		Unlabelled	
	#Image	#Class	#Image	#Class
CIFAR10	12.5K	5	37.5K	10
CIFAR100	20.0K	80	30.0K	100
ImageNet-100	31.9K	50	95.3K	100
ImageNet-1K	321K	500	960K	1000
CUB	1.5K	100	4.5K	200
Stanford Cars	2.0K	98	6.1K	196
Oxford Pets	1.9K	19	5.5K	37
Flowers102	0.3K	51	0.8K	102

A.2. Experiments Settings

Network Architecture. In Figure A.1, we present a detailed overview of our proposed Retrieval-based Text Aggregation (RTA). In Table A.2, we present the learnable parameters of network architecture. The visual and textual branches share the same architecture. Specifically, when using CLIP-B/16 as the backbone, we fine-tune the last residual attention block (which includes the multi-head self-attention mechanism, feed-forward network, and layer normalization), along with the image and text projectors of CLIP. Additionally, the dual-branch classifiers are learned with the fine-tuning of CLIP jointly.

Data Preparation. For each mini-batch, we generate a set of text embeddings for query images by integrating four similar tags and four similar attributes through the proposed RTA strategy by using the encoders of CLIP-H/14 for all test datasets. Then, both images and text are augmented into two views, and the augmentation strategies are the same across datasets, as detailed in Tables A.3 and A.4. The embeddings of augmented images and text are used for representation learning.

Training Settings. We use Stochastic Gradient Descent (SGD) with the momentum of 0.9, the weight decay of 1×10^{-4}, and using cosine annealing learning rate decay for the training process. We train the model for 200 epochs and set the batch size to 128. The random seeds for the

Table A.2. Model architectures. "*CLIP*" denotes the learnable parameters in CLIP and "*Cls*" denotes the classifiers.

CLIP	Last residual attention block: $\mathbb{R}^{512} \rightarrow \mathbb{R}^{512}$ Image/text projector: $\mathbb{R}^{512} \rightarrow \mathbb{R}^{512}$
Cls.	Linear projection: $\mathbb{R}^{512} \rightarrow \mathbb{R}^{K}$ Softmax function

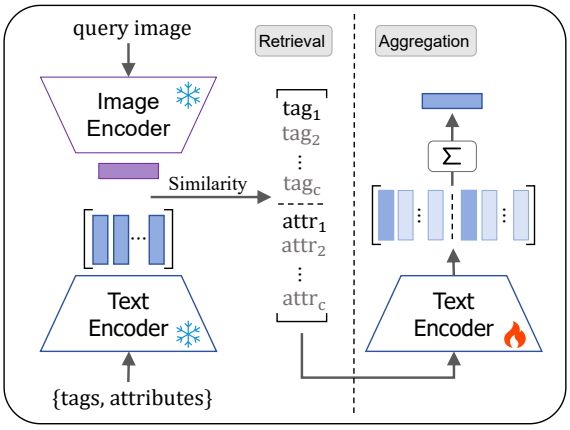


Figure A.1. An overview of Retrieval-based Text Aggregation (RTA).

Table A.3. Python code for the image augmentation.

```
from torchvision.transforms import *

Compose([
  RandomResizedCrop(32, BILINEAR)
  RandomHorizontalFlip(p=0.5),
  ColorJitter()
  ToTensor(),
  Normalize([0.485, 0.456, 0.406],
            [0.229, 0.224, 0.225])
  )])
```

Table A.4. Pseudo-code for the text augmentation.

```
Input: text

For each word in text:
  If len(word) ≥ 3:
    index ← random(1, len(word)−2)
    action ← random({replace, delete, add, none})
    Case action:
      replace: word ← replace random char at index
      delete: word ← remove char at index
      add: word ← insert random char at index
      none: continue

Output: augmented text
```

Table A.5. The mean ± std ACC (%) of TextGCD, GET and our approach on generic datasets.

	CIFAR-10			CIFAR-100			ImageNet-100			ImageNet-1k		
Method	All	Old	New	All	Old	New	All	Old	New	All	Old	New
TextGCD	98.2±0.0	98.0±0.2	**98.6**±0.1	85.7±0.9	**86.3**±0.6	84.6±2.2	88.0±0.6	92.4±0.9	85.2±1.2	64.8±0.2	**77.8**±0.5	58.3±0.4
GET	97.2±0.1	94.6±0.1	98.5±0.1	82.1±0.4	85.5±0.5	75.5±0.5	91.7±0.3	95.7±0.0	89.7±0.4	62.4±0.0	74.0±0.2	56.6±0.1
Ours	**98.5**±0.2	**98.3**±0.2	**98.6**±0.3	**86.4**±0.5	86.2±0.6	**86.9**±1.6	**92.1**±0.3	**96.0**±0.6	**90.2**±0.8	**66.7**±0.1	77.3±0.3	**61.1**±0.2

Table A.6. The mean ± std ACC (%) of TextGCD, GET and our approach on fine-grained datasets.

	CUB			Stanford Cars			Oxford Pets			Flowers102		
Method	All	Old	New	All	Old	New	All	Old	New	All	Old	New
TextGCD	76.6±0.6	**80.6**±2.0	74.7±1.7	86.1±0.9	91.8±0.4	83.9±1.3	93.7±0.6	93.2±1.1	94.0±0.9	87.2±2.3	90.7±1.3	85.4±3.8
GET	77.0±0.5	78.1±1.6	76.4±1.2	78.5±1.3	86.8±1.5	74.5±2.2	91.1±1.0	89.7±1.6	92.4±1.2	85.5±0.5	90.8±1.5	81.3±1.7
Ours	**78.3**±0.8	78.5±1.2	**78.2**±0.9	**89.2**±0.3	**93.1**±0.9	**87.3**±0.2	**95.7**±0.2	**95.1**±0.5	**96.0**±0.4	**93.5**±0.4	**93.3**±0.8	**93.9**±1.0

three trials are set to $\{0, 1, 2\}$. For representation learning, the initial learning rate for CLIP is set to 0.001, and our proposed objective $\mathcal{L}_{\text{SSR}^2}$ does not need an extra balancing hyper-parameter, except for ϵ which is set as 0.5. The initial learning rate for the two classifiers is set to 0.1, the epochs for warm-up stage is set to 10, and 60% of pseudo-labels of each categories are selected for co-teaching. We use the same setting for all test datasets. All experiments are conducted on a single NVIDIA GeForce RTX3090 GPU.

B. More Experiments

B.1. Evaluation on Model Stability

In Tables A.5 and A.6, we report the mean accuracy with standard deviation (±std) over 3 trials across the test datasets, and compare to the multi-modal counterparts TextGCD [32] and GET [24]. As can be seen, the performance of our proposed SSR^2-GCD is relatively stable.

B.2. Effect of Retrieval-based Text Aggregation

Table B.7. The number of identical tags between the candidate pool and the unknown categories in four benchmark datasets.

Dataset	# Unknown categories	# Removed tags
CUB-200	100	90
Stanford Cars	98	83
Flowers102	51	43
Oxford Pets	19	17

In Table B.7, we report the results of our SSR^2-GCD compared to TextGCD [32] when removing the prompts of the *unknown* categories for each dataset from the candidate pool, since that the prompts in candidate pool may semantically identical to the unknown categories. As can be seen in Table B.8, though the performance of both methods is slightly dropped, our SSR^2-GCD is still leading.

Table B.8. Effect of retrieval-based methods with or without (w/o) prompts from unknown categories on four test datasets. ACC (%) on "All" categories is reported.

Datasets	CUB	Cars	Pets	Flowers
TexGCD [32]	76.6	86.1	93.7	87.2
TextGCD (w/o)	75.7	85.5	91.5	84.0
SSR^2-GCD (w/o)	**77.6**	**88.1**	**94.0**	**91.6**

B.3. Evaluation on Modality Alignments for Representation Learning

Evaluation on More Inter-Modal Alignment Methods. To further evaluate the necessity of inter-modal alignment, we report the performance of using different inter-modal alignment loss for training our SSR^2-GCD. Specifically, we introduce a Cross-modal Instance Consistency Objective (CICO) which is proposed in GET [24] as an inter-modal alignment constraint. CICO is defined as follows:

$$\mathcal{L}_{\text{CICO}} = \frac{1}{2|B|} \sum_{i \in B} \left(D_{\text{KL}}(\boldsymbol{s}_i^{\text{T}} \| \boldsymbol{s}_i^{\text{I}}) + D_{\text{KL}}(\boldsymbol{s}_i^{\text{I}} \| \boldsymbol{s}_i^{\text{T}}) \right), \quad (13)$$

where D_{KL} is the Kullback-Leibler divergence, B denotes the mini-batch data, $\boldsymbol{s}_i^{\text{I}} = \text{softmax}(\boldsymbol{z}_i^{\text{I}\top} \mathcal{A}^{\text{I}})$ and $\boldsymbol{s}_i^{\text{T}} = \text{softmax}(\boldsymbol{z}_i^{\text{T}\top} \mathcal{A}^{\text{T}})$ measure the similarity between the i-th image/text embeddings and prototypes, and $\mathcal{A}^{\text{I}}$ and $\mathcal{A}^{\text{T}}$ are the prototypes determined by the labeled anchors for each modality.

In Table B.9, we report the performance of using $\mathcal{L}_{\text{CICO}}$ and its combination with the intra-modal alignment losses for representation learning, in which the results are marked in gray. As can be seen that, both inter-modal alignment

losses $\mathcal{L}_{\text{CLIP}}$ and $\mathcal{L}_{\text{CICO}}$ fail to bring performance improvements when combining with the intra-modal alignment loss. This further confirms that performing extra inter-modal alignment is not necessary.

Table B.9. Evaluation of different representation learning methods. Average ACC (%) on "All" categories is reported. "N/A" denotes using a frozen CLIP.

Rep. Losses	Inter	Intra	C-10	C-100	CUB	Cars	Pets	Flowers
N/A	×	×	97.9	84.1	74.5	86.0	91.9	87.4
$\mathcal{L}_{\text{CLIP}}$	✓	×	98.3	86.0	76.6	86.7	93.9	89.7
$\mathcal{L}_{\text{CICO}}$	✓	×	98.0	85.0	76.4	86.1	94.9	87.2
$\mathcal{L}_{\text{con}}$	×	✓	98.4	**86.7**	77.5	87.9	94.9	91.8
$\mathcal{L}_{\text{SSR}^2}$	×	✓	**98.5**	86.4	**78.3**	**89.2**	**95.7**	**93.5**
$\mathcal{L}_{\text{CLIP}}+\mathcal{L}_{\text{con}}$	✓	✓	98.2	86.3	78.0	86.7	95.0	90.9
$\mathcal{L}_{\text{CICO}}+\mathcal{L}_{\text{con}}$	✓	✓	98.4	85.9	76.8	87.0	94.4	88.6
$\mathcal{L}_{\text{CLIP}}+\mathcal{L}_{\text{SSR}^2}$	✓	✓	98.3	86.1	77.2	88.1	95.0	92.9
$\mathcal{L}_{\text{CICO}}+\mathcal{L}_{\text{SSR}^2}$	✓	✓	98.3	86.1	76.7	87.5	95.5	92.1

To further verify that extra inter-modal alignment may be not necessary, and jointly optimizing inter-modal alignment and intra-modal alignment significantly impairs feature learning, we validate the role of inter-modal alignment under different text prompt qualities (i.e., which associate with different noise levels). Specifically, in the RTA strategy, we employ three frozen CLIP models [18]—namely CLIP-H/14, CLIP-L/14, and CLIP-B/16—to perform prompt searching, in order to evaluate how inter-modal alignment affects model performance under varying prompt qualities. The results in Table B.10 show that, on Stanford Cars and Flowers102, introducing an extra $\mathcal{L}_{\text{CLIP}}$ loss for joint training with $\mathcal{L}_{\text{SSR}^2}$ consistently degrades clustering performance at all levels of prompt quality.

Evaluation on Uni-Modal Representation Learning. To evaluate its effectiveness of the proposed $\mathcal{L}_{\text{SSR}^2}$ as an intra-modal alignment loss, we apply $\mathcal{L}_{\text{SSR}^2}$ to uni-modal GCD counterparts and report their clustering performance. Specifically, for GCD and SimGCD frameworks, we keep their pre-trained models and clustering algorithms unchanged, but replace the supervised and unsupervised contrastive loss $\mathcal{L}_{\text{con}}$ with our proposed $\mathcal{L}_{\text{SSR}^2}$. As can be read from Table B.11 that, using our $\mathcal{L}_{\text{SSR}^2}$ for representation learning achieves improvements on most cases.

Evaluation on Effect of Inter-Modal Alignment against Intra-Modal Alignment. To evaluate the effect of performing inter-modal alignment against intra-modal alignment, we combine our proposed intra-modal alignment loss $\mathcal{L}_{\text{SSR}^2}$ with the inter-modal alignment loss $\mathcal{L}_{\text{CLIP}}$ by introducing a tradeoff parameter $\nu > 0$, i.e.,

$$\mathcal{L}_{\text{SSR}^2} + \nu \cdot \mathcal{L}_{\text{CLIP}},$$

where using a larger ν means to emphasizing more on the inter-model alignment. In Figure B.2, we report the accuracy as a function with respect to a varying ν on Flowers102. Existing multi-modal GCD frameworks, such as GET [24], assume that enforcing an inter-modal alignment does not affect that of intra-modal alignment, and simply treat these two alignments in representation learning as independent and equally important. However, our experiments reveal here that the more emphasizing upon inter-modal alignment (i.e., via $\mathcal{L}_{\text{CLIP}}$ with a larger ν), rather than upon the intra-modality alignment (i.e., via $\mathcal{L}_{\text{SSR}^2}$), the lower the clustering accuracy is.

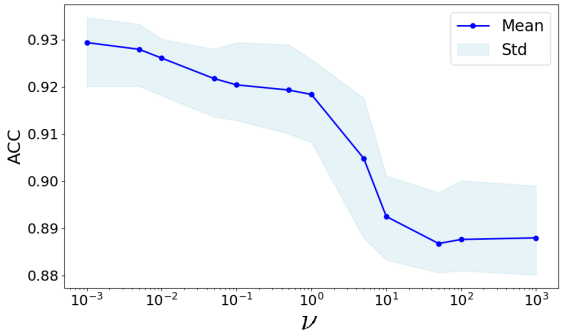
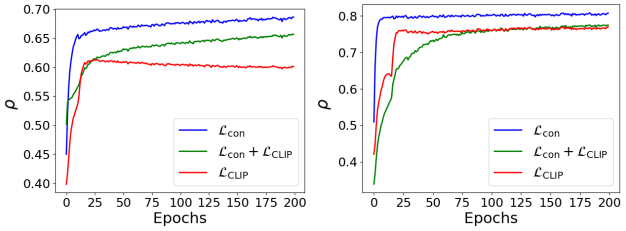


Figure B.2. Mean accuracy with standard deviation over 3 trials of "All" categories with varying penalty weight ν on Flowers102.

More Results on Consistency Measure ρ. Recall that we define a consistency measure ρ in Eq.(12) to quantify the intra-modal consistency of the embeddings to explain why extra inter-modal alignment could be unnecessary. Here, we additionally report the consistency measure ρ when training via inter-modal alignment loss (i.e., $\mathcal{L}_{\text{CLIP}}$), intra-modal alignment losses (i.e., $\mathcal{L}_{\text{con}}$) and their combinations (i.e., $\mathcal{L}_{\text{con}} + \mathcal{L}_{\text{CLIP}}$) on Stanford Cars. We display the experimental results in Figure B.3. As can As can be seen that, the inter-modal alignment also damages the learning of other intra-modal losses such as the widely adopted supervised and unsupervised contrastive loss $\mathcal{L}_{\text{con}}$.



(a) Image embeddings (b) Text embeddings

Figure B.3. Consistency measure ρ as a function of training epoch under different losses on Stanford Cars.

Table B.10. Accuracy comparison of models using different feature learning methods under various text prompt qualities. Δ denotes the accuracy difference before and after adding the inter-modal alignment loss $\mathcal{L}_{\text{CLIP}}$, and "$\downarrow$" indicates accuracy drop.

Text retrieval models	Prompt qualities	Stanford Cars			Flowers102		
		$\mathcal{L}_{\text{SSR}^2}$	$\mathcal{L}_{\text{SSR}^2} + \mathcal{L}_{\text{CLIP}}$	Δ	$\mathcal{L}_{\text{SSR}^2}$	$\mathcal{L}_{\text{SSR}^2} + \mathcal{L}_{\text{CLIP}}$	Δ
CLIP-H/14	High	89.2	88.1	1.1 $\downarrow$	93.5	92.9	0.6 $\downarrow$
CLIP-L/14	Medium	87.5	86.1	1.4 $\downarrow$	92.4	91.8	0.6 $\downarrow$
CLIP-B/16	Low	85.2	84.0	1.2 $\downarrow$	90.1	89.8	0.3 $\downarrow$

Table B.11. Comparison to uni-modal counterparts. ACC (%) on generic and fine-grained datasets is reported.

	CIFAR-10			CIFAR-100			CUB			Stanford Cars			Oxford Pets			Flowers102		
Method	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
GCD	91.5	**97.9**	88.2	73.0	76.2	66.5	51.3	56.6	48.7	39.0	57.6	29.9	80.2	85.1	77.6	74.4	74.9	74.1
GCD+SSR^2	92.5	96.4	91.6	73.9	79.0	63.2	51.9	55.0	47.1	47.9	56.1	47.3	83.6	87.7	79.8	80.0	83.3	78.5
SimGCD	97.1	95.1	**98.1**	80.1	81.2	77.8	60.3	**65.6**	57.7	53.8	**71.9**	45.0	87.7	85.9	88.6	71.3	80.9	66.5
SimGCD+SSR^2	**97.6**	97.5	97.7	**81.1**	**82.5**	**78.9**	**60.8**	64.7	**59.0**	**57.1**	66.8	**53.9**	**90.0**	**89.8**	**91.2**	**81.6**	**83.5**	**80.1**

B.4. Evaluation on Parameter ϵ in SSR^2

To evaluate the impact of using different hyper-parameters ϵ in our SSR^2-GCD, we keep all other hyper-parameters fixed, and conduct experiments under different parameter ϵ on six benchmark datasets, i.e., CIFAR-10, CIFAR-100, CUB, Stanford Cars, Oxford Pets and Flowers102. Experimental results are reported in Figure B.4. As can be seen, our SSR^2-GCD framework is not sensitive to ϵ and achieves the best performance when ϵ is in the range of $[0.2, 0.5]$.

B.5. Visualization

In Figure B.5, we use t-SNE to visualize the image embeddings and text embeddings of our SSR^2-GCD framework using different representation learning methods on Oxford Pet, which comprises 12 categories of cats and 25 categories of dogs. As can be seen, the image embeddings produced by using the frozen CLIP image encoder exhibit a distribution that can only be roughly partitioned into two classes (i.e., "cat" and "dog"). In contrast, thanks to our proposed RTA strategy, the distribution of text embeddings produced by using the frozen CLIP text encoder assisted with our RTA exhibits significant discriminability. Meanwhile, training via $\mathcal{L}_{\text{CLIP}}$ fails to learn well-aligned intra-modal relationships (See, e.g., Figure B.5a and Figure B.5b for comparison). In contrast, our SSR^2-GCD learns discriminative and well-balanced representations for both modalities.

B.6. Learning Curves

We compute the clustering accuracy (ACC) as a function of epoch during training period, and display the ACC curves on different datasets in Figure B.6. We can observe that our SSR^2-GCD converges and achieves stable clustering performance on "All" categories roughly within 50 epochs.

Table B.12. Running time and memory costs on Flowers102. "†" denotes to use the method in TextGCD [32] to produce text features.

Methods	Running Time (sec/iter)			Memory (MB)	
	Forward	Backward	Overall	w/o RTA†	w/ RTA
TextGCD [32]	0.34	6.7×10^{-3}	0.69	7,622	-
SSR^2-GCD	0.57	7.5×10^{-2}	0.92	8,035	11,080

B.7. Computation Time and Memory Costs

We report the running time and memory costs on Flowers102 of our SSR^2-GCD in Table B.12. Note that the coding rate term $\log\det(\cdot)$ is cheap to handle because its scale is kept to $\min\{|B|, d\}$ due to the fact that $\log\det\left(\mathbf{I} + \mathbf{Z}\mathbf{Z}^\top\right) = \log\det\left(\mathbf{I} + \mathbf{Z}^\top\mathbf{Z}\right)$. Comparing to TextGCD [32], the increased costs of our SSR^2-GCD is due to the RTA step in forward pass.

B.8. Performance Evaluation on Using Varying K

To evaluate the sensitivity of our SSR^2-GCD to the category number K, we change the output dimensions of the classifiers d_{cls} to deviate from the total number of categories K, and report the clustering accuracy (ACC), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) of our SSR^2-GCD with varying d_{cls} on Flowers102 in Figure B.7. As can be observed that, our SSR^2-GCD achieves relatively high clustering performance when $d_{\text{cls}} \approx K$ (e.g., $d_{\text{cls}} \in \{100, 102, 104\}$). Specifically, our SSR^2-GCD is sensitive to under-specification, where all metrics degrade sharply when $d_{\text{cls}} < K$; whereas our SSR^2-GCD exhibits good robustness to over-specification, where all metrics degrades slightly. We notice of that NMI remains nearly unchanged as d_{cls} increases beyond K.

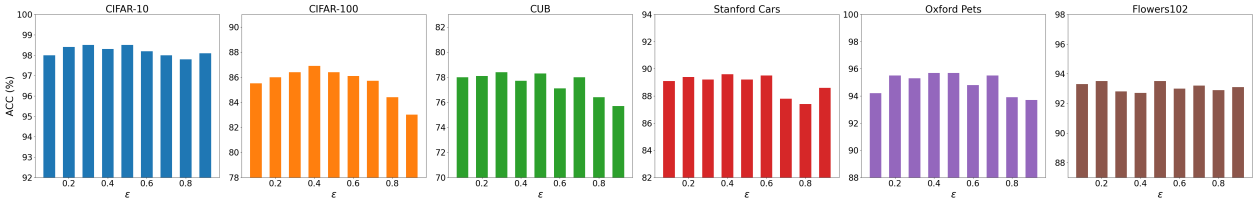
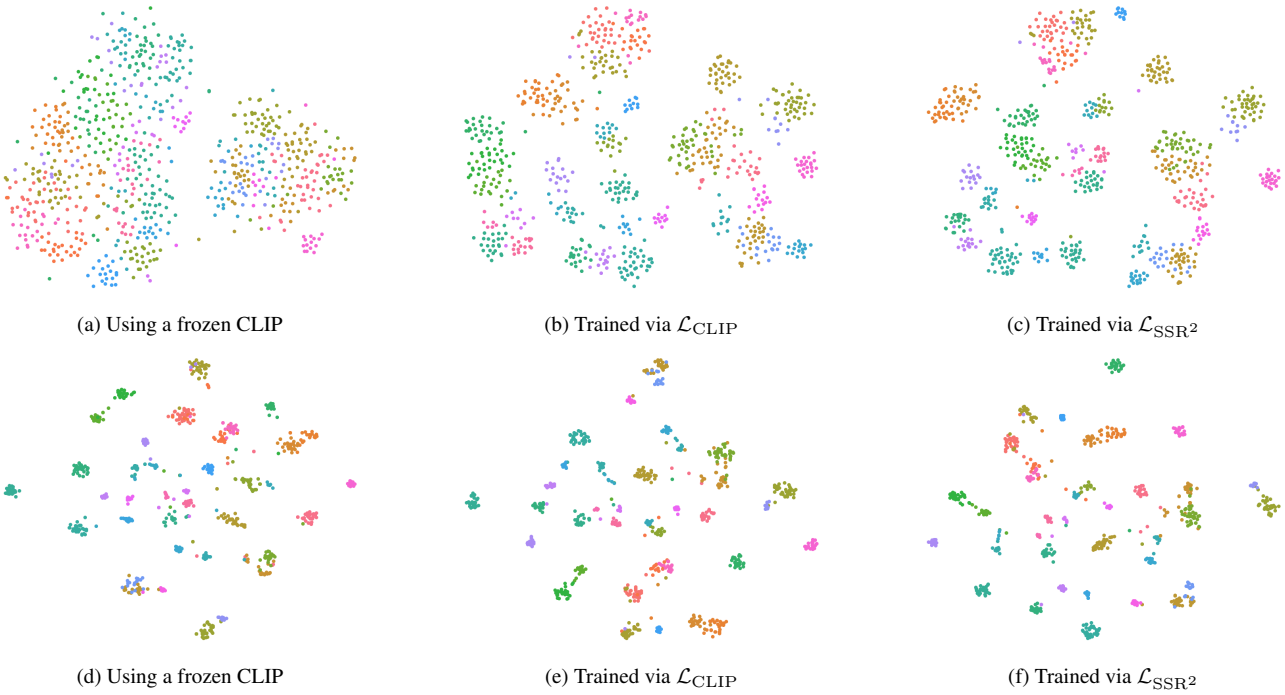


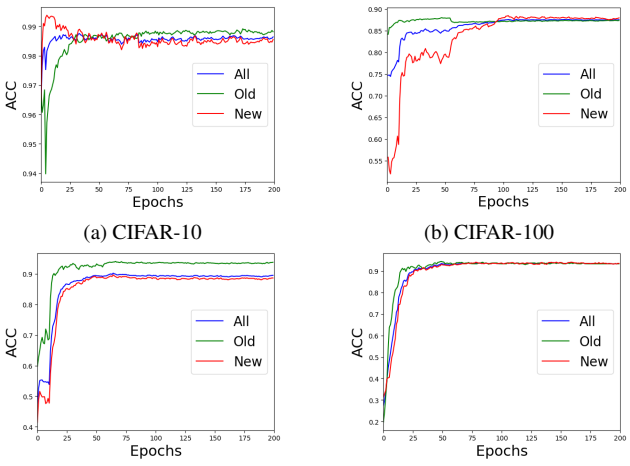
Figure B.4. ACC (%) with varying ϵ in SSR^2 across six benchmark datasets.



(a) Using a frozen CLIP (b) Trained via $\mathcal{L}_{\text{CLIP}}$ (c) Trained via $\mathcal{L}_{\text{SSR}^2}$

(d) Using a frozen CLIP (e) Trained via $\mathcal{L}_{\text{CLIP}}$ (f) Trained via $\mathcal{L}_{\text{SSR}^2}$

Figure B.5. Visualization of image embeddings (**top**) and text embeddings (**bottom**) on Oxford Pets.



(a) CIFAR-10 (b) CIFAR-100

(c) Stanford Cars (d) Flowers102

Figure B.6. ACC curves on benchmark datasets.

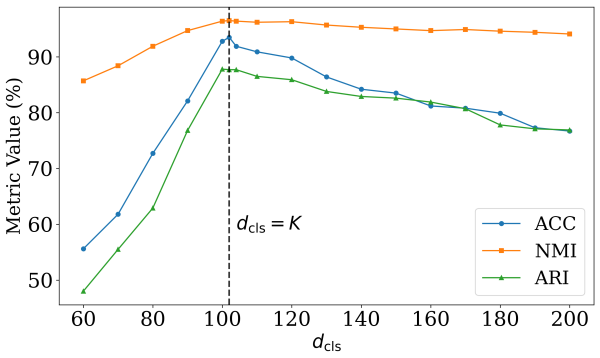


Figure B.7. Evaluation on Clustering performance ACC, NMI and ARI of our SSR^2-GCD with varying d_{cls} on Flowers102.