SCAN: Bootstrapping Contrastive Pre-training for Data Efficiency – Supplementary Material

Yangyang Guo, Mohan Kankanhalli
National University of Singapore, Singapore
guoyang.eric@gmail.com, mohan@comp.nus.edu.sg

Abstract

This supplementary material is structured as follows. First, we present the detailed algorithm of the proposed method. Next, we describe the baseline methods used for comparison. Finally, we provide an in-depth analysis of additional experiments.

1. SCAN Algorithm

We present a detailed algorithm of our proposed SCAN in Algorithm 1. This algorithm is applicable to contrastive pre-training models including CLIP and MoCo.

2. More Experimental Settings

2.1. Pre-Training Details

Our primary objective in this study is to assess the efficacy of our proposed data-efficient method. Consequently, we did not conduct an extensive parameter search and instead utilized a universal setting across different models.

Due to limitations in computational resources, most of our pre-training experiments were conducted using four NVIDIA A5000 GPUs. Specifically, for CLIP models, we employed 32 epochs, a learning rate of 1e-3, and a weight decay of 0.1. Various batch sizes are detailed in Table 1. For the downstream image classification task, we fine-tuned the pre-trained models on a single NVIDIA A100-40G GPU. Fine-tuning comprises 10 epochs with a learning rate of 1e-3 and a weight decay of 0.1.

Regarding the pre-training of MoCo, we utilized the original implementation[1]. We employed batch sizes of 600 and 370 for ViT-16/S and ViT-B/16, respectively.

2.2. Compared Baselines

We compared with the following four baselines in this work:

- **Random** prunes ρ samples with randomness for each epoch. Notably, it falls under dynamic pruning methods as the pruned samples vary across epochs.
- **SemDeDup** [1] identifies the semantic duplicates based on embedding similarities. We used one public implementation[2]. This method is applicable only to multi-modal models such as CLIP.
- **D-Pruning** [3] estimate the parameter influence of a training example through the removal of it. We utilized the official implementation[3] for CLIP models only. We abandoned the use of MoCo due to its hard-to-configure running environment.
- **Info-Batch** [2] is a recent robust dataset pruning baseline. It prunes a portion of less informative samples and then rescales the gradients of the remaining samples to approximate the original gradients. We followed the original code[4] to re-implement it for our experiments.

3. More Experimental Results

We present additional fine-tuning results of CLIP in Table 3 and Table 4. Furthermore, Table 5 shows the results of linear probing for CLIP. It is evident that our proposed SCAN method consistently achieves superior performance across various settings.

Experimental Results on CLIP-Benchmark. We utilized the CLIP-Benchmark tool to assess the performance of both CLIP and our SCAN method across 19 additional datasets. For this evaluation, we employed models pre-trained on the CC12M+ datasets. The results, presented in Table 6, demonstrate that our SCAN method delivers performance competitive with the original CLIP.

Results *w.r.t.* Pre-defined Thresholds. To assess the impact of varying thresholds, we evaluated two model architectures, RN50 and ViT-B/32, using threshold values from 0.1 to 0.7, with a step size of 0.2. The ImageNet zero-shot

[1] https://github.com/facebookresearch/moco-v3.

[2] https://github.com/BAAI-DCAI/Dataset-Pruning/tree/main.

[3] https://github.com/BAAI-DCAI/Dataset-Pruning/tree/main.

[4] https://github.com/henryqin1997/InfoBatch.

Algorithm 1: Dataset Pruning of SCAN.

Input: Full training data $\mathcal{D}$, Number of training epochs τ_{stop}, Number of mutation epochs τ_{cos}, Pre-initialized losses $\mathcal{L}_{pre}$ and $\mathcal{L}_{cur}$, Threshold value T_{td} and an infinitesimal value ϵ.

Output: Pre-trained model $\mathcal{M}$

while $\tau_{cur} < \tau_{stop}$ **do**
- `// Pre-Pruning Warm-Up`
- **if** $(\hat{\mathcal{L}}_{pre} - \hat{\mathcal{L}}_{cur})/(\hat{\mathcal{L}}_{pre} + \epsilon) \geq T_{td}$ **then**
  - **for** *Batched sample* $\mathcal{D}_t \in \mathcal{D}$ **do**
    - Forward and update $\mathcal{M}$ on $\mathcal{D}_t$;
  - **end**
  - $\hat{\mathcal{L}}_{pre} \leftarrow \hat{\mathcal{L}}_{cur}$;
  - Get the updated current epoch loss $\hat{\mathcal{L}}_{cur}$;
- **end**
- **else**
  - `// Pruning Data Preparation`
  - **if** τ_{cur} *mod* $(\tau_{cos} + 1) = 0$ **then**
    - **for** *Batched sample* $\mathcal{D}_t \in \mathcal{D}$ **do**
      - Forward and update $\mathcal{M}$ on $\mathcal{D}_t$;
      - Obtain *redundant* set $\mathcal{D}_t^{red}$ and *ill-matched* set $\mathcal{D}_t^{ill}$;
      - Obtain the overall pruning subset $\mathcal{D}_t^{'} = \mathcal{D}_t^{red} \,|\, \mathcal{D}_t^{ill}$;
    - **end**
    - Accumulate all the candidate pruning data $\mathcal{D}^{'}$;
  - **end**
  - `// Dataset Mutation`
  - **else**
    - Obtain the pruning ratio ρ_{cur};
    - Randomly prune $\rho_{cur}|\mathcal{D}^{'}|$ samples from $\mathcal{D}^{'}$;
    - **for** *Batched sample* $\mathcal{D}_t \in \mathcal{D} \setminus \mathcal{D}_{\rho}^{'}$ **do**
      - Forward and update $\mathcal{M}$ on $\mathcal{D}_t$
    - **end**
  - **end**
- **end**
- $\tau_{cur} \leftarrow \tau_{cur} + 1$

end

performance results are summarized in the table below. As indicated, the models perform optimally at threshold values of 0.3 or 0.5. For simplicity and consistency, we selected a threshold of 0.3 for subsequent model evaluations.

Different Pruning Ratios of MoCo. The performance variations with different pruning ratios (ρ) for the MoCo model are depicted in Fig. 1. It is evident that as the pruning ratios increase, there is a general degradation in performance.

More Visualization of Ill-matched Samples from CLIP. We further visualize some ill-matched samples as indicated by SCAN in Fig. 2.

References

[1] Amro Abbas, Kushal Tirumala, Daniel Simig, Surya Ganguli, and Ari S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *CoRR*, 2023. 2, 4, 5, 6

[2] Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Daquan Zhou, and Yang You. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *ICLR*, 2024. 2, 4, 5, 6

[3] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. In *ICLR*, 2023. 2, 4, 5, 6

Table 1. Batch sizes for pre-training and fine-tuning CLIP models.

PT	RN50	RN101	ViT-S/32	ViT-S/16	ViT-B/32	ViT-B/16	Swin-Base
✓	256×4	200×4	800×4	400×4	480×4	200×4	100×4
✗	384	225	1024	600	768	300	160

Base	ViT-B/16	RN101	RN50	ViT-S/16	ViT-B/32	ViT-S/32
CLIP	87	79	60	49	37	32
Static	61	55	42	34	26	23
Info-Batch	64	60	44	34	28	25
SCAN	64	62	45	34	30	26

Table 2. Pre-training time in hours.

Table 3. Performance comparison of CLIP models on the **CC3M+** pre-trained datasets. All methods utilize **30% fewer pre-trained data samples** than CLIP. Consequently, they also require approximately **30% less pre-training time**. The best results (excluding the original CLIP model) are highlighted in **bold**.

Architecture	Method	IN Zero-Shot		CIFAR10	CIFAR100	IN	IN-V2	IN-R
		Top-1	Top-5					
RN50	CLIP	17.06	36.21	95.32	80.01	73.81	61.89	36.09
	Random	11.02	25.23	94.01	75.12	70.22	58.04	31.80
	SemDeDup [1]	11.98	26.30	94.53	76.81	71.51	58.79	32.31
	D-Pruning [3]	11.72	26.65	94.48	76.73	71.11	58.79	31.88
	Info-Batch [2]	16.44	**36.74**	95.30	79.40	**73.01**	**61.49**	**35.04**
	SCAN	**16.91**	35.79	**95.30**	**80.24**	72.91	60.59	34.53
ViT-S/32	CLIP	13.70	29.33	90.59	71.74	55.60	42.81	23.91
	Random	06.57	16.19	86.61	60.18	48.87	34.48	17.98
	SemDeDup [1]	05.33	14.05	85.16	59.87	47.39	35.56	17.70
	D-Pruning [3]	04.78	12.91	84.21	57.96	46.53	34.77	16.88
	Info-Batch [2]	10.89	26.91	90.02	69.99	50.53	39.61	19.69
	SCAN	**14.88**	**31.47**	**90.12**	**70.33**	**54.13**	**41.29**	**22.70**
ViT-S/16	CLIP	18.41	37.41	96.09	81.31	68.49	55.79	29.52
	Random	07.80	21.53	93.58	72.11	62.13	49.63	19.01
	SemDeDup [1]	09.57	22.00	93.43	74.37	62.30	48.89	23.04
	D-Pruning [3]	08.60	20.35	93.26	73.72	61.70	48.97	22.46
	Info-Batch [2]	16.19	35.06	**95.64**	80.03	67.57	53.52	**27.64**
	SCAN	**17.31**	**35.51**	95.53	**80.27**	**66.86**	**53.59**	27.34
ViT-B/32	CLIP	14.97	32.02	94.43	77.72	58.33	45.70	25.59
	Random	07.44	18.88	89.96	69.41	50.43	40.62	18.07
	SemDeDup [1]	07.20	17.50	90.88	70.13	50.99	38.34	19.76
	D-Pruning [3]	06.51	16.13	60.07	69.11	50.01	38.43	19.03
	Info-Batch [2]	12.44	30.98	93.57	75.44	55.99	43.30	**24.64**
	SCAN	**16.48**	**33.60**	**93.77**	**77.63**	**56.64**	**44.25**	24.10

Table 4. Performance comparison of CLIP models on the **CC12M+** pre-trained datasets. All methods utilize **30% fewer pre-trained data samples** than CLIP. Consequently, they also require approximately **30% less pre-training time**. The best results (excluding the original CLIP model) are highlighted in **bold**.

Architecture	Method	IN Zero-Shot		CIFAR10	CIFAR100	IN	IN-V2	IN-R
		Top-1	Top-5					
RN50	CLIP	20.95	44.41	95.68	80.75	74.93	62.81	38.36
	Random	12.39	35.96	94.89	76.96	71.65	59.71	32.03
	SemDeDup [1]	15.89	36.76	95.00	78.12	72.46	60.01	33.86
	D-Pruning [3]	11.19	26.53	94.31	77.69	71.96	59.19	33.44
	Info-Batch [2]	20.63	45.10	**95.68**	79.88	73.53	61.23	36.67
	SCAN	**23.03**	**47.83**	95.63	**81.03**	**74.28**	**62.20**	**38.14**
ViT-S/32	CLIP	26.48	51.32	93.23	76.32	61.53	48.60	30.57
	Random	08.79	16.93	87.79	63.04	50.12	38.09	21.11
	SemDeDup [1]	05.04	13.49	86.43	61.67	49.46	37.37	19.29
	D-Pruning [3]	04.54	12.43	85.86	61.81	48.39	36.57	18.62
	Info-Batch [2]	10.07	26.63	91.11	67.94	53.47	40.91	20.77
	SCAN	**25.27**	**50.08**	**91.86**	**75.27**	**59.87**	**46.96**	**27.86**
ViT-S/16	CLIP	27.09	53.57	96.62	84.05	71.40	58.40	34.24
	Random	16.58	35.43	95.00	79.90	67.78	54.12	26.23
	SemDeDup [1]	10.56	26.52	94.46	76.65	65.32	51.37	25.52
	D-Pruning [3]	09.37	22.16	93.42	75.52	63.53	50.79	24.43
	Info-Batch [2]	21.28	45.56	96.09	82.13	68.87	55.90	29.58
	SCAN	**28.46**	**54.56**	**96.24**	**83.32**	**70.40**	**57.10**	**31.85**

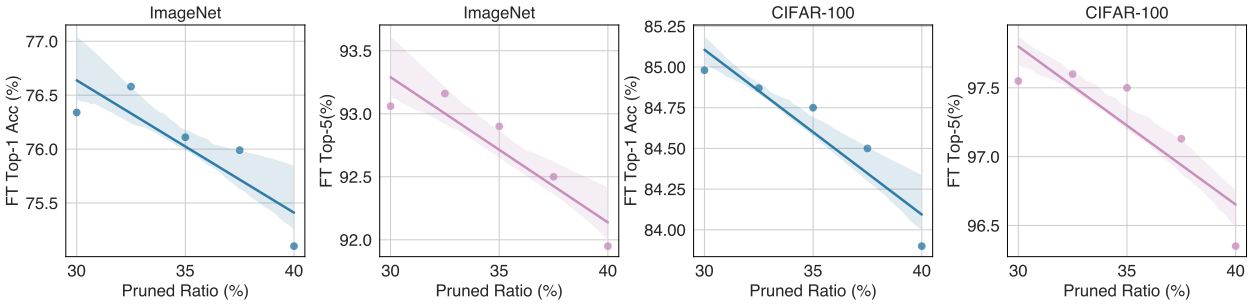


Figure 1. Downstream performance variation of ViT-S/16 MoCo model *w.r.t.* different pruning ratios.

Table 5. Linear probing results of six CLIP models. All methods utilize **30% fewer pre-trained data samples** than CLIP. Consequently, they also require approximately **30% less pre-training time**. The best results (excluding the original CLIP model) are highlighted in **bold**. A dash (-) indicates the collapse of pre-training, resulting in impaired evaluation of downstream tasks.

Arc	Method	CC3M+					CC12M+				
		CF-10	CF-100	IN	IN-V2	IN-R	CF-10	CF-100	IN	IN-V2	IN-R
RN50	CLIP	95.58	80.31	73.96	61.60	35.59	95.69	81.88	74.96	62.85	38.57
	Random	93.89	75.45	70.25	58.05	31.78	94.00	76.43	70.99	58.78	32.09
	SemDeDup [1]	94.92	77.16	71.62	58.99	32.44	94.88	78.00	72.22	59.70	33.16
	D-Pruning [3]	94.50	76.78	71.00	57.98	31.70	94.30	77.70	71.77	59.01	33.20
	Info-Batch [2]	95.29	79.39	73.07	61.03	**34.66**	**95.66**	79.84	73.23	61.10	36.63
	SCAN	**95.46**	**80.35**	**73.07**	**61.25**	34.59	95.62	**81.28**	**74.27**	**62.66**	**37.30**
RN101	CLIP	95.92	82.04	75.10	63.61	38.78	96.03	82.73	75.78	63.93	40.09
	Random	95.00	78.13	73.79	60.20	36.12	95.02	78.34	73.99	60.27	36.13
	SemDeDup [1]	94.84	79.25	74.08	61.94	36.74	95.01	78.02	73.89	59.91	33.80
	D-Pruning [3]	94.79	72.12	73.74	61.66	35.64	94.78	78.83	74.08	61.28	37.09
	Info-Batch [2]	95.08	80.76	74.13	62.89	37.57	95.82	81.56	75.02	63.21	39.21
	SCAN	**95.67**	**81.36**	**74.42**	**63.07**	**37.86**	**95.93**	**82.12**	**75.61**	**63.87**	**39.32**
ViT-S/32	CLIP	91.65	72.23	55.52	43.00	23.48	93.29	77.06	61.73	48.84	30.40
	Random	87.00	61.31	49.97	36.07	20.88	87.79	63.04	50.12	38.09	21.11
	SemDeDup [1]	83.46	60.06	47.65	35.51	17.61	86.23	61.77	49.20	37.10	19.11
	D-Pruning [3]	84.21	58.73	46.57	35.03	16.95	85.82	61.09	47.99	36.58	18.00
	Info-Batch [2]	89.30	70.02	50.51	39.58	19.78	91.02	68.90	53.49	40.69	20.71
	SCAN	**89.37**	**71.05**	**54.24**	**41.30**	**22.65**	**91.88**	**74.86**	**59.90**	**46.90**	**27.90**
ViT-S/16	CLIP	96.09	81.39	68.49	55.19	29.06	96.66	84.35	71.53	58.56	33.85
	Random	93.62	73.37	63.02	49.96	20.62	94.90	79.91	67.90	54.10	26.24
	SemDeDup [1]	93.21	73.85	62.34	49.40	22.54	94.00	77.01	64.45	51.40	25.51
	D-Pruning [3]	93.28	73.09	61.67	48.99	22.48	93.41	75.43	63.42	50.77	24.41
	Info-Batch [2]	95.26	**80.46**	**67.76**	53.49	27.11	96.03	82.11	68.78	55.78	29.59
	SCAN	**95.31**	80.00	67.04	**53.75**	**27.41**	**96.37**	**82.71**	**70.32**	**57.17**	**31.89**
ViT-B/32	CLIP	94.36	77.84	58.43	45.79	25.50	95.65	81.62	63.40	50.33	31.28
	Random	90.05	69.26	50.23	40.54	18.03	90.13	69.98	51.99	41.01	20.08
	SemDeDup [1]	90.44	69.86	50.89	38.15	19.89	90.77	70.00	51.19	39.80	20.91
	D-Pruning [3]	90.06	69.08	50.04	37.87	19.11	90.07	69.65	51.23	37.99	20.43
	Info-Batch [2]	93.54	75.49	**56.98**	44.03	24.08	-	-	-	-	-
	SCAN	**94.00**	**76.91**	56.72	**44.12**	**24.21**	**95.05**	**81.21**	**61.96**	**48.42**	**29.53**
ViT-B/16	CLIP	96.27	82.74	70.87	57.77	29.82	96.77	84.48	72.37	59.07	33.24
	Random	91.60	73.61	50.59	40.52	21.72	94.56	76.67	67.57	54.40	27.10
	SemDeDup [1]	94.16	76.34	66.60	53.13	25.60	94.17	76.66	67.10	53.39	27.11
	D-Pruning [3]	93.48	75.41	65.90	52.69	24.57	93.88	75.99	65.98	53.00	26.05
	Info-Batch [2]	96.10	81.06	**70.30**	56.10	28.48	96.12	81.78	71.34	56.25	31.12
	SCAN	**96.16**	**81.10**	69.55	**56.48**	**28.76**	**96.12**	**83.97**	**71.82**	**58.31**	**32.48**

Text: during sports team vs game

Text: look out onto the blue waters while taking a dip in pool

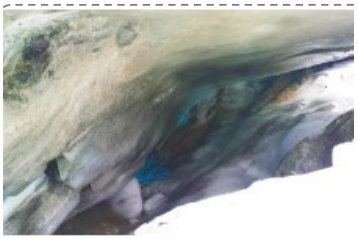
Text: an aerial view of home

Text: the first passengers disembark the flight

Text: a fountain of an embracing young couple under an umbrella

Text: biological species crawling on a banana leaf

Text: the fish pond and remains in the grounds



Text: magic blue glow under the glacier, photo by person

Text: select a wallpaper for children's rooms-wall to feel

Figure 2. More *ill-matched* samples obtained by our SCAN approach.

Table 6. Comparison of ViT-B/32 and ViT-B/16 using CLIP and SCAN on CLIP-Benchmark.

Dataset	ViT-B/32		ViT-B/16	
	CLIP	SCAN	CLIP	SCAN
FER2013	18.50	22.27	18.36	20.77
ImageNet-O	30.70	30.55	33.05	31.20
ImageNet-R	29.23	31.91	31.08	29.67
ImageNetv2	20.19	21.80	21.39	20.90
ObjectNet	15.13	13.93	14.84	15.03
rendered-sst2	50.08	49.92	51.12	50.02
STL-10	85.18	86.06	85.11	85.04
SUN397	40.55	41.02	41.95	41.29
VOC-2007	47.22	42.62	52.59	48.48
Caltech-101	64.93	68.56	65.63	65.46
Dmlab	20.02	11.81	17.77	16.19
DTD	15.66	16.44	16.24	13.83
EuroSat	21.92	29.81	34.20	29.67
Flowers	18.63	24.70	20.80	20.13
KITTI	32.63	32.77	35.49	35.59
PCam	50.33	52.23	50.32	52.69
Pet	31.28	43.06	36.41	35.84
RESISC45	23.41	23.05	21.28	19.38
SVHN	16.99	06.97	09.73	07.86

Table 7. Performance comparison of RN50 and ViT-B/32 at different thresholds.

Threshold	RN50		ViT-B/32	
	Top-1	Top-5	Top-1	Top-5
0.1	15.80	35.21	14.75	31.58
0.3	16.91	35.79	16.48	33.60
0.5	18.22	37.79	16.04	33.19
0.7	18.20	37.78	16.48	33.23