Hardness aware positive pairing

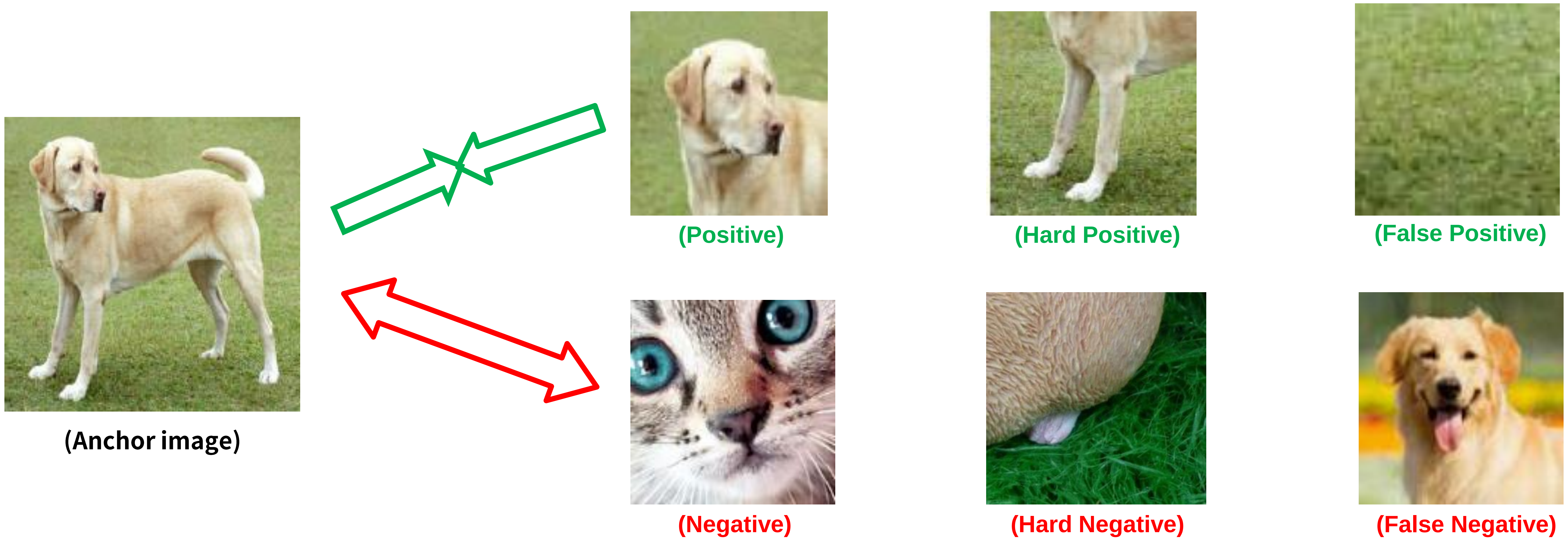
김성윤, 박재승
Deep Representation Learning Group 인턴

Table of contents

- Review on hard/false samples of contrastive learning
- Hardness aware property of contrastive learning
- Objectives of the experiments
- Results
- TODO/ On going

Hard/ False samples in contrastive learning

- Given an anchor image, contrastive learning pulls the “positive images” together, and pushes the “negative” images apart



- Concentrate more on hard samples, ignore false samples for better representation

Figure from:
Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." *International conference on machine learning*. PMLR, 2020.

Hardness aware property of contrastive learning

For a given anchor image x_i , the contrastive loss is given as

$$L(x_i) = -\log\left(\frac{\exp(\frac{s_{i,i}}{\tau})}{\sum_{k=1}^{N}\exp(\frac{s_{i,k}}{\tau})}\right)$$

Its gradient with respect to $s_{i,j}$ shows that contrastive loss is actually a “hardness-aware” loss function, meaning it repels similar negatives harder

$$\frac{\partial L(x_i)}{\partial s_{i,j}} = \frac{1}{\tau}\frac{\exp(\frac{s_{i,j}}{\tau})}{\sum_{k=1}^{N}\exp(\frac{s_{i,k}}{\tau})}$$

By controlling tau, the hardness-aware property can be controlled:

- As tau approaches 0, the loss approaches to triplet loss
- As tau approaches infinity, the loss concentrates on all negative samples

Objectives

Checking that we can control the hardness aware property of Sup-con loss in multi-positive setting using tau_positive

$$L_{sup}^{in}(x_i) = -\log\left(\frac{\sum_{j\in P(i)} \exp(\frac{s_{i,j}}{\tau_{pos}})}{\sum_{j=1}^{N} \exp(\frac{s_{i,j}}{\tau_{neg}})}\right)$$

$$\frac{\partial L_{sup}^{in}(x_i)}{\partial s_{i,k}} = \frac{\lambda_{pos}}{\tau_{pos}} \frac{\exp(\frac{s_{i,k}}{\tau_{pos}})}{\sum_{j\in P(i)} \exp(\frac{s_{i,j}}{\tau_{pos}})} - \frac{\lambda_{neg}}{\tau_{neg}} \frac{\exp(\frac{s_{i,k}}{\tau_{neg}})}{\sum_{j=1}^{N} \exp(\frac{s_{i,j}}{\tau_{neg}})}$$

A similar gradient analysis for Sup-con-in loss shows that

- smaller tau_pos -> Concentrate more on similar positive pairs
- larger tau_pos -> Uniform concentration on every positives

For weaker augmentations, large tau_pos will be beneficial(as using harder positives is more beneficial)
For stronger augmentations, smaller tau_pos is beneficial(as stronger augmentations will possibly make more false positives)

- Experiment with Sup-con model + various tau_pos & augmentation strength

Experiment settings

- Used default setting of SupCon for positives: Construct positive batch by merging two different augmentations of a single batch

- Used true marginal negatives for cifar100: Construct negative batch independently from positive batch

- Fixed tau_neg & lambda_neg, changed lambda_pos proportional to tau_pos

$$\frac{\partial L^{in}_{sup}(x_i)}{\partial s_{i,k}} = \frac{\lambda_{pos}}{\tau_{pos}} \frac{\exp(\frac{s_{i,k}}{\tau_{pos}})}{\sum_{j \in P(i)} \exp(\frac{s_{i,j}}{\tau_{pos}})} - \frac{\lambda_{neg}}{\tau_{neg}} \frac{\exp(\frac{s_{i,k}}{\tau_{neg}})}{\sum_{j=1}^{N} \exp(\frac{s_{i,j}}{\tau_{neg}})}$$

Similarity Analysis

Objective: Check if augmentations actually change the similarity distributions & how augmentation strength and distribution is related

Using a pretrained network, compute the similarity of various image pairs:

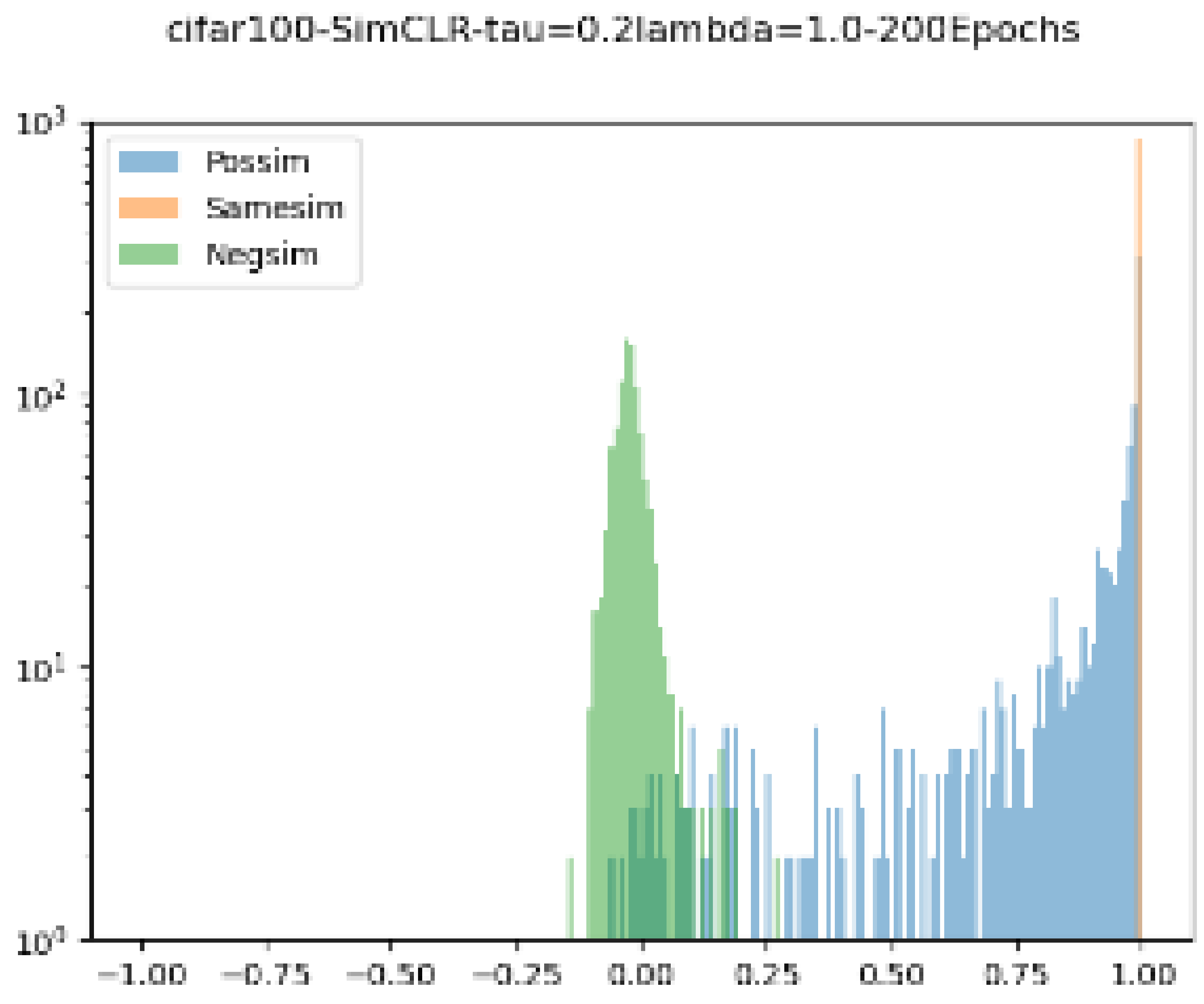
- anchor vs Aug(anchor)
- anchor vs Aug(Imgs from the same class)
- anchor vs Aug(Imgs from different classes)

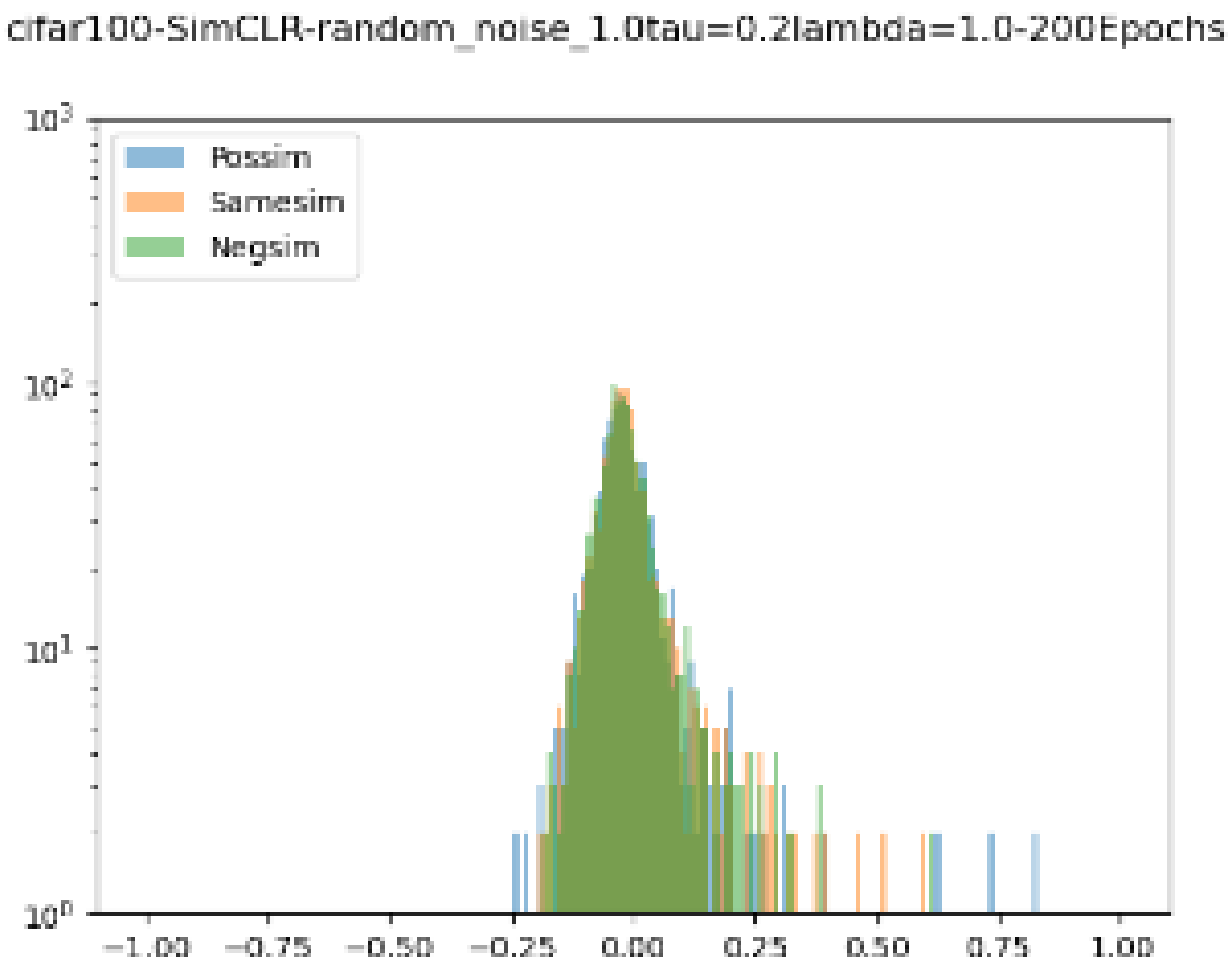
Pretrained networks include networks trained with SupCon Loss and Xent Loss.
Augmentations used in training the networks include SimCLR Aug, Supervised Aug, and no Aug.
Only the results with SimCLR Aug + SupCon Loss will be shown.

Random noise “augmentation”

A special type of augmentation that definitely creates false positives.

Similarity analysis result(cifar100)





Random noise “augmentation”

A special type of augmentation that definitely creates false positives.

(Cifar10 results)

Random Noise Probability and tau_pos experiment				
Random noise & Tau_positive				
Probability	Baseline (no aug)	0.2	0.5	0.8
tau_pos: 0.05	**24.8**	82.29	83.36	66.13
tau_pos: 0.1	**26.67**	82.75	84.39	52.64
tau_pos: 0.2	87.72	88.03	88.24	87.51
tau_pos: 3	87.99	87.87	87.84	85.35
tau_pos: 5	88.15	87.62	87.18	84.51

- When tau_pos is low, the model fails to learn as they learn from “the most similar” positive pairs
- Random noise helps erase the “most similar” positive pairs, helping the model recover accuracy: Nicely understood by hardness aware property

Random noise “augmentation”

A special type of augmentation that definitely creates false positives.

(Cifar100 results)

Random Noise Probability and tau_pos experiment					
Random noise & Tau_positive					
Probability	**Baseline (no aug)**	**0.1**	**0.3**	**0.5**	**0.7**
tau_pos: 0.1	**6.33**	17.62	57.58	58.29	42.61
tau_pos: 0.2	**48.17**	55.88	59.74	58.72	31.33
tau_pos: 0.4	64.25	62.85	61.37	29.22	9.93
tau_pos: 1	63.50	62.95	57.21	27.16	7.87
tau_pos: 3	63.06	61.92	53.62	21.97	10.25
tau_pos: 10	63.46	62.60	53.64	29.88	12.70

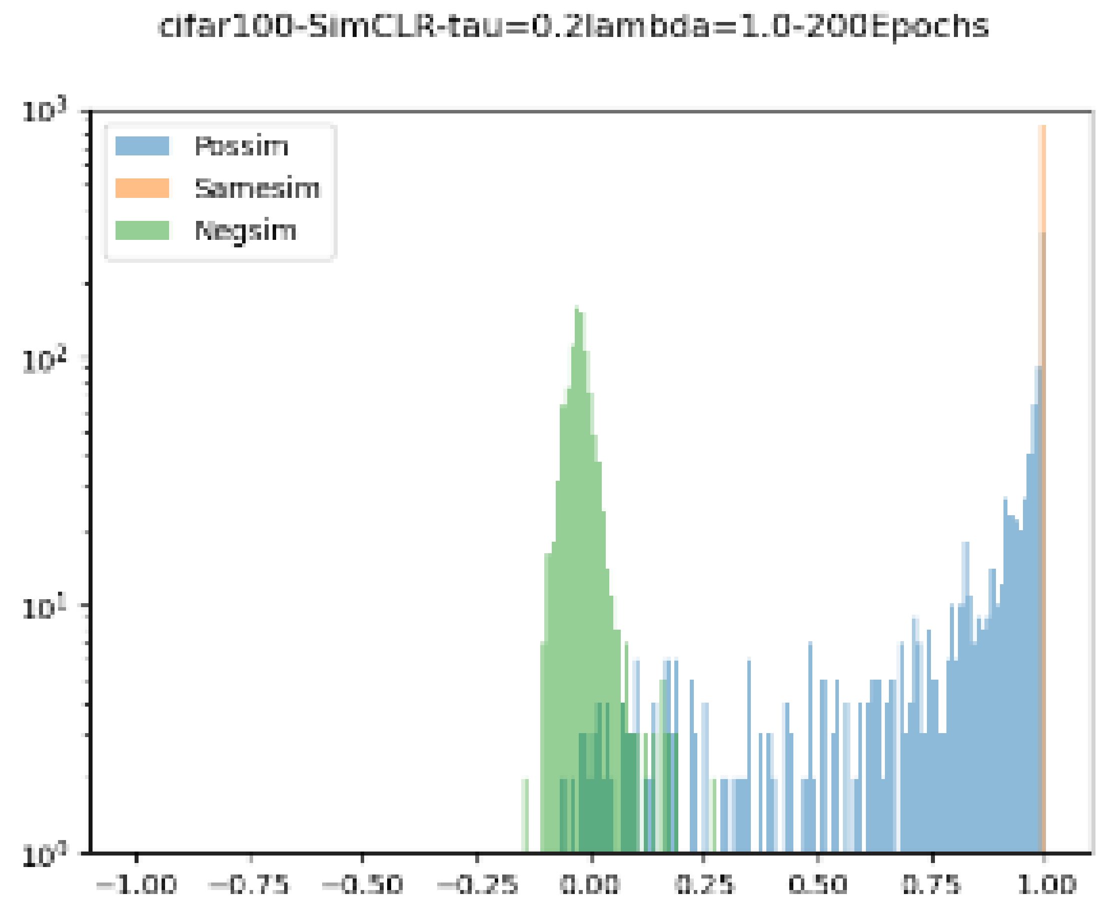
- Similar recovery in accuracy is found.

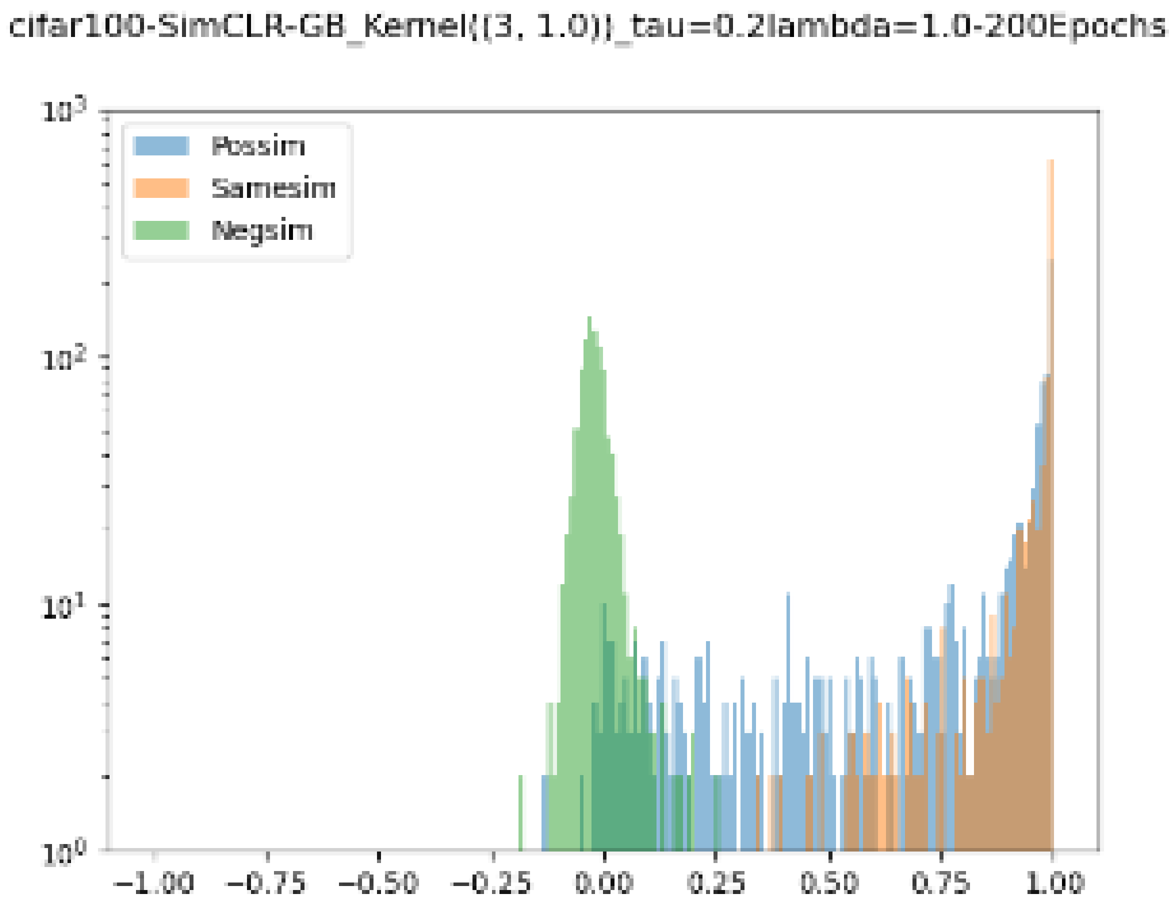
Gaussian Blur

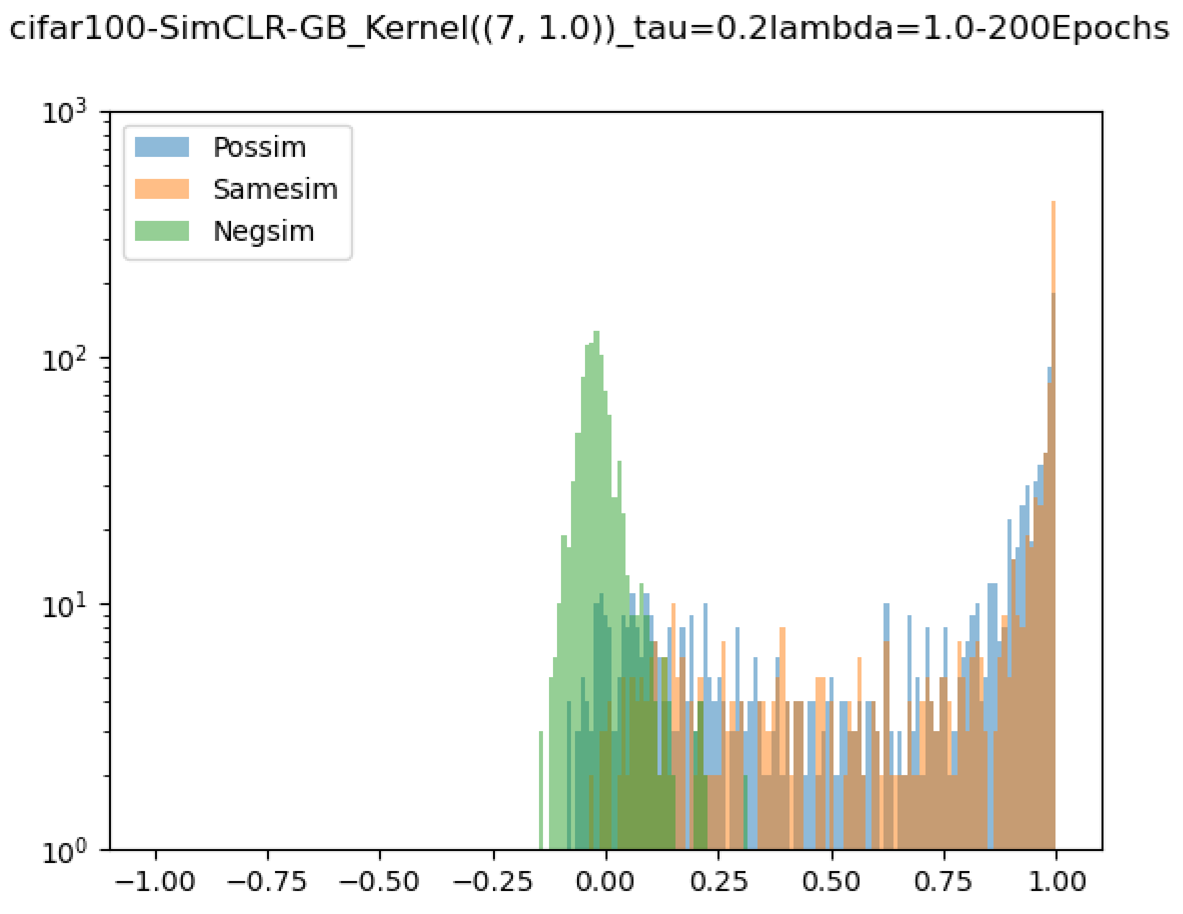
Apply Gaussian Blur with given kernel size

Strength controlled by kernel size & probability

Similarity analysis(cifar100)







Gaussian Blur

Apply Gaussian Blur with given kernel size

Strength controlled by kernel size & probability

(cifar100 results)

Gaussian Blur kernel, probability and tau_pos experiment

Gaussian Blur, kernel_size = 3, Probability control

Probability	Baseline(no aug)	0.3	0.5
tau_pos: 0.4	64.25	59.35	57.88
tau_pos: 1	63.50	60.79	57.34
tau_pos: 3	63.06	60.78	59.20
tau_pos: 10	63.46	**60.99**	**59.47**

Gaussian Blur, Probability = 0.3, Kernel size control

Kernel size	Baseline(no aug)	3	7
tau_pos: 0.4	64.25	59.35	55.28
tau_pos: 1	63.50	60.79	57.72
tau_pos: 3	63.06	60.78	58.09
tau_pos: 10	63.46	**60.99**	**58.52**

Gaussian Blur, kernel_size = 7, Probability control

Probability	Baseline(no aug)	0.3	0.5
tau_pos: 0.4	64.25	55.28	53.70
tau_pos: 1	63.50	57.72	55.53
tau_pos: 3	63.06	58.09	55.85
tau_pos: 10	63.46	**58.52**	**56.43**

Gaussian Blur, Probability = 0.5, Kernel size control

Kernel size	Baseline(no aug)	3	7
tau_pos: 0.4	64.25	57.88	53.70
tau_pos: 1	63.50	57.34	55.53
tau_pos: 3	63.06	59.2	55.85
tau_pos: 10	63.46	**59.47**	**56.43**

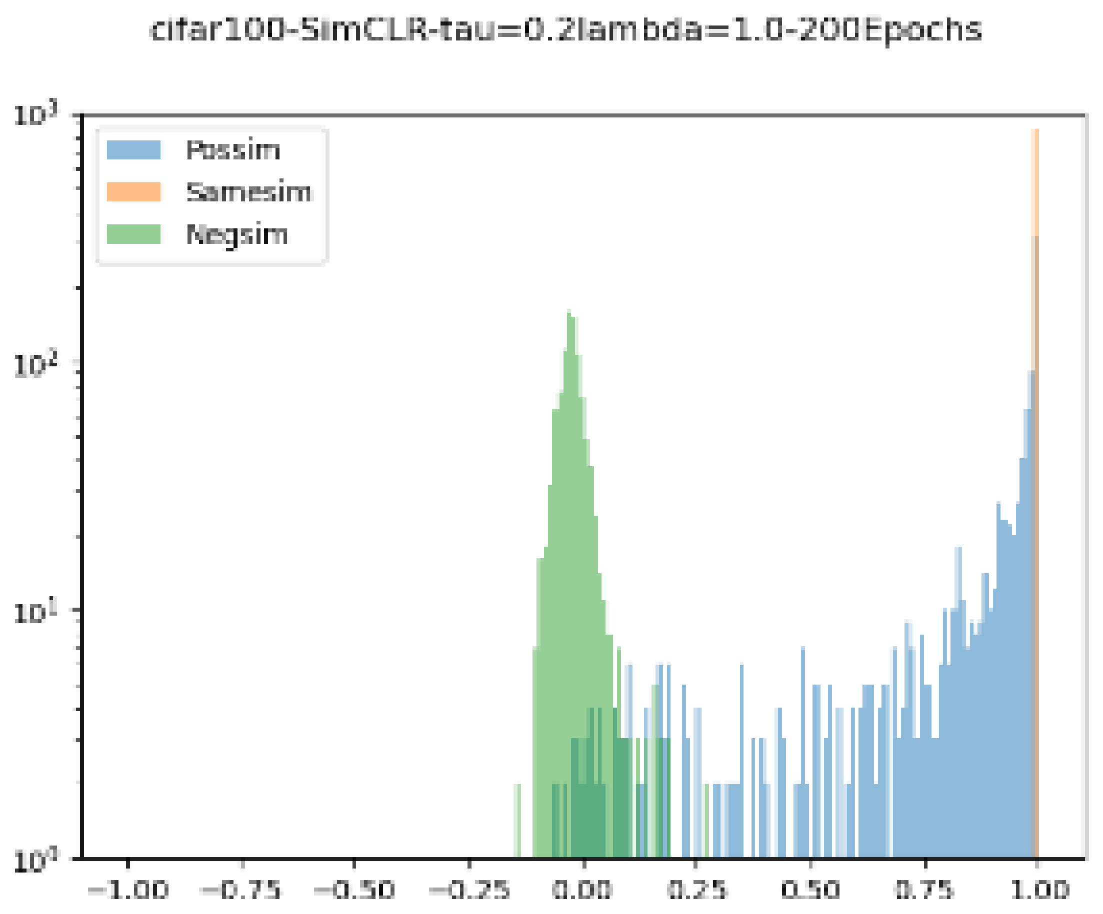
- Benefits from larger tau_pos, even when the augmentation gets stronger

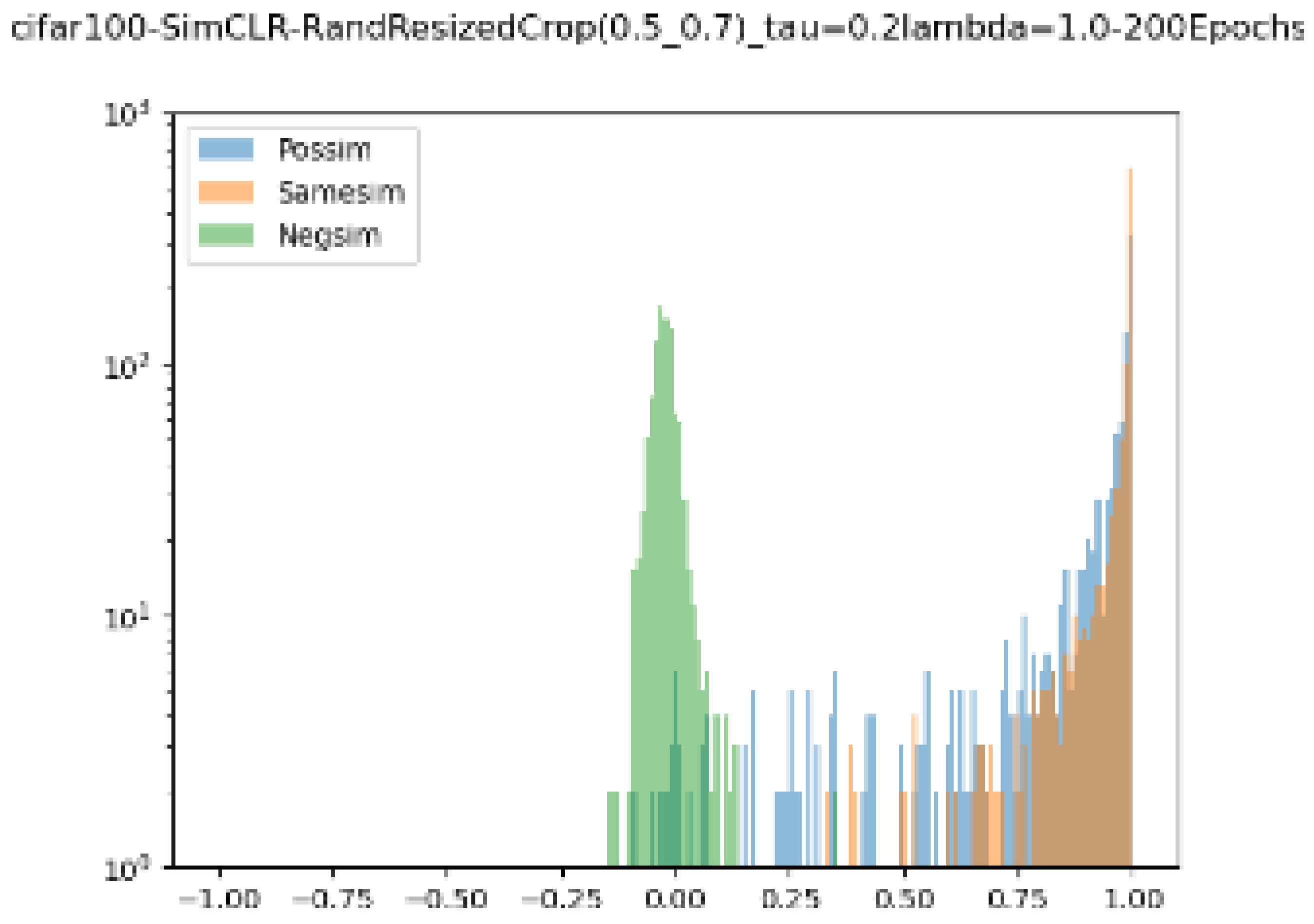
Random Resized Crop

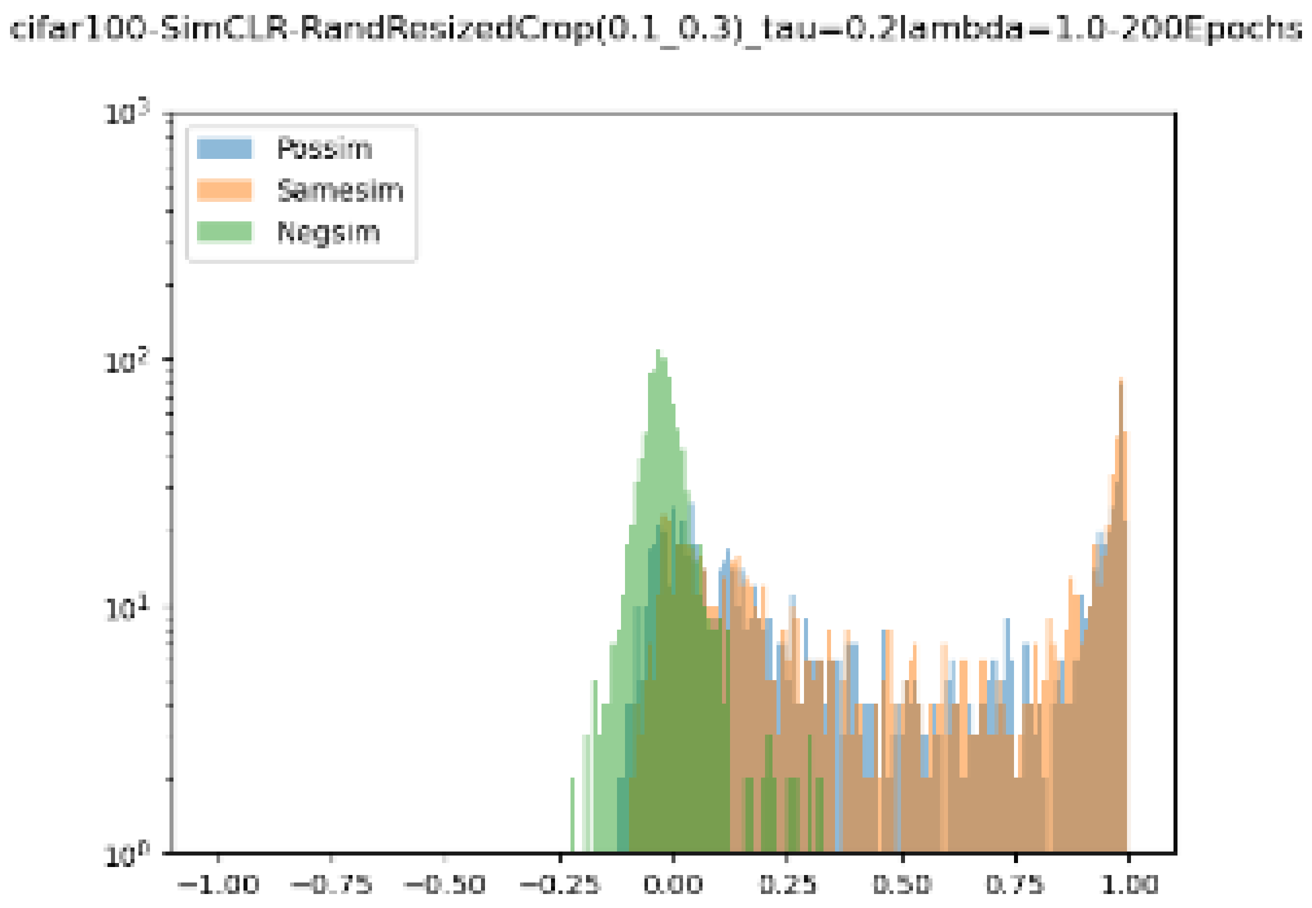
Randomly crop the image with given scale, and resize the image

Strength controlled by scale

Similarity analysis(cifar100)







Random Resized Crop

Randomly crop the image with given scale, and resize the image

Strength controlled by scale

(cifar100 results)

Random Resized Crop and tau_pos(prob = 1)

Scale	Baseline(no aug)	(0.7,1)	(0.5,1)	(0.3,1)	(0.1,1)
tau_pos: 0.3	62.65	**72.57**	**71.96**	70.87	68.61
tau_pos: 0.4	64.25	72.15	71.57	**71.11**	70.10
tau_pos: 1	63.50	72.04	71.50	70.83	70.96
tau_pos: 3	63.06	71.08	71.53	70.93	**71.60**

Scale	Baseline(no aug)	(0.7,1)	(0.3,0.7)	(0.3,0.5)	(0.1,0.3)	(0.01,0.1)
tau_pos: 0.3	62.65	**72.57**	67.11	61.06	46.12	31.40
tau_pos: 0.4	64.25	72.15	**67.98**	60.94	47.99	32.47
tau_pos: 1	63.50	72.04	67.34	60.71	49.35	35.90
tau_pos: 3	63.06	71.08	67.30	**61.19**	**51.88**	**37.11**

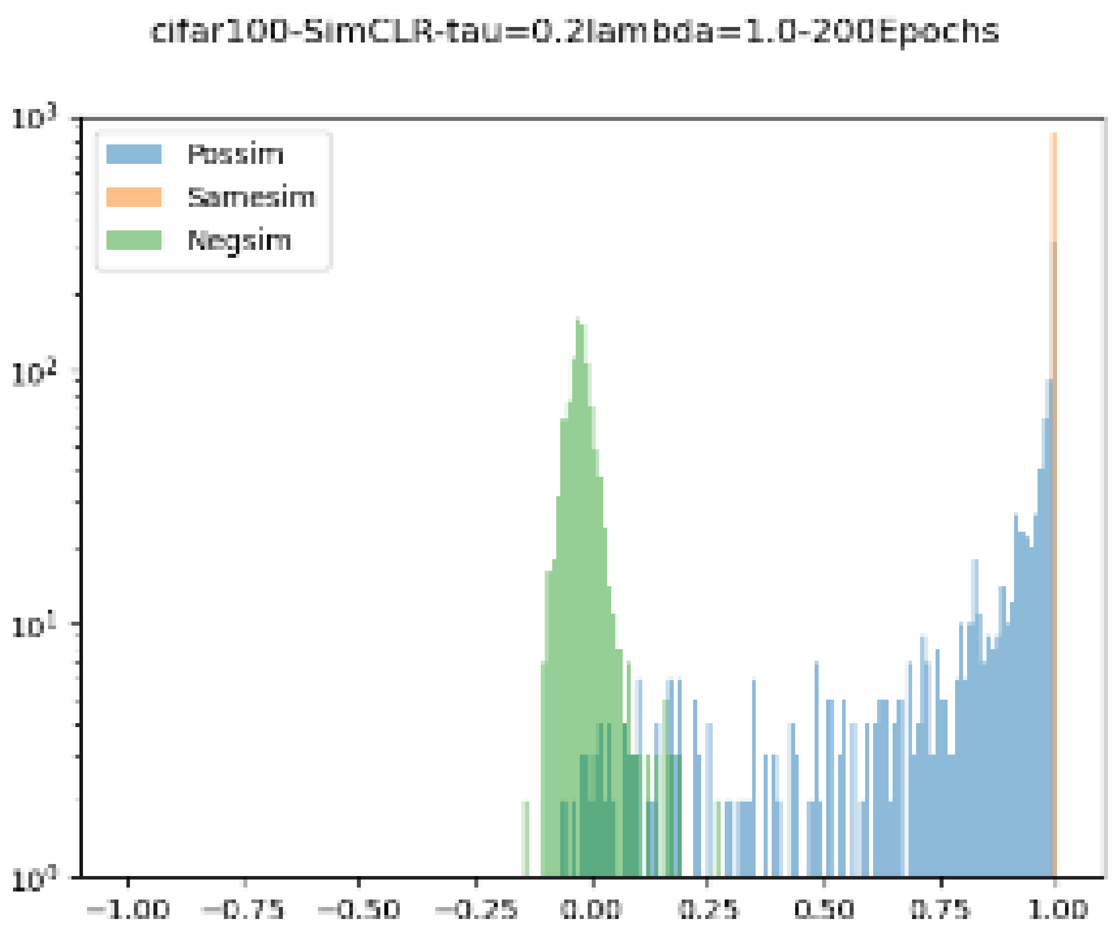
- Weaker augmentation benefits from small tau_pos, stronger augmentation benefits from large tau_pos: Opposite from our hypothesis

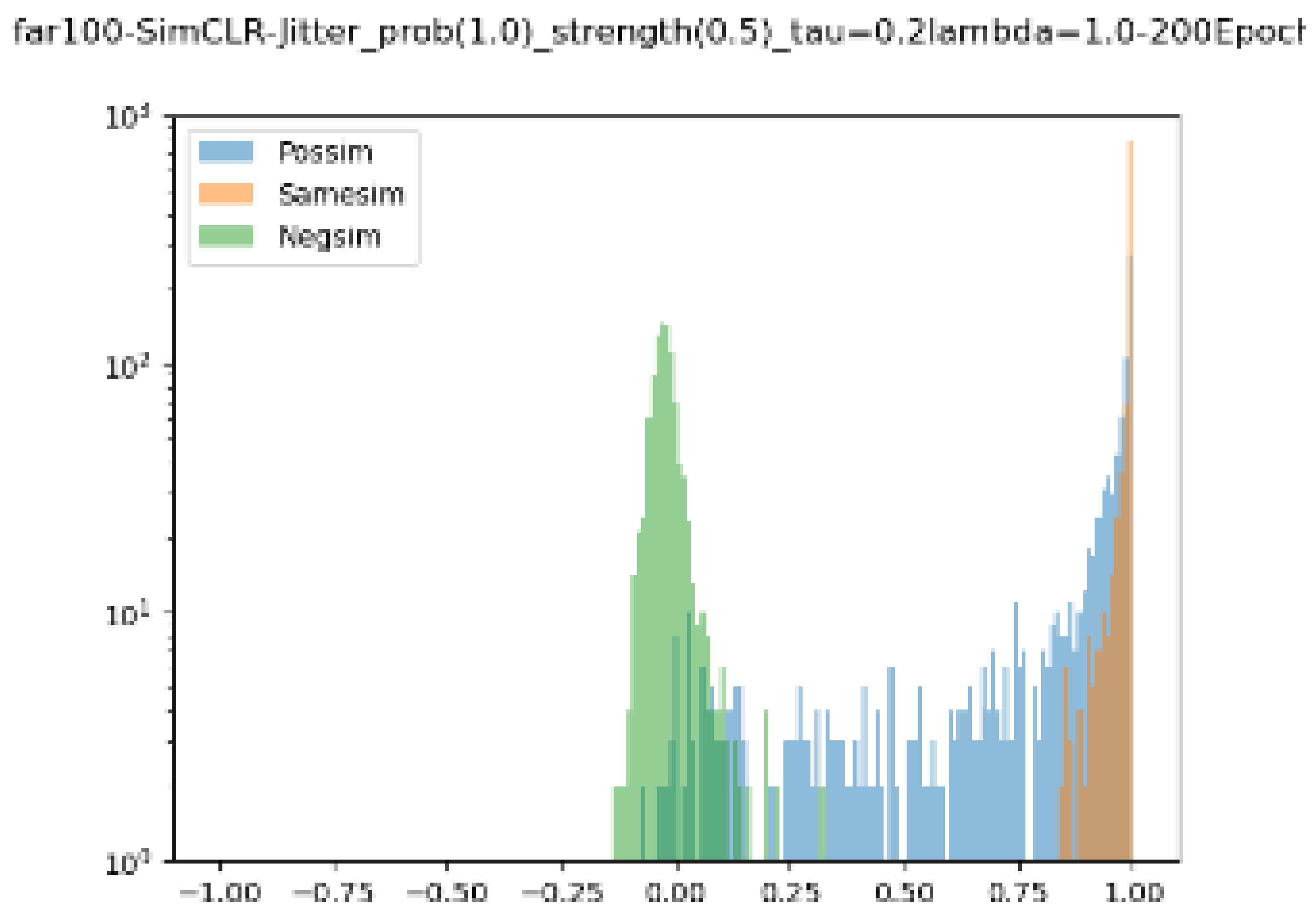
Color jitter

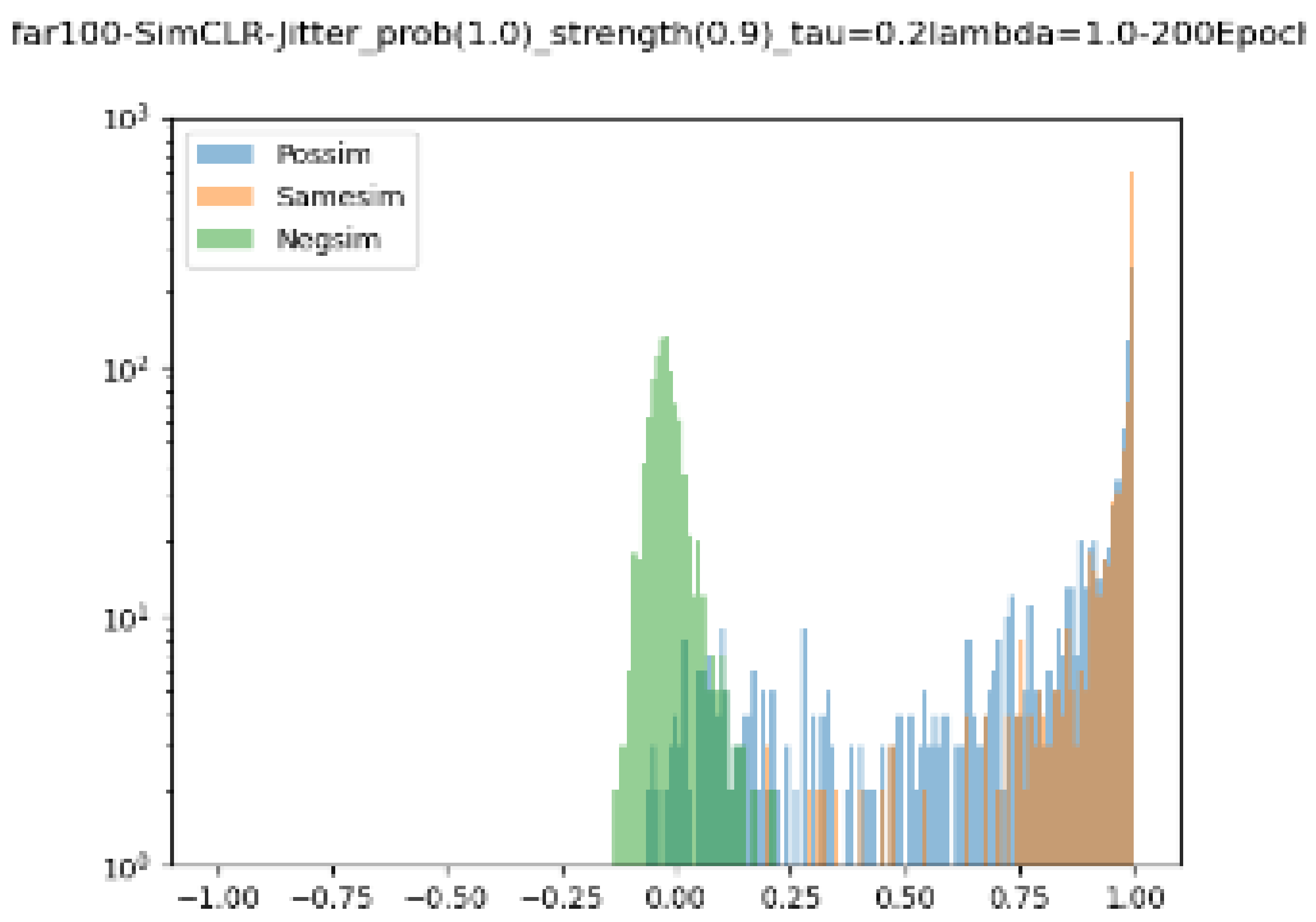
Distort the brightness/contrast/saturation/hue of a given image

Strength controlled by controlling range of the above parameters

Similarity analysis(cifar100)







Color jitter

Distort the brightness/contrast/saturation/hue of a given image

Strength controlled by controlling range of the above parameters

(cifar100 results)

Color Jitter p=0.8 (default) and tau_pos						
CJ Strength	**Baseline(no aug)**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
tau_pos: 0.3	62.65	63.17	62.72	61.00	59.68	57.67
tau_pos: 0.4	**64.25**	**64.22**	**63.89**	62.07	60.32	58.37
tau_pos: 1	63.50	63.11	62.74	**62.74**	**60.99**	58.16
tau_pos: 3	63.06	63.03	62.78	62.61	60.29	**58.62**

- Weaker augmentation benefits from small tau_pos, stronger augmentation benefits from large tau_pos: Maybe colorjitter is just a weak augmentation

TODO/ On Going

Isolating the effects positive pairs by replacing the negative samples with random noise images, as in NCE

- Collapses to a trivial solution

Analyzing the effects of augmentations on top of more commonly used setups, like SimCLR augmentation

- SimCLR Aug done; trying Supervised Training Augmentations
- No significant difference in the trend

If there does not exist a clear relationship between augmentation strength and tau_pos, it may be more plausible to examine the overall distribution as in ‘Similarity Analysis’ without considering the effect of different augmentations or augmentation strengths

- more of a learning dynamics approach