Improving Contrastive Learning by Visualizing Feature Transformation

Rui Zhu*, Bingchen Zhao*, Jingen Liu, Zhenglong Sun, Chang Wen Chen

ICCV 2021 (Oral)

STRUCT Group Seminar

Presenter: Wenjing Wang

2020.10.08

OUTLINE

- Authorship
- Background
- Proposed Method
- Experimental Results
- Conclusion

OUTLINE

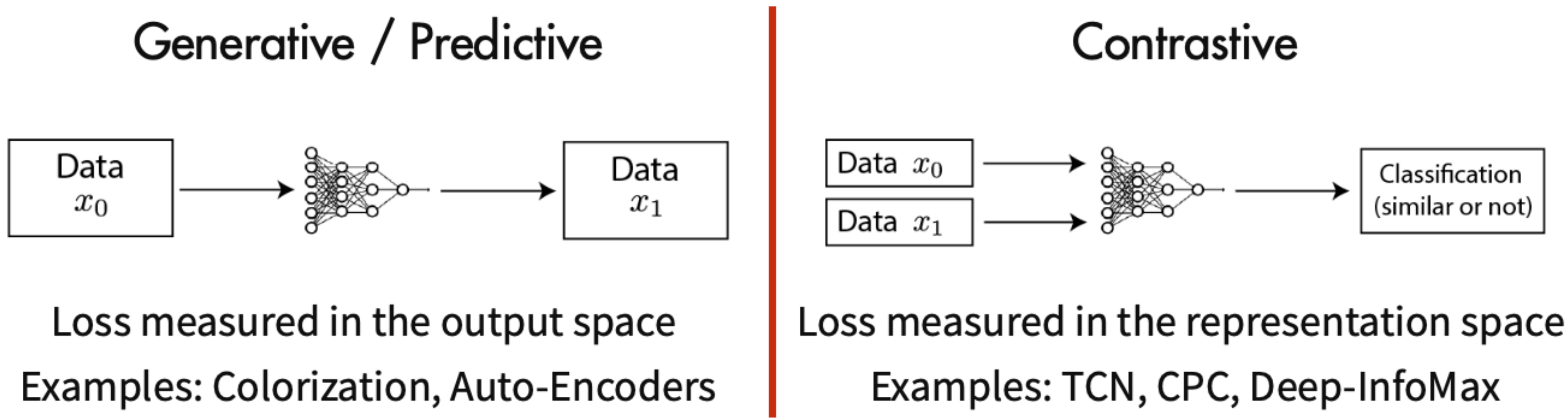
- Authorship
- Background
- Proposed Method
- Experimental Results
- Conclusion

OUTLINE

- Authorship
- Background
- Proposed Method
- Experimental Results
- Conclusion

BACKGROUND

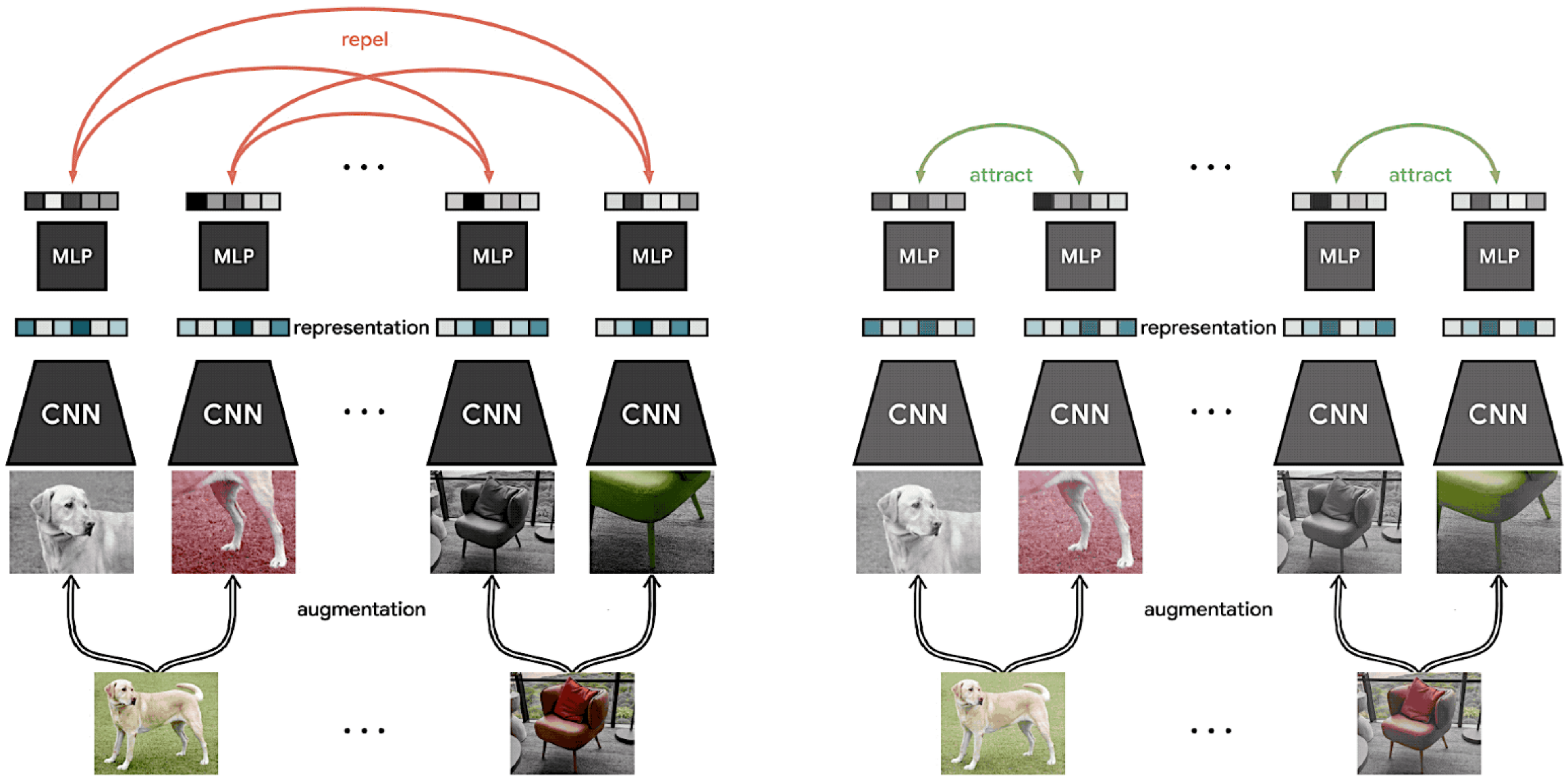
➤ Contrastive



[1] https://ankeshanand.com/blog/2020/01/26/contrative-self-supervised-learning.html

BACKGROUND

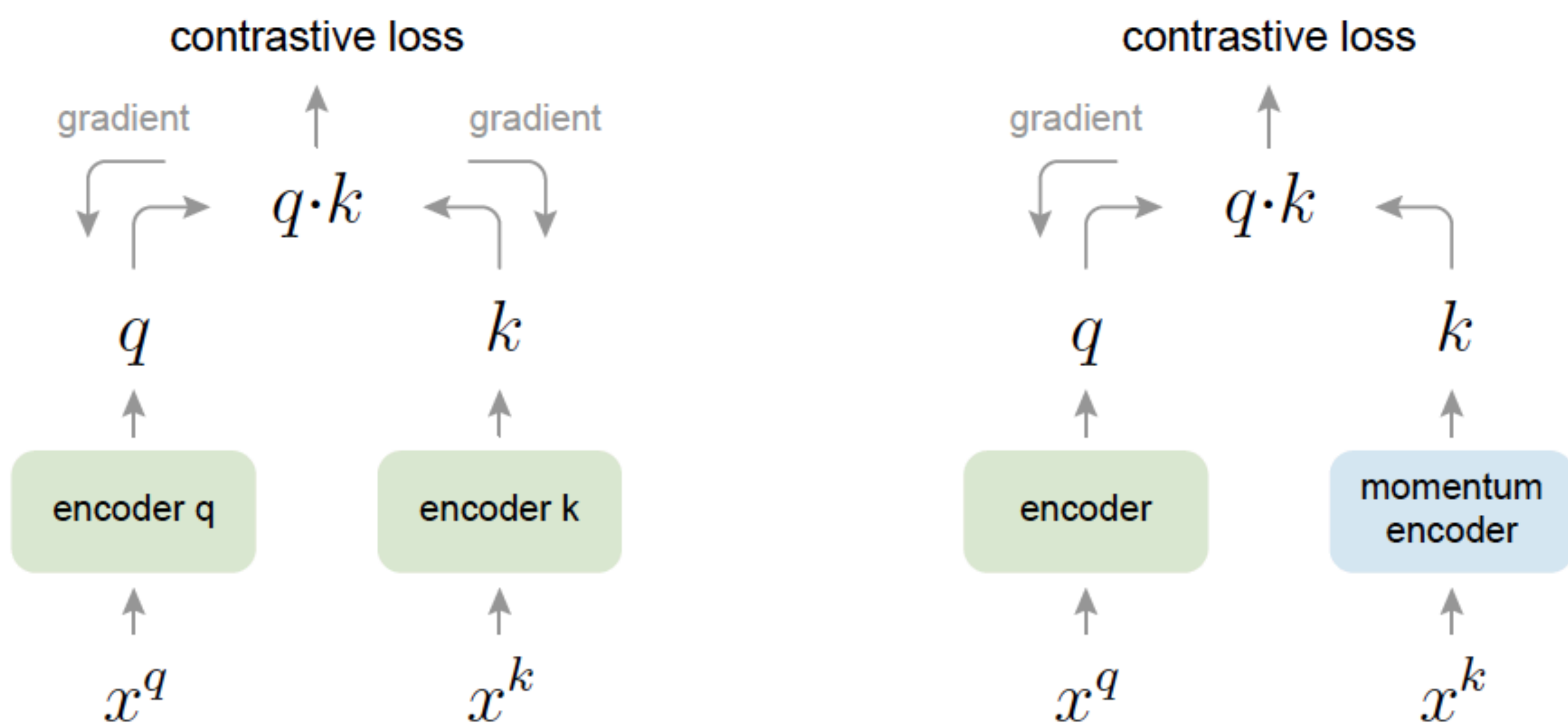
➤ SimCLR



[1] A Simple Framework for Contrastive Learning of Visual Representations (ICML20)

BACKGROUND

- MoCo

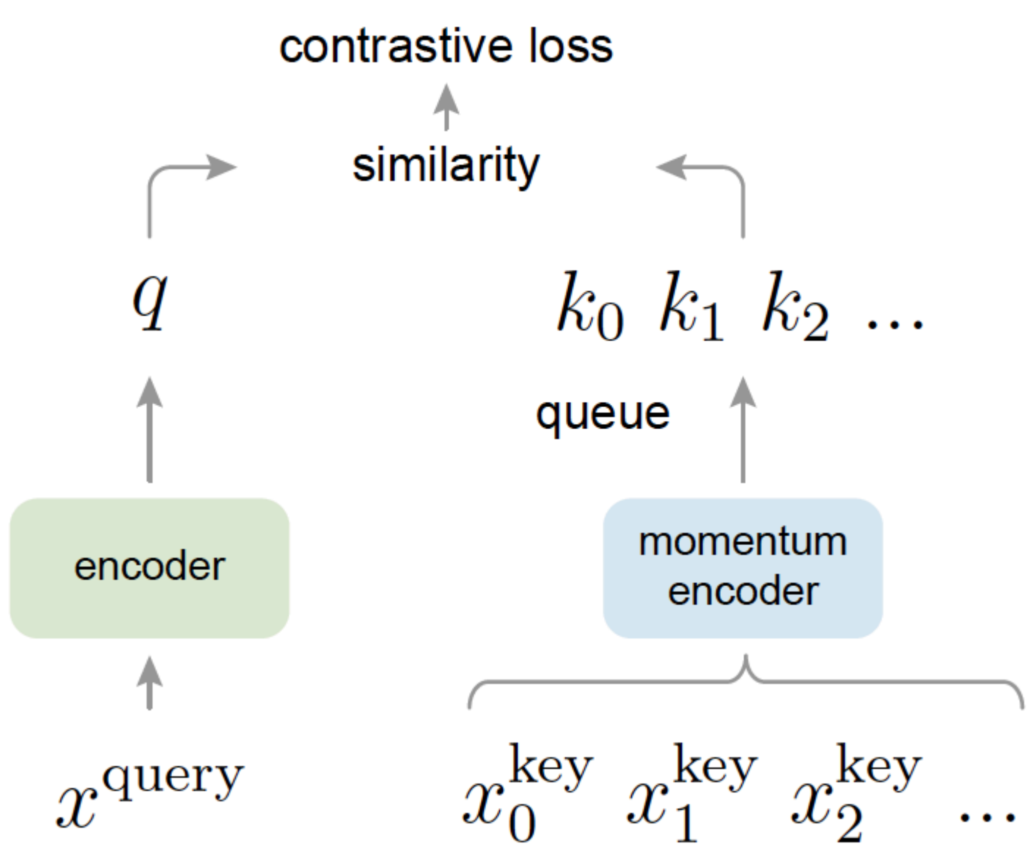


(a) end-to-end (c) MoCo

[1] Momentum Contrast for Unsupervised Visual Representation Learning (CVPR20)

BACKGROUND

➤ MoCo



$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q$$

θ_k : weights of the encoder for negative examples

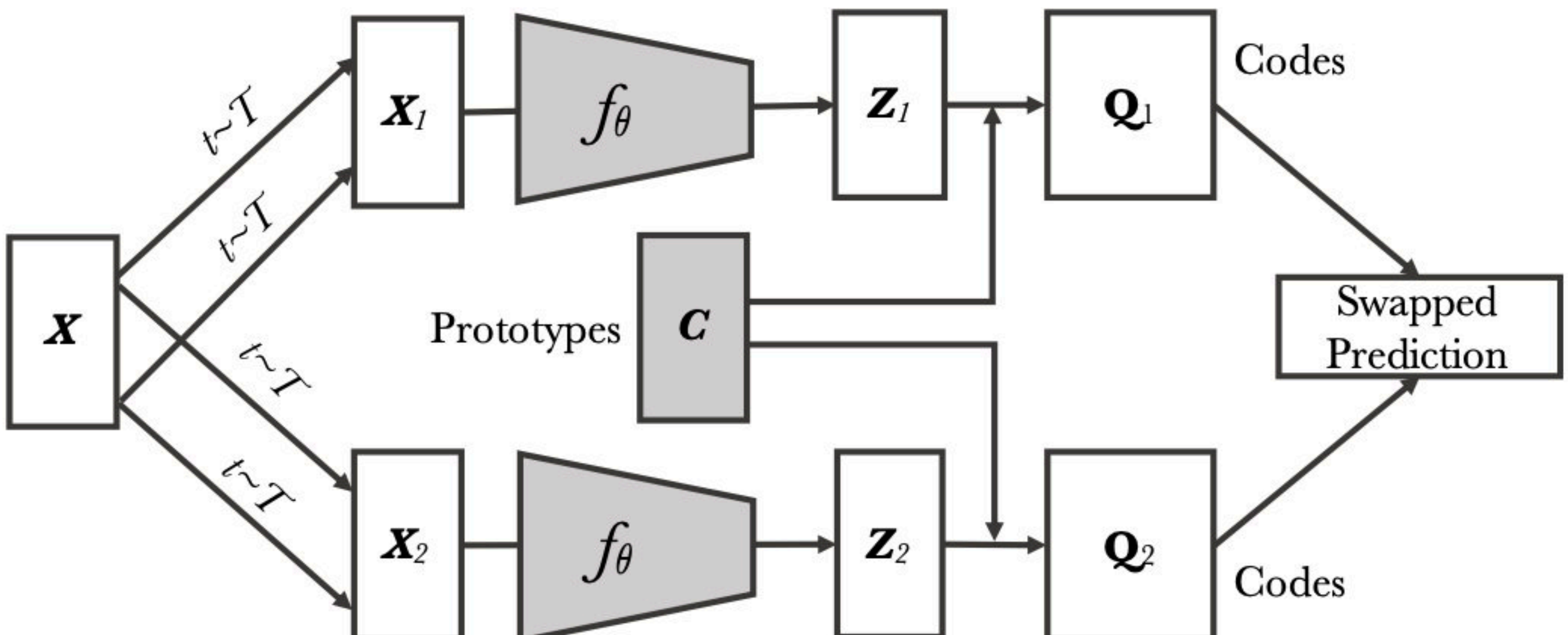
θ_q : weights of the encoder for positive examples

only θ_q update through BP

[1] Momentum Contrast for Unsupervised Visual Representation Learning (CVPR20)

BACKGROUND

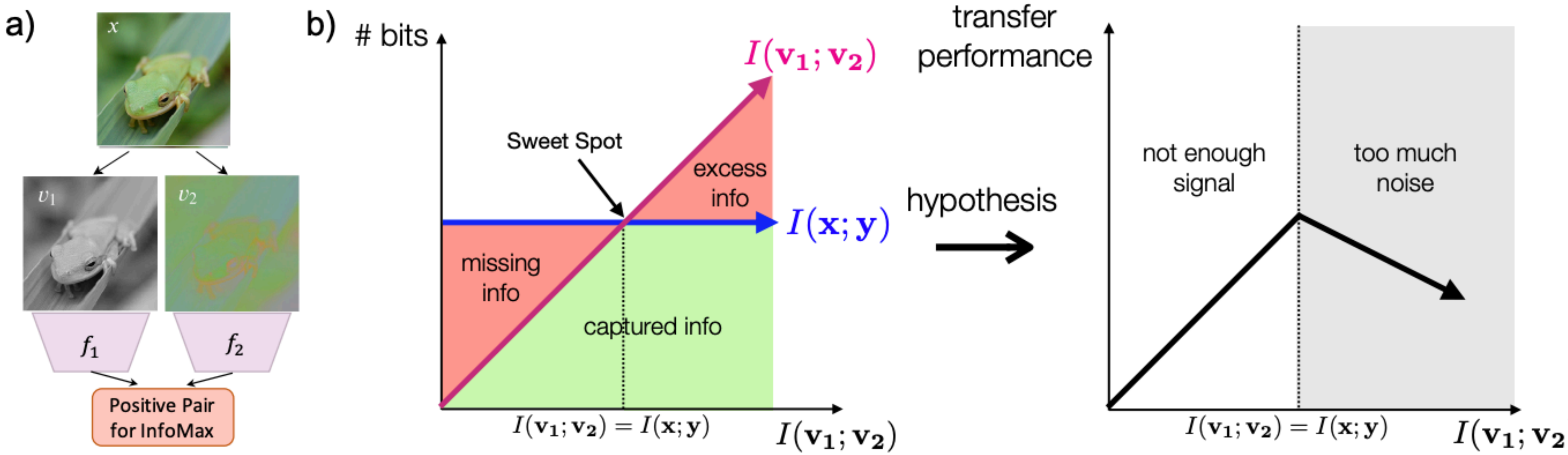
➤ SwAV



[1] Unsupervised Learning of Visual Features by Contrasting Cluster Assignments (NeurIPS20)

BACKGROUND

➤ Infomin



[1] What makes for good views for contrastive learning (NeurIPS20)

BACKGROUND

- Infomin

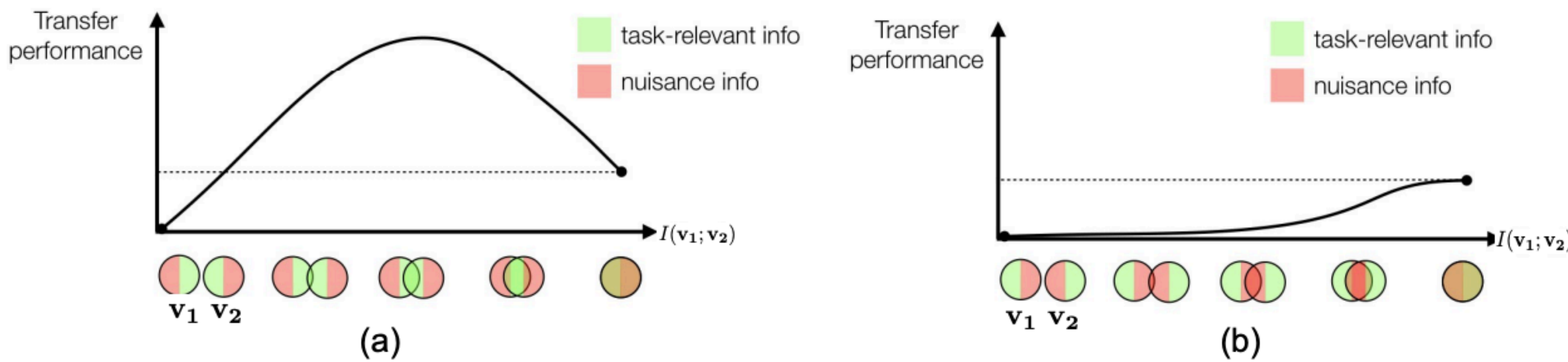
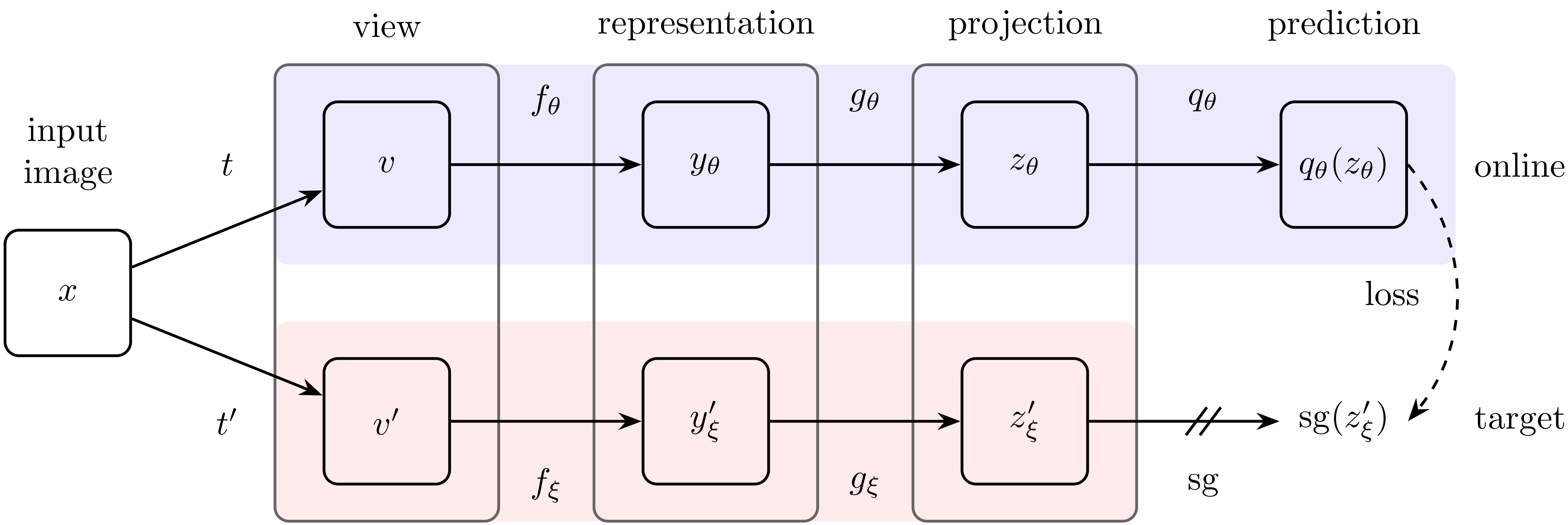


Figure 2: As the mutual information between views is changed, information about the downstream task (green) and nuisance variables (red) can be selectively included or excluded, biasing the learned representation. **(a)** depicts a scenario where views are chosen to preserve downstream task information between views while throwing out nuisance information, while in **(b)** reducing MI always throws out information relevant for the task leading to decreasing performance as MI is reduced.

[1] What makes for good views for contrastive learning (NeurIPS20)

BACKGROUND

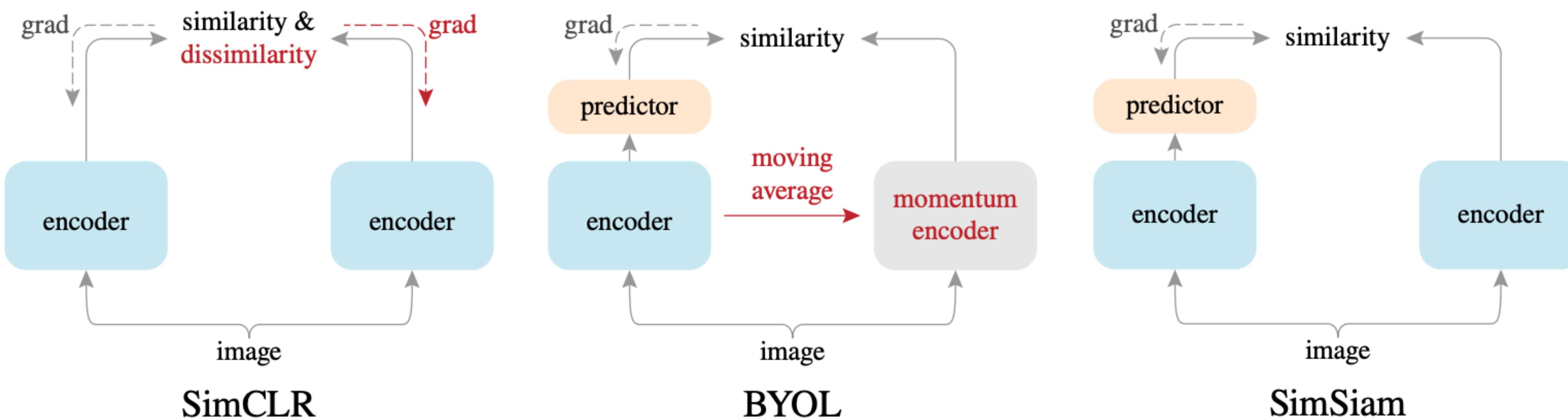
- BYOL



[1] Bootstrap Your Own Latent A New Approach to Self-Supervised Learning (NeurIPS20)

BACKGROUND

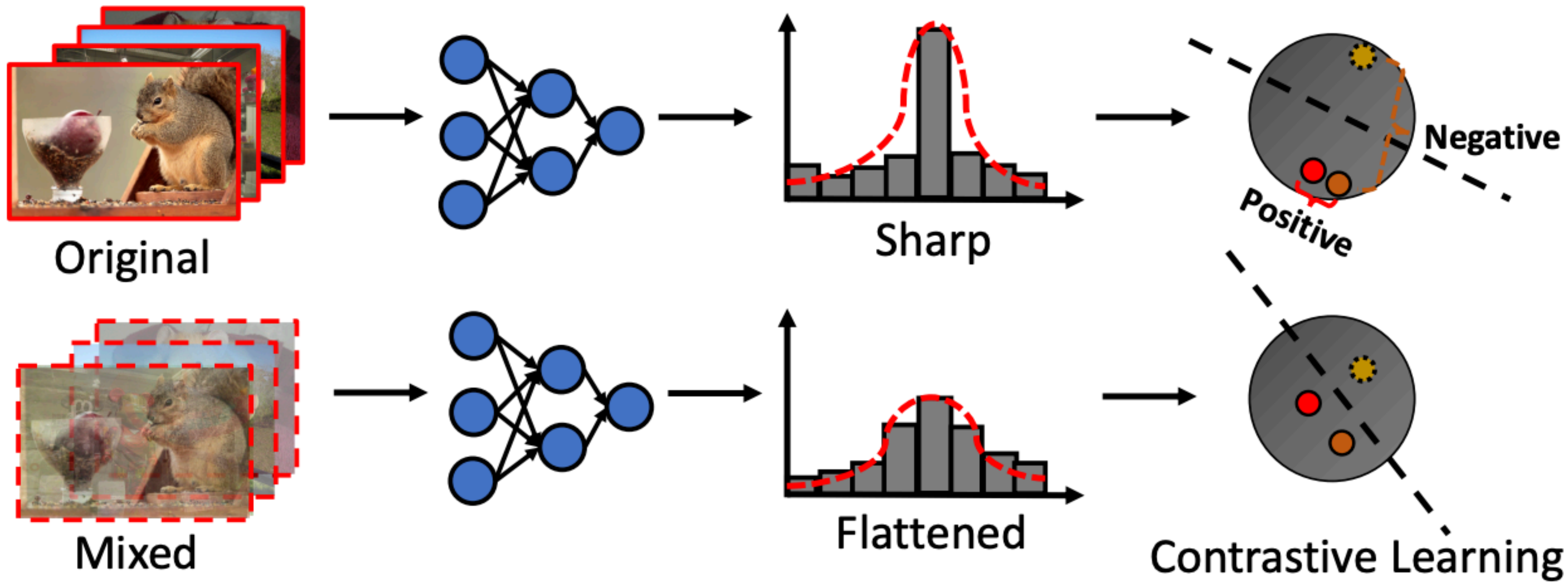
- ➤ SimSiam



[1] Exploring Simple Siamese Representation Learning (CVPR21)

BACKGROUND

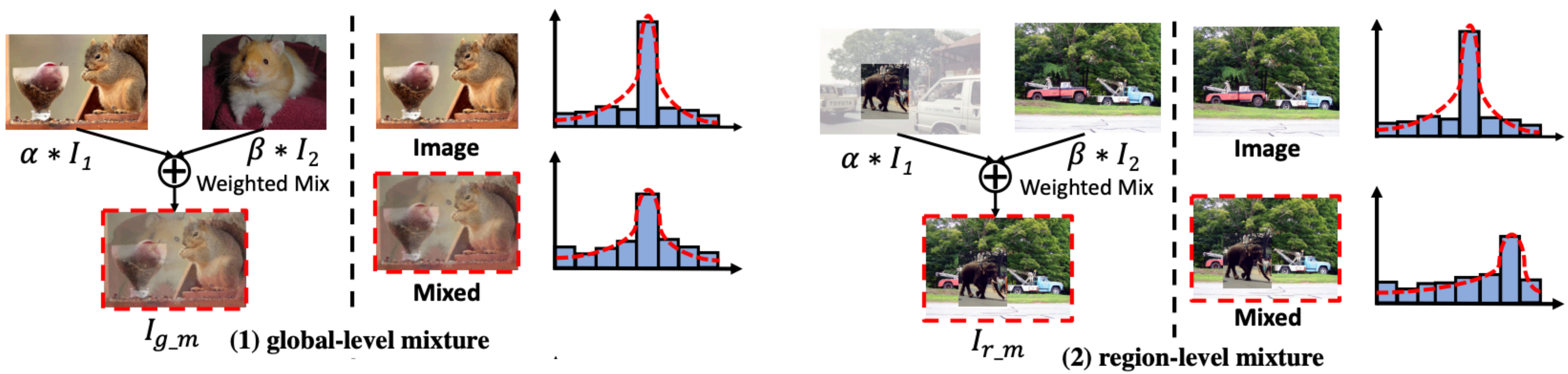
➤ UnMix: Image-level mixing



[1] Exploring Simple Siamese Representation Learning (arXiv 2020)

BACKGROUND

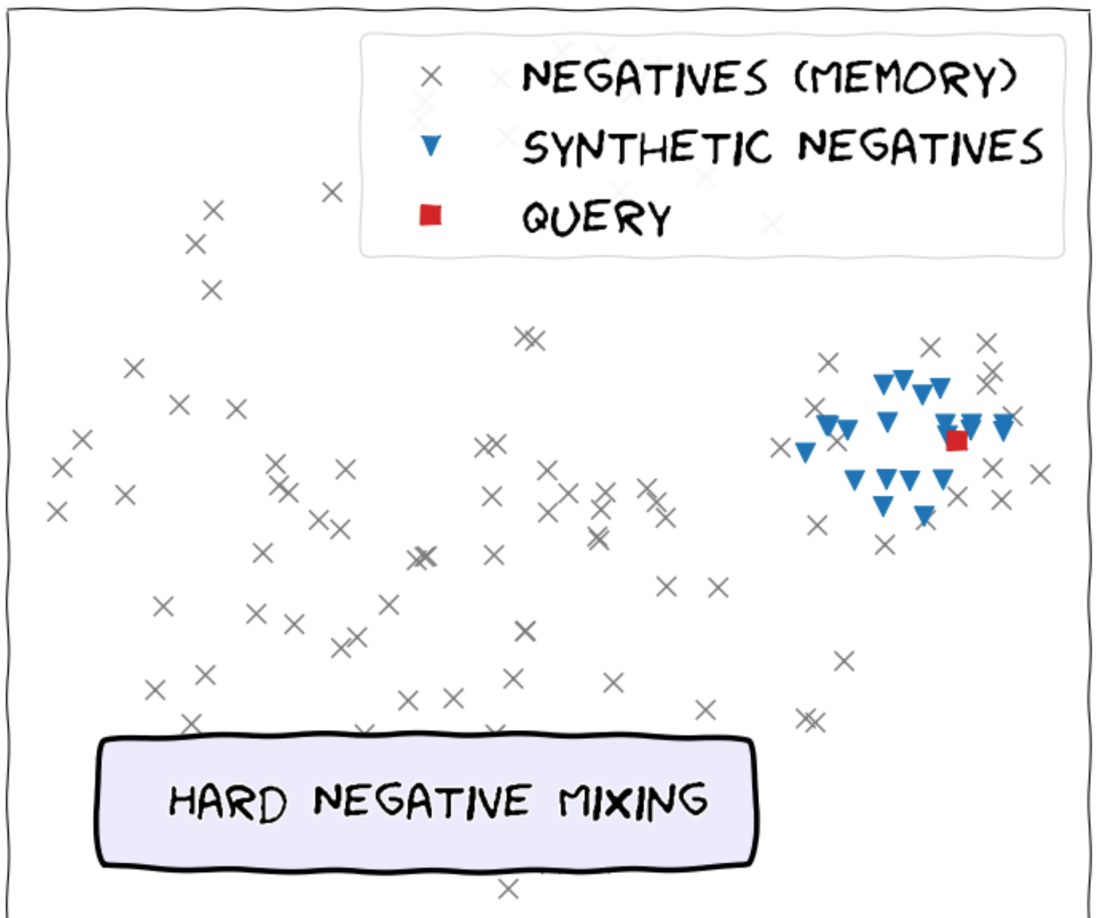
➤ UnMix: Image-level mixing



[1] Exploring Simple Siamese Representation Learning (arXiv 2020)

BACKGROUND

- MoCHi: Feature-level mixing



Creating convex linear combinations of pairs of its "hardest" existing negatives

[1] Hard negative mixing for contrastive learning (NeurIPS20)

OUTLINE

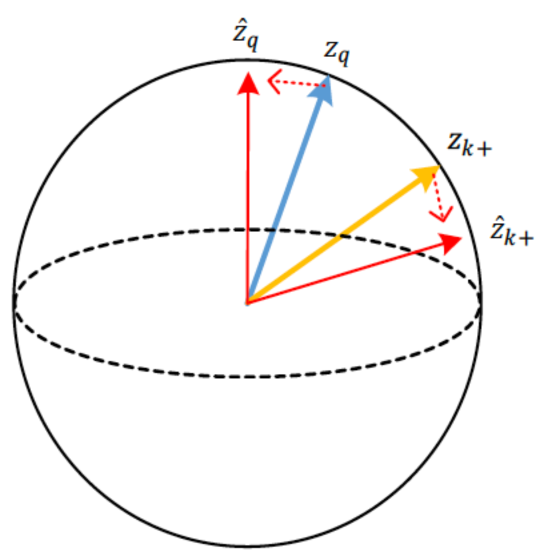
- Authorship
- Background
- Proposed Method
- Experimental Results
- Conclusion

PROPOSED METHOD

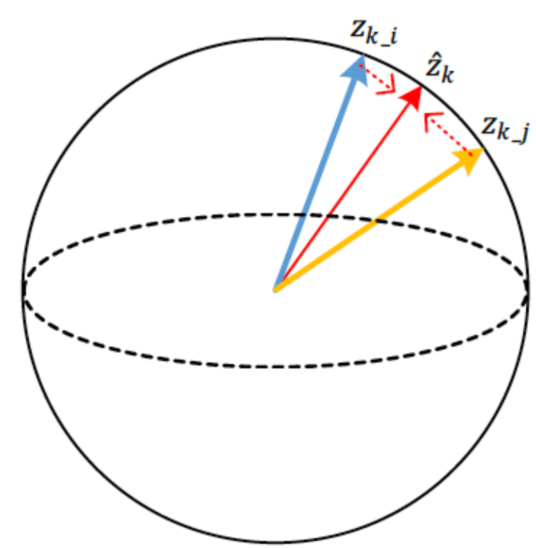
➤ Differing from data augmentation, **feature-level** data manipulation

➤ Proposed strategies:

- Positive extrapolation



- Negative interpolation



PROPOSED METHOD

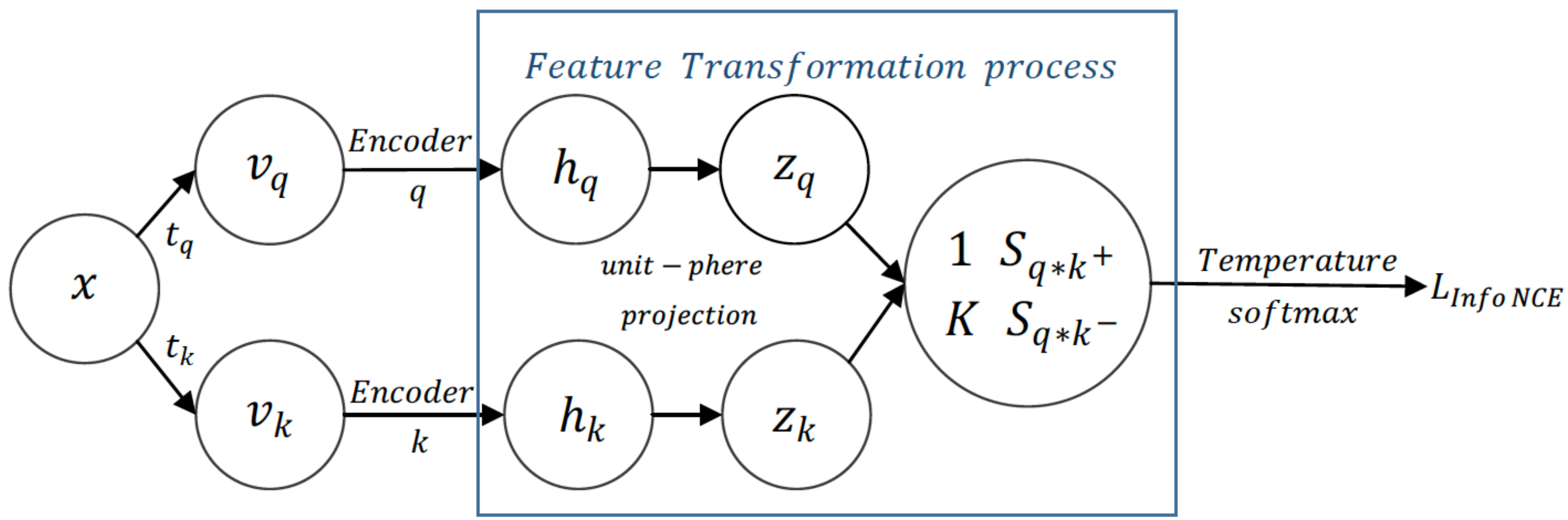
- ➤ Motivation
- ➤ Detailed Method

PROPOSED METHOD

➤ Motivation: visualization of contrastive learning

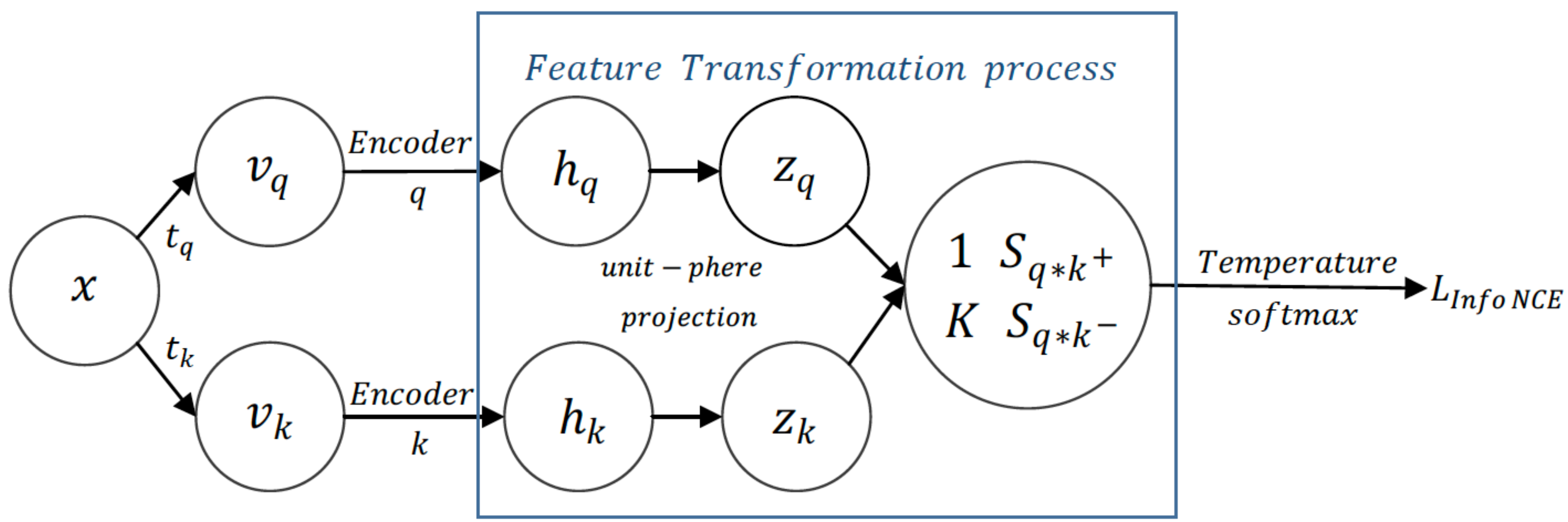
- Preliminaries

$$\mathcal{L} = -\log\left[\frac{\exp\left(S_{q\cdot k^+}/\tau\right)}{\exp\left(S_{q\cdot k^+}/\tau\right) + \sum_K \exp\left(S_{q\cdot k^-}/\tau\right)}\right]$$



PROPOSED METHOD

- ➤ Motivation: visualization of contrastive learning
- Preliminaries
- Cos similarity: one score $S_{q \cdot k^+}$ and K scores $S_{q \cdot k^-}$



PROPOSED METHOD

➤ Motivation: visualization of contrastive learning

- Score distribution visualization
  - Factor: m for momentum

$$\boxed{\theta_{f_k} \leftarrow m\theta_{f_k} + (1-m)\theta_{f_q}}$$

m	≤ 0.5	0.6	0.7	0.8	0.9	0.99	0.999	1
acc (%)	*collapse*	21.2	32.8	39.3	46.5	56.2	53.1	31.2

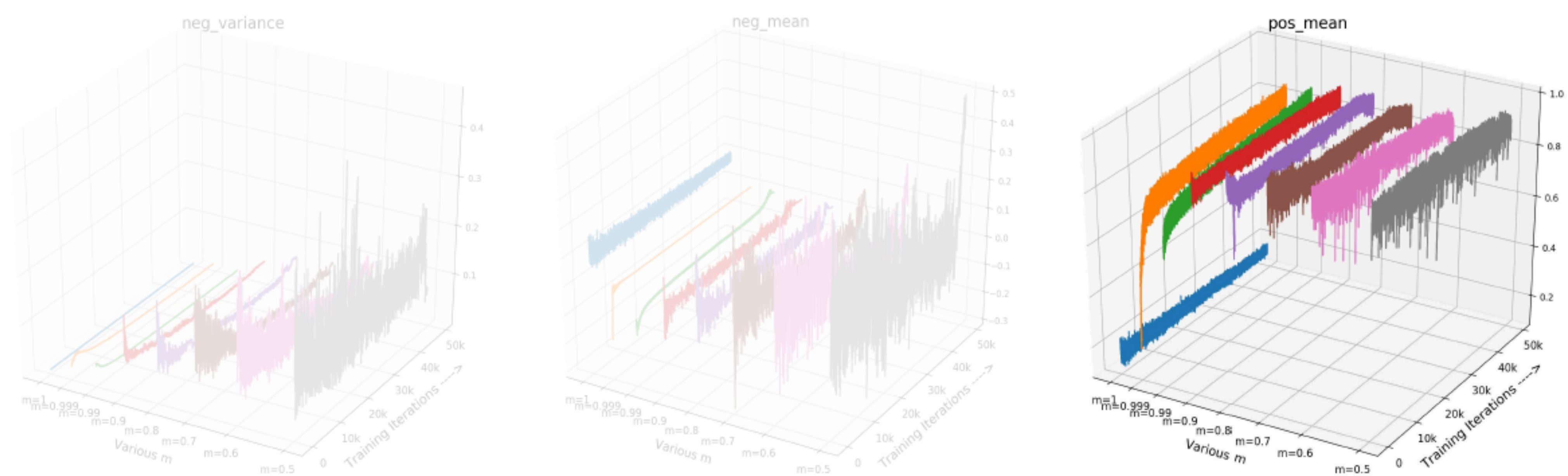
Table 1. The parameter experiments of m on MoCo ($\tau = 0.07$).

  - Target 1: Mean of pos/neg scores (indicating the approximate average of the pos/neg pair distance)
  - Target 2: Variance of negative scores (indicating the fluctuation degree of the negative samples in the memory queue)

PROPOSED METHOD

➤ Motivation: visualization of contrastive learning

- Score distribution visualization



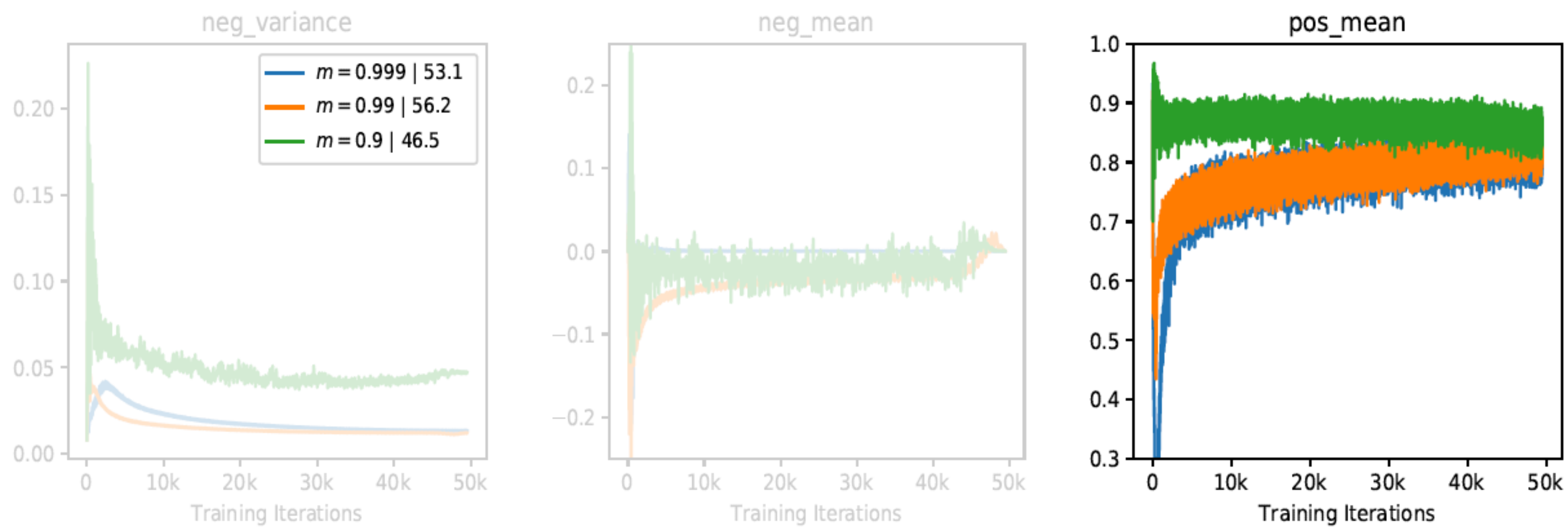
(a) Var of neg scores (b) Mean of neg scores (c) Mean of pos scores

Figure 3. Pos/neg score statistics of various m in MoCo training

PROPOSED METHOD

➤ Motivation: visualization of contrastive learning

- Score distribution visualization



(a) Var of neg scores (b) Mean of neg scores (c) Mean of pos scores

Figure 5. 2D view of pos/neg score statistics of various m

PROPOSED METHOD

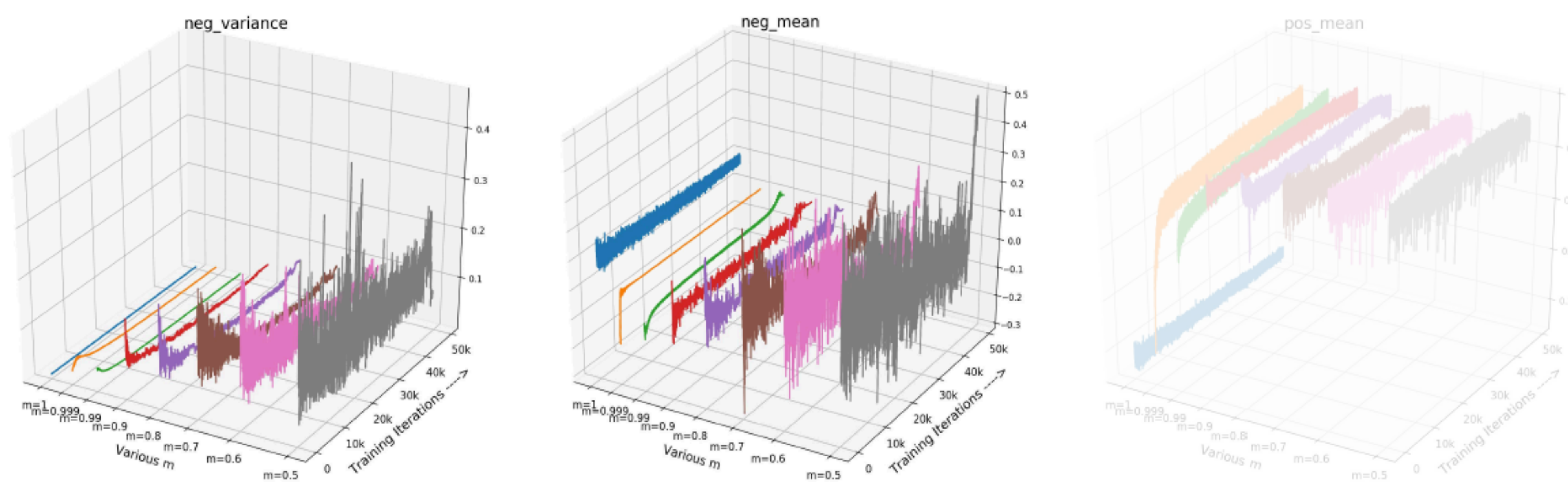
➤ Motivation: visualization of contrastive learning

- Conclusion
  - [Positive] In the guarantee of stable and smooth score distribution and gradient, we can adopt some feature transformation methods which create **hard ones** by decreasing easy positive scores

PROPOSED METHOD

➤ Motivation: visualization of contrastive learning

- Score distribution visualization



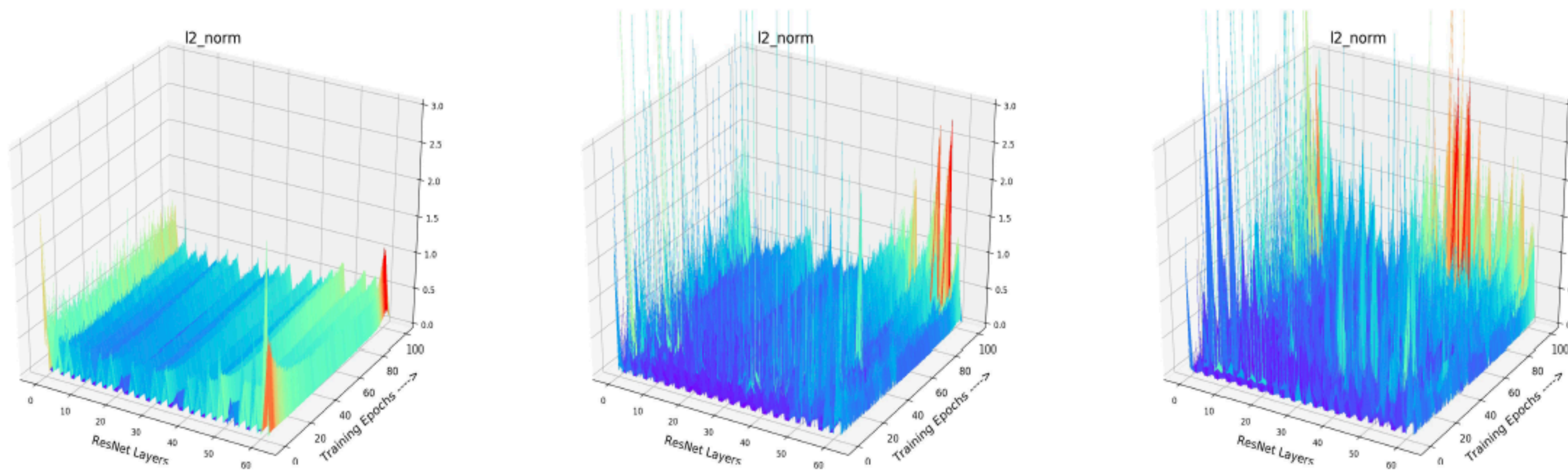
(a) Var of neg scores (b) Mean of neg scores (c) Mean of pos scores

Figure 3. Pos/neg score statistics of various m in MoCo training

PROPOSED METHOD

➤ Motivation: visualization of contrastive learning

- Score distribution visualization



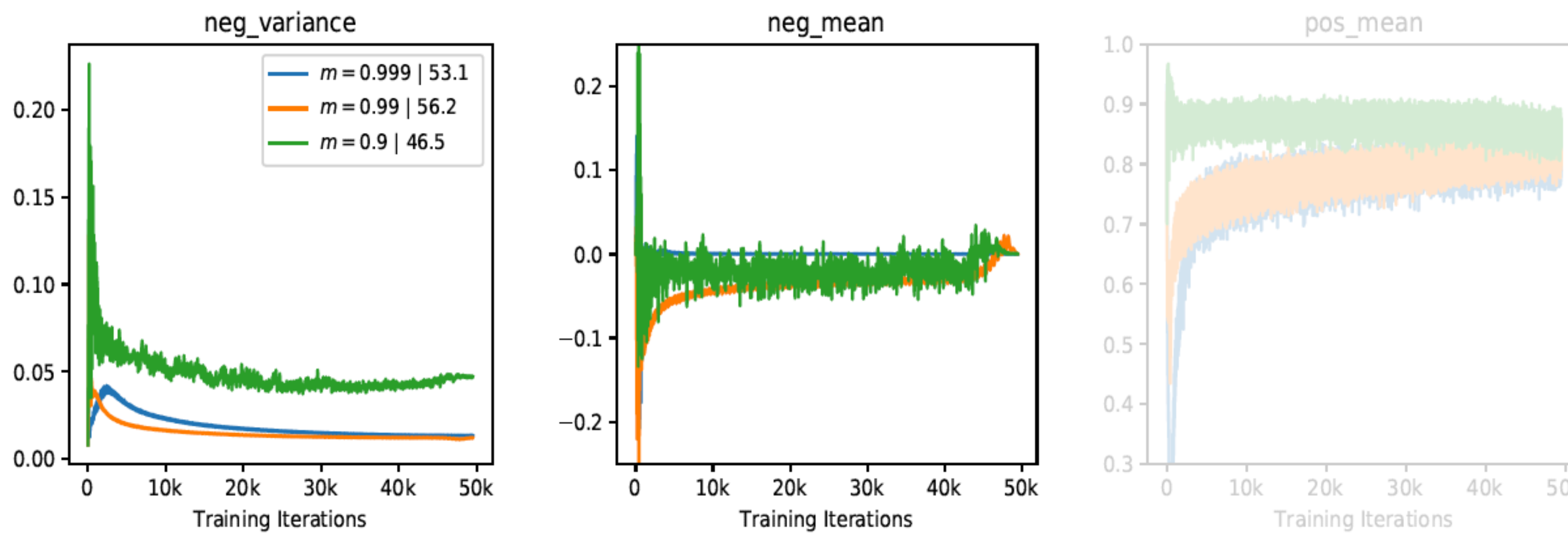
(a) $m = 0.99 \mid 56.2\%$ (b) $m = 0.6 \mid 21.2\%$ (c) $m = 0.5 \mid$ collapse

Figure 4. Gradient (ℓ_2 norm) landscape of various m

PROPOSED METHOD

➤ Motivation: visualization of contrastive learning

- Score distribution visualization



(a) Var of neg scores (b) Mean of neg scores (c) Mean of pos scores

Figure 5. 2D view of pos/neg score statistics of various m

PROPOSED METHOD

- Motivation: visualization of contrastive learning
- Conclusion
  - [Positive] In the guarantee of stable and smooth score distribution and gradient, we can adopt some feature transformation methods which create **hard ones** by decreasing easy positive scores
  - [Negative] We need to prepare negative pairs that can maintain the **stability and smoothness** of score distribution and gradient for the training process

PROPOSED METHOD

- Motivation
- Detailed Method

PROPOSED METHOD

➤ Feature Transformation

- Draw the positive pair z_q and z_{k+} closer
- Pushing away negative pairs z_q and all the z_{k-} in memory queue

- Positive extrapolation: increase the hardness
- Negative interpolation: increase the diversity

PROPOSED METHOD

➤ Positive

- Weighted addition

$$\hat{z}_q = \lambda_{ex} z_q + (1 - \lambda_{ex}) z_{k+}$$
$$\hat{z}_{k+} = \lambda_{ex} z_{k+} + (1 - \lambda_{ex}) z_q$$

- More importantly, we should guarantee than the transformed pos score $\hat{S}_{q \cdot k+}$ is smaller than the original pos score $S_{q \cdot k+}$, namely $\hat{z}_q \hat{z}_{k+} \leq z_q z_{k+}$.

$$\hat{S}_{q \cdot k+} = 2\lambda_{ex}(1 - \lambda_{ex})(1 - S_{q \cdot k+}) + S_{q \cdot k+} \leq S_{q \cdot k+}$$

PROPOSED METHOD

$$\hat{S}_{q\cdot k^+} = 2\lambda_{ex}(1-\lambda_{ex})(1-S_{q\cdot k^+}) + S_{q\cdot k^+} \le S_{q\cdot k^+}$$

Because $S_{q\cdot k^+} \in [-1, 1]$ and thus $(1 - S_{q\cdot k^+}) \ge 0$. To make sure the lower score $\hat{S}_{q\cdot k^+} \le S_{q\cdot k^+}$, we need to set $\lambda_{ex} \ge 1$ to let $2 \cdot \lambda_{ex}(1 - \lambda_{ex}) \le 0$. So we choose $\lambda_{ex} \sim Beta(\alpha_{ex}, \alpha_{ex}) + 1$ [4] is sampled from a beta distribution and then adding 1 results in a range of $(1, 2)$. And the range of transformed pos score will be $\hat{S}_{q\cdot k^+} \in [-4 + 5S_{q\cdot k^+}, S_{q\cdot k^+}]$.

PROPOSED METHOD

$$\hat{S}_{q\cdot k^+} = 2\lambda_{ex}(1-\lambda_{ex})(1 -$$

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,du}$$
$$= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$
$$= \frac{1}{\mathrm{B}(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

Because $S_{q\cdot k^+} \in [-1,1]$ a

To make sure the lower score $\hat{S}_{q\cdot k^+} \leq S_{q\cdot k^+}$, we need to set $\lambda_{ex} \geq 1$ to let $2\cdot\lambda_{ex}(1-\lambda_{ex}) \leq 0$. So we choose $\lambda_{ex} \sim Beta(\alpha_{ex},\alpha_{ex}) + 1$ [4] is sampled from a beta distribution and then adding 1 results in a range of $(1,2)$. And the range of transformed pos score will be $\hat{S}_{q\cdot k^+} \in [-4+5S_{q\cdot k^+}, S_{q\cdot k^+}]$.

PROPOSED METHOD

$$\hat{S}_{q \cdot k^+} = 2\lambda_{ex}(1 - \lambda_{ex})(1 - S_{q \cdot k^+})$$

Because $S_{q \cdot k^+} \in [-1, 1]$ and thus

To make sure the lower score $\hat{S}_{q \cdot k^+} \leq S_{q \cdot k^+}$, we need to set $\lambda_{ex} \geq 1$ to let $2 \cdot \lambda_{ex}(1 - \lambda_{ex}) \leq 0$. So we choose $\lambda_{ex} \sim Beta(\alpha_{ex}, \alpha_{ex}) + 1$ [4] is sampled from a beta distribution and then adding 1 results in a range of $(1, 2)$. And the range of transformed pos score will be $\hat{S}_{q \cdot k^+} \in [-4 + 5S_{q \cdot k^+}, S_{q \cdot k^+}]$.

PROPOSED METHOD

$$\hat{S}_{q\cdot k^+} = 2\lambda_{ex}(1-\lambda_{ex})(1-S_{q\cdot k^+})$$

Because $S_{q\cdot k^+} \in [-1, 1]$ and thus

To make sure the lower score $\hat{S}_{q\cdot k^+} \leq S_{q\cdot k^+}$, we need to set $\lambda_{ex} \geq 1$ to let $2 \cdot \lambda_{ex}(1-\lambda_{ex}) \leq 0$. So we choose $\lambda_{ex} \sim Beta(\alpha_{ex}, \alpha_{ex}) + 1$ [4] is sampled from a beta distribution and then adding 1 results in a range of $(1, 2)$. And the range of transformed pos score will be $\hat{S}_{q\cdot k^+} \in [-4 + 5S_{q\cdot k^+}, S_{q\cdot k^+}]$.

PROPOSED METHOD

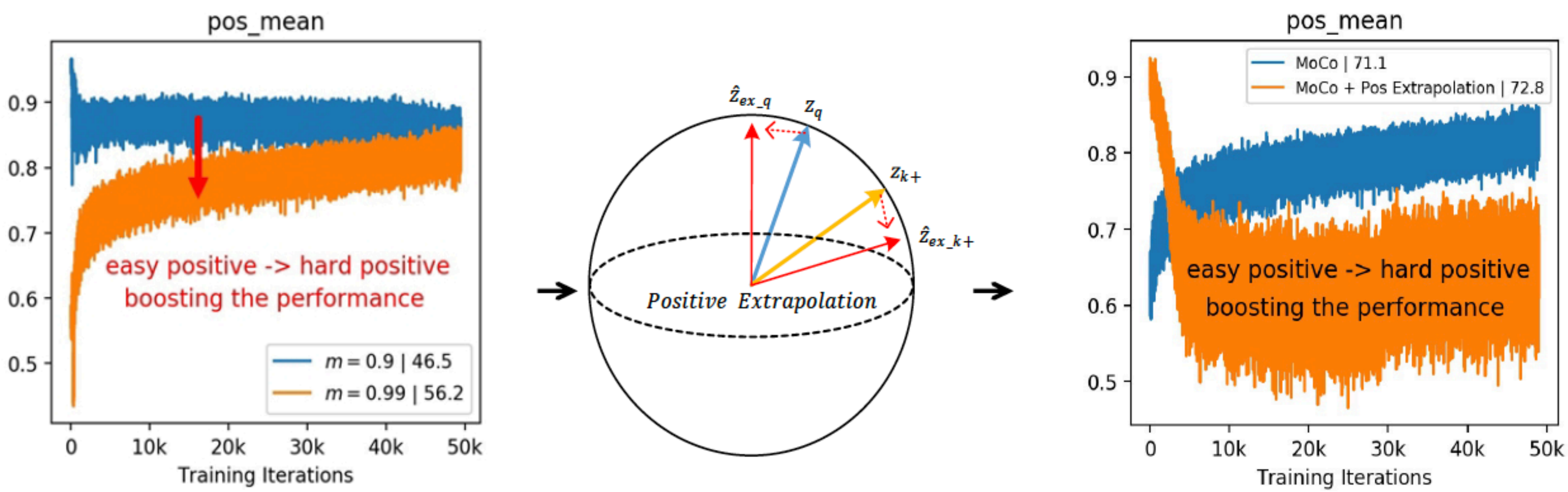
➤ Positive

α_{ex}	-	0.2	0.4	0.6	1.4	1.6	2.0
acc (%)	71.1	71.6	71.8	71.9	72.7	72.4	**72.8**

Table 2. Various α_{ex} for positive extrapolation, the best result is marked in bold. We employ ResNet-50 [16] for the results. '-' indicates MoCo baseline without using extrapolation.

PROPOSED METHOD

➤ Positive



(a) Observation (b) Proposed Method (c) Performance Gain

PROPOSED METHOD

➤ Positive

- Why not extrapolation?

Method	α_{ex}	pos interpolation/extrapolation
MoCo	0.2	69.1 / 71.6
(baseline: 71.1)	2.0	67.4 / 72.8

Table 3. Positive extrapolation v.s. interpolation. Interpolation hurts the performance while extrapolation improves.

PROPOSED METHOD

- Negative

Specifically, we denote the negative memory queue of MoCo as $Z_{neg} = \{z_1, z_2, \ldots, z_K\}$ where K is the size of the memory queue, and Z_{perm} as the random permutation of Z_{neg}. We propose to use a simple interpolation between two memory queue to create a new queue $\hat{Z}_{neg} = \{\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_K\}$:

$$\hat{Z}_{neg} = \lambda_{in} \cdot Z_{neg} + (1 - \lambda_{in}) \cdot Z_{perm} \tag{5}$$

where $\lambda_{in} \sim Beta(\alpha_{in}, \alpha_{in})$ is in the range of $(0, 1)$

PROPOSED METHOD

- Negative

α_{in}	-	0.2	0.4	0.6	1.4	1.6	2.0
acc (%)	71.1	73.3	74.1	74.2	73.5	**74.6**	74.1

Table 4. Various α_{in} for negative interpolation, the best result is marked in bold. We employ ResNet-50 [16] for the results. '-' indicates MoCo baseline without using negative interpolation.

PROPOSED METHOD

- ➤ Discussions
- What if extending memory queue instead of FT
- When to add FT?
- Dimension-level mixing rather than linear mixup?
- Could the gains brought by FT vanish if training longer?

PROPOSED METHOD

➤ Discussions

- What if extending memory queue instead of FT

Method	α_{in}	Z_n	queue size	Acc
moco+ original queue	-	Z_{neg}	K	71.10
moco+ original queue	-	Z_{neg}	$2K$	71.40
moco+ Neg FT queue	1.6	$\hat{Z}_{neg}$	K	74.64
moco+ Neg FT+original	1.6	$\tilde{Z}_{neg}$	$2K$	74.73

Table 5. Ablation results for using different queue of negative features (Res50). The transformed queue $\hat{Z}_{neg}$ can completely replace the extended queue $\tilde{Z}_{neg}$ with small computations.

PROPOSED METHOD

➤ Discussions

- When to add FT?

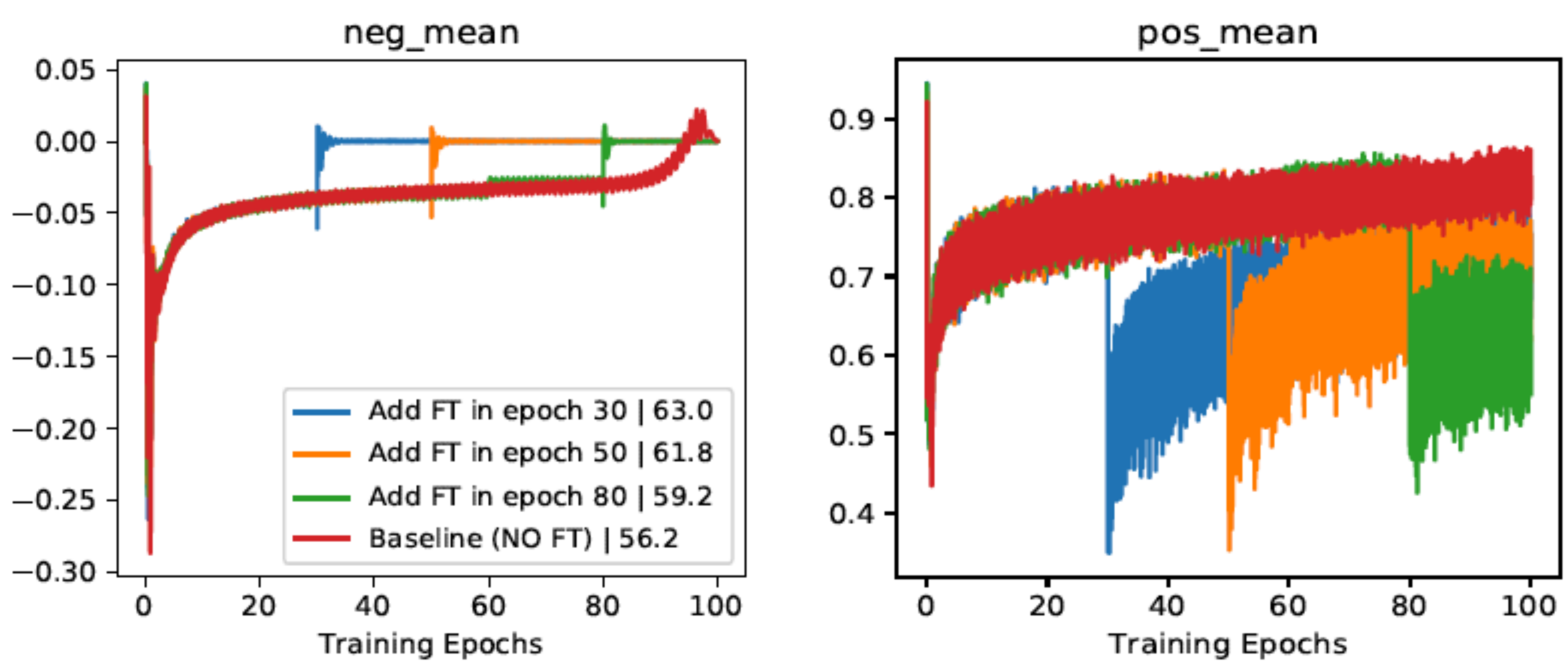
FT begin epoch	0	2	30	50	80	-
Res18 acc (%)	62.6	**63.3**	62.9	61.8	59.2	56.2
Res50 acc (%)	**76.9**	76.4	75.9	74.0	72.2	71.1

Table 6. When to add feature transformation. We employ Res-18 (total 100 epochs) and Res-50 (total 200 epochs) on IN-100 for the results. '-' indicates MoCo baseline without using any FT.

PROPOSED METHOD

➤ Discussions

- When to add FT?



(a) Mean of neg scores (b) Mean of pos scores

PROPOSED METHOD

➤ Discussions

- Dimension-level mixing rather than linear mixup?

$$\hat{z}_{new} = \lambda \odot z_i + (1-\lambda) \odot z_j$$

where $\odot$ stands for Hadamard product, and $\lambda \in [0,1]^n$ is a vector with the same dimension as the feature vector.

PROPOSED METHOD

➤ Discussions

- Could the gains brought by FT vanish if training longer?

- 200 epoch: 75.6% $\rightarrow$ 78.3%
- 500 epoch: 80.7% $\rightarrow$ 81.5%

- Longer training minimizes the improvement over the baseline

OUTLINE

- Authorship
- Background
- Proposed Method
- Experimental Results
- Conclusion

EXPERIMENTAL RESULTS

➤ ImageNet-100

Method	MoCov1	MoCov2	simCLR	Infomin	swav	SimSiam
baseline*	71.10	75.61	74.32	81.9	82.1	77.1
+pos FT	72.80	76.22	75.80	-	-	77.8
+neg FT	74.64	77.12	76.71	-	-	
+both	76.87	78.33	78.25	83.2	83.2	
+both$_{dim}$	**77.21**	**79.21**	**78.81**	-	-	

Table 7. Ablation studies of proposed methods on various contrastive models. The models are pre-trained for 200 epochs with Res50 on IN-100. * indicates reproduced baseline results.

EXPERIMENTAL RESULTS

- ImageNet-1000 and Fine-Grained Classification

pre-train	IN-1k	inat-18	CUB200	FGVC-aircraft
supervised	76.1	66.1	81.9*	82.6*
mocov1[14]	60.6	65.6	82.8*	83.5*
mocov1+ours	61.9	67.3	83.2	84.0
mocov2[7]	67.5	66.8*	82.9*	83.6*
mocov2+ours	**69.6**	**67.7**	**83.1**	**84.1**
mocov2+MoCHi[20]	68.0	-	-	-
mocov2+UnMix[38]	68.6	-	-	-

Table 8. Classification results. * indicates our reproduced results.

EXPERIMENTAL RESULTS

➤ Object Detection

pre-train	IN-1k Top-1	Faster [35] R50-C4 VOC AP	AP_{50}	AP_{75}	Mask R-CNN [15] R50-C4 COCO AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
random init*	-	33.8	60.2	33.1	26.4	44.0	27.8	29.3	46.9	30.8
supervised*	76.1	53.5	81.3	58.8	38.2	58.2	41.2	33.3	54.7	35.2
infomin*	70.1	57.6	82.7	64.6	39.0	58.5	42.0	34.1	55.2	36.3
mocoV1[14]	60.6	55.9	81.5	62.6	38.5	58.3	41.6	33.6	54.8	35.6
mocoV1+ours	61.9	56.1	82.0	62.0	39.0	58.7	42.1	34.1	55.1	36.0
mocoV2[7]	67.5	57.0	82.4	63.6	39.0	58.6	41.9	34.2	55.4	36.2
mocoV2+ours	**69.6**	**58.1**	**83.3**	65.1	**39.5**	**59.2**	42.1	**34.6**	55.6	36.5
mocoV2+mochi[20]	68.0	57.1	82.7	64.1	39.4	59.0	42.7	34.5	55.7	36.7
DetCo[53]	68.6	57.8	82.6	64.2	39.4	59.2	42.3	34.4	55.7	36.6
InsLoc[55]	-	57.9	82.9	65.3	39.5	59.1	**42.7**	34.5	56.0	36.8

Table 9. Object detection. All model are pre-trained for 200 epochs on ImageNet-1k. * means that the results are followed from respective papers [14, 42]. The COCO results of mocoV2 is from [20]. Our results are reported using the average of 5 runs.

OUTLINE

- Authorship
- Background
- Proposed Method
- Experimental Results
- Conclusion

CONCLUSION

- Feature-level data manipulation
  - Visualization scheme for pos/neg score distribution
  - Extrapolation of positives
  - Interpolation among negatives