Quantization meets Self-supervised Learning

MEGVII 旷视

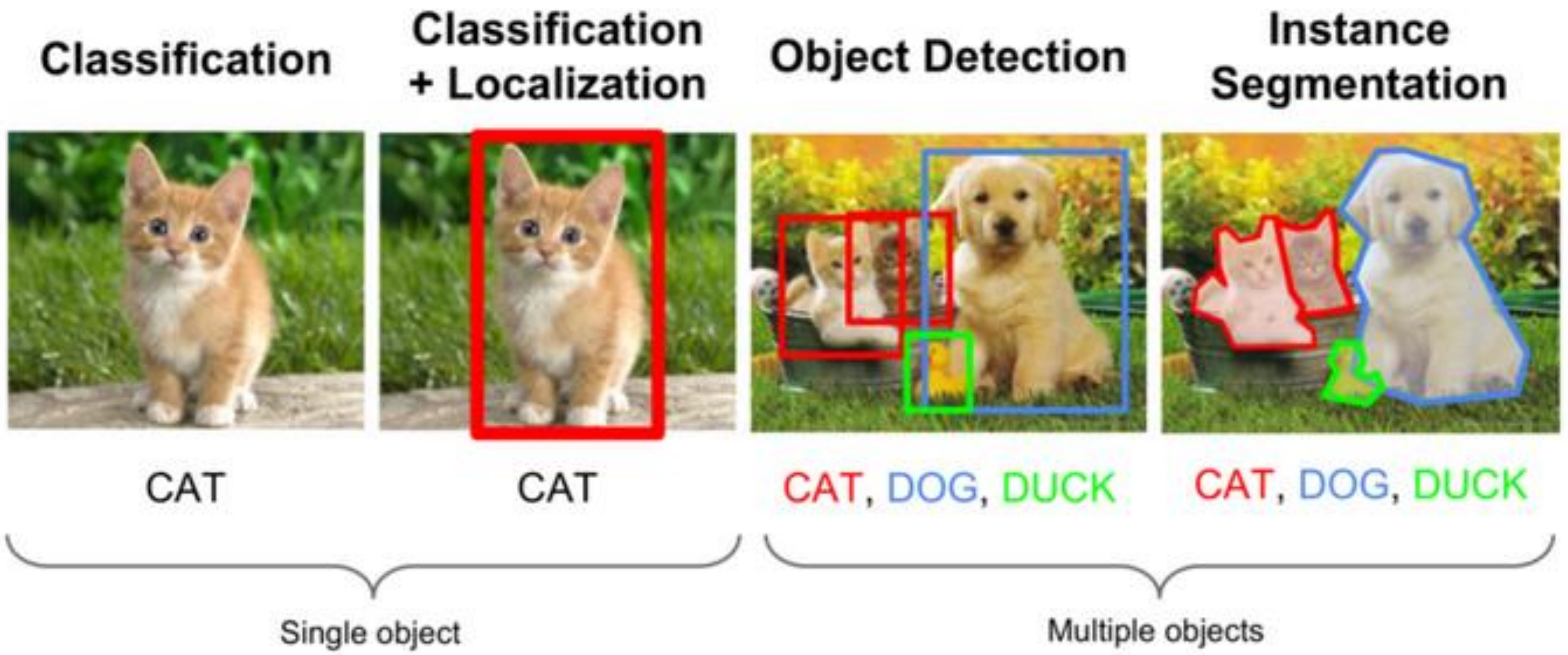
Outline

- **Background**
  - Why need unsupervised learning?
  - What is unsupervised learning?
- **Self-Supervised-Learning (SSL)**
  - What is self-supervised learning?
  - Some pretext tasks
  - Contrastive Learning
- **SSQL** (this course)
  - Synergistic Self-supervised and Quantization Learning
- **Q&A**

Background - Why need unsupervised learning?

MEGVII 旷视

- **Supervised learning**
  - More datas, More intellgence
  - The annotated of vision tasks is different. More complex task, more exspensive, more times

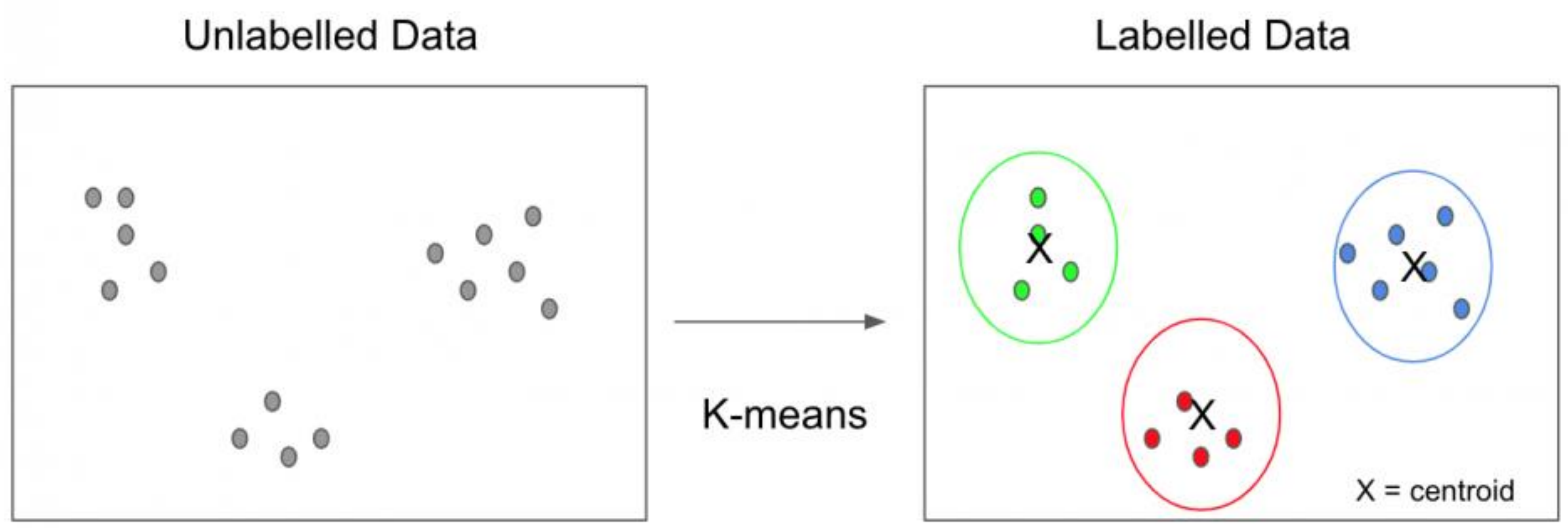


Can AI learns from data without labels? **Yes !!!**

Background - What is unsupervised learning?

MEGVII 旷视

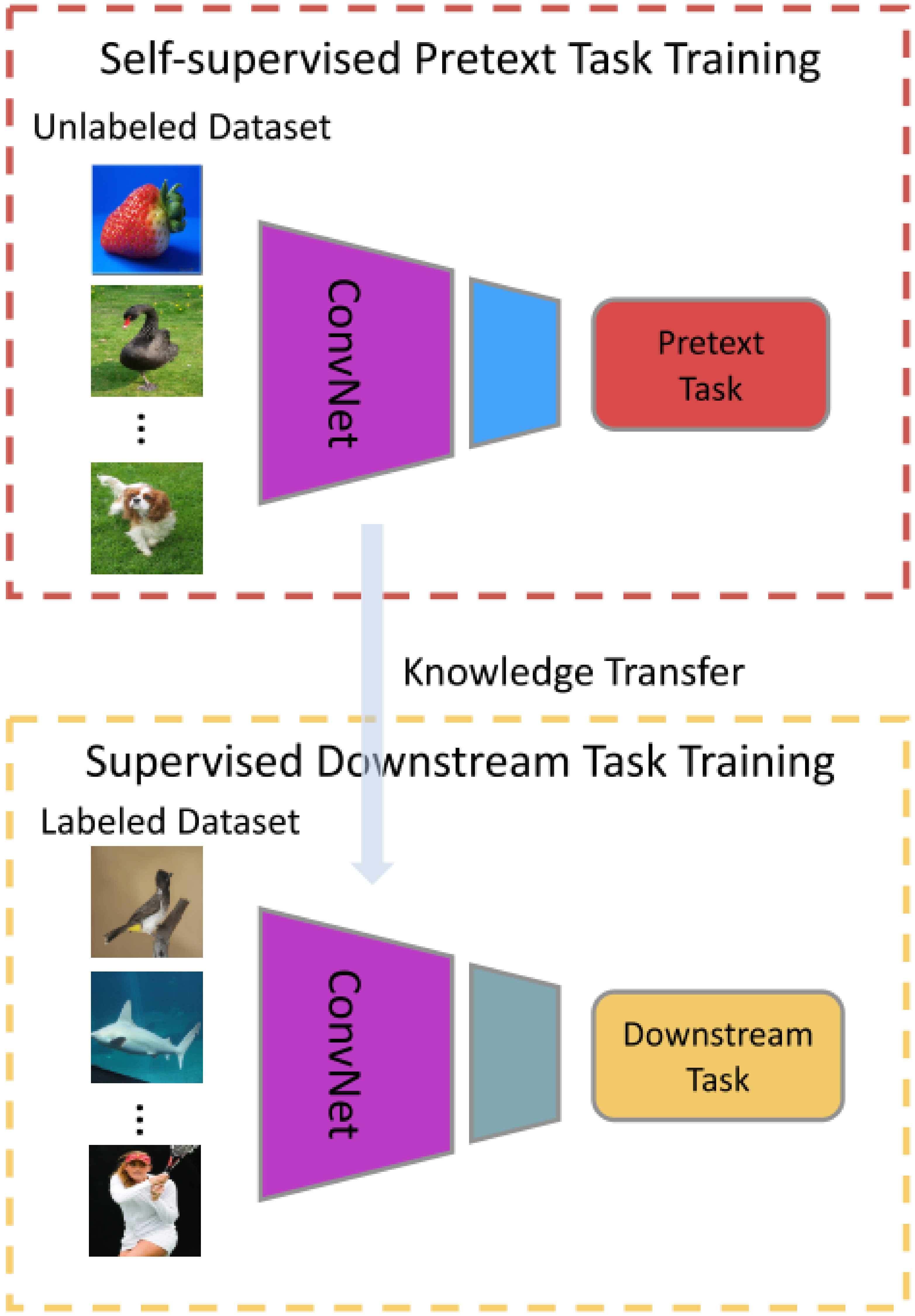
- **Definition:**
  - uses machine learning algorithms to analyze and cluster *unlabeled data sets*
  - discover *hidden patterns* in data without the need for human intervention



- **Categories:**
  - Generative models: reconstruct the distribution of data as faithfully as possible, i.e. Variational Auto-Encoder(VAE), Generative Adversarial networks(GAN).
  - Self-Supervised Learning: exploits internal structures of data and formulates pretext tasks to train a model, i.e. Masked Language Model, Contrastive Learning

Self-Supervised-Learning

- **Definition**
  - a subset of unsupervised learning because it learns from unlabeled sample data.
  - *an intermediate form* between supervised and unsupervised learning
  - A pretrained model trained from a pseudo-label based pretext, then fine-tuning on a downstream dataset with labels



L. Jing, Y. Tian, Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey

The Future: Self-Supervised-Learning

MEGVII 旷视

Y. LeCun

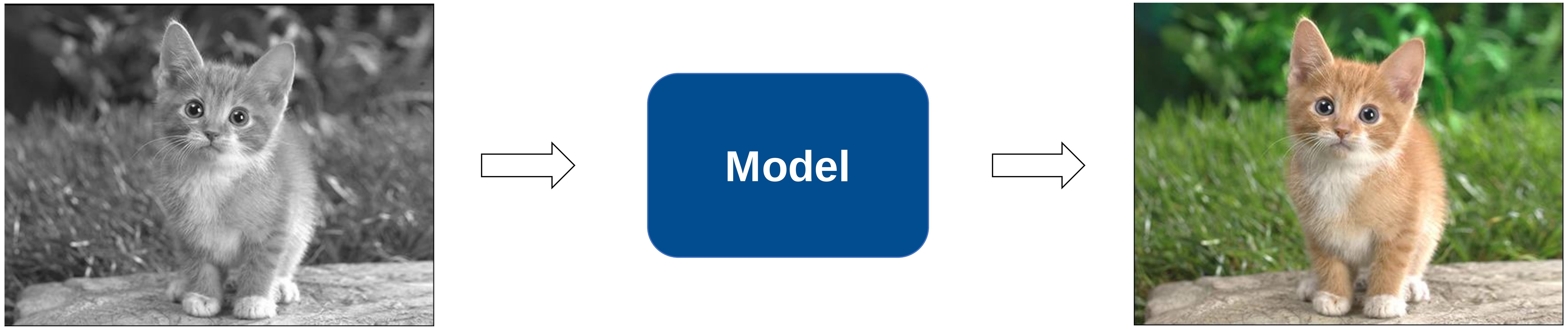
How Much Information is the Machine Given during Learning?

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

https://drive.google.com/file/d/1nHmGL6lRsjZdRRAT_VxLGYpnVV6HHSHc/view?usp=sharing

SSL – Some Pretext Tasks

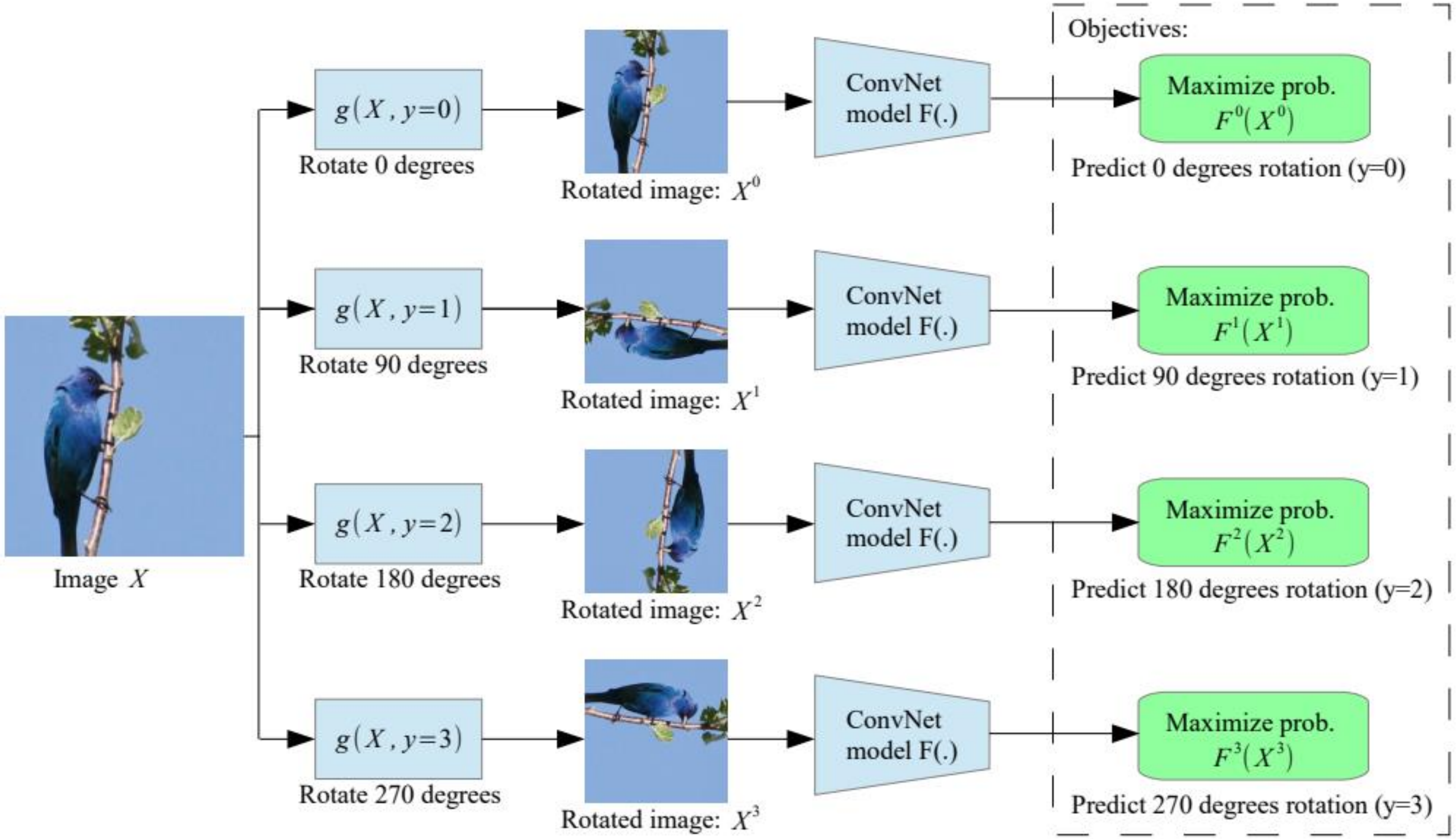
- **Introduction**
  - The term "pretext" implies that the task being solved is not of genuine interest, but is solved only for the true purpose of learning a good data presentation.
  - These pseudo labels are generated automatically based on the attributes found in the data
- **Colorization**
  - Given a grayscale photograph as input, produces a plausible colorization that could potentially fool a human observer



R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In ECCV, 2016.

SSL – Some Pretext Tasks

- **Geometric Transformer -- Rotation**
  - Learn image features by training a model to recognize the 2d rotation that is applied to the image that it gets as input.



S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In ICLR, 2018.

SSL - Some Pretext Tasks

- **Context-based -- Jigsaw puzzle**
  - Studies in psychonomic show that jigsaw puzzles can be used to assess visuospatial processing in humans.

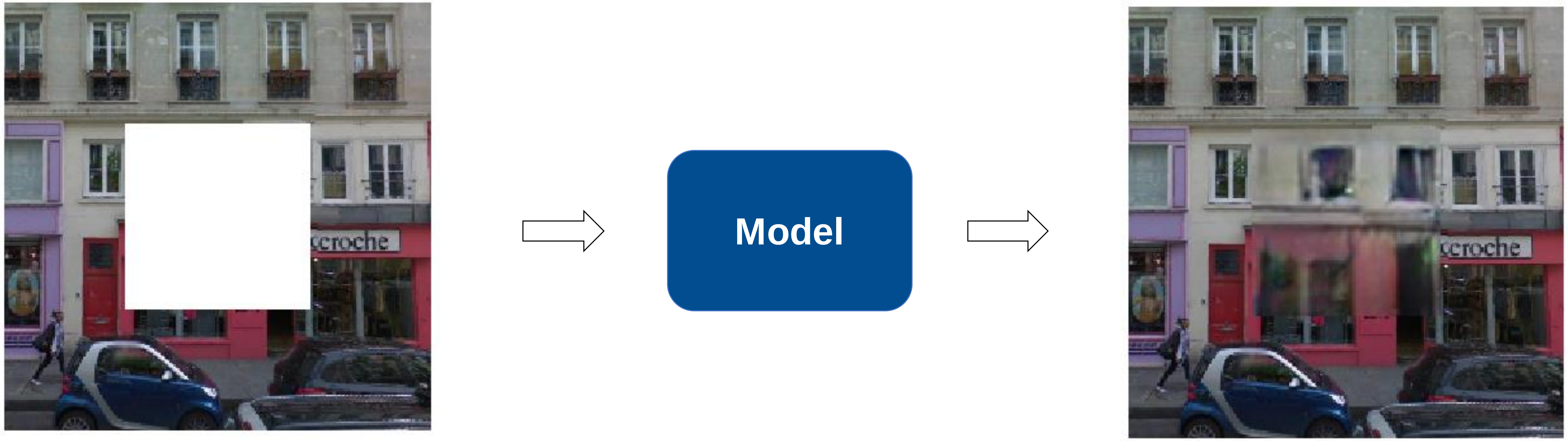
(a) (b) (c)

Fig. 1: Learning image representations by solving Jigsaw puzzles. (a) The image from which the tiles (marked with green lines) are extracted. (b) A puzzle obtained by shuffling the tiles. Some tiles might be directly identifiable as object parts, but others are ambiguous (*e.g.*, have similar patterns) and their identification is much more reliable when all tiles are jointly evaluated. In contrast, with reference to (c), determining the relative position between the central tile and the top two tiles from the left can be very challenging [10].

M. Noroozi and P. Favaro. Unsupervised learning of visual representions by solving jigsaw puzzles. In ECCV, 2016.

SSL – Some Pretext Tasks

- **Image inpainting**
  - a neural network trained to generate the contents of an arbitrary image region conditioned on its surroundings



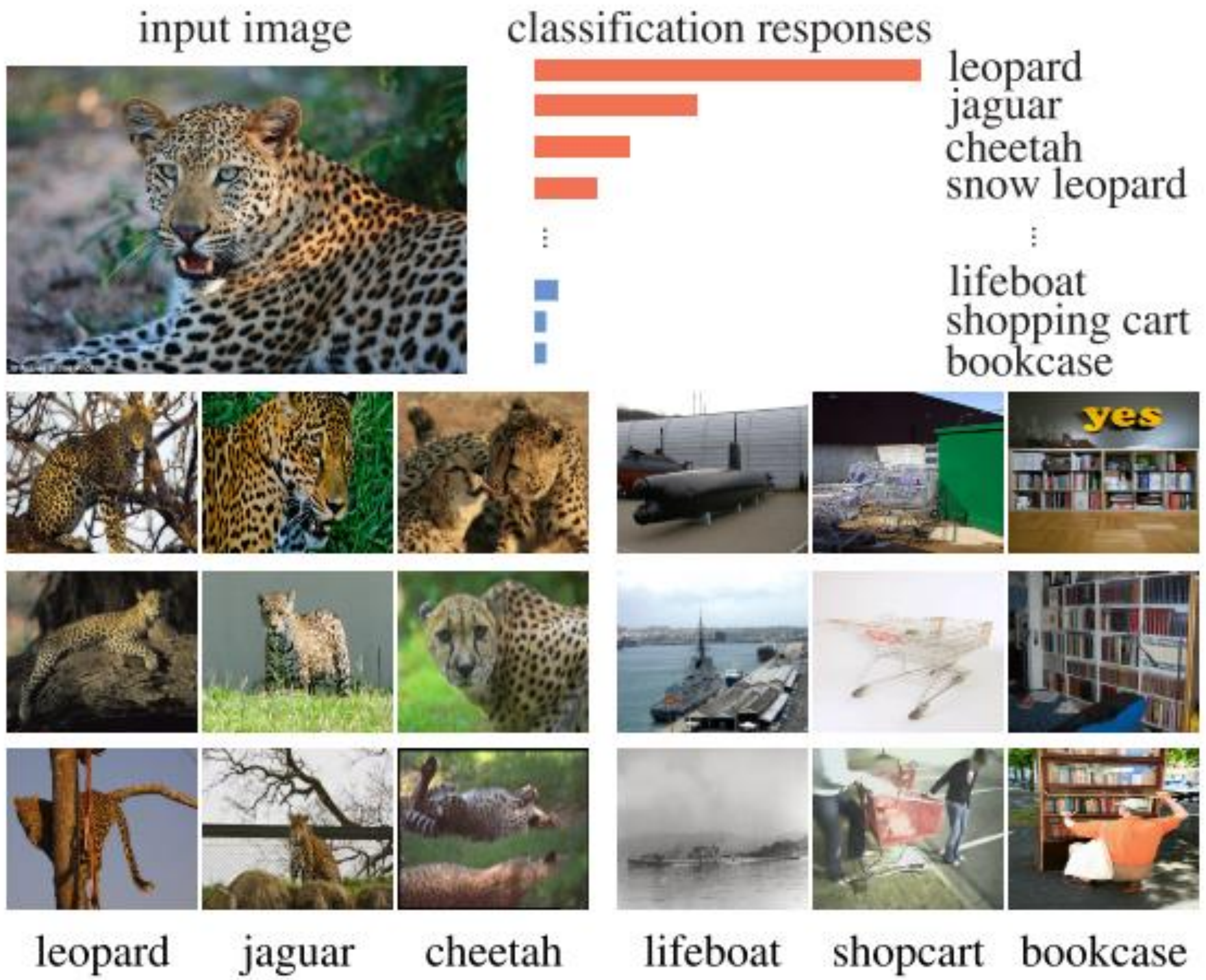
(a) Input context

(d) Context Encoder

D. Pathak, P. Kr¨ahenb¨uhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In CVPR 2016.

SSL – Some Pretext Tasks

- **Instance Discrimination**
  - A typical discriminative learning method can automatically discover apparent similarity among semantic categories
  - An image is distinctive in its own right, and each could differ significantly from other images in the same semantic category
  - Learn to discriminate between individual instances, without any notion of semantic categories, a representation that captures apparent similarity among instances



Z. Wu, Y. Xiong, X. Y. Stella, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In CVPR, 2018.

SSL - Some Pretext Tasks

- **Instance Discrimination**
  - The goal of instance discrimination is to learn a embedding function $\mathbf{v} = f_\theta(x)$
  - A good embedding should mapping visually similar images closer to each other and away from dissimilar images
  - The features are store in a memory bank not model weights, and update by momentum

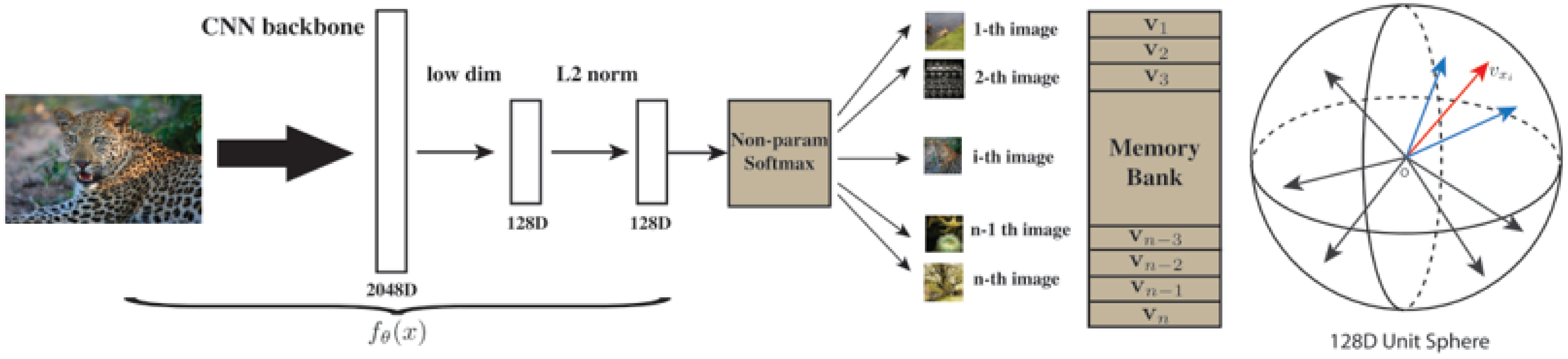


Figure 2: The pipeline of our unsupervised feature learning approach. We use a backbone CNN to encode each image as a feature vector, which is projected to a 128-dimensional space and L2 normalized. The optimal feature embedding is learned via instance-level discrimination, which tries to maximally scatter the features of training samples over the 128-dimensional unit sphere.

$$P(\mathrm{i}|\mathbf{v}) = \frac{\exp(\frac{\mathbf{v}_i^T\mathbf{v}}{\tau})}{\sum_{j=1}^{n}\exp(\frac{\mathbf{v}_j^T\mathbf{v}}{\tau})}$$

$$h(i,\mathbf{v}) = P(D=1|i,\mathbf{v}) = \frac{P(i|\mathbf{v})}{P(i|\mathbf{v}) + mP_n(i)}$$

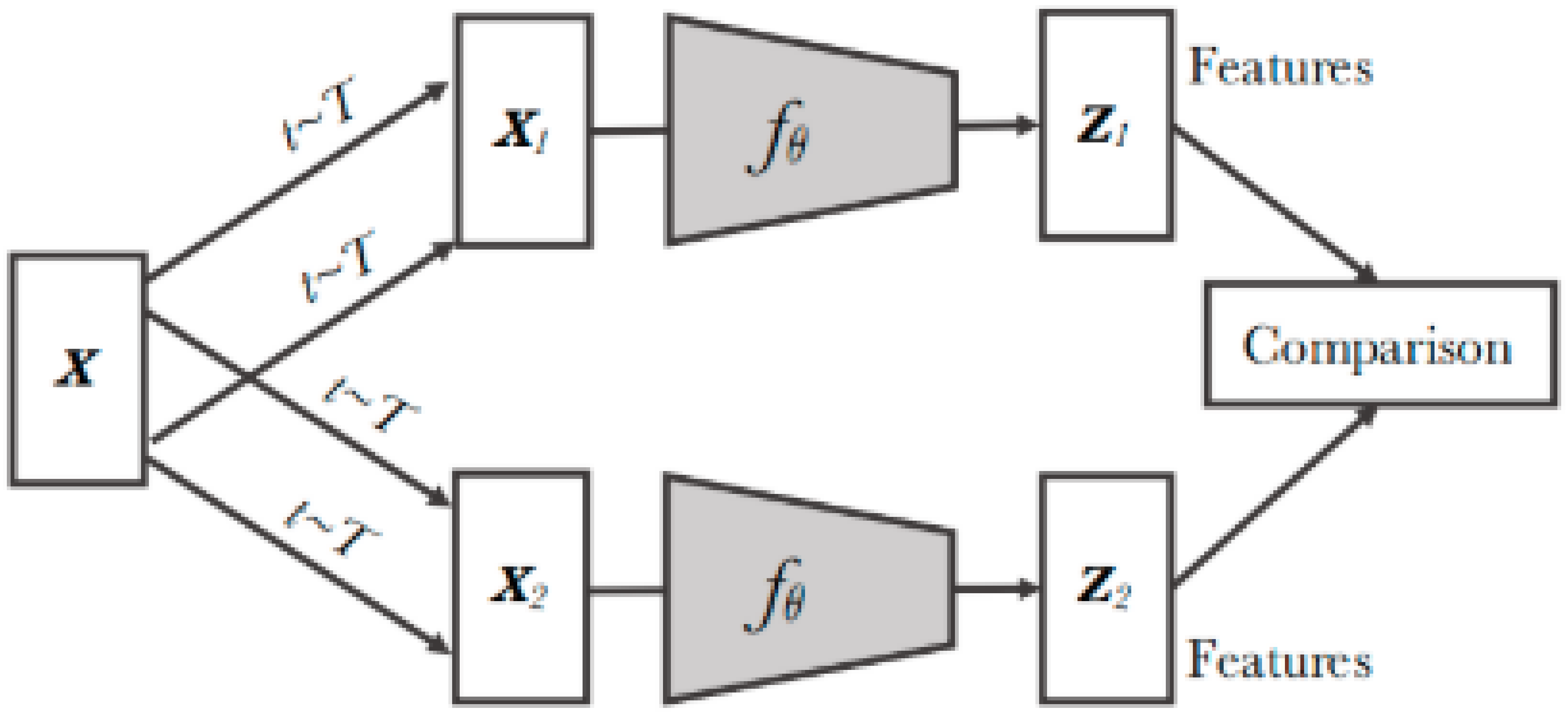
$$Z \simeq Z_i \simeq nE_j[\exp(\mathbf{v}_j^T f_i/\tau)] = \frac{n}{m}\sum_{k=1}^{m}\exp(\mathbf{v}_{jk}^T f_i/\tau)$$

$$J(\boldsymbol{\theta}) = -E_{P_d}[\log h(i,\mathbf{v}_i)] - m * E_{P_n}[\log(1 - h(i,\mathbf{v}'))]$$

Z. Wu, Y. Xiong, X. Y. Stella, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In CVPR, 2018.

Contrastive Learning - Introduction

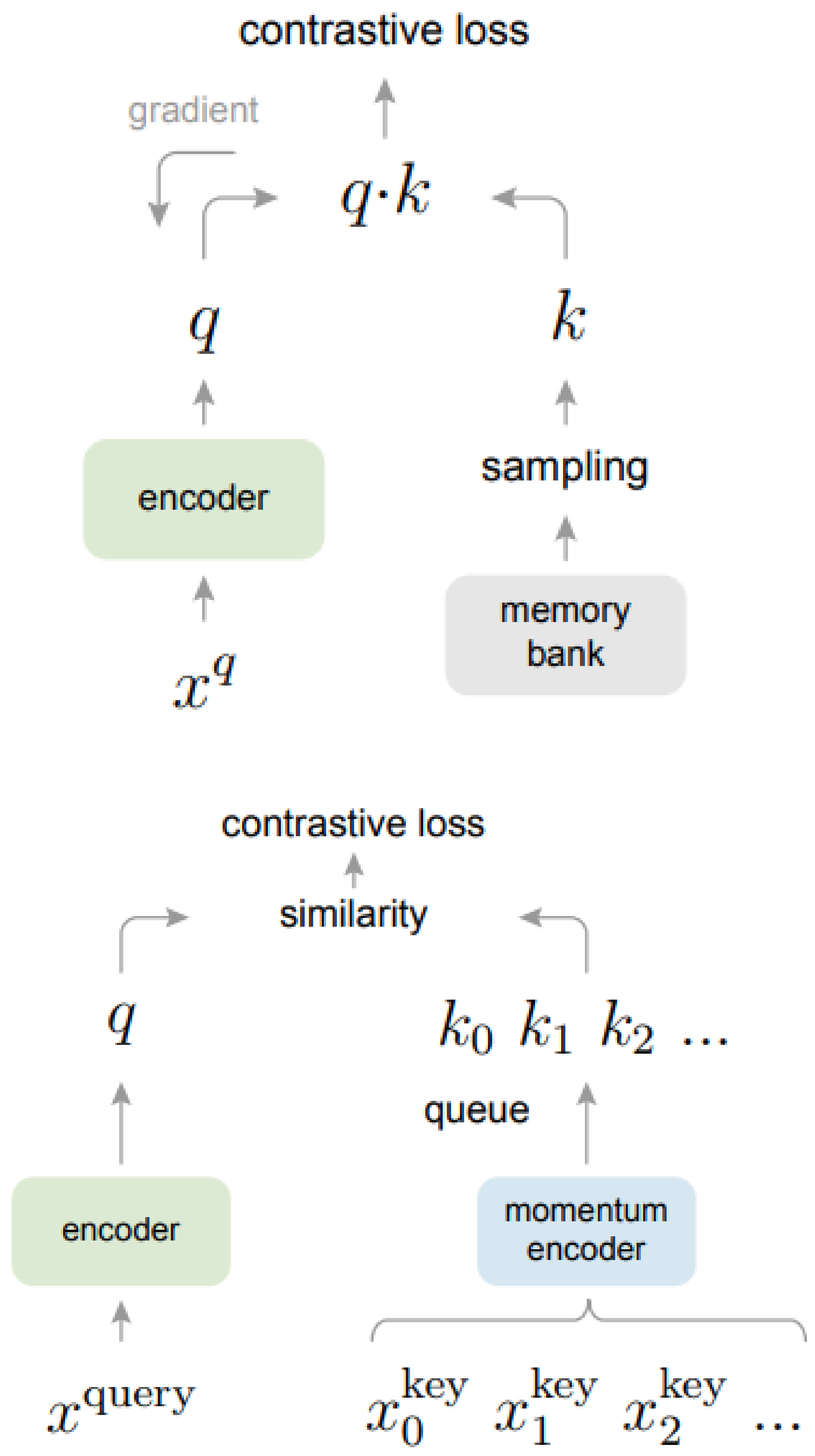
- One sample, Two views.
- Push away representations from different images while pulling together those from transformations



Contrastive instance learning

Contrastive Learning - MoCo

- **Momentum Contrast**
  - Build a dynamic dictionary with a queue and a moving-averaged encoder from a perspective on contrastive learning as dictionary look-up.
  - An encoded query should be similar to its matching key and dissimilar to others.
  - The dictionaries should be **large** and **consistent** as they evolve during training
  - The encoded representations of the current mini-batch are enqueued, and the oldest are dequeued.
  - The queue **decouples** the dictionary size from the mini-batch size, allowing it to be large.

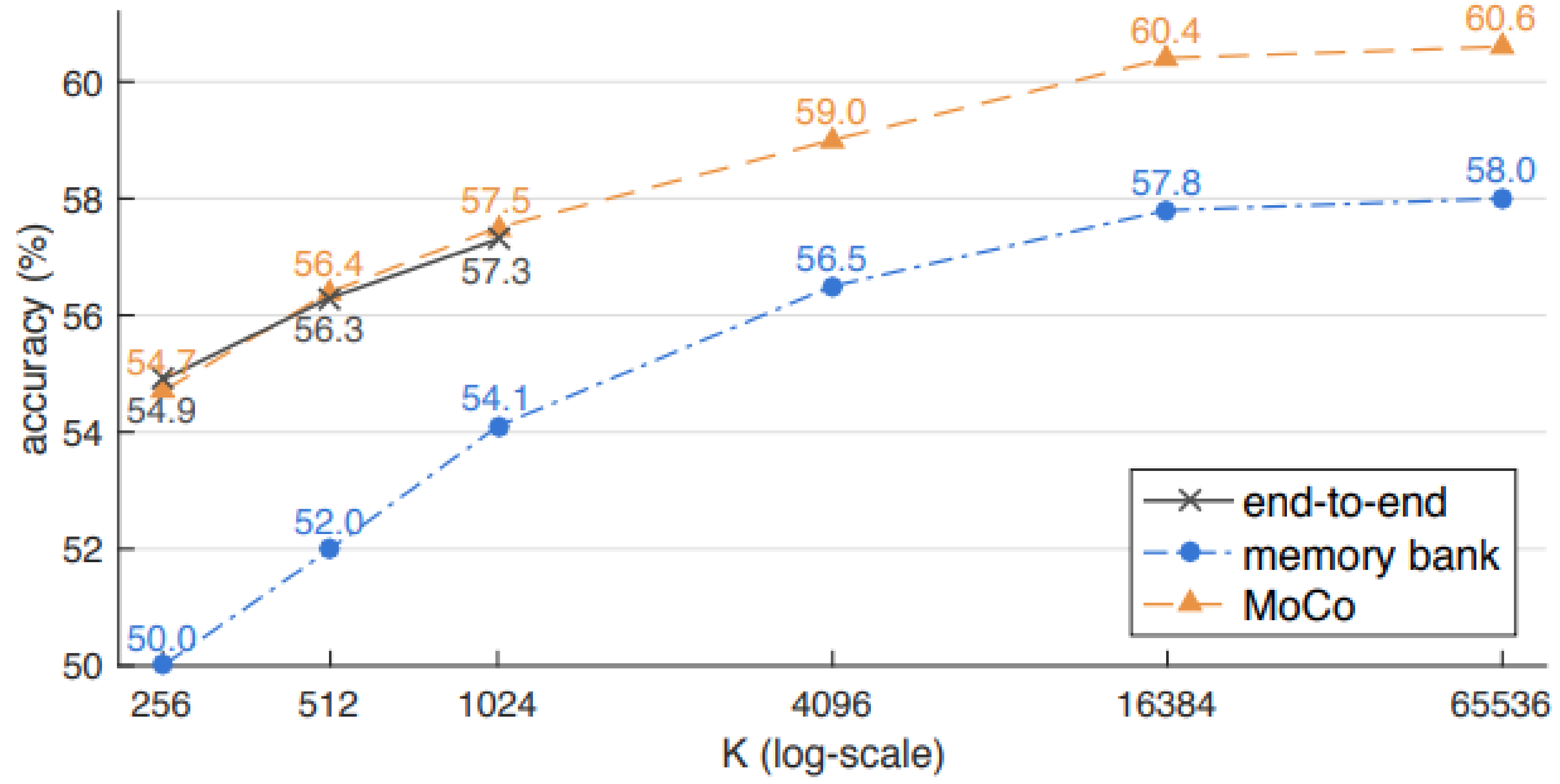


Contrastive Learning - MoCo

- **Loss**
  - Contrastive loss is a function whose value is low when the query is similar to its positive key and dissimilar from the negative keys
  - **InfoNCE** is a form of a contrastive loss with similarity measured by dot product

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

- **Compared with memory-bank**
  - More negative samples, better performance under ImageNet linear classification protocol
  - Performance is saturated at a large number of negative samples



Contrastive Learning - SimCLR

MEGVII 旷视

- **A simple framework for contrast learning**
  - Composition of multiple **data augmentation** operations is crucial in defining the contrastive prediction tasks that yield effective representations.
  - A learnable nonlinear transformation between the representation and the contrastive loss substantially improves the quality of the learned representations.
  - Representation learning with contrastive cross entropy loss benefits from **normalized embeddings** and an appropriately **adjusted temperature** parameter.
  - Contrastive learning benefits from larger batch-size and longer training.

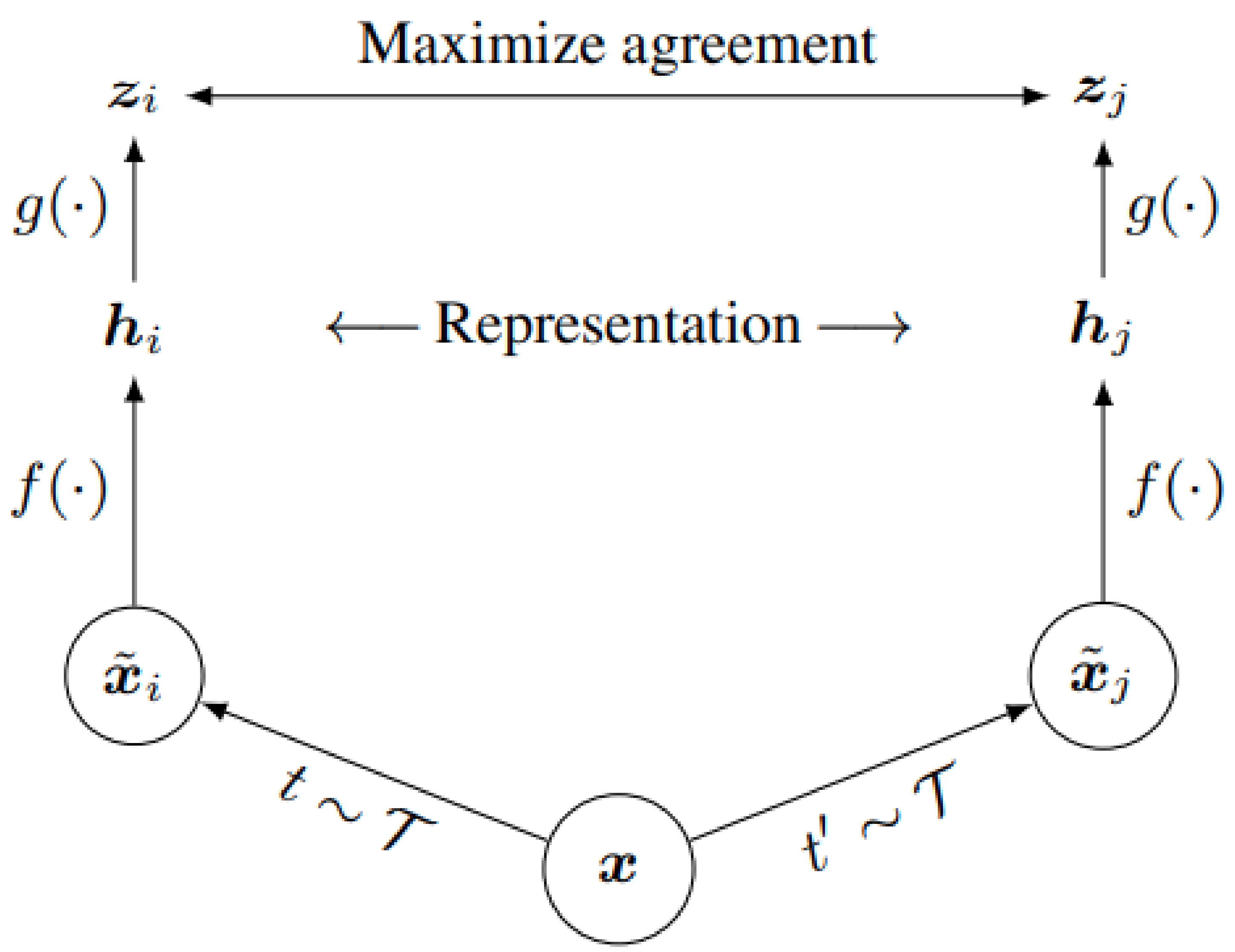
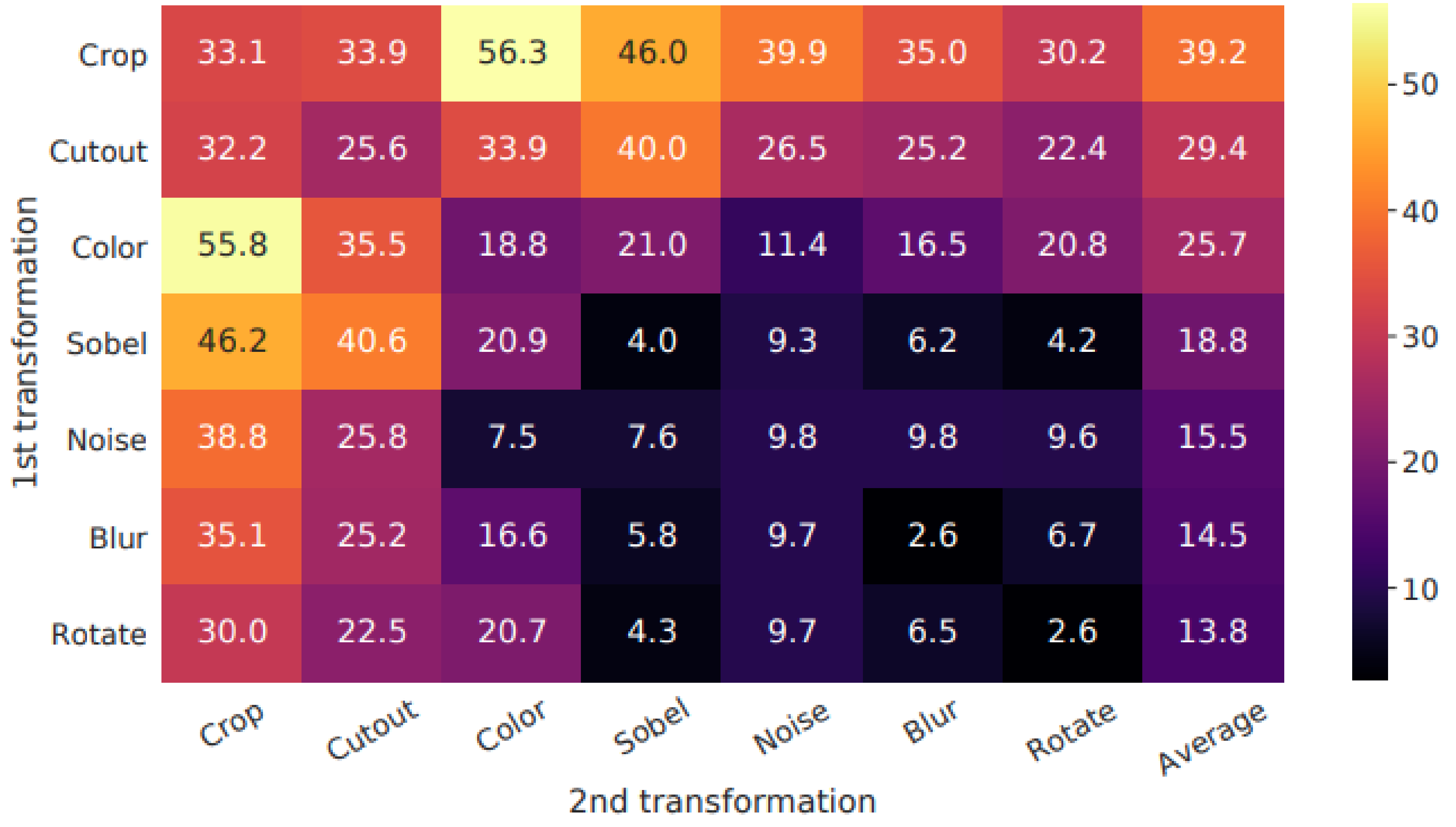


Figure 2. A simple framework for contrastive learning of visual representations. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, we throw away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation $\boldsymbol{h}$ for downstream tasks.

Contrastive Learning - SimCLR

- **Data augmentation**
  - No single transformation suffices to learn good representations
  - One composition of augmentations stands out: random cropping and random color distortion
  - Contrastive learning needs stronger data augmentation than supervised learning

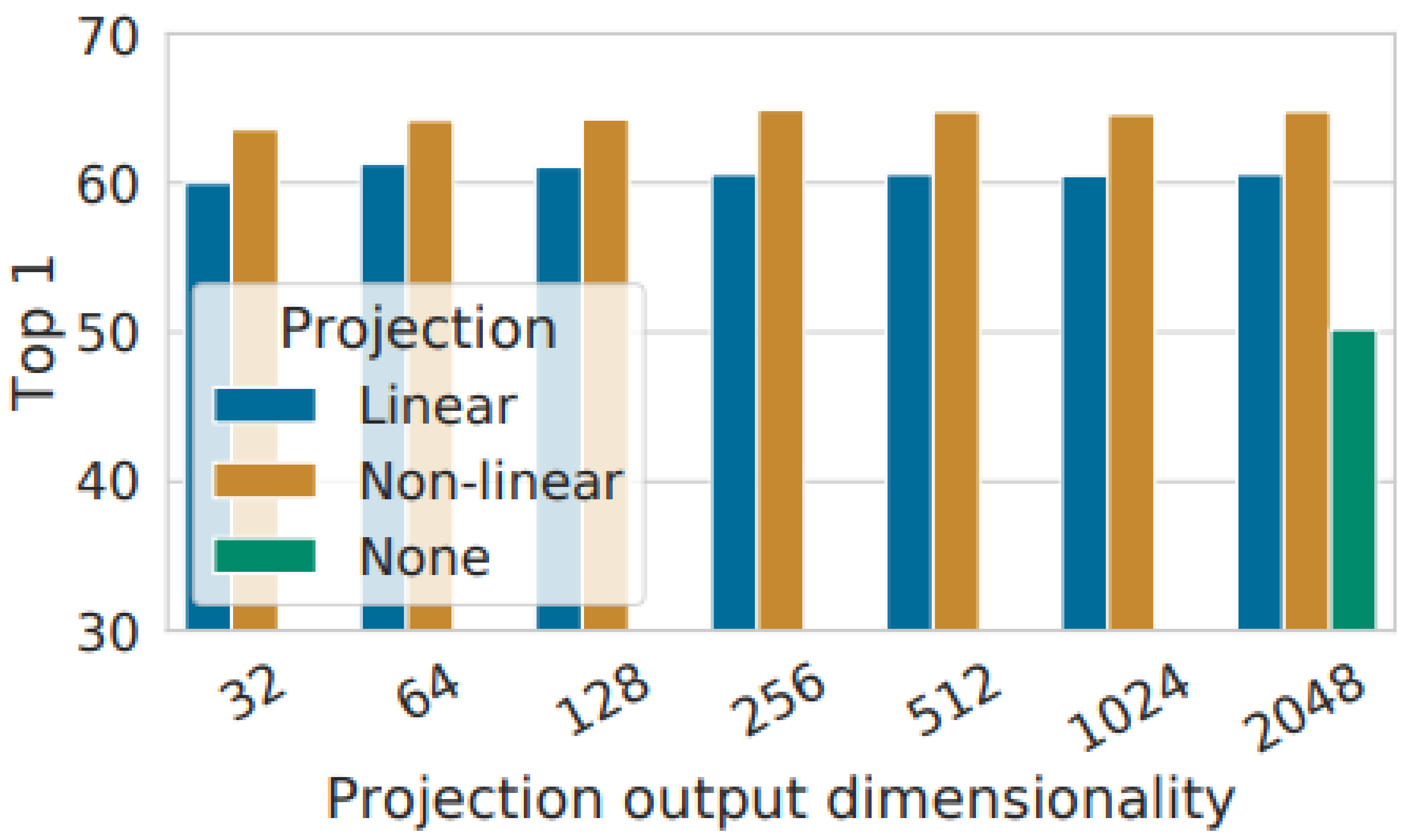


Methods	Color distortion strength 1/8	1/4	1/2	1	1 (+Blur)	AutoAug
SimCLR	59.6	61.0	62.6	63.2	64.5	61.1
Supervised	77.0	76.7	76.5	75.7	75.4	77.1

Table 1. Top-1 accuracy of unsupervised ResNet-50 using linear evaluation and supervised ResNet-50[5], under varied color distortion strength (see Appendix A) and other data transformations. Strength 1 (+Blur) is our default data augmentation policy.

Contrastive Learning - SimCLR

- **Architectures for Encoder and Head**
  - unsupervised learning benefits more from bigger models than its supervised counterpart
  - a nonlinear projection is better than a linear projection (+3%), and much better than no projection (>10%).



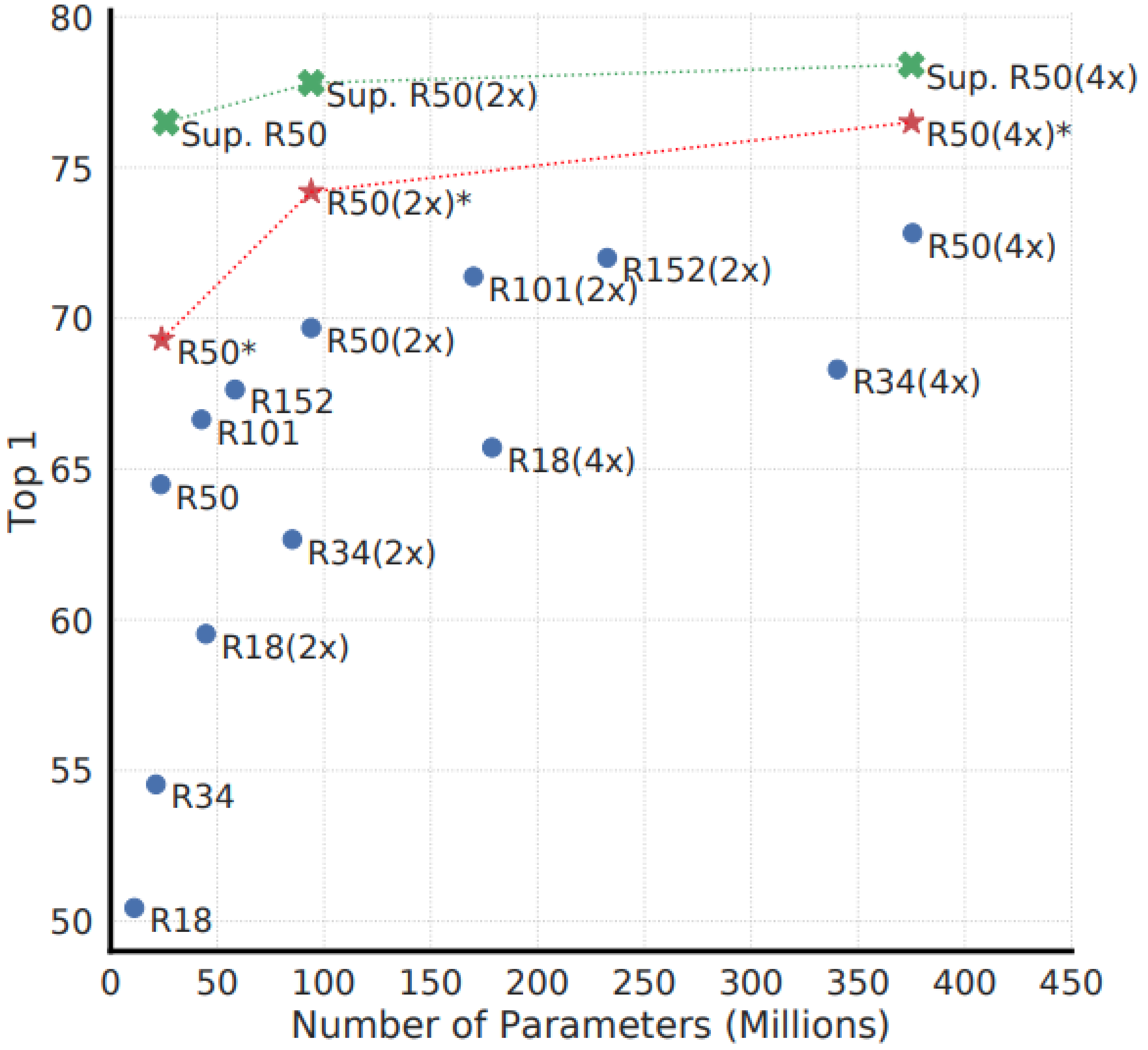


Figure 7. Linear evaluation of models with varied depth and width. Models in blue dots are ours trained for 100 epochs, models in red stars are ours trained for 1000 epochs, and models in green crosses are supervised ResNets trained for 90 epochs[7] (He et al., 2016).

Contrastive Learning - SimSiam

- **Simple Siamese Representation Learning**
  - No negative sample pairs
  - No large batches
  - No momentum encoders
  - Siamese networks can naturally introduce inductive biases for modeling invariance, as by definition "invariance" means that two observations of the same concept should produce the same outputs
  - A stop-gradient operation plays an essential role in preventing collapsing

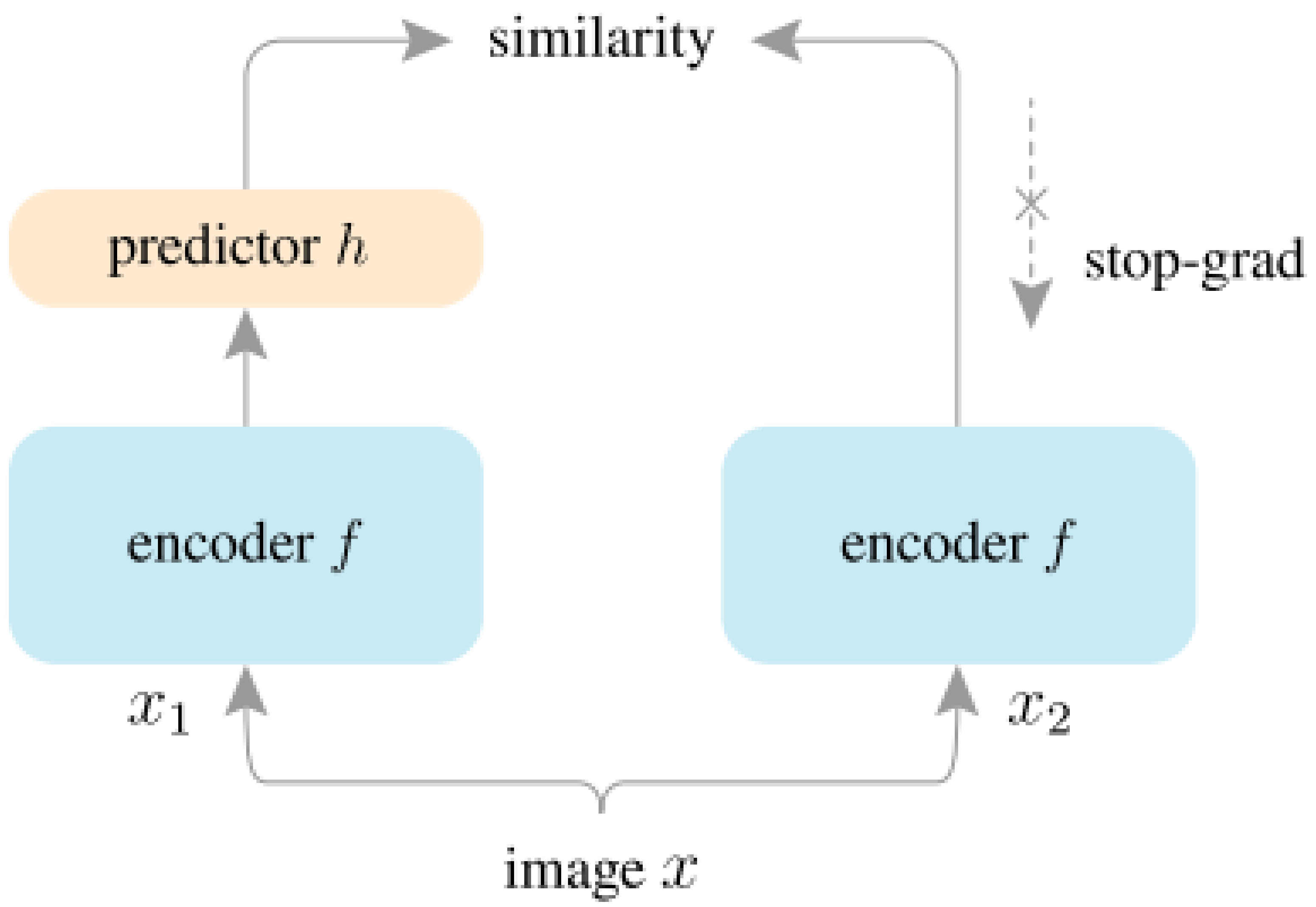


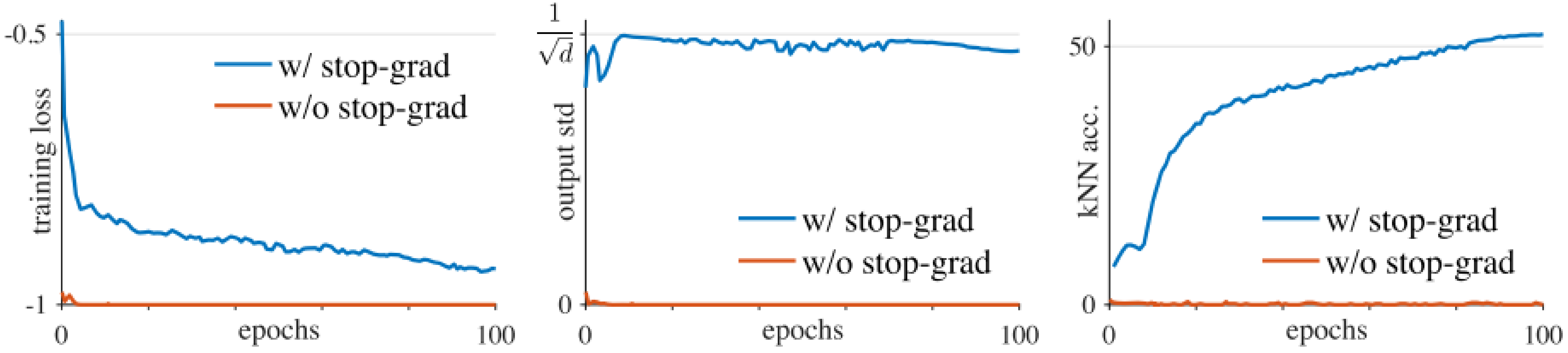
Figure 1. **SimSiam architecture**. Two augmented views of one image are processed by the same encoder network f (a backbone plus a projection MLP). Then a prediction MLP h is applied on one side, and a stop-gradient operation is applied on the other side. The model maximizes the similarity between both sides. It uses neither negative pairs nor a momentum encoder.

$$\mathcal{D}(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \cdot \frac{z_2}{\|z_2\|_2}, \tag{1}$$

$$\mathcal{L} = \frac{1}{2}\mathcal{D}(p_1, \texttt{stopgrad}(z_2)) + \frac{1}{2}\mathcal{D}(p_2, \texttt{stopgrad}(z_1)). \tag{4}$$

Contrastive Learning - SimSiam

- **Stop Gradient is important to prevent collapsing**

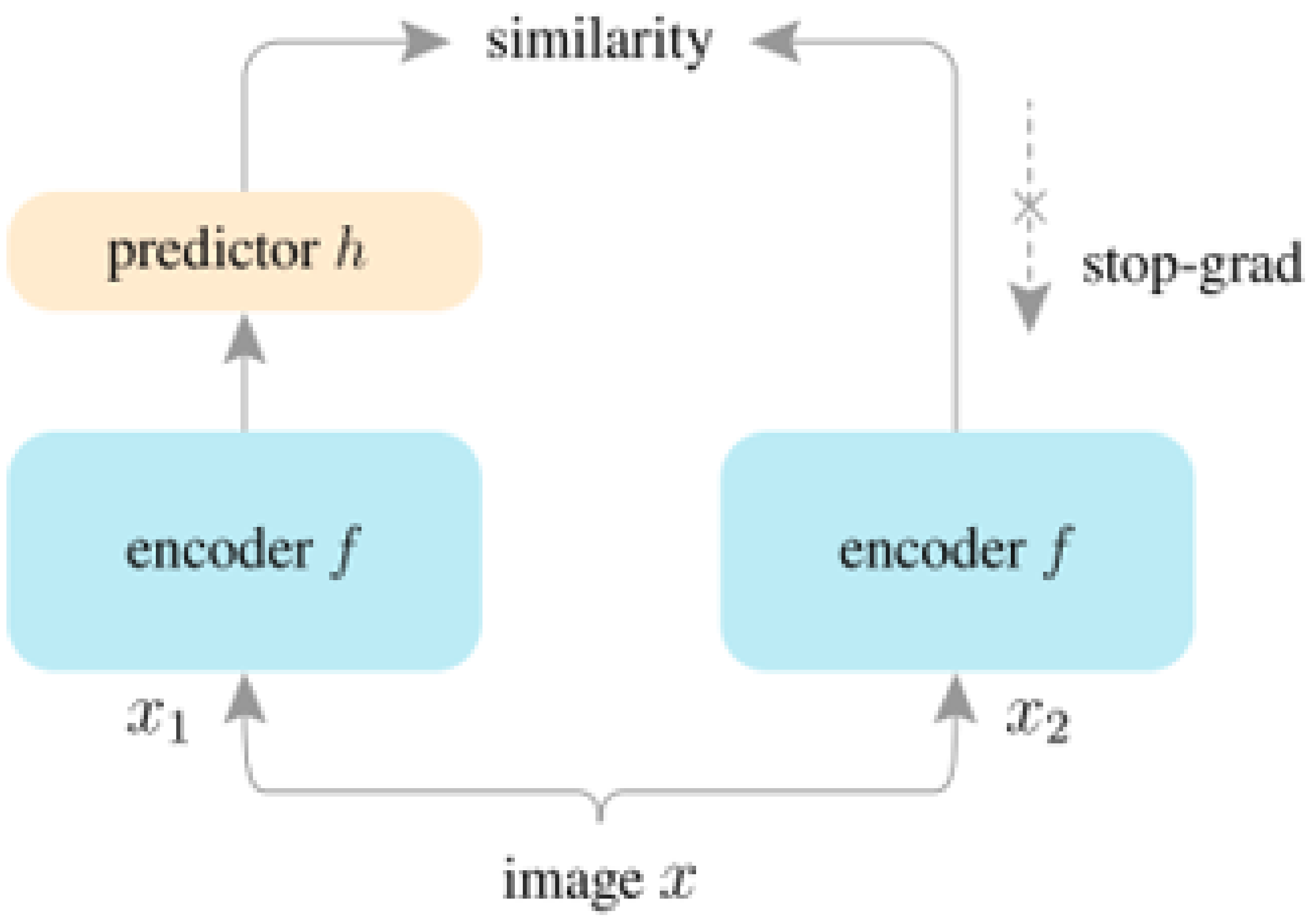


	acc. (%)
w/ stop-grad	67.7±0.1
w/o stop-grad	0.1

Figure 2. **SimSiam with *vs.* without stop-gradient. Left plot**: training loss. Without stop-gradient it degenerates immediately. **Middle plot**: the per-channel std of the ℓ_2-normalized output, plotted as the averaged std over all channels. **Right plot**: validation accuracy of a kNN classifier [36] as a monitor of progress. **Table**: ImageNet linear evaluation ("w/ stop-grad" is mean±std over 5 trials).

Contrastive Learning - SimSiam

- **Hypothesis**
  - Introduced another set of variables η. The size of η is proportional to the number of images.



$$\mathcal{L}(\theta,\eta) = \mathbb{E}_{x,T}\left[\| \mathcal{F}_\theta\big(T(x)\big) - \eta_x \|_2^2\right]$$

$$\min_{\theta,\eta} \mathcal{L}(\theta,\eta)$$

$$\theta^t \leftarrow \arg\min_\theta \mathcal{L}(\theta,\eta^{t-1})$$

$$\eta^t \leftarrow \arg\min_\eta \mathcal{L}(\theta^t,\eta)$$

$$\eta_x^t \leftarrow \mathbb{E}_T[\mathcal{F}_{\theta^t}(T(x))] \qquad \eta_x^t \leftarrow \mathcal{F}_{\theta^t}(T'(x))$$

$$\theta^{t+1} \leftarrow \arg\min_\theta \mathbb{E}_{x,T}\left[\| \mathcal{F}_\theta\big(T(x)\big) - \mathcal{F}_{\theta^t}(T'(x)) \|_2^2\right]$$

$$h(z_1) = \mathbb{E}_z[z_2] = \mathbb{E}_T[f(T(x))]$$

SSQL: Synergistic Self-supervised and Quantization

MEGVII 旷视

Synergistic Self-supervised and Quantization Learning

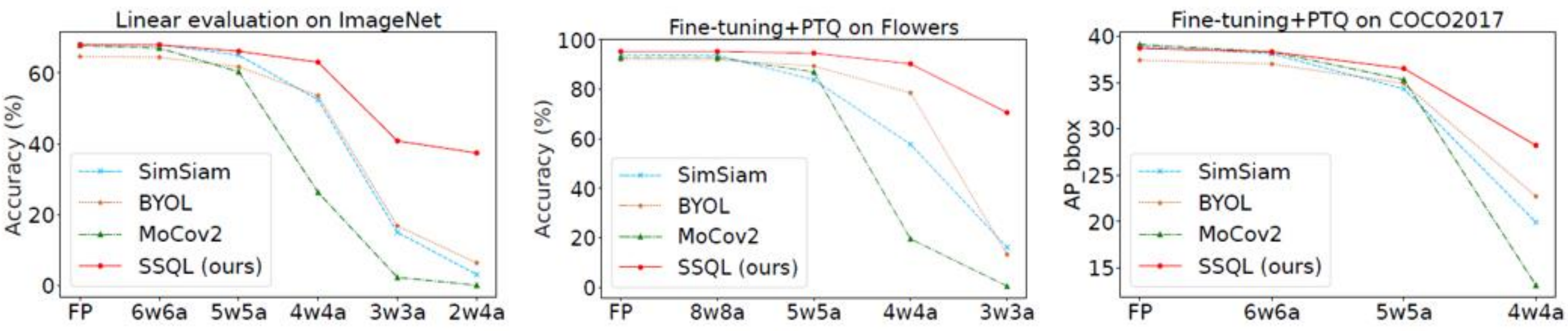
Yun-Hao Cao[1], Peiqin Sun[2*], Yechang Huang[2], Jianxin Wu[1], and Shuchang Zhou[2]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, China
[2] MEGVII Technology
{caoyunhao1997, wujx2001}@gmail.com, {sunpeiqin, huangyechang, zsc}@megvii.com

https://github.com/megvii-research/SSQL-ECCV2022

SSQL: Synergistic Self-supervised and Quantization

- Current SSL models suffer severe accuracy drops when performing low-bit quantization.
- Can we learn a quantization-friendly representation in SSL such that the pretrained model can be quantized more easily to facilitate deployment when transferring to different downstream tasks?

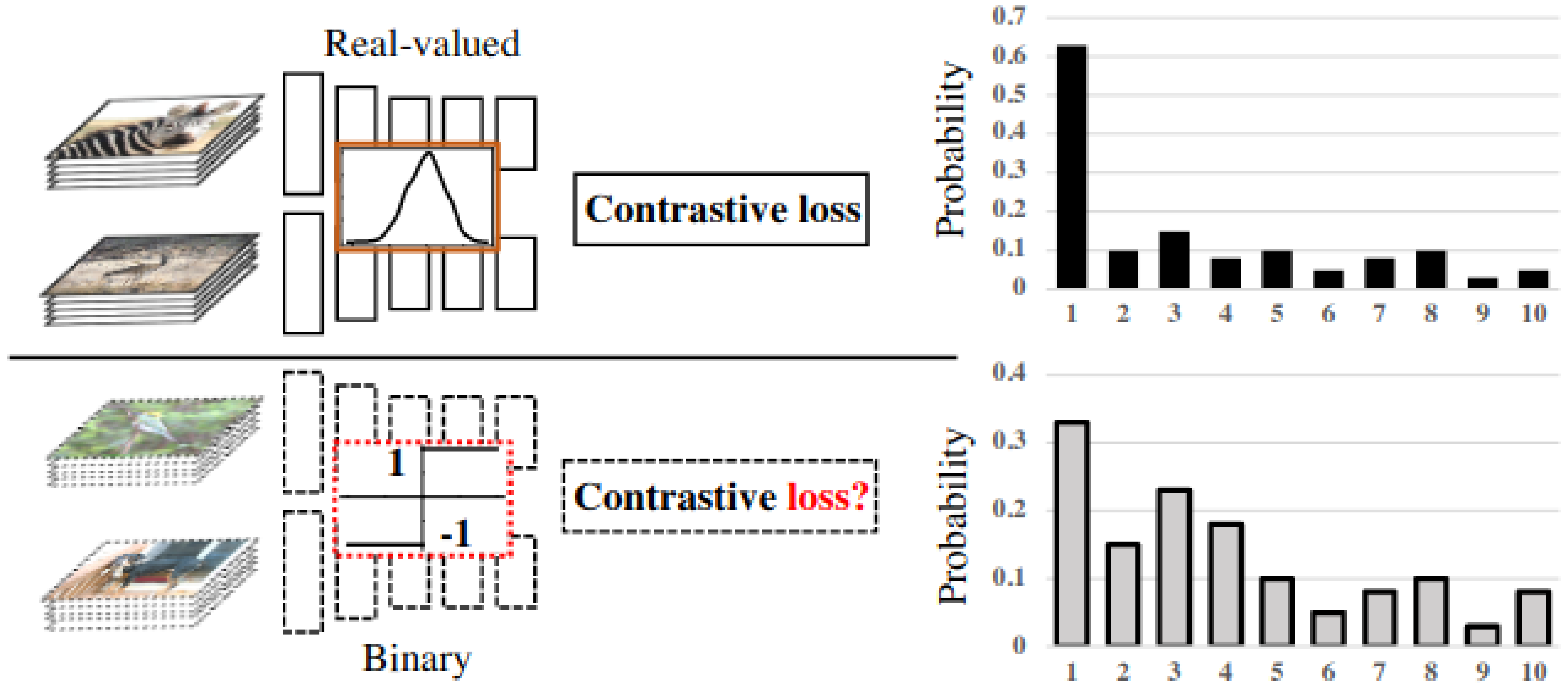


(a) ImageNet (b) Flowers (c) COCO2017 detection

only training once, only one copy of weights

SSQL – Related works

- **S²-BNN:**
  - follow the real-valued self-supervised method to train a teacher
  - enforce representations of BNNs to be similar to the real-valued reference network
  - solely optimize BNNs with the guided learning paradigm and the performance



- **SEED**

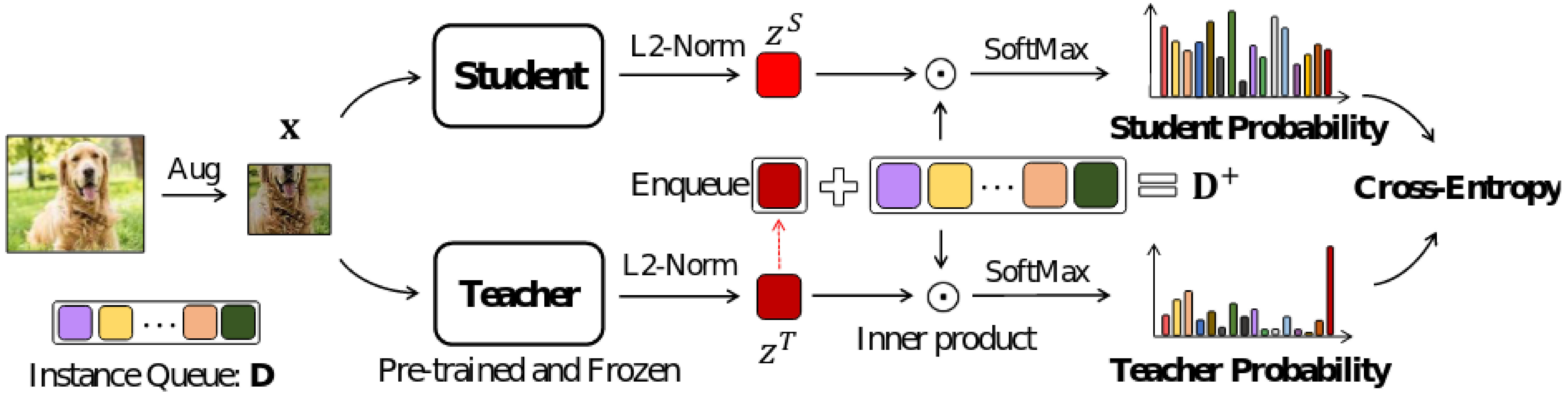


Figure 2: Illustration of our self-supervised distillation pipeline. The teacher encoder is pre-trained by *SSL* and kept frozen during the distillation. The student encoder is trained by minimizing the cross entropy of probabilities from teacher & student for an augmented view of an image, computed over a dynamically maintained queue.

S²-BNN: Bridging the Gap Between Self-Supervised Real and 1-bit Neural Networks via Guided Distribution Calibration
SEED: SELF-SUPERVISED DISTILLATION FOR VISUAL REPRESENTATION

SSQL: Synergistic Self-supervised and Quantization

- **Preliminary**
  - Replace one branch to quantized version (contrast quantized and FP)

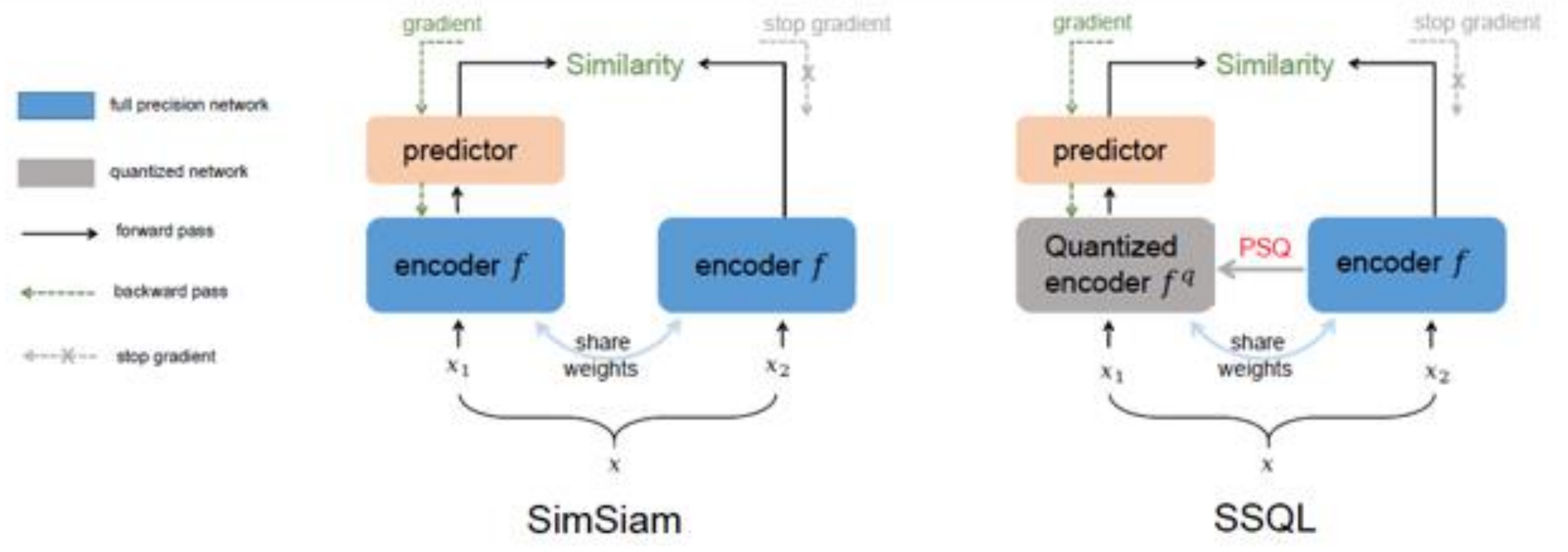


Fig. 2: Illustration of our method. Left: SimSiam [6]. Right: The proposed method SSQL. 'PSQ' denotes post step quantization, see Sec. 3.2 for details.

$$L_{\text{SimSiam}} = D(\boldsymbol{p}_1, SG(\boldsymbol{z}_2)) + D(\boldsymbol{p}_2, SG(\boldsymbol{z}_1))$$

$$L_{SSQL} = D(\boldsymbol{p}_1^q, SG(\boldsymbol{z}_2)) + D(\boldsymbol{p}_2^q, SG(\boldsymbol{z}_1))$$

SSQL - The devils are in the details

- **How to quantize the network?**
  - Adopt uniform quantization for its simplicity
  - Proposed PSQ which calculate scale & zero-point in each step

$$X_{int} = clip\left(\lfloor \frac{X}{S} + Z \rceil, 0, 2^q - 1\right),$$

$$X_q = (X_{int} - Z)S,$$

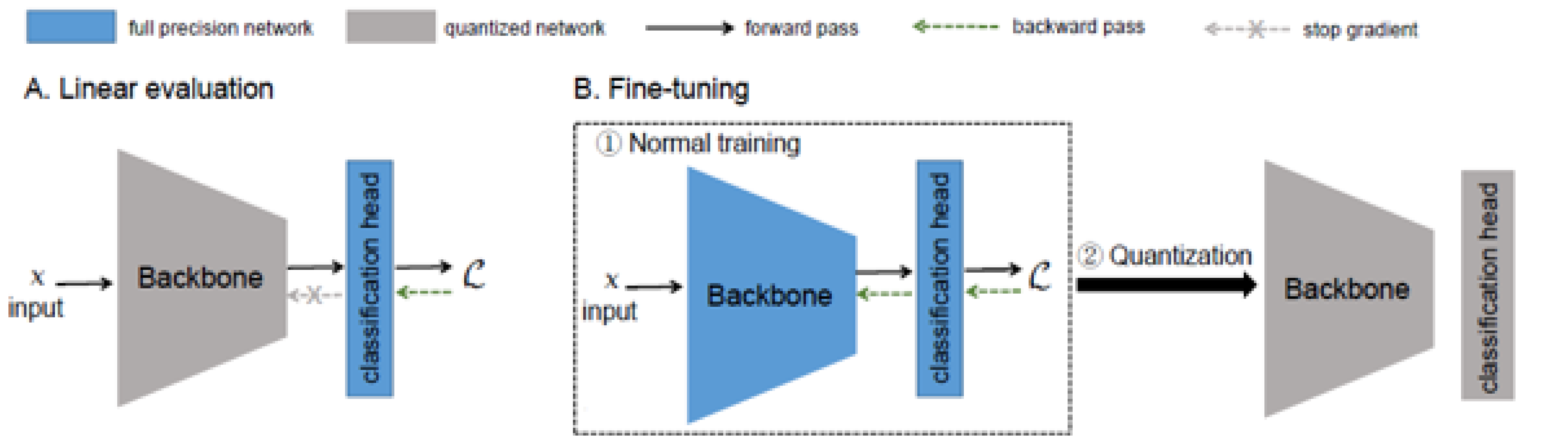
- **How to update weights in quantized network?**
  - The quantized network f_q and full-precision network **share weight**
  - Only backprop on quantized network using STE

$$\frac{\partial L}{\partial x} \approx \frac{\partial L}{\partial x_q}$$

- **How to achieve bit-width flexibility?**
  - Random select a value from a set of candidate bit-widths in each-step for the assignment of q.

SSQL – Evaluation Protocols

- **Linear evaluation**
  - Fixed the backbone quantized weights, only training classification
- **Fine-tuning**
  - Training update with float backbone, then executes PTQ.
  - In QAT, we use the pre-trained float backbone as initialization.



SSQL – Experiment Results

MEGVII 旷视

- achieves better performance than SSL in most bit-widths.
- achieves better performance than PACT in most bit-widths.

Table 1: Linear evaluation results on CIFAR-10. All pretrained for 400 epochs. SimSiam-PACT trains 7 models separately and we color it grey.

Backbone	Method	Linear evaluation accuracy (%)							
		FP	8w8a	6w6a	5w5a	4w4a	3w3a	2w8a	2w4a
ResNet-18	SimSiam [6]	**90.7**	90.7	**90.6**	90.3	88.9	66.0	70.1	63.8
	BYOL [15]	89.3	89.3	89.4	89.3	88.0	75.1	71.9	63.3
	SimSiam-PACT [8]	-	89.2	89.2	89.3	89.2	88.2	89.3	88.3
	SSQL (ours)	**90.7**	**90.8**	**90.6**	**90.6**	**90.1**	**85.6**	**88.0**	**86.5**
	SimCLR [4]	**89.4**	**89.3**	**89.2**	**88.8**	87.1	73.9	65.6	55.6
	MoCov2 [5]	88.9	88.8	88.4	88.2	86.8	72.2	66.4	50.7
	SSQL-NCE (ours)	89.0	89.0	89.0	**88.8**	**87.9**	**82.9**	**87.1**	**84.9**
ResNet-50	SimSiam [6]	90.9	90.9	91.0	90.6	89.5	74.1	55.1	57.1
	BYOL [15]	90.3	90.3	90.0	89.7	87.5	58.5	82.4	67.8
	SSQL (ours)	**91.1**	**91.1**	**91.1**	**91.1**	**90.0**	**77.4**	**89.5**	**87.2**
	SimCLR [4]	91.5	91.4	91.3	90.5	88.1	59.6	63.5	42.4
	MoCov2 [5]	90.2	90.2	90.2	89.4	87.9	72.1	68.8	49.5
	SSQL-NCE (ours)	**92.1**	**92.1**	**92.0**	**91.9**	**89.8**	**74.0**	**88.6**	**84.9**

SSQL – Experiment Results

MEGVII 旷视

- achieves better performance at all bit-widths with only training once and one copy of weights.

Table 3: Linear evaluation results on ImageNet. All pretrained for 100 epochs, except for MoCov2. † denotes that we use the official MoCov2 200ep checkpoint. SimSiam-PACT trains 5 models separately and we color it grey.

Backbone	Method	Linear evaluation accuracy (%)					
		FP	8w8a	5w5a	4w4a	3w3a	2w4a
ResNet-18	SimSiam [6]	55.0	54.7	53.9	36.7	6.3	1.5
	BYOL [15]	54.1	54.0	51.9	42.4	13.6	3.6
	SimSiam-PACT [8]	-	52.8	52.8	52.3	51.0	51.6
	SSQL (ours)	**57.6**	**57.6**	**56.7**	**52.8**	**41.0**	**43.1**
ResNet-50	SimSiam [6]	**68.1**	**67.9**	65.0	52.4	15.0	3.1
	BYOL [15]	64.6	64.4	61.7	53.6	16.8	6.4
	MoCov2† [5]	67.7	67.0	60.3	26.3	2.3	0.1
	SSQL (ours)	67.9	**67.9**	**66.1**	**63.0**	**40.8**	**37.4**

Table 4: Fine-tuning+PTQ results on ImageNet subsets. Here we adopt the fine-tuning settings on 1%/10% labeled data and report Top-5 accuracy (%).

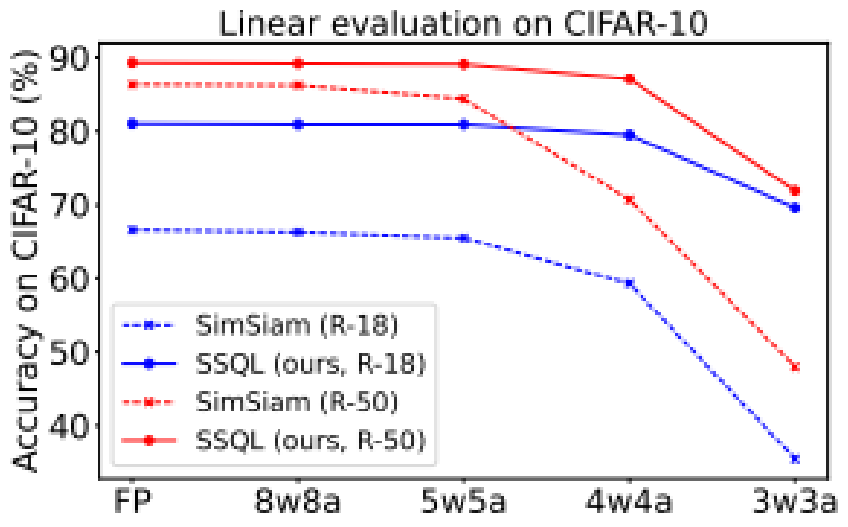
Backbone	Method	1% labels				10% labels			
		FP	6w6a	5w5a	4w4a	FP	6w6a	5w5a	4w4a
ResNet-18	SimSiam [6]	43.7	43.4	42.4	37.5	**76.1**	75.8	74.3	64.5
	BYOL [15]	36.7	36.5	35.5	31.2	75.5	75.1	73.9	65.2
	SSQL (ours)	**47.7**	**47.6**	**47.1**	**45.0**	**76.1**	**75.9**	**75.0**	**70.7**
ResNet-50	SimSiam [6]	53.2	52.8	51.5	36.4	82.5	81.7	79.0	67.9
	BYOL [15]	47.3	47.2	46.4	40.4	81.1	80.7	79.6	69.9
	SSQL (ours)	**55.2**	**55.0**	**54.4**	**51.8**	**83.0**	**82.7**	**81.0**	**76.7**

SSQL – Experiment Results

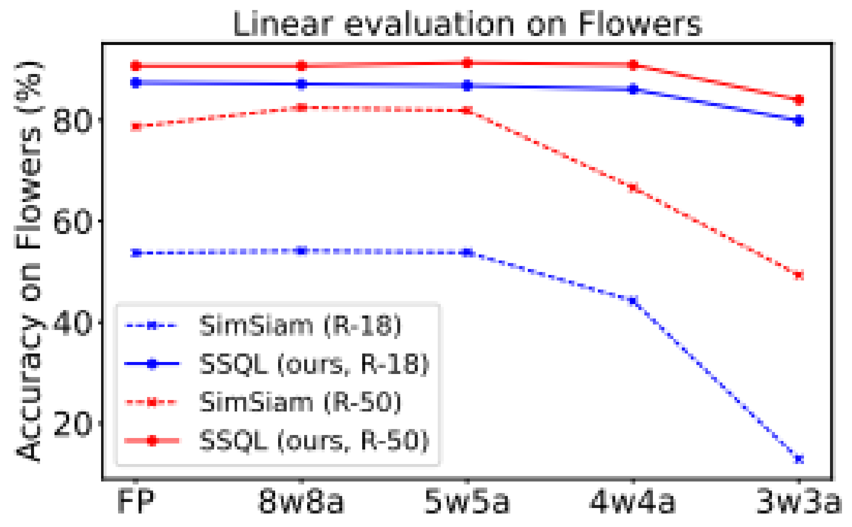
- boosts the transfer results of full-precision model significantly
- achieves better performance than SimSiam on various downstream tasks at all bit-widths.

Table 5: ImageNet transfer results on recognition benchmarks under R-50.

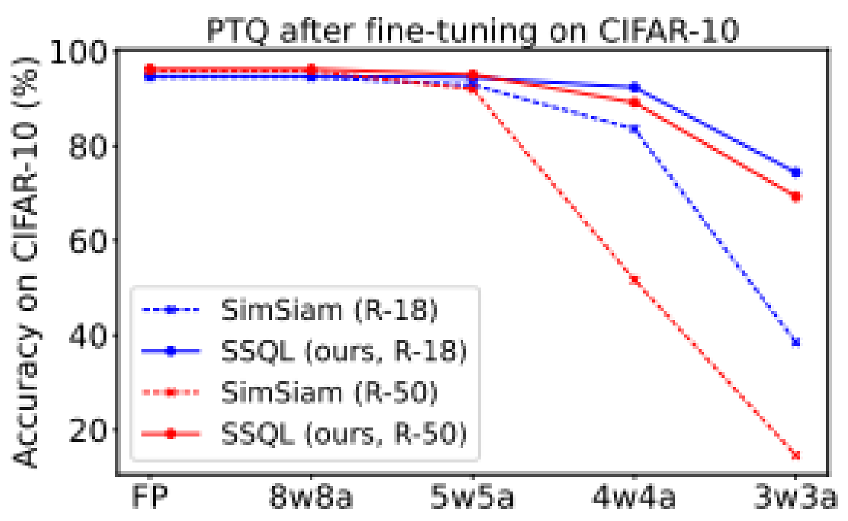
Datasets	Method	Linear evaluation					Fine-tuning				
		FP	8w8a	5w5a	4w4a	3w3a	FP	8w8a	5w5a	4w4a	3w3a
CIFAR-10	SimSiam	86.3	86.2	84.4	70.7	48.0	95.9	95.9	92.0	51.7	14.6
	SSQL (ours)	**89.3**	**89.2**	**89.1**	**87.1**	**71.9**	**96.3**	**96.3**	**95.1**	**89.2**	**69.3**
CIFAR-100	SimSiam	58.9	58.7	52.5	39.0	20.2	82.9	82.5	76.7	66.0	5.3
	SSQL (ours)	**68.7**	**68.6**	**68.8**	**66.4**	**49.7**	**83.3**	**83.3**	**82.0**	**74.7**	**39.3**
Flowers	SimSiam	78.7	82.5	81.9	66.6	49.3	94.0	93.8	83.8	57.8	16.2
	SSQL (ours)	**90.7**	**90.7**	**91.3**	**90.9**	**84.0**	**95.3**	**95.3**	**94.6**	**90.3**	**70.6**
Food-101	SimSiam	67.1	67.1	64.7	56.0	27.7	**86.2**	**86.2**	80.4	54.4	2.2
	SSQL (ours)	**72.6**	**72.5**	**71.5**	**68.4**	**51.6**	85.5	85.5	**84.5**	**70.4**	**11.2**
Pets	SimSiam	79.7	79.6	74.3	70.9	32.2	**87.5**	**87.4**	81.3	59.3	10.9
	SSQL (ours)	**83.6**	**83.9**	**83.3**	**82.3**	**73.8**	86.9	86.8	**85.9**	**84.6**	**73.6**
Dtd	SimSiam	69.9	69.7	69.1	63.4	46.6	73.4	73.6	70.5	60.4	8.8
	SSQL (ours)	**74.4**	**74.3**	**74.3**	**73.4**	**64.4**	**73.7**	**73.7**	**71.9**	**70.1**	**56.6**
Caltech-101	SimSiam	80.2	80.4	78.6	66.7	31.4	**86.9**	**86.6**	85.0	76.8	7.9
	SSQL (ours)	**86.9**	**87.2**	**85.2**	**83.8**	**65.9**	86.4	86.3	**85.5**	**82.9**	**59.7**



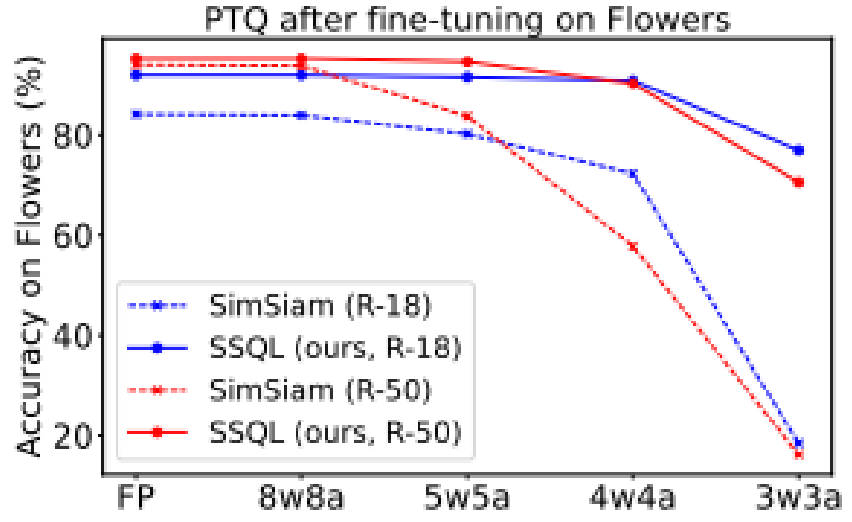
(a) CIFAR-10+Lin



(b) Flowers+Lin



(e) CIFAR-10+FT



(f) Flowers+FT

SSQL – Experiment Results

- exceeds the other methods at all bit-widths by a large margin on VOC2007 & COCO2017

Table 6: Object detection results on VOC2007 under R18-C4. The best results are in **boldface** and the second best results are underlined.

Method	FP			8w8a			6w6a			5w5a			4w4a		
	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}
random init.	58.9	32.1	30.5	58.7	31.9	30.2	58.4	31.6	30.2	57.0	30.4	28.9	42.4	20.8	17.0
IN supervised	**73.9**	44.6	46.5	**74.1**	44.2	46.2	73.0	43.4	44.5	68.9	39.4	39.3	33.1	16.7	14.0
BYOL	72.8	44.7	46.3	72.4	44.4	46.0	72.2	44.2	45.6	62.7	38.0	39.4	52.7	28.9	27.8
SimSiam	72.8	44.4	46.6	72.9	44.4	46.3	72.4	44.0	46.0	69.7	42.0	43.1	50.4	26.4	23.8
SimSiam-200ep	72.5	44.3	46.5	72.5	44.3	46.5	72.0	43.9	46.3	69.2	41.4	42.7	53.7	29.8	29.0
SSQL (ours)	73.4	44.7	46.8	73.5	**45.0**	46.8	**73.1**	44.5	46.4	**71.6**	42.8	44.4	**61.2**	34.1	33.4
SSQL-200ep (ours)	73.2	**45.0**	**47.3**	73.2	**45.0**	**47.0**	72.9	**44.8**	**46.8**	71.3	**43.3**	**45.0**	**61.2**	**35.1**	**35.0**

Table 7: Object detection/segmentation results on COCO2017 under R50-FPN.

Method	FP						6w6a					
	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
IN supervised	38.2	56.0	42.0	34.8	56.0	37.2	37.6	58.3	41.4	34.3	55.2	36.8
SimSiam	**38.9**	**59.8**	**42.3**	**35.2**	**56.7**	**37.7**	38.1	58.7	41.5	34.5	55.7	36.8
BYOL	37.4	57.9	40.6	34.1	54.9	36.4	37.0	57.4	40.2	33.7	54.3	36.0
SSQL (ours)	38.7	59.2	**42.3**	**35.2**	56.2	**37.7**	**38.3**	**58.8**	**41.7**	**34.8**	**55.8**	**37.3**
	5w5a						4w4a					
IN supervised	35.2	55.5	38.4	31.9	52.3	34.0	23.4	38.6	24.6	21.4	36.3	22.1
SimSiam	34.3	54.0	36.7	30.9	50.6	32.6	19.9	33.6	20.6	18.1	31.3	18.3
BYOL	34.9	54.4	37.7	31.8	51.4	33.8	22.7	37.4	24.0	20.9	35.2	21.7
SSQL (ours)	**36.5**	**56.9**	**39.4**	**33.3**	**53.6**	**35.5**	**28.2**	**43.1**	**27.5**	**26.0**	**43.1**	**27.5**

SSQL – Combined with QAT

- The pretrained model from SSQL can serve as a better initialization when combined with QAT methods to boost performance.

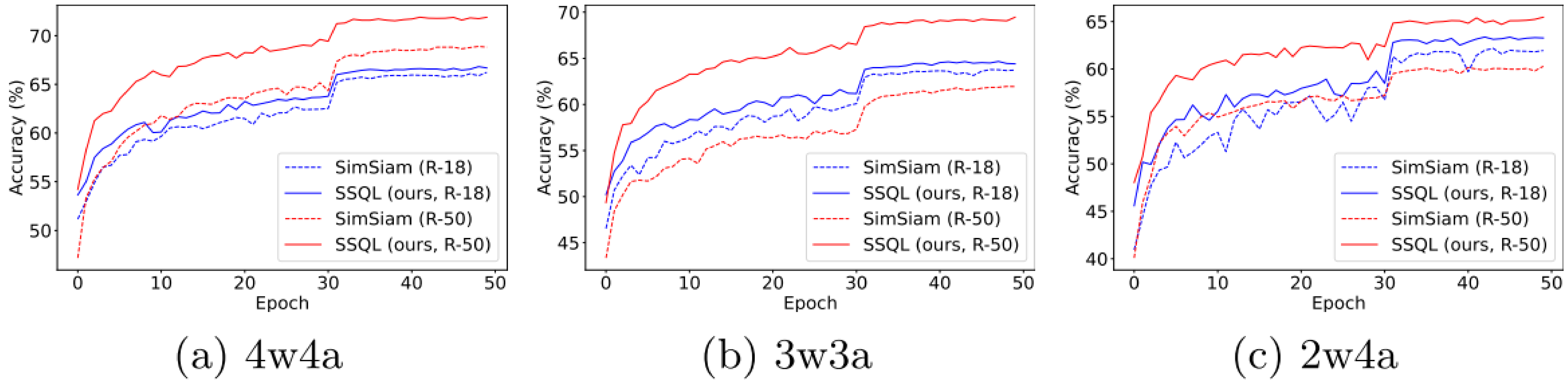
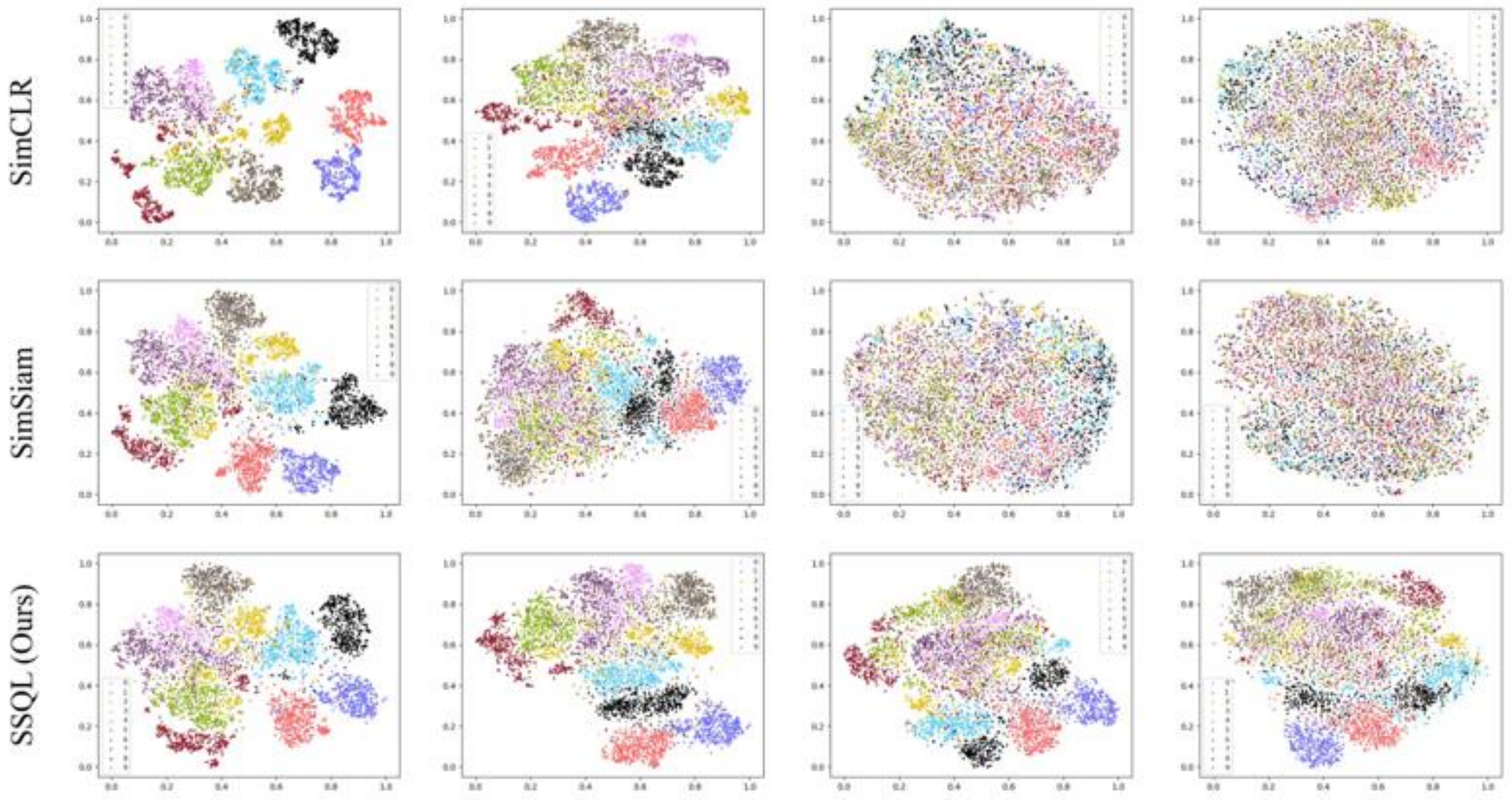


Fig. 6: ImageNet results using LSQ [11], initialized with the linear evaluation models in Table 3. See appendix for details of the settings for LSQ.

SSQL – Experiment Results

- T-SNE visualization on CIFAR-10
  - All methods have learned separable representations in full precision.
  - The learned representations via SSQL is more separable in ultra-low bit-widths.



(a) Full precision (b) 4w4a (c) 2w8a (d) 2w4a

SSQL – Experiment Results

- **Weight distribution**
  - More uniformly distributed, have smaller dynamic ranges, less outliers

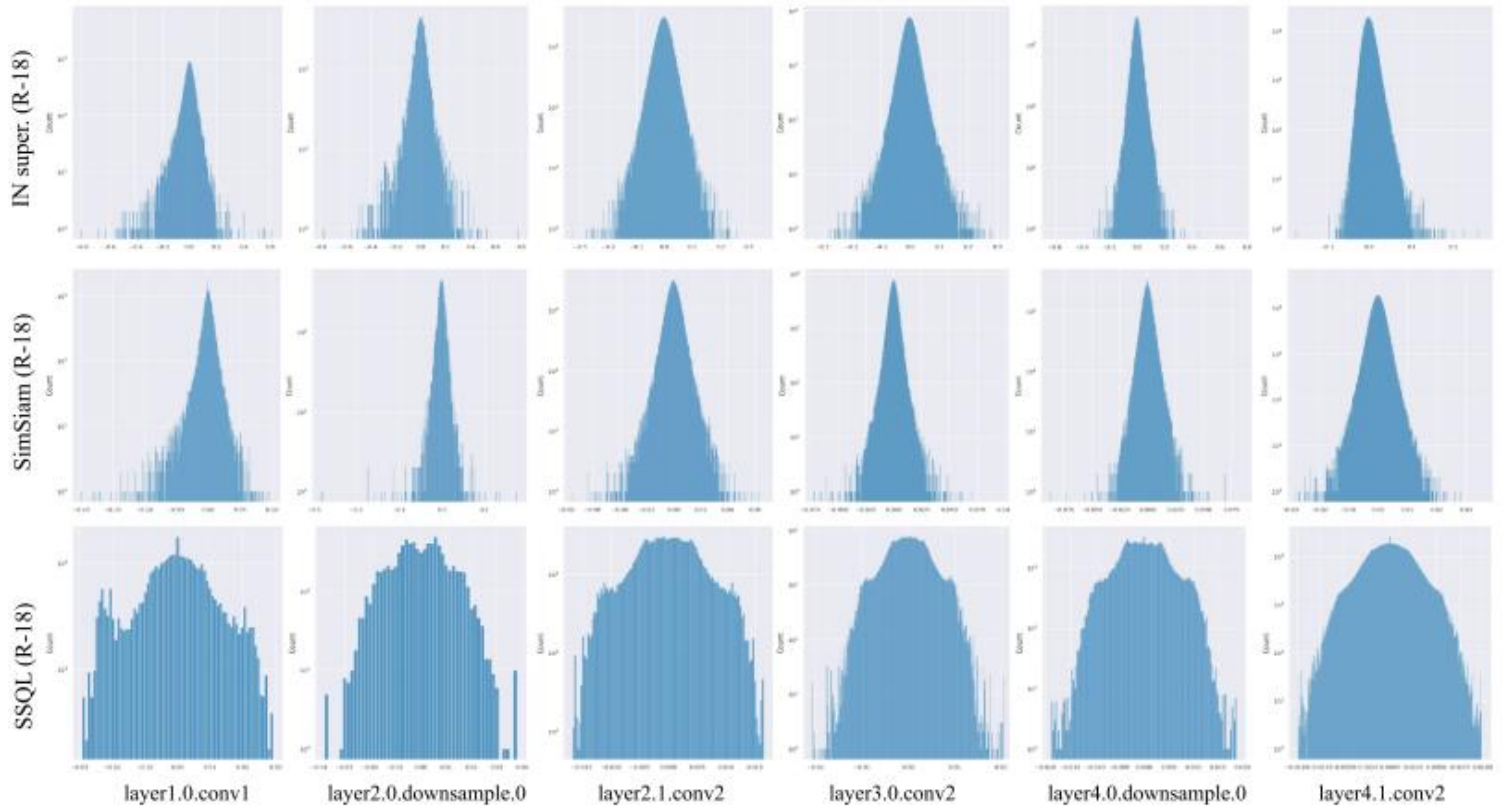


Fig. 5: Visualization of weight distribution for ResNet-18. The first, second and third row are the results of ImageNet supervised, SimSiam and ours, respectively.

-0.5 ~ 0.5

-0.3 ~ 0.1

-0.05 ~ 0.05

SSQL – Ablation studies

- Quantizing the target branch only degenerates the performance.
- Random selection of bit-widths for training is better than training with a single bit-width.
- Using a reasonable bit perturbation range further improves the performance at lower bit-widths.

Table 8: Ablation studies on CIFAR-10 using ResNet-34.

ID	Q Pred	Q Target	Aux	W Bit	A Bit	Linear evaluation accuracy (%)					
						FP	6w6a	4w4a	3w3a	2w4a	Avg.
(a)	×	×	×	-	-	89.0	89.0	87.2	75.6	55.3	79.2
(b)	×	✓	×	4∼16	4∼16	87.6	87.5	85.8	70.4	58.5	78.0
(c)	✓	✓	×	4∼16	4∼16	90.5	90.4	88.9	79.2	73.7	84.5
(d)	✓	×	×	4∼16	4∼16	**91.0**	**91.0**	89.5	83.0	65.2	83.9
(e)	✓	×	×	6	6	90.0	89.9	87.9	69.1	62.1	79.8
(f)	✓	×	×	4	4	36.0	35.9	36.4	29.2	29.7	33.4
(g)	✓	✓	×	2∼8	4∼8	88.3	88.2	86.9	80.3	85.4	85.8
(h)	✓	×	×	2∼8	4∼8	89.6	89.5	88.2	82.9	81.5	86.3
(i)	✓	×	✓	2∼8	4∼8	90.9	90.8	**89.6**	**83.2**	**86.8**	**88.3**

SSQL – Theoretic analysis

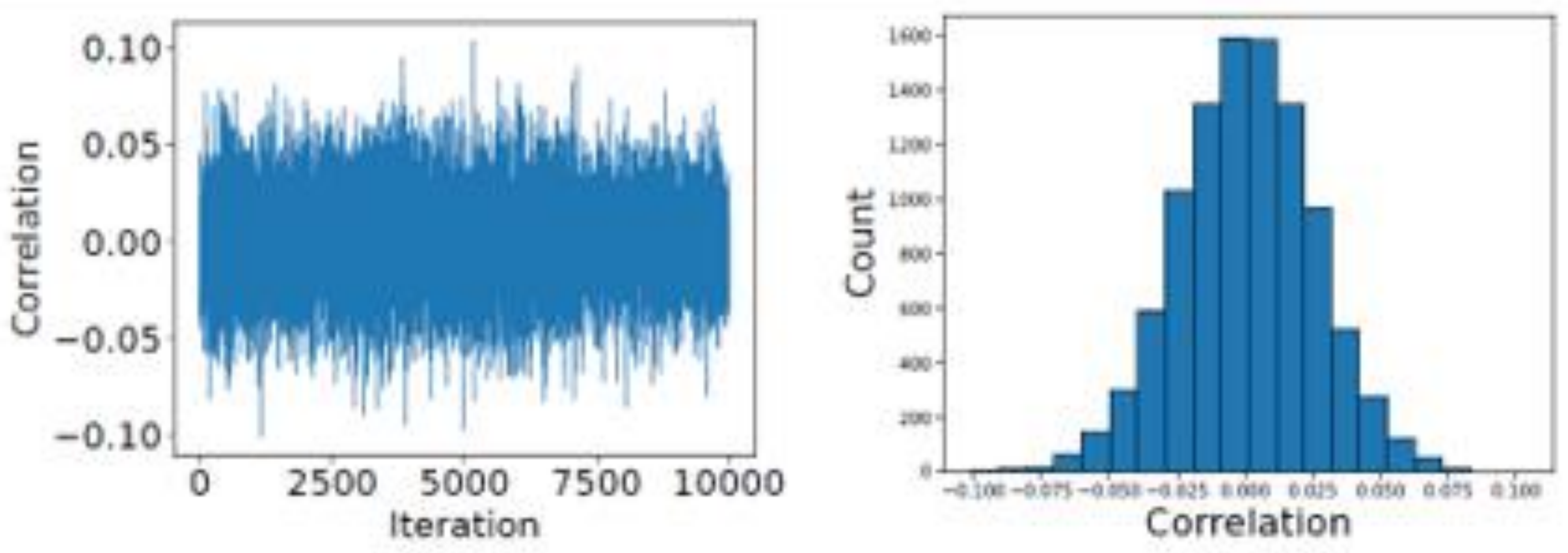
MEGVII 旷视

- Reformulate the loss function in an Expectation-Maximization

$$\theta^{t+1} \leftarrow \arg\min_{\theta} \mathbb{E}_{x,\mathcal{T},q}\left[\|\mathcal{F}_{\theta}^{q}(\mathcal{T}(x)) - \mathcal{F}_{\theta^t}(\mathcal{T}'(x))\|_2^2\right]$$

- The synergy between self-supervised and quantization learning

$$\begin{aligned}
&\mathbb{E}_{x,\mathcal{T},q}\left[\|\mathcal{F}_{\theta}^{q}(\mathcal{T}(x)) - \mathcal{F}_{\theta^t}(\mathcal{T}'(x))\|_2^2\right] \\
=&\mathbb{E}_{x,\mathcal{T},q}\left[\|\mathcal{F}_{\theta}^{q}(\mathcal{T}(x)) - \mathcal{F}_{\theta}(\mathcal{T}(x)) + \mathcal{F}_{\theta}(\mathcal{T}(x)) - \mathcal{F}_{\theta^t}(\mathcal{T}'(x))\|_2^2\right] \\
=&\underbrace{\mathbb{E}_{x,\mathcal{T},q}\left[\|\mathcal{F}_{\theta}^{q}(\mathcal{T}(x)) - \mathcal{F}_{\theta}(\mathcal{T}(x))\|_2^2\right]}_{\text{Q term (quantization term)}} + \underbrace{\mathbb{E}_{x,\mathcal{T},q}\left[\|\mathcal{F}_{\theta}(\mathcal{T}(x)) - \mathcal{F}_{\theta^t}(\mathcal{T}'(x))\|_2^2\right]}_{\text{CL term (contrastive learning term)}} \\
&+ \underbrace{2\mathbb{E}_{x,\mathcal{T},q}\left[\left(\mathcal{F}_{\theta}^{q}(\mathcal{T}(x)) - \mathcal{F}_{\theta}(\mathcal{T}(x))\right)^T \left(\mathcal{F}_{\theta}(\mathcal{T}(x)) - \mathcal{F}_{\theta^t}(\mathcal{T}'(x))\right)\right]}_{\text{cross term}} .
\end{aligned}$$



SSQL – Versatile

- Effectiveness when applied to vision transformer? **Yes!!!**

Table 13: Linear evaluation results on CIFAR-10 under ViT-Small.

Backbone	Method	FP	8w8a	6w6a	5w5a	4w4a
ViT-Small	MoCov3	88.0	87.6	87.2	82.2	82.0
	MoCov3+SSQL	**88.6**	**88.6**	**88.3**	**88.2**	**86.9**

- Effectiveness when applied to other self-supervised methods? **Yes!!!**

Table 12: Linear evaluation results on CIFAR-10.

Backbone	Method	FP	6w6a	5w5a	4w4a	3w3a	2w4a
ResNet-18	BYOL	89.3	89.4	89.3	88.0	75.1	63.3
	BYOL+SSQL	**90.8**	**90.7**	**90.6**	**89.8**	**85.0**	**85.7**
	MoCov2	88.9	88.4	88.2	86.8	72.2	50.7
	MoCov2+SSQL	**89.6**	**89.6**	**89.5**	**88.5**	**83.4**	**85.2**

Q & A

Contrastive Learning - SimSiam

MEGVII 旷视

- **ImageNet Linear classification**
  - No large batch size
  - No negative pairs
  - No momentum encoder

method	batch size	negative pairs	momentum encoder	100 ep	200 ep	400 ep	800 ep
SimCLR (repro.+)	4096	✓		66.5	68.3	69.8	70.4
MoCo v2 (repro.+)	**256**	✓	✓	67.4	69.9	71.0	72.2
BYOL (repro.)	4096		✓	66.5	**70.6**	**73.2**	**74.3**
SwAV (repro.+)	4096			66.5	69.1	70.7	71.8
SimSiam	**256**			**68.1**	70.0	70.8	71.3

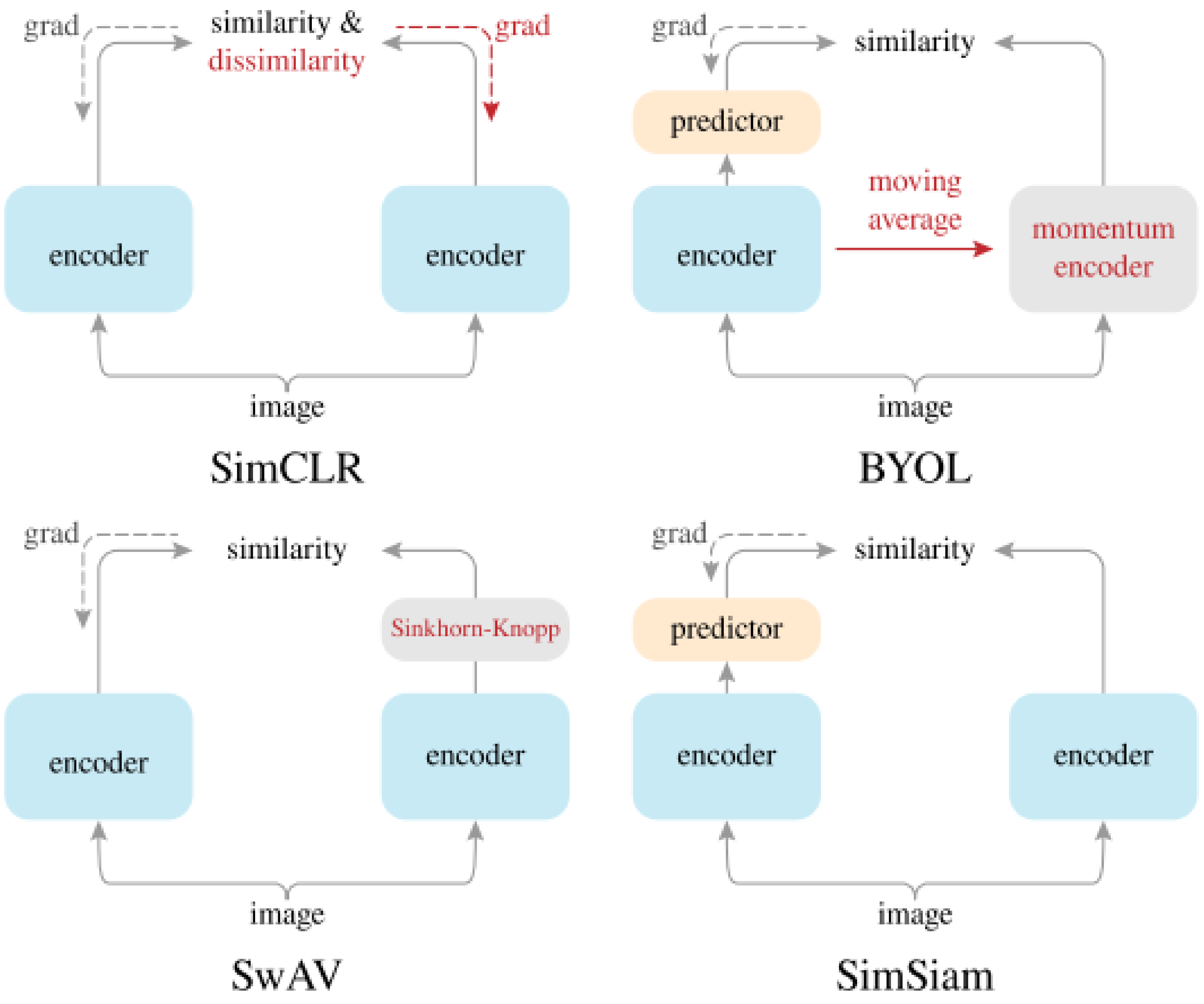


Figure 3. **Comparison on Siamese architectures**. The encoder includes all layers that can be shared between both branches. The dash lines indicate the gradient propagation flow. In BYOL, SwAV, and SimSiam, the lack of a dash line implies stop-gradient, and their symmetrization is not illustrated for simplicity. The components in red are those missing in SimSiam.

Contrastive Learning - SimCLR

- **Larger batch sizes & Longer training**
  - Larger batch sizes provide more negative examples, facilitating convergence
  - Training longer also provides more negative examples, improving the results

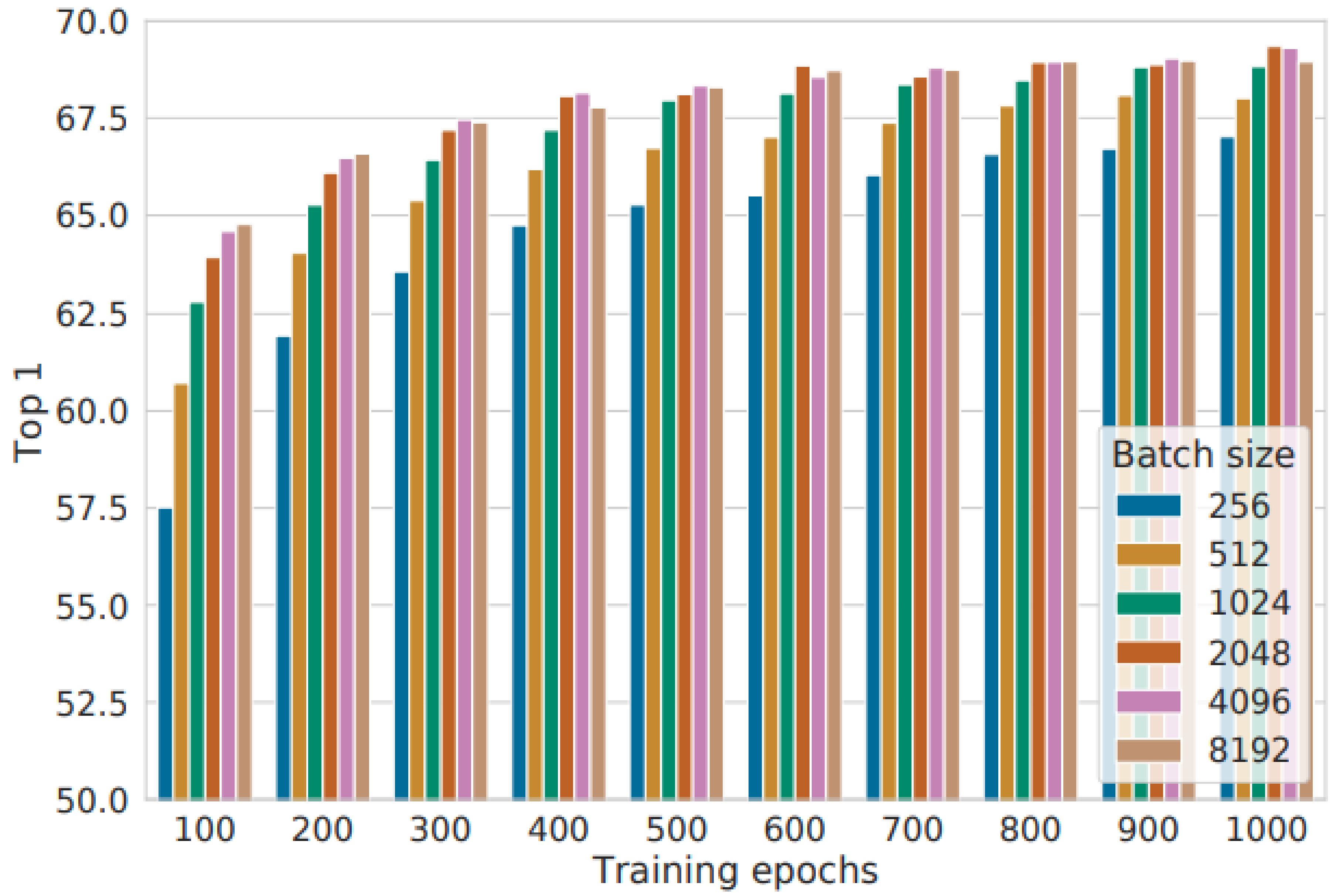


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

Contrastive Learning - MoCo

MEGVII 旷视

- **Experiments**
  - ImageNet supervised pre-training is most influential when serving as initialization for fine-tuning in downstream tasks.

pre-train	AP_{50}	AP	AP_{75}
random init.	64.4	37.9	38.6
super. IN-1M	81.4	54.0	59.1
MoCo IN-1M	81.1 (−0.3)	54.6 (+0.6)	59.9 (+0.8)
MoCo IG-1B	81.6 (+0.2)	55.5 (+1.5)	61.2 (+2.1)

(a) Faster R-CNN, R50-**dilated-C5**

Object detection fine-tuned on PASCAL VOC

pre-train	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
random init.	36.7	56.7	40.0	33.7	53.8	35.9
super. IN-1M	40.6	61.3	44.4	36.8	58.1	39.5
MoCo IN-1M	40.8 (+0.2)	61.6 (+0.3)	44.7 (+0.3)	36.9 (+0.1)	58.4 (+0.3)	39.7 (+0.2)
MoCo IG-1B	41.1 (+0.5)	61.8 (+0.5)	45.1 (+0.7)	37.4 (+0.6)	59.1 (+1.0)	40.2 (+0.7)

(b) Mask R-CNN, R50-**FPN**, **2**× schedule

Object detection fine-tuned on COCO

pre-train	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
random init.	26.4	44.0	27.8	29.3	46.9	30.8
super. IN-1M	38.2	58.2	41.2	33.3	54.7	35.2
MoCo IN-1M	38.5 (+0.3)	58.3 (+0.1)	41.6 (+0.4)	33.6 (+0.3)	54.8 (+0.1)	35.6 (+0.4)
MoCo IG-1B	39.1 (+0.9)	58.7 (+0.5)	42.2 (+1.0)	34.1 (+0.8)	55.4 (+0.7)	36.4 (+1.2)

(c) Mask R-CNN, R50-**C4**, **1**× schedule

Object segmentation fine-tuned on COCO

pre-train	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}
random init.	65.9	86.5	71.7
super. IN-1M	65.8	86.9	71.9
MoCo IN-1M	66.8 (+1.0)	87.4 (+0.5)	72.5 (+0.6)
MoCo IG-1B	66.9 (+1.1)	87.8 (+0.9)	73.0 (+1.1)

Keypoint detection fine-tuned on COCO