VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning

Mehmet F. Demirel

Oct 12, 2021

Some Examples

SimCLR

- Contrastive learning
- Positive/negative pairs
- Requires large batches

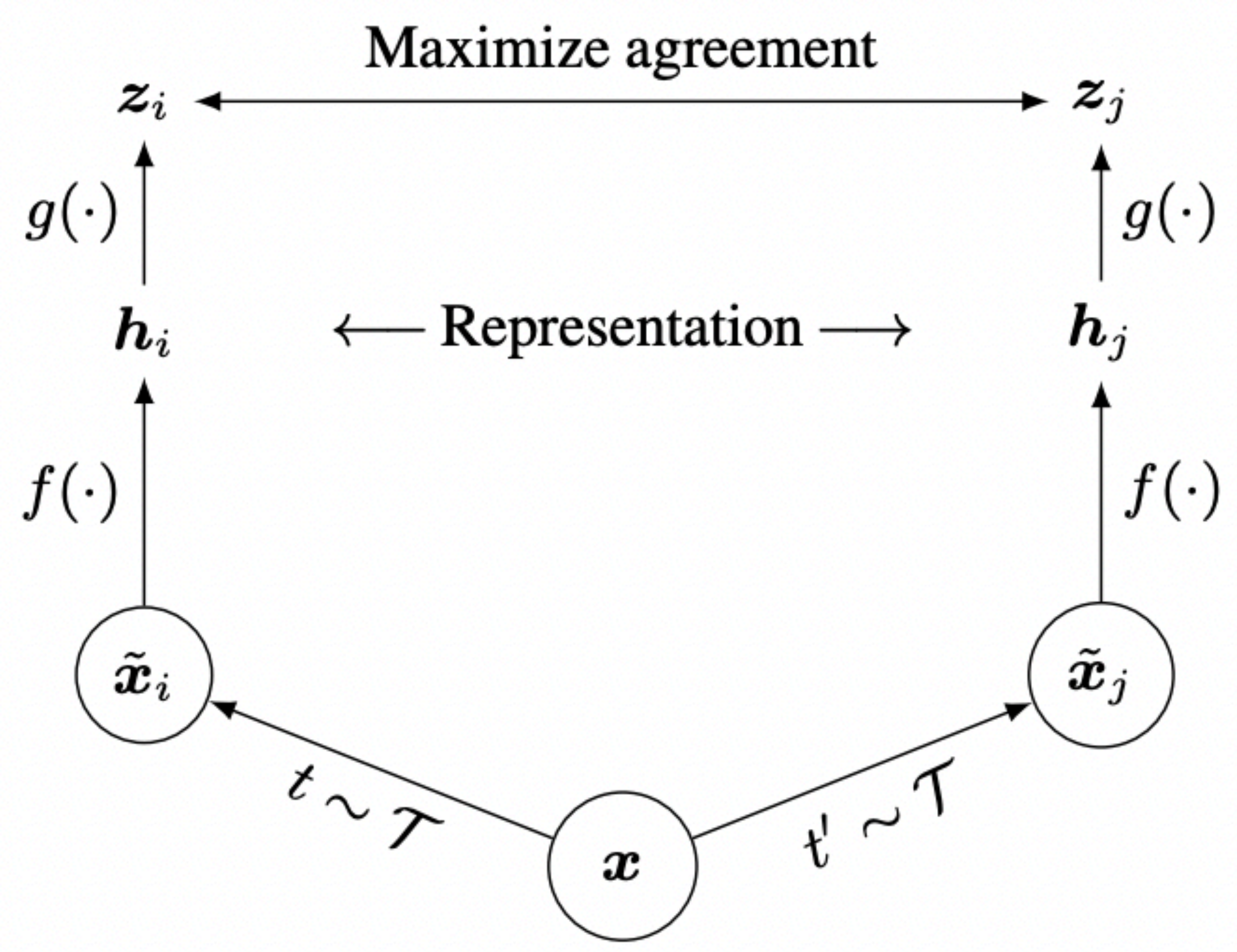


Figure 2. A simple framework for contrastive learning of visual representations. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder 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.

Some Examples

BYOL

- Target network is updated via a slow-moving average of the online network.
- No negative pairs.

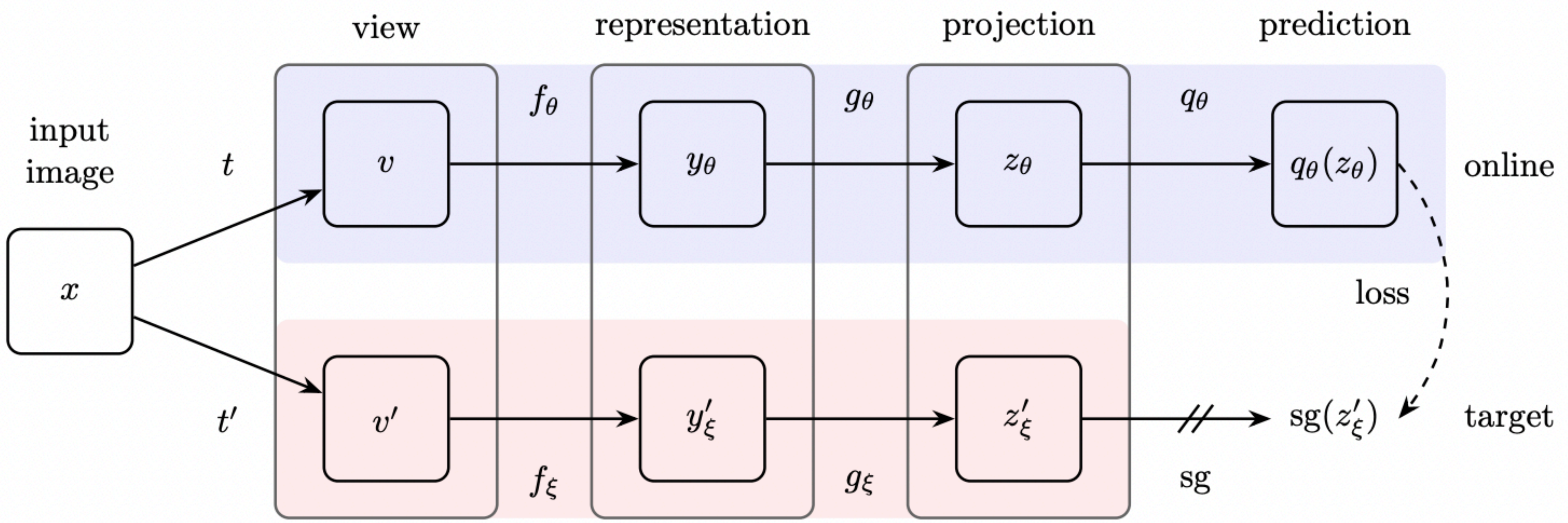


Figure 2: BYOL's architecture. BYOL minimizes a similarity loss between $q_\theta(z_\theta)$ and $\mathrm{sg}(z'_\xi)$, where θ are the trained weights, ξ are an exponential moving average of θ and sg means stop-gradient. At the end of training, everything but f_θ is discarded, and y_θ is used as the image representation.

$$\xi \leftarrow \tau\xi + (1-\tau)\theta.$$

Some Examples

SimSiam

- Weight sharing on two branches.
- Stop-gradient on one.
- No negative pairs
- No large batches
- No momentum encoders

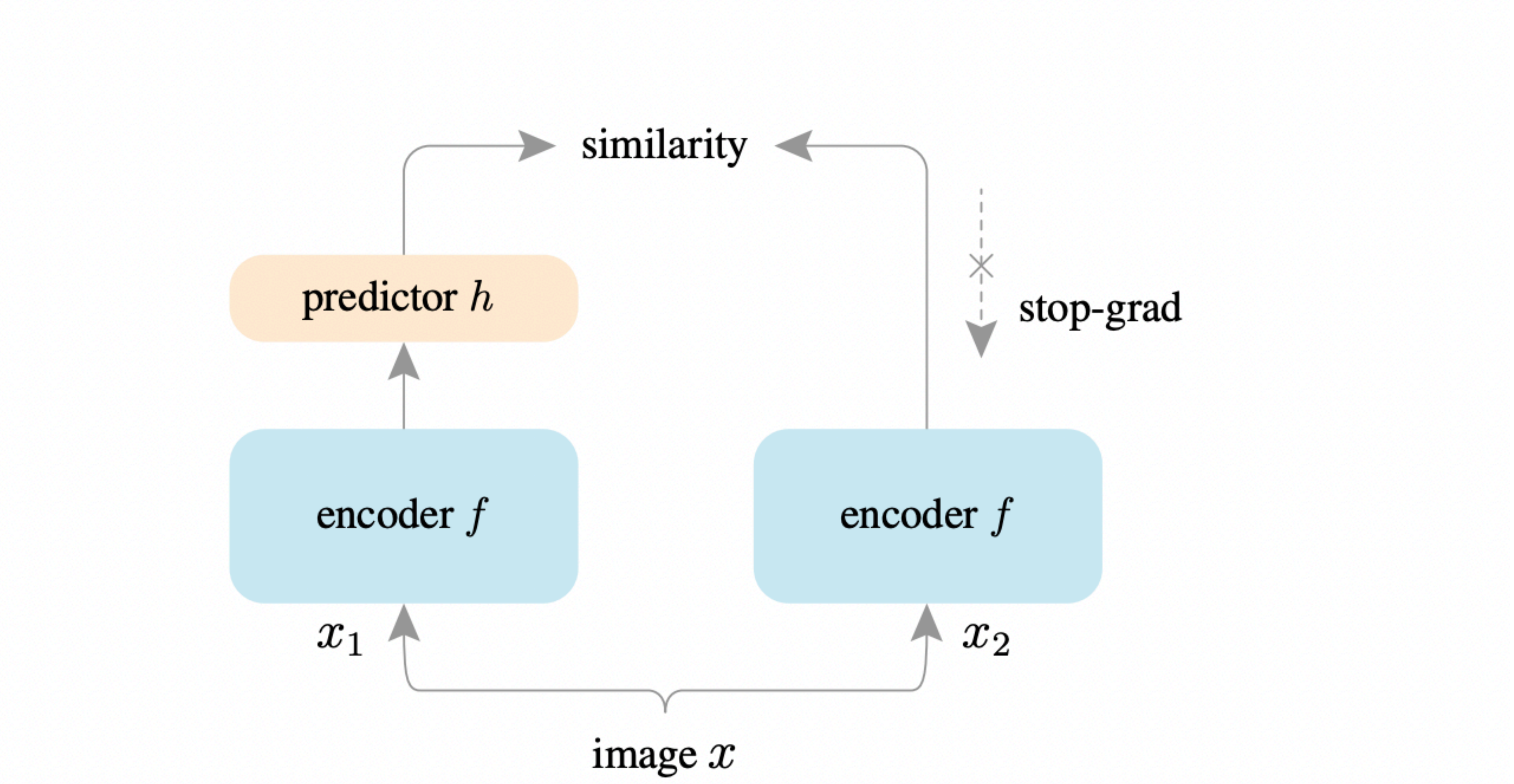
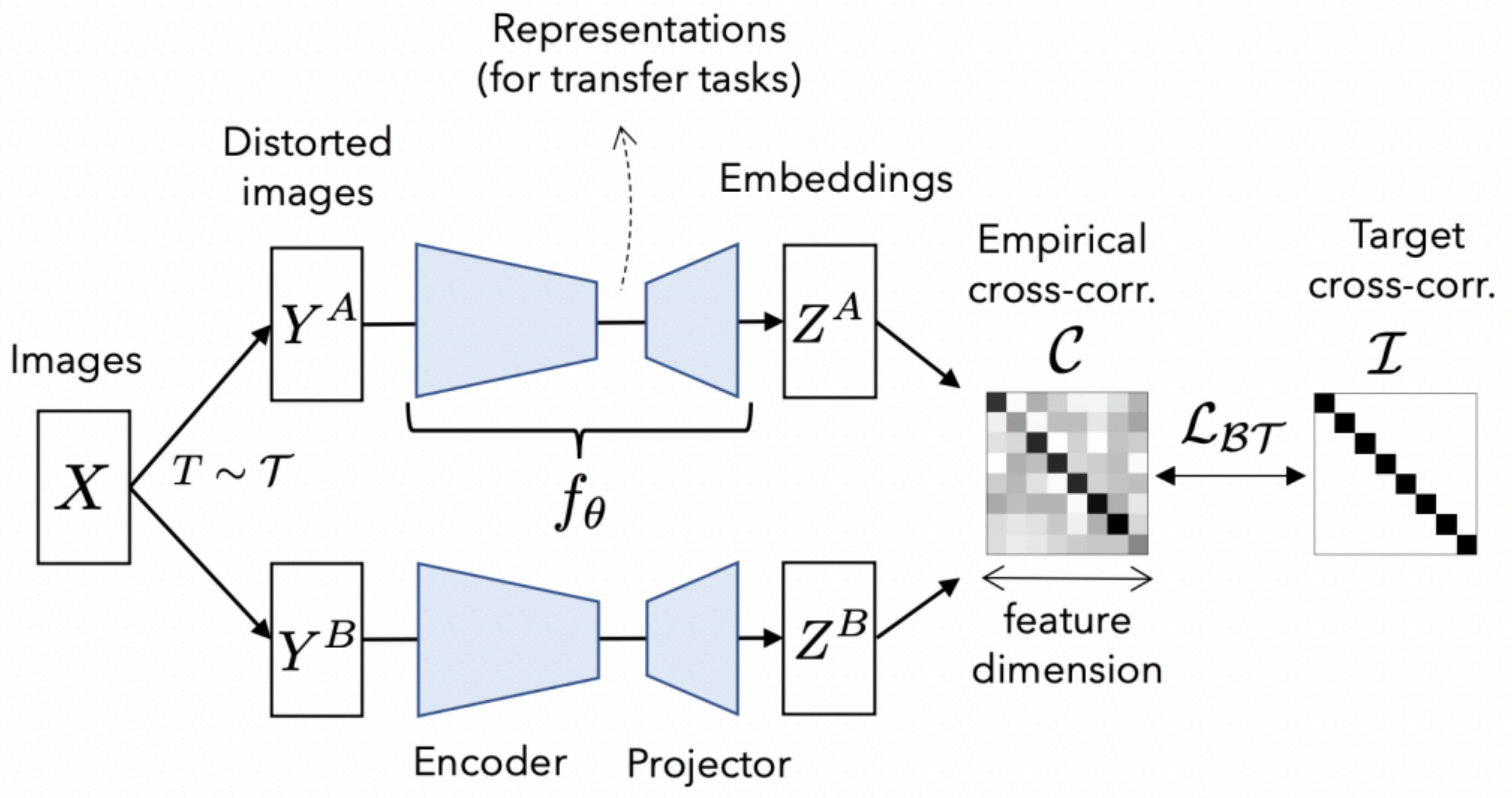


Figure 1. **SimSiam architecture**. Two augmented views of one image are processed by the same encoder network f (a backbone plus a projection MLP). Then a prediction MLP h is applied on one side, and a stop-gradient operation is applied on the other side. The model maximizes the similarity between both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

Some Examples

Barlow Twins

- Forces the cross-correlation matrix between the outputs of two identical networks towards identity.
- No large batches, stop-gradient, asymmetric networks, or momentum encoding.



$$\mathcal{L}_{\mathcal{BT}} \triangleq \underbrace{\sum_i (1-\mathcal{C}_{ii})^2}_{} + \lambda \underbrace{\sum_i \sum_{j \neq i} {\mathcal{C}_{ij}}^2}_{} \qquad (1)$$

$$\mathcal{C}_{ij} \triangleq \frac{\sum_b z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_b \left(z^A_{b,i}\right)^2} \sqrt{\sum_b \left(z^B_{b,j}\right)^2}} \qquad (2)$$

VICReg

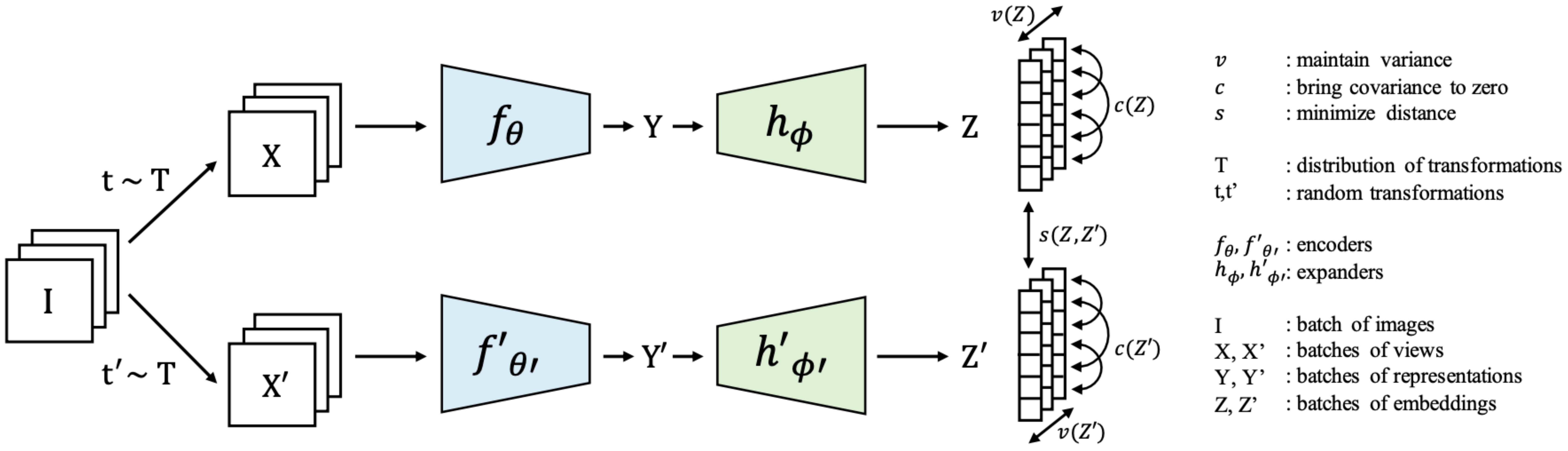
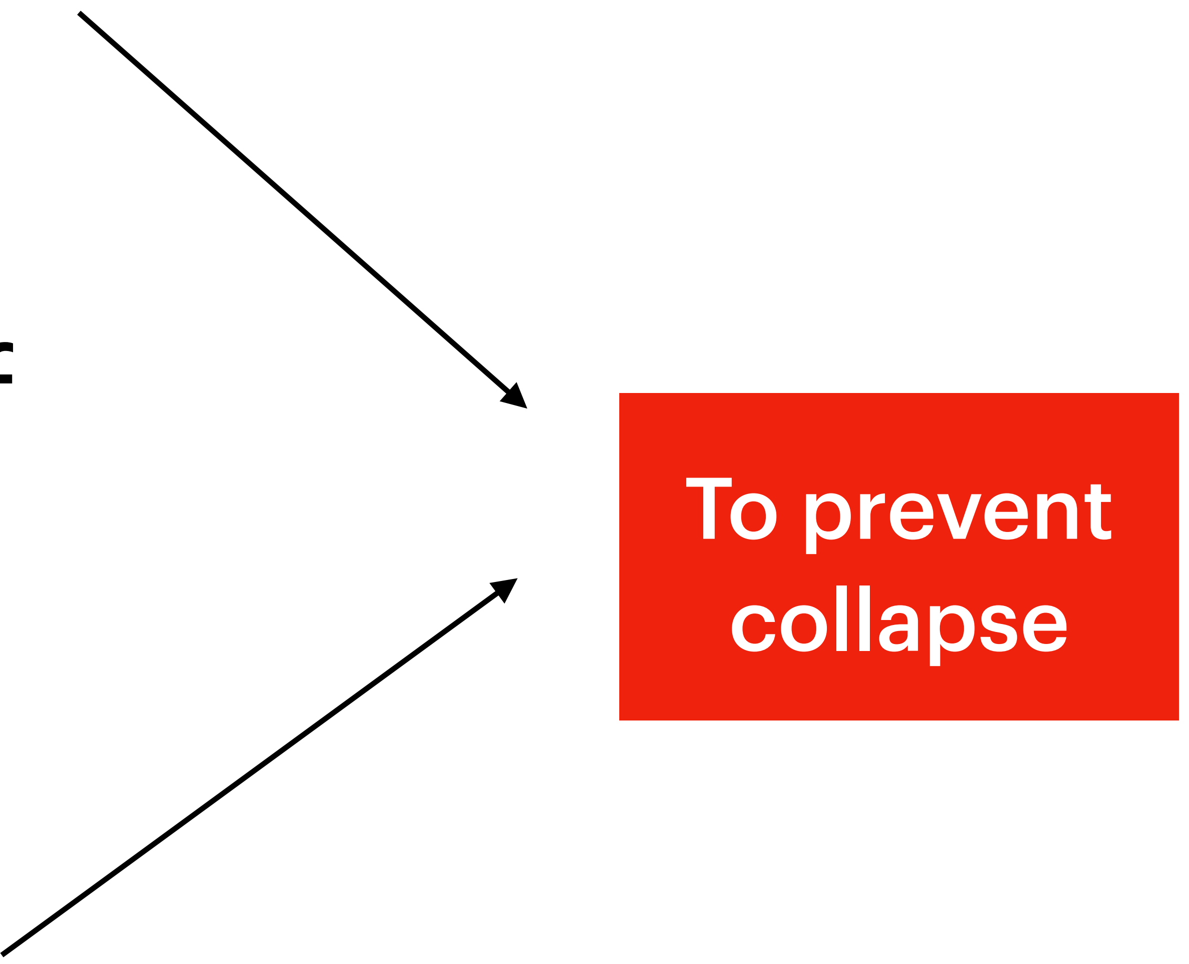

I
t ~ T
t′ ~ T
X
X′
f_θ
$f'_{\theta'}$
Y
Y′
h_ϕ
$h'_{\phi'}$
Z
Z′
$v(Z)$
$c(Z)$
$s(Z,Z')$
$c(Z')$
$v(Z')$
v : maintain variance
c : bring covariance to zero
s : minimize distance
T : distribution of transformations
t,t' : random transformations
$f_\theta, f'_{\theta'}$: encoders
$h_\phi, h'_{\phi'}$: expanders
I : batch of images
X, X' : batches of views
Y, Y' : batches of representations
Z, Z' : batches of embeddings

Figure 1: **VICReg: joint embedding architecture with variance, invariance and covariance regularization.** Given a batch of images I, two batches of different views X and X' are produced and are then encoded into representations Y and Y'. The representations are fed to an expander producing the embeddings Z and Z'. The distance between two embeddings from the same image is minimized, the variance of each embedding variable over a batch is maintained above a threshold, and the covariance between pairs of embedding variables over a batch are attracted to zero, decorrelating the variables from each other. Although the two branches do not require identical architectures nor share weights, in most of our experiments, they are Siamese with shared weights: the encoders are ResNet-50 backbones with output dimension 2048. The expanders have 3 fully-connected layers of size 8192.

VICReg

- **Variance**: a constraint on the embedded vectors along each dimension so that the variance in each dimension is close to some value.
- **Invariance**: force the embeddings from different views of the same image to be close to each other
- **Covariance**: prevent the network from encoding similar information in different dimensions in the embedded space.



To prevent collapse

VICReg

- **Variance**: $$v(Z) = \frac{1}{d}\sum_{j=1}^{d} \max(0, \gamma - S(z^j, \epsilon)), \quad S(x, \epsilon) = \sqrt{\mathrm{Var}(x) + \epsilon},$$

- **Invariance**: $$s(Z, Z') = \frac{1}{n}\sum_{i} \|z_i - z'_i\|_2^2.$$

- **Covariance**: $$C(Z) = \frac{1}{n-1}\sum_{i=1}^{n}(z_i - \bar{z})(z_i - \bar{z})^T, \quad \text{where} \quad \bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i.$$

$$c(Z) = \frac{1}{d}\sum_{i \neq j}[C(Z)]_{i,j}^2.$$

VICReg

- **Variance**: $$v(Z) = \frac{1}{d} \sum_{j=1}^{d} \max(0, \gamma - S(z^j, \epsilon)), \quad S(x, \epsilon) = \sqrt{\mathrm{Var}(x) + \epsilon},$$

- **Invariance**: $$s(Z, Z') = \frac{1}{n} \sum_{i} \|z_i - z'_i\|_2^2.$$

- **Covariance**: $$C(Z) = \frac{1}{n-1} \sum_{i=1}^{n} (z_i - \bar{z})(z_i - \bar{z})^T, \quad \text{where} \quad \bar{z} = \frac{1}{n} \sum_{i=1}^{n} z_i.$$

$$c(Z) = \frac{1}{d} \sum_{i \neq j} [C(Z)]_{i,j}^2.$$

Loss: $$\ell(Z, Z') = \lambda s(Z, Z') + \mu[v(Z) + v(Z')] + \nu[c(Z) + c(Z')],$$

Main Results

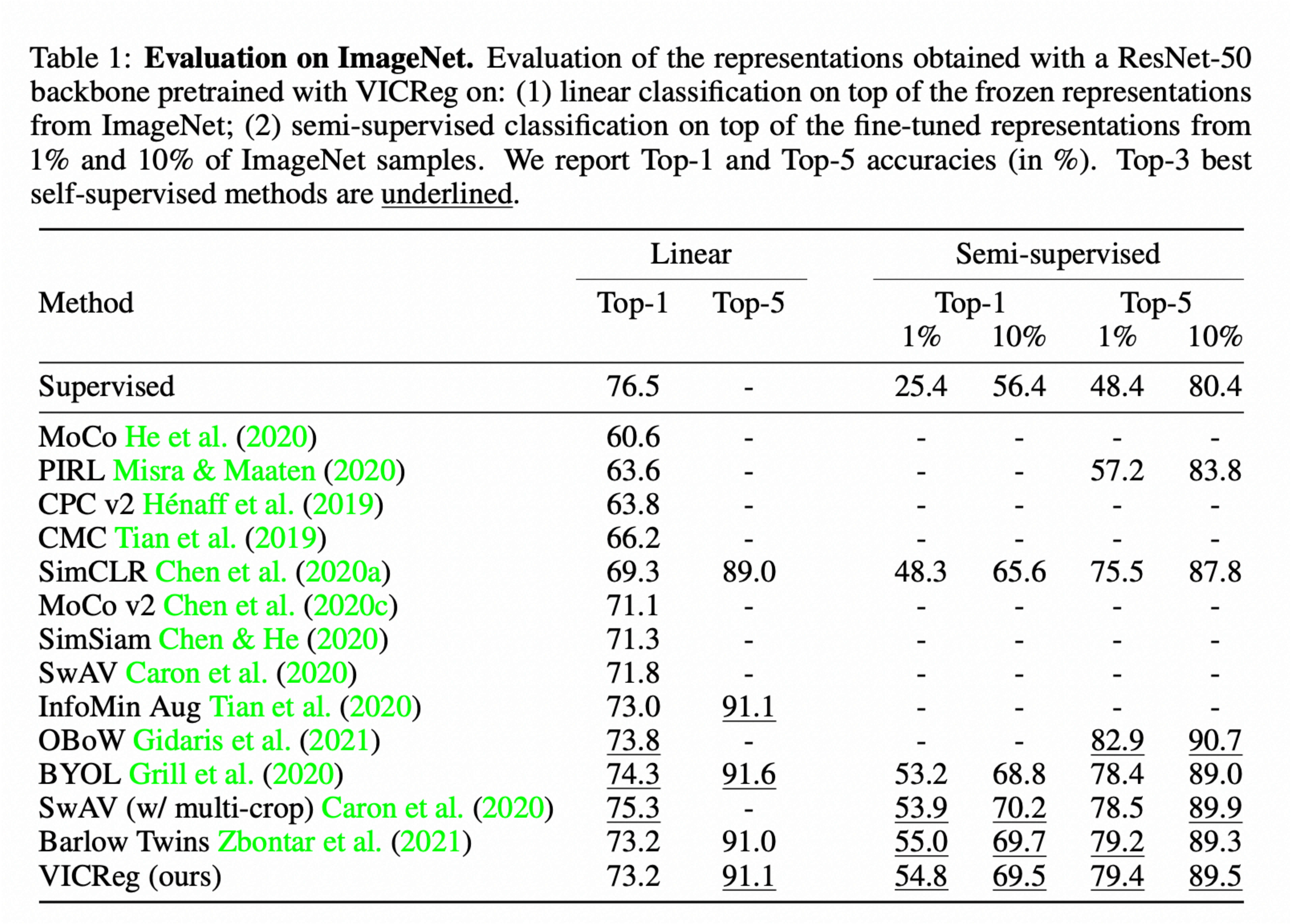


Table 1: **Evaluation on ImageNet.** Evaluation of the representations obtained with a ResNet-50 backbone pretrained with VICReg on: (1) linear classification on top of the frozen representations from ImageNet; (2) semi-supervised classification on top of the fine-tuned representations from 1% and 10% of ImageNet samples. We report Top-1 and Top-5 accuracies (in %). Top-3 best self-supervised methods are underlined.

	Linear		Semi-supervised			
Method	Top-1	Top-5	Top-1		Top-5	
			1%	10%	1%	10%
Supervised	76.5	-	25.4	56.4	48.4	80.4
MoCo He et al. (2020)	60.6	-	-	-	-	-
PIRL Misra & Maaten (2020)	63.6	-	-	-	57.2	83.8
CPC v2 Hénaff et al. (2019)	63.8	-	-	-	-	-
CMC Tian et al. (2019)	66.2	-	-	-	-	-
SimCLR Chen et al. (2020a)	69.3	89.0	48.3	65.6	75.5	87.8
MoCo v2 Chen et al. (2020c)	71.1	-	-	-	-	-
SimSiam Chen & He (2020)	71.3	-	-	-	-	-
SwAV Caron et al. (2020)	71.8	-	-	-	-	-
InfoMin Aug Tian et al. (2020)	73.0	91.1	-	-	-	-
OBoW Gidaris et al. (2021)	73.8	-	-	-	82.9	90.7
BYOL Grill et al. (2020)	74.3	91.6	53.2	68.8	78.4	89.0
SwAV (w/ multi-crop) Caron et al. (2020)	75.3	-	53.9	70.2	78.5	89.9
Barlow Twins Zbontar et al. (2021)	73.2	91.0	55.0	69.7	79.2	89.3
VICReg (ours)	73.2	91.1	54.8	69.5	79.4	89.5

Transfer Learning

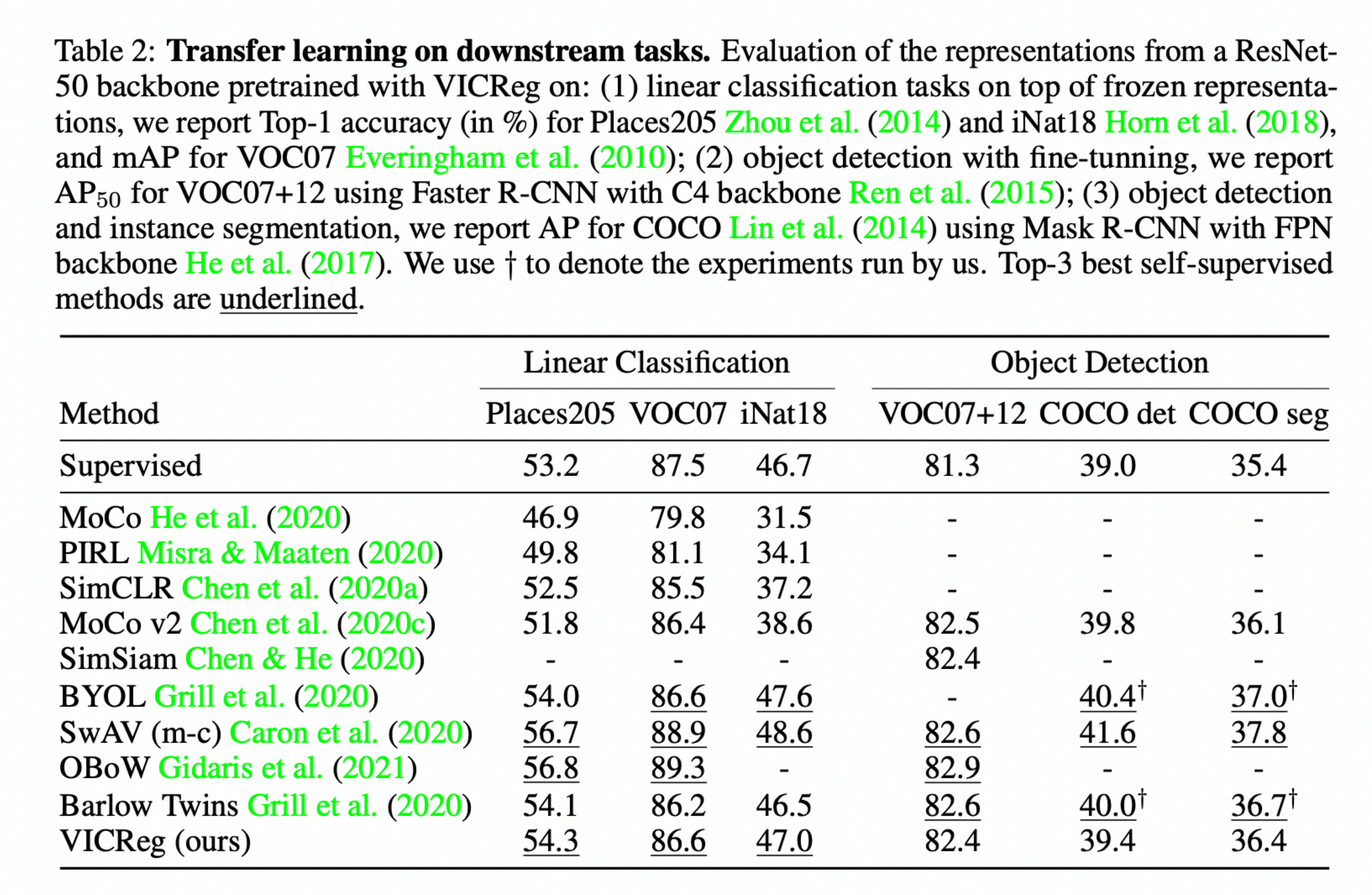


Table 2: **Transfer learning on downstream tasks.** Evaluation of the representations from a ResNet-50 backbone pretrained with VICReg on: (1) linear classification tasks on top of frozen representations, we report Top-1 accuracy (in %) for Places205 Zhou et al. (2014) and iNat18 Horn et al. (2018), and mAP for VOC07 Everingham et al. (2010); (2) object detection with fine-tunning, we report AP_{50} for VOC07+12 using Faster R-CNN with C4 backbone Ren et al. (2015); (3) object detection and instance segmentation, we report AP for COCO Lin et al. (2014) using Mask R-CNN with FPN backbone He et al. (2017). We use † to denote the experiments run by us. Top-3 best self-supervised methods are underlined.

	Linear Classification			Object Detection		
Method	Places205	VOC07	iNat18	VOC07+12	COCO det	COCO seg
Supervised	53.2	87.5	46.7	81.3	39.0	35.4
MoCo He et al. (2020)	46.9	79.8	31.5	-	-	-
PIRL Misra & Maaten (2020)	49.8	81.1	34.1	-	-	-
SimCLR Chen et al. (2020a)	52.5	85.5	37.2	-	-	-
MoCo v2 Chen et al. (2020c)	51.8	86.4	38.6	82.5	39.8	36.1
SimSiam Chen & He (2020)	-	-	-	82.4	-	-
BYOL Grill et al. (2020)	54.0	86.6	47.6	-	$40.4^{\dagger}$	$37.0^{\dagger}$
SwAV (m-c) Caron et al. (2020)	56.7	88.9	48.6	82.6	41.6	37.8
OBoW Gidaris et al. (2021)	56.8	89.3	-	82.9	-	-
Barlow Twins Grill et al. (2020)	54.1	86.2	46.5	82.6	$40.0^{\dagger}$	$36.7^{\dagger}$
VICReg (ours)	54.3	86.6	47.0	82.4	39.4	36.4

V-C applied on other methods

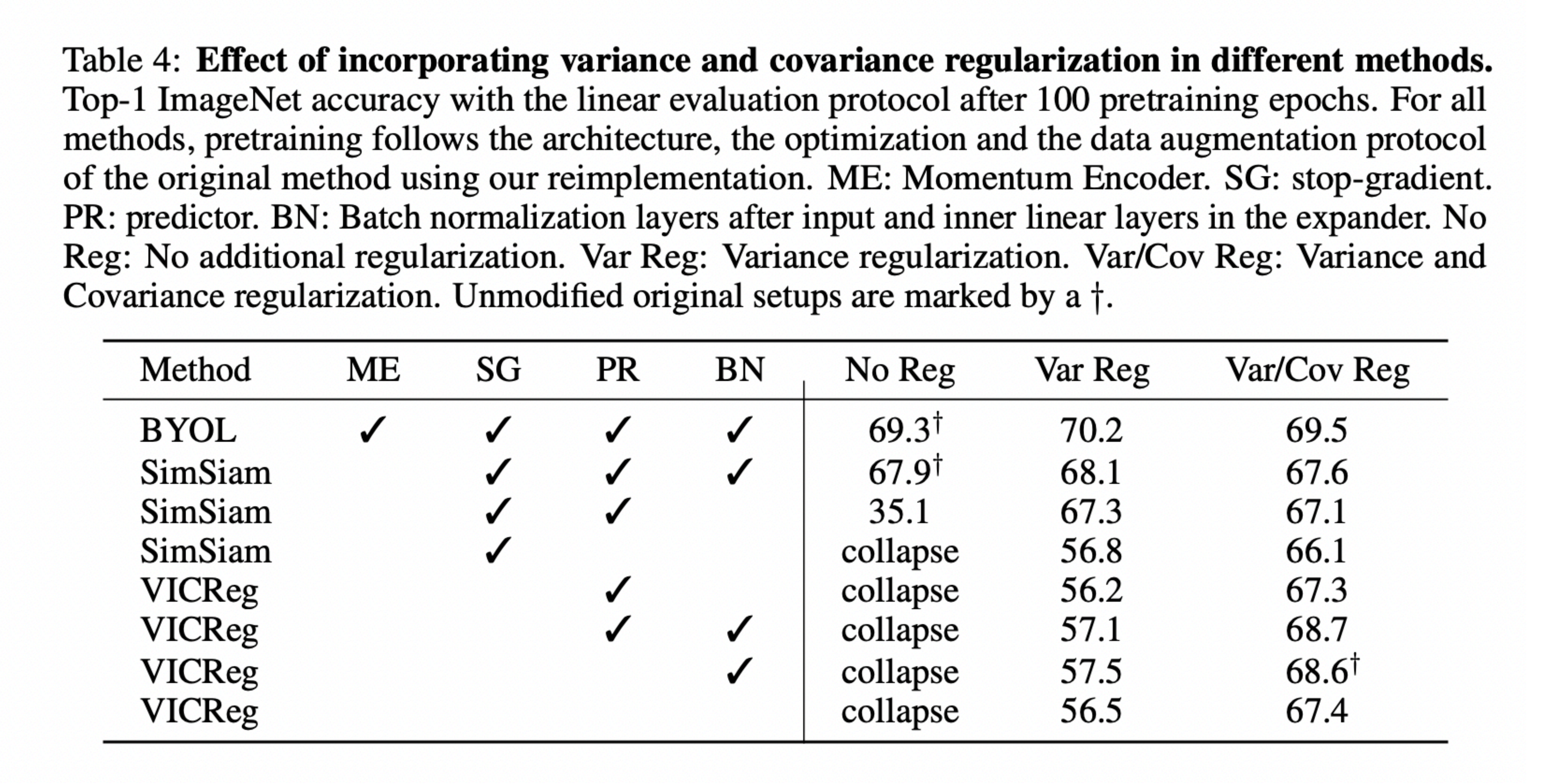


Table 4: **Effect of incorporating variance and covariance regularization in different methods.** Top-1 ImageNet accuracy with the linear evaluation protocol after 100 pretraining epochs. For all methods, pretraining follows the architecture, the optimization and the data augmentation protocol of the original method using our reimplementation. ME: Momentum Encoder. SG: stop-gradient. PR: predictor. BN: Batch normalization layers after input and inner linear layers in the expander. No Reg: No additional regularization. Var Reg: Variance regularization. Var/Cov Reg: Variance and Covariance regularization. Unmodified original setups are marked by a †.

Method	ME	SG	PR	BN	No Reg	Var Reg	Var/Cov Reg
BYOL	✓	✓	✓	✓	69.3†	70.2	69.5
SimSiam		✓	✓	✓	67.9†	68.1	67.6
SimSiam		✓	✓		35.1	67.3	67.1
SimSiam		✓			collapse	56.8	66.1
VICReg			✓		collapse	56.2	67.3
VICReg			✓	✓	collapse	57.1	68.7
VICReg				✓	collapse	57.5	68.6†
VICReg					collapse	56.5	67.4

Effect of different parts

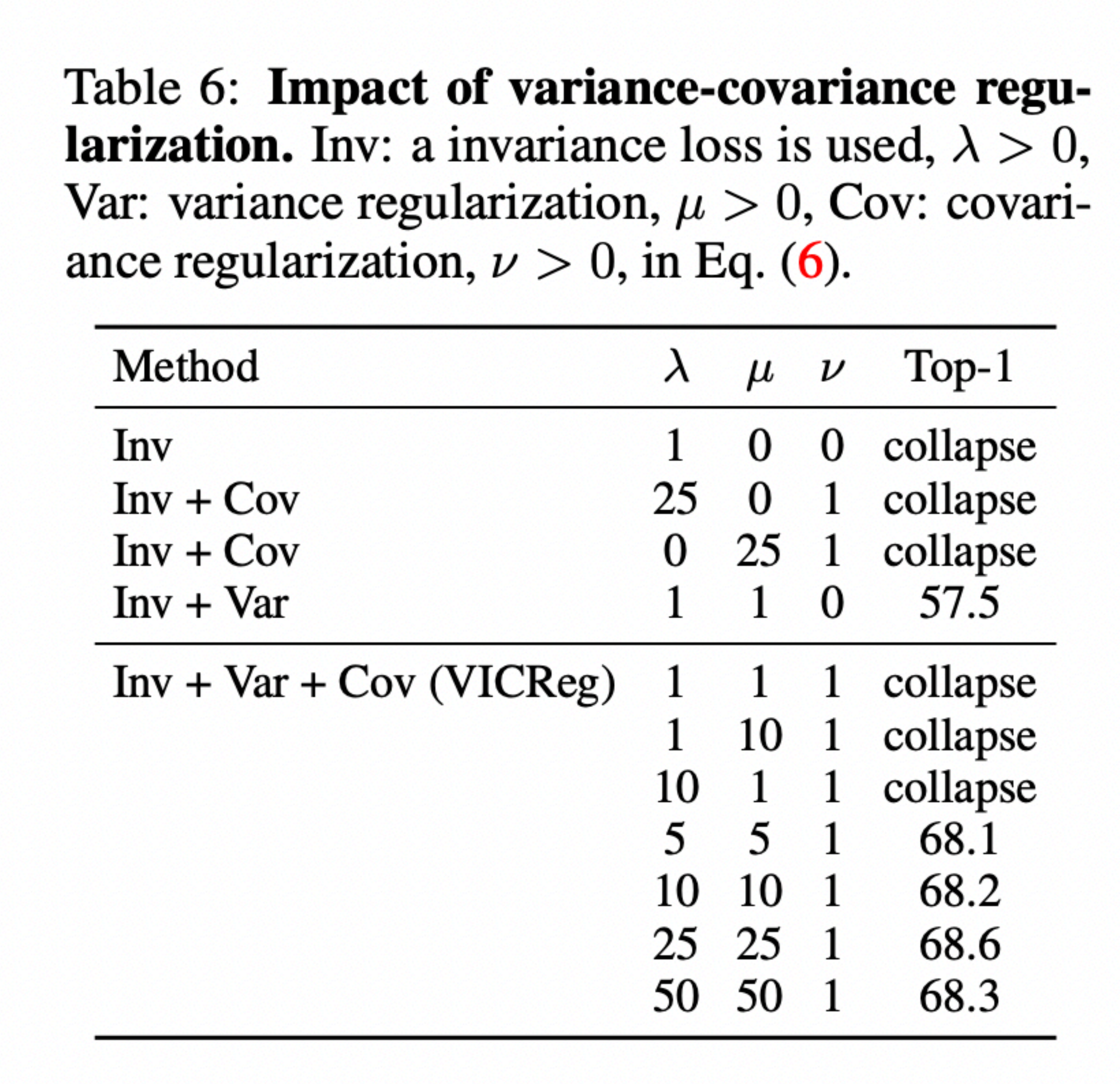


Table 6: **Impact of variance-covariance regularization.** Inv: a invariance loss is used, $\lambda > 0$, Var: variance regularization, $\mu > 0$, Cov: covariance regularization, $\nu > 0$, in Eq. (6).

Method	λ	μ	ν	Top-1
Inv	1	0	0	collapse
Inv + Cov	25	0	1	collapse
Inv + Cov	0	25	1	collapse
Inv + Var	1	1	0	57.5
Inv + Var + Cov (VICReg)	1	1	1	collapse
	1	10	1	collapse
	10	1	1	collapse
	5	5	1	68.1
	10	10	1	68.2
	25	25	1	68.6
	50	50	1	68.3