BYOL works *even* without batch statistics

Pierre H. Richemond[*,1,2] **Jean-Bastien Grill**[*,1] **Florent Altché**[*,1] **Corentin Tallec**[*,1] **Florian Strub**[*,1]

Andrew Brock[1] **Samuel Smith**[1] **Soham De**[1] **Razvan Pascanu**[1]

Bilal Piot[1] **Michal Valko**[1]

[1]DeepMind [2]Imperial College

phr17@ic.ac.uk [jbgrill,fstrub,altche,corentint]@google.com

Abstract

Bootstrap Your Own Latent (BYOL) is a self-supervised learning approach for image representation. From an augmented view of an image, BYOL trains an online network to predict a target network representation of a different augmented view of the same image. Unlike contrastive methods, BYOL does not explicitly use a repulsion term built from *negative pairs* in its training objective. Yet, it avoids collapse to a trivial, constant representation. Thus, it has recently been hypothesized that batch normalization (BN) is critical to prevent collapse in BYOL. Indeed, BN flows gradients across batch elements, and could leak information about negative views in the batch, which could act as an implicit negative (contrastive) term. However, we experimentally show that replacing BN with a batch-independent normalization scheme (namely, a combination of group normalization and weight standardization) achieves performance comparable to vanilla BYOL (73.9% vs. 74.3% top-1 accuracy under the linear evaluation protocol on ImageNet with ResNet-50). Our finding disproves the hypothesis that the use of batch statistics is a crucial ingredient for BYOL to learn useful representations.

1 Introduction

Self-supervised image representation methods [1, 2, 3, 4] have achieved downstream performance that rivals those of supervised pre-training on ImageNet [5]. Current self-supervised methods rely on image transformations to generate different *views* from an input image while preserving semantic information. Among the most successful algorithms, contrastive methods [6, 7, 8, 1, 3, 9] use a loss function that balances out two terms: a term associated to the positive pairs (that we refer to as the *positive* term) encouraging representations from views of the same image to be similar, and a term associated to negative pairs (a *negative* term) which encourages representations to be spread out.

Taking a different route, other approaches manage to avoid the contrastive paradigm [10, 11, 12, 13]. Among them, BYOL [4] learns its representation by predicting the target network representation of a view from the online representation of another view of the same image. However, such a setup has obvious *collapsed* equilibria where the representation is constant, and thus can be predicted from any input. This has raised the question of how BYOL could even work without a negative term nor an explicit mechanism to prevent collapse. Experimental reports [14, 15] suggest that the use of batch normalization, BN [16], in BYOL's network is crucial to achieve good performance. These reports hypothesise that the BN used in BYOL's network could implicitly introduce a negative term.

[*]Equal contribution; the order of first authors was randomly selected.

Preprint. Under review.

We experimentally confirm the particular importance of BN in BYOL: removing all instances of BN in the network prevents BYOL from learning anything at all in the classic setting, see Section 3.1 and Table 1. However, our experimental results given in Table 2 go against some interpretations proposed notably in [14, 15]. In particular, we **refute the following hypotheses**:

- (H1) *BYOL needs BN because BN provides an implicit negative term required to avoid collapse.* In Section 3.2, we show that BYOL avoids collapse and achieves 65.7% top-1 accuracy on ImageNet under the linear evaluation protocol [17] *without* using any normalization during training, by using both a better initialization scheme and retaining the additional trainable parameters scaling and bias (γ and β) introduced by BN.

  Therefore, unlike (H1), we hypothesize that the main role of BN is to make the network more robust to cases when the initialization is scaled improperly. Indeed, proper initialization is critical for deep nets [18, 19, 20, 21] and BYOL suffers from a bad initialization in two ways: (i) as for any deep network, it makes optimization difficult and (ii) BYOL's target network outputs will be ill-conditioned, which initially provides poor targets for the online network.
- (H2) *BYOL cannot achieve competitive performance without the implicit contrastive effect provided by batch statistics.* In Section 3.3, we show that most of this performance gap—65.7% achieved without BN vs. 74.3% achieved *with* BN—can be bridged without using batch statistics. Specifically, if we replace BN with a combination of group normalization, GN [22], and weight standardization, WS [23], while keeping standard initializations, BYOL achieves 73.9% top-1 accuracy.

2 Background

In this section, we adopt the notation of [4]. Recall that x denotes an image and $v = t(x)$ and $v' = t'(x)$ are two views of x obtained from two independent transformations t and t' sampled from a distribution $\mathcal{T}$. These views are used as input to an encoder network to obtain representations $y_\theta = f_\theta(v)$ and $y'_\theta = f_\theta(v')$; and projections $z_\theta = g_\theta(y_\theta)$ and $z'_\theta = g_\theta(y'_\theta)$. We continue by a brief recap of standard contrastive methods and BYOL.

InfoNCE Most contrastive methods use variants of the InfoNCE [6] loss to train their representation,

$$\text{InfoNCE}_\theta = \underbrace{-\frac{\langle z_\theta, z'_\theta \rangle}{\tau \cdot \|z_\theta\| \cdot \|z'_\theta\|}}_{\text{positive term}} + \underbrace{\log \sum_i \exp \frac{\langle z_\theta, z^i_\theta \rangle}{\tau \cdot \|z_\theta\| \cdot \|z^i_\theta\|}}_{\text{negative term}},$$

where the z^i_θ are projections from views of all images (including z'_θ but not z_θ), and τ is a temperature parameter. The first term of this loss (the *positive* term) encourages projections of views of the same image to become similar, while the second term (the *negative* term) makes projections of views from different images more dissimilar. Such a loss has a strong theoretical underpinning: minimizing this loss is equivalent to maximizing a lower bound on the mutual information between the representation of two views [24] which is tight when the function approximator is sufficiently expressive.

BYOL BYOL trains its representation using both an online network (parameterized by θ) and a target network (parameterized by ξ). As a part of the online network, it further defines a predictor network q_θ that is used to predict target projections z'_ξ using online projections z_θ as inputs. Accordingly, the parameters of the online projection are updated following the gradients of the prediction loss

$$\text{BYOL}_\theta = -\frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\|q_\theta(z_\theta)\| \cdot \|z'_\xi\|}.$$

In turn, the target network weights ξ are updated as an exponential moving average of the online network's weights, i.e. $\xi \leftarrow (1-\eta)\xi + \eta\theta$, with η being a decay parameter. As $q_\theta(z_\theta)$ is a function of v and z'_ξ is a function of v', BYOL's loss can be seen as a measure of similarity between the views v and v' and therefore resembles the positive term of the InfoNCE loss.

Group normalization (GN) GN [22] is an activation normalization method, like BN [16], layer normalization (LN [25]), and instance normalization (IN [26]). For an activation tensor X of dimensions (N, H, W, C), GN first splits channels into G equally-sized groups, then normalizes activations with

the mean and standard deviation computed over disjoint slices of size $(1, H, W, C/G)$. The number of groups G thus trades off between normalization over all channels ($G = 1$, equivalent to `LN`), and normalization over a single one ($G = C$, equivalent to `IN`). Importantly, `GN` operates independently on each batch element and therefore *it does not rely on batch statistics*.

Weight standardization (WS) `WS` normalizes the weights corresponding to each activation using weight statistics. Each row of the weight matrix W is normalized to get a new weight matrix $\widehat{W}$ which is directly used in place of W during training. Only the normalized weights $\widehat{W}$ are used to compute convolution outputs but the loss is differentiated with respect to non-normalized weights W,

$$\widehat{W}_{i,j} = \frac{W_{i,j} - \mu_i}{\sigma_i}, \quad \text{with} \quad \mu_i = \frac{1}{\mathcal{I}} \sum_{j=1}^{\mathcal{I}} W_{i,j} \quad \text{and} \quad \sigma_i = \sqrt{\varepsilon + \frac{1}{\mathcal{I}} \sum_{j=1}^{\mathcal{I}} (W_{i,j} - \mu_i)^2},$$

where $\mathcal{I}$ is the input dimension (product of input channel dimension and kernel spatial dimension); we set $\varepsilon = 10^{-4}$. Contrary to `BN`, `LN`, and `GN`, `WS` does not create additional trainable weights.

3 Experimental results

While many metrics can be used to evaluate self-supervised representations, we focus on classification accuracy on ImageNet [5] under the standard linear evaluation protocol [17] with a ResNet-50 architecture, with the same setup as [6, 8, 1, 4]. Unless otherwise specified, we follow the training setup and hyperparameters described in [4] when training `BYOL`.

3.1 Removing `BN` causes collapse

In Table 1, we explore the impact of using different normalization schemes in `SimCLR` and `BYOL`, by using either `BN`, `LN`, or removing normalization in each component of `BYOL` and `SimCLR`, i.e., the encoder, the projector (for `SimCLR` and `BYOL`), and the predictor (for `BYOL` only). First, we observe that removing all instances of `BN` in `BYOL` leads to performance that is no better than random. Noticeably, this is specific to `BYOL` as `SimCLR` still performs reasonably well in this regime. Nevertheless, solely applying `BN` to the ResNet encoder is enough for `BYOL` to achieve high performance.[2]

From these observations, [14] hypothesizes that `BN` implicitly introduces a negative contrastive term, which acts as a crucial component to stabilize training (H1). This hypothesis may seem further supported by the performance difference between `SimCLR` and `BYOL` when replacing `BN` (which uses batch statistics) with `LN` which does not.

However, we observe that `BN` seems to be mainly useful in the ResNet encoder, for which standard initializations are known to lead to poor conditioning [28, 29]. Also `BYOL` might be even more affected by improper initialization as it creates its own targets. Rather than (H1), we therefore hypothesize that the main contribution of `BN` in `BYOL` is to compensate for improper initialization.

Table 1: *Ablation results on normalization, per network component:* The numbers correspond to top-1 linear accuracy (%), 300 epochs on ImageNet, averaged over 3 seeds.

Encoder	BN				LN				-				-		
Projector	BN		-		LN		-		BN		LN		-		
Predictor	BN	-	BN	-	LN	-	LN	-	BN	-	LN	-	BN	LN	-
`BYOL`	73.2	73.2	72.0	72.1	0.1	5.4	0.1	0.1	62.6	0.1	0.1	0.1	61.1	0.1	0.1
`SimCLR`	69.3		68.5		68.0		67.8		53.8[3]		56.7		0.1		

[2] Some of these observations differ from the ones initially reported in [14]. Specifically, the authors observed a collapse when removing `BN` in `BYOL`'s predictor and projector. This difference could be linked to the use of the `SGD` optimizer instead of `LARS` [27].

[3] Unstable in late training: three seeds ending at 48.4%, 57.9%, 56.1%.

3.2 Proper initialization allows working without BN

To confirm this assumption, we design the following protocol to mimic the effect of BN on initial scalings and training dynamics, without using or backpropagating through batch statistics. Before training, we compute per-activation BN statistics for each layer by running a single forward pass of the network with BN on a batch of augmented data. We then remove then batch normalization layers, but retain the scale and offset parameters γ and β trainable, and initialize them as

$$\gamma^k_{\text{init}} = \frac{\gamma^k_0}{\sigma^k} \qquad \text{and} \qquad \beta^k_{\text{init}} = -\mu^k \cdot \gamma^k_{\text{init}},$$

where γ^k_{init} and β^k_{init} are the initialization of the additional trainable parameters corresponding to the k-th removed BN, and μ^k and σ^k are the batch statistics computed during the first pass for the k-th removed BN. Additionally, we set $\gamma^k_0 = 0$ if the k-th removed BN corresponds to the final BN layer in a residual block, and $\gamma^k_0 = 1$ otherwise. This is similar to what is done in [30], except that we further rescale the initialization of the scale and offset parameters by a data-dependent quantity, in the spirit of [31]. Such setup keeps the initial scaling effect of BN, while avoiding the computation of any batch statistic during training, thus discarding any potential implicit contrastive term.

We use the exact same hyperparameters as for vanilla BYOL (*i.e.*, base learning rate of 0.2, weight decay of $1.5 \cdot 10^{-6}$ and decay rate of 0.996), except that we increase the number of warmup epochs from 10 to 50. After 1000 epochs, this representation achieves 65.7% top-1 accuracy in the linear evaluation setting compared to 74.3% for the baseline. These results are reported in Table 2.

Despite its comparatively low performance, the trained representation still provides considerably better classification results than a random ResNet-50 backbone, and is thus necessarily not collapsed. This confirms that BYOL does not need BN to prevent collapse. It also confirms that one of the effects of BN is to provide better initial scalings and training dynamics, and that, contrary to SimCLR, these are required for BYOL to perform well.

Table 2: *Summary of our results*: top-1 accuracy with linear evaluation on ImageNet, at 1000 epochs.

BYOL variant	Vanilla BN	No BN	Modified init.	GN + WS
Uses batch statistics	Yes	No	No	No
Top-1 accuracy (%)	74.3	0.1	65.7	73.9

3.3 Using GN with WS leads to competitive performance

In the previous section, we have shown that BYOL can learn a non-collapsed representation without using BN. Yet, BYOL performs worse in this regime. This only disproves (H1), but BN could still both provide better initial scaling *and* an implicit contrastive term, responsible for some of the performance. To study this hypothesis, we explore other refined element-wise normalization procedures. More precisely, we apply weight standardization to convolutional and linear parameters by weight standardized alternatives, and replace all BN by GN layers.

To train the network, we use the same hyperparameters as in BYOL except for the weight decay, set to $3 \cdot 10^{-8}$ instead of $1.5 \cdot 10^{-6}$, the base learning rate set to 0.24 instead of 0.2 and the target update rate, set to 0.999 instead of 0.996; we also set the number of groups for GN to $G = 16$. With this setup, BYOL (+GN +WS) achieves 73.9% top-1 accuracy after 1000 epochs.

As neither GN nor WS compute batch statistics, this version of BYOL cannot compare elements from the batch, and therefore it likewise cannot implement a batch-wise implicit contrastive mechanism. Therefore, we experimentally show that BYOL can maintain most of its performance even without a hypothetical implicit contrastive term provided by BN.

4 Conclusion

Unlike contrastive methods, the loss used in BYOL does not explicitly include a negative term that would encourage its representations to spread apart. Nonetheless, BYOL's representation does not collapse during training, and BN has been hypothesized to fill the crucial role of an implicit negative

term by leaking batch statistics into the gradient. We refute this hypothesis, and show that BYOL can achieve competitive results without using batch statistics. In particular, BYOL achieves 65.7% top-1 accuracy when removing BN and changing the initialization. Moreover, BYOL achieves a competitive 73.9% top-1 accuracy by replacing BN with a normalization scheme operating element-wise.

Acknowledgement

The authors would like to thank the following people for their help throughout the process of writing this paper, in alphabetical order: Jean-Baptiste Alayrac, Bernardo Avila Pires, Nathalie Beauguerlange, Elena Buchatskaya, Jeffrey De Fauw, Sander Dieleman, Carl Doersch, Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Olivier Henaff, Koray Kavukcuoglu, Pauline Luc, Katrina McKinney, Rémi Munos, Aaron van den Oord, Jason Ramapuram, Adria Recasens, Karen Simonyan, Oriol Vinyals and the DeepMind team. We would like to also thank the authors of the following papers for fruitful discussions: [1, 9, 14, 32].

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of International Conference on Machine Learning (ICML)*, 2020.

[2] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[3] Rishabh Jain, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[4] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[6] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[7] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aäron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of International Conference on Machine Learning (ICML)*, 2019.

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[9] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[10] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017.

[11] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of European Conference on Computer Vision (ECML)*, 2018.

[12] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[13] Uri Shaham, Kelly Stanton, Henry Li, Ronen Basri, Boaz Nadler, and Yuval Kluger. Spectralnet: Spectral clustering using deep neural networks. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[14] Abe Fetterman and Josh Albrecht. Understanding self-supervised and contrastive learning with bootstrap your own latent (BYOL). *https://untitled-ai.github.io/understanding-self-supervised-contrastive-learning.html*, 2020.

[15] Yuandong Tian, Lantao Yu, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning with dual deep networks, 2020.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of International Conference on Machine Learning (ICML)*, 2015.

[17] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *Proceedings of European Conference on Computer Vision (ECML)*, 2016.

[18] Dmytro Mishkin and Jiri Matas. All you need is a good init, 2016.

[19] Ge Yang and Samuel Schoenholz. Mean field residual networks: On the edge of chaos. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[20] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S. Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[21] Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Residual learning without normalization via better initialization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[22] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECML)*, 2018.

[23] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Weight standardization. *arXiv preprint arXiv:1903.10520*, 2019.

[24] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. In *Proceedings of International Conference on Machine Learning (ICML)*, 2019.

[25] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[26] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[27] Yang You, Igor Gitman, and Boris Ginsburg. Scaling SGD batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*, 2017.

[28] Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[29] Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. A mean field theory of batch normalization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[30] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[31] Philipp Krähenbühl, Carl Doersch, Jeff Donahue, and Trevor Darrell. Data-dependent initializations of convolutional neural networks. *arXiv preprint arXiv:1511.06856*, 2015.

[32] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with momentum predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.