Revisiting Self-Supervised Visual Representation Learning

Alexander Kolesnikov*, Xiaohua Zhai*, Lucas Beyer*

Google Brain
Zu¨rich, Switzerland {akolesnikov,xzhai,lbeyer}@google.com

Presented by: Xi Fang
Date: 9/25/2019

Content

1. Background and target
2. Related works
3. Experiments and results
4. Conclusion

Background and target

Background

1. Unsupervised visual representation learning remains a largely unsolved problem in computer vision research.

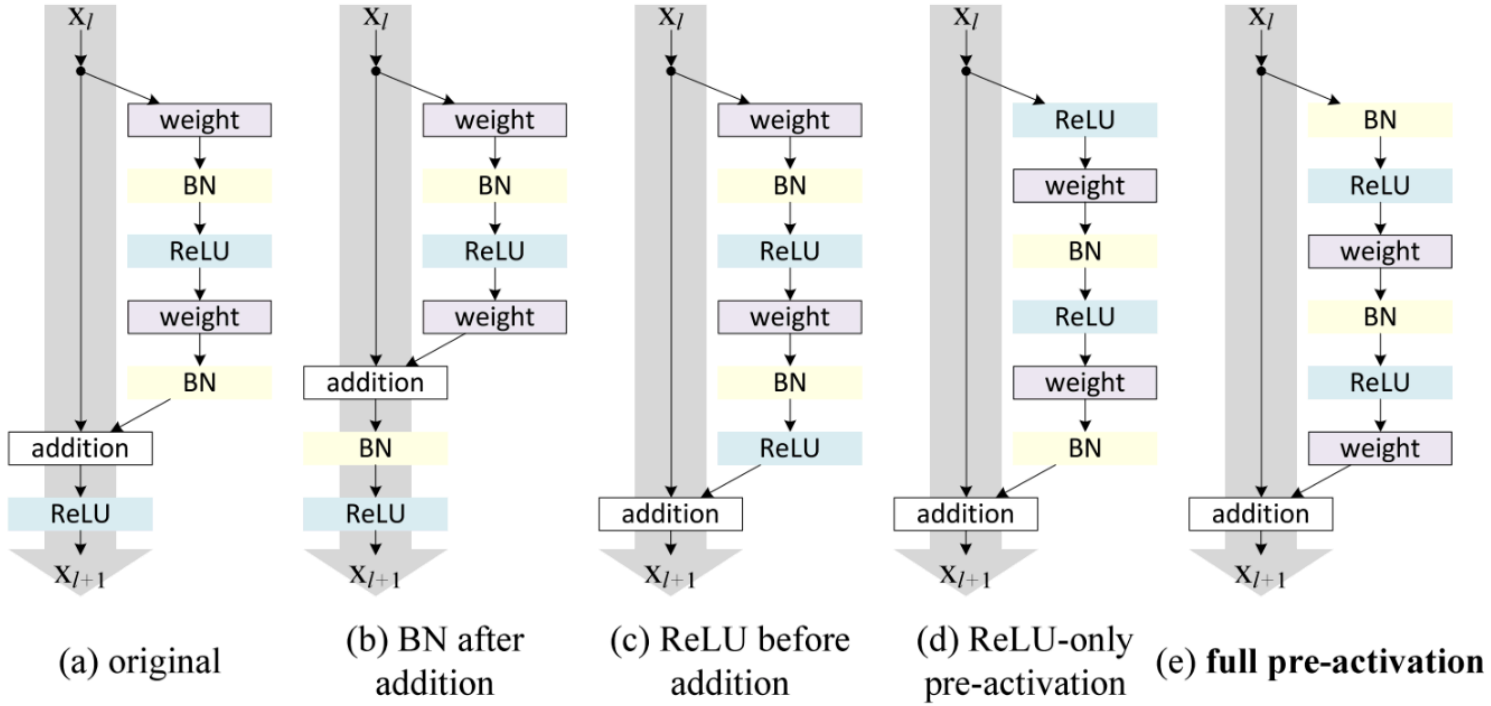
2. A large number of the pretext tasks for self-supervised learning have been studied, but other important aspects, such as the choice of convolutional neural networks (CNN), has not received equal attention.

Target:

Revisit numerous previously proposed self-supervised models, conduct a thorough large scale study to uncover multiple crucial insights

Related works

- Self supervised learning methods:
  - Rotation
  - Exemplar
  - Jigsaw
  - Relative patch location
- Popular networks:
  - ResNet
  - ResNet-v1
  - ResNet-v2
  - RevNet
  - VGG19



variants of residual blocks

Experiments and Results

- Datasets:
  - ImageNet
    - 1.3million natural images that represent 1000 various semantic classes
    - 50 000 images in the official validation and test sets
    - To avoid overfitting to the official validation split, we report numbers on our own validation split (50 000 random images from the training split)
  - Place205
    - 2.5million images depicting 205 different scene types such as *airfield*, *kitchen*, *coast*, etc.
    - Qualitatively different from *ImageNet*

Experiments and Results

Table 1. Evaluation of representations from self-supervised techniques based on various CNN architectures. The scores are accuracies (in %) of a linear logistic regression model trained on top of these representations using *ImageNet* training split. Our validation split is used for computing accuracies. The architectures marked by a "(-)" are slight variations described in Section 3.1. Sub-columns such as 4× correspond to widening factors. Top-performing architectures in a column are bold; the best pretext task for each model is underlined.

Model	Rotation				Exemplar			RelPatchLoc		Jigsaw	
	4×	8×	12×	16×	4×	8×	12×	4×	8×	4×	8×
RevNet50	**47.3**	**50.4**	**53.1**	**53.7**	**42.4**	45.6	46.4	40.6	45.0	40.1	43.7
ResNet50 v2	43.8	47.5	47.2	47.6	**43.0**	45.7	46.6	42.2	46.7	38.4	41.3
ResNet50 v1	41.7	43.4	43.3	43.2	**42.8**	**46.9**	**47.7**	**46.8**	**50.5**	**42.2**	**45.4**
RevNet50 (-)	45.2	**51.0**	**52.8**	**53.7**	38.0	42.6	44.3	33.8	43.5	36.1	41.5
ResNet50 v2 (-)	38.6	44.5	47.3	48.2	33.7	36.7	38.2	38.6	43.4	32.5	34.4
VGG19-BN	16.8	14.6	16.6	22.7	26.4	28.3	29.0	28.5	29.4	19.8	21.1

Experiments and Results

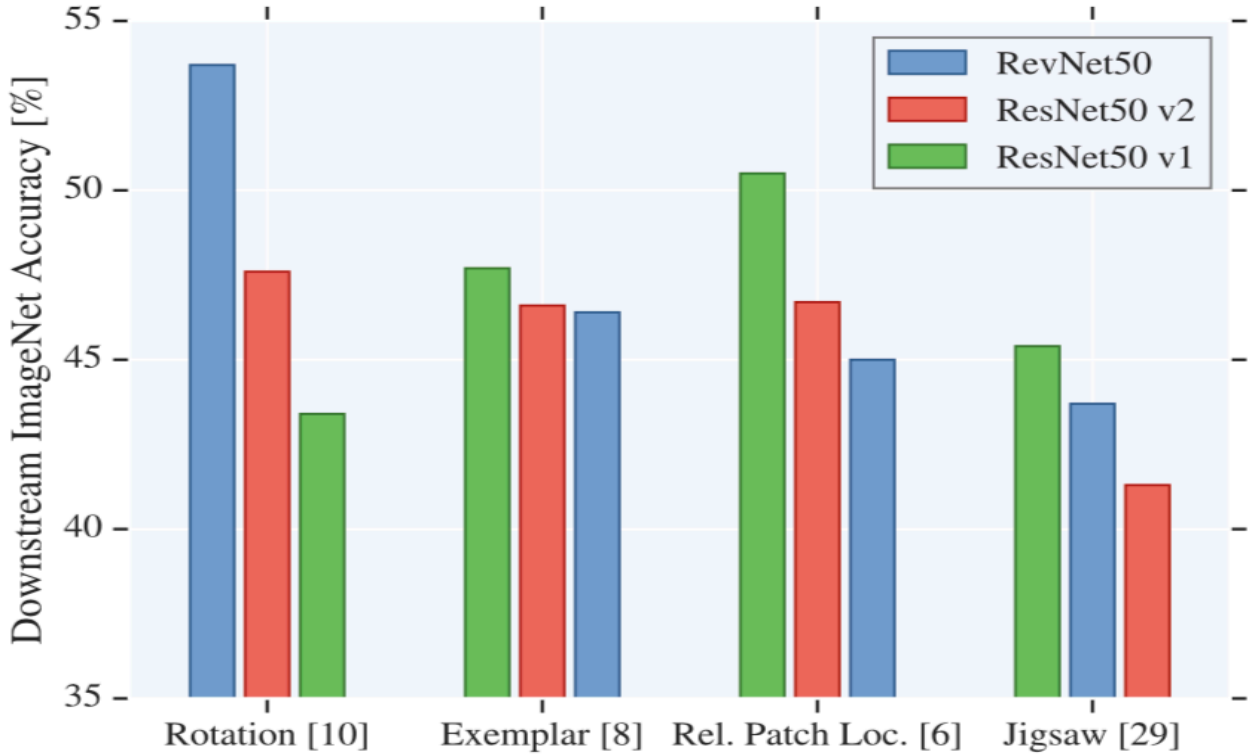


Figure 1. Quality of visual representations learned by various self-supervised learning techniques significantly depends on the convolutional neural network architecture that was used for solving the self-supervised learning task. In our paper we provide a large scale in-depth study in support of this observation and discuss its implications for evaluation of self-supervised models.

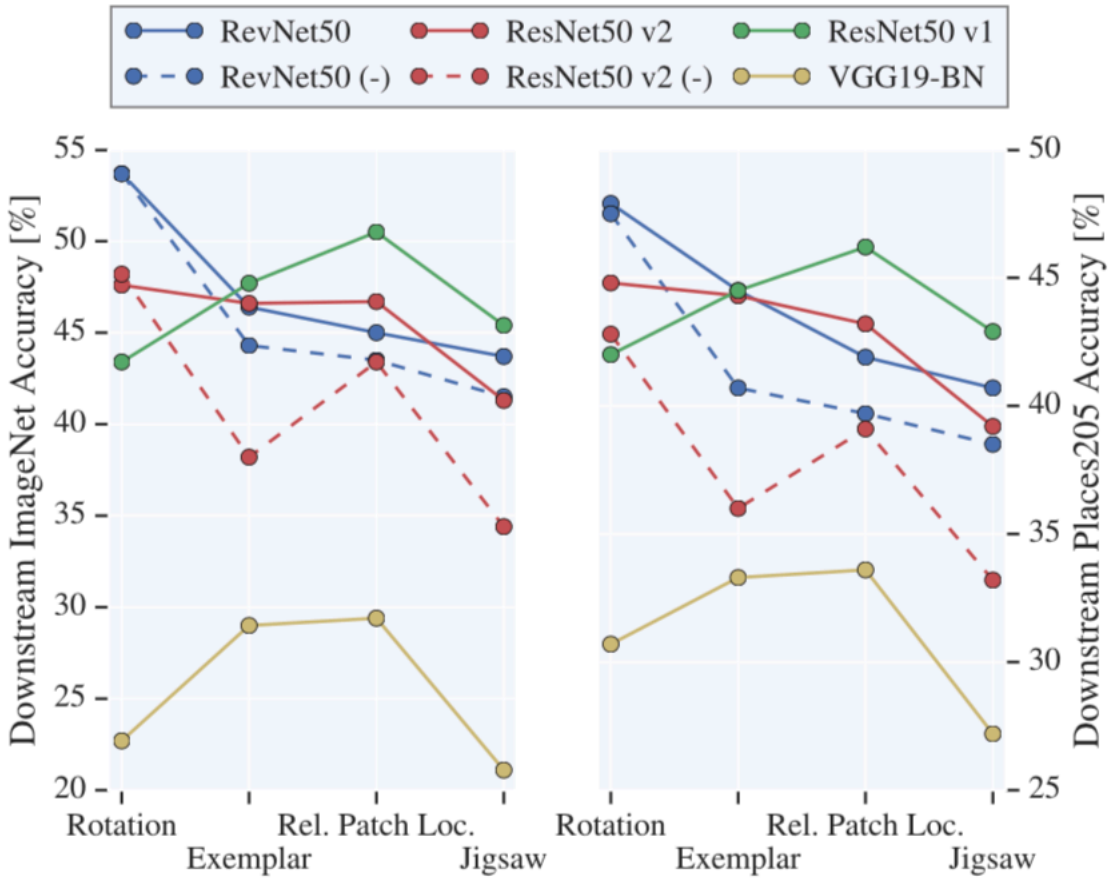


Figure 2. Different network architectures perform significantly differently across self-supervision tasks. This observation generalizes across datasets: *ImageNet* evaluation is shown on the left and *Places205* is shown on the right.

- Neither is the ranking of architectures consistent across different methods, nor is the ranking of methods consistent across architectures.

Experiments and Results

Family		ImageNet		Places205	
		Prev.	Ours	Prev.	Ours
A	Rotation[11]	38.7	**55.4**	35.1	**48.0**
R	Exemplar[8]	31.5	46.0	-	42.7
R	Rel. Patch Loc.[8]	36.2	51.4	-	45.3
A	Jigsaw[34, 51]	34.7	44.6	35.5	42.2
V	CC+vgg-Jigsaw++[36]	37.3	-	37.5	-
A	Counting[35]	34.3	-	36.3	-
A	Split-Brain[51]	35.4	-	34.1	-
V	DeepClustering[3]	**41.0**	-	**39.8**	-
R	CPC[37]	48.7†	-	-	-
R	Supervised RevNet50	74.8	74.4	-	58.9
R	Supervised ResNet50 v2	76.0	75.8	-	61.6
V	Supervised VGG19	72.7	75.0	58.9	61.5

† *marks results reported in unpublished manuscripts.*

Table 2. Comparison of the published self-supervised models to our best models. The scores correspond to accuracy of linear logistic regression that is trained on top of representations provided by self-supervised models. Official validation splits of *ImageNet* and *Places205* are used for computing accuracies. The "Family" column shows which basic model architecture was used in the referenced literature: **A**lexNet, **V**GG-style, or **R**esidual.

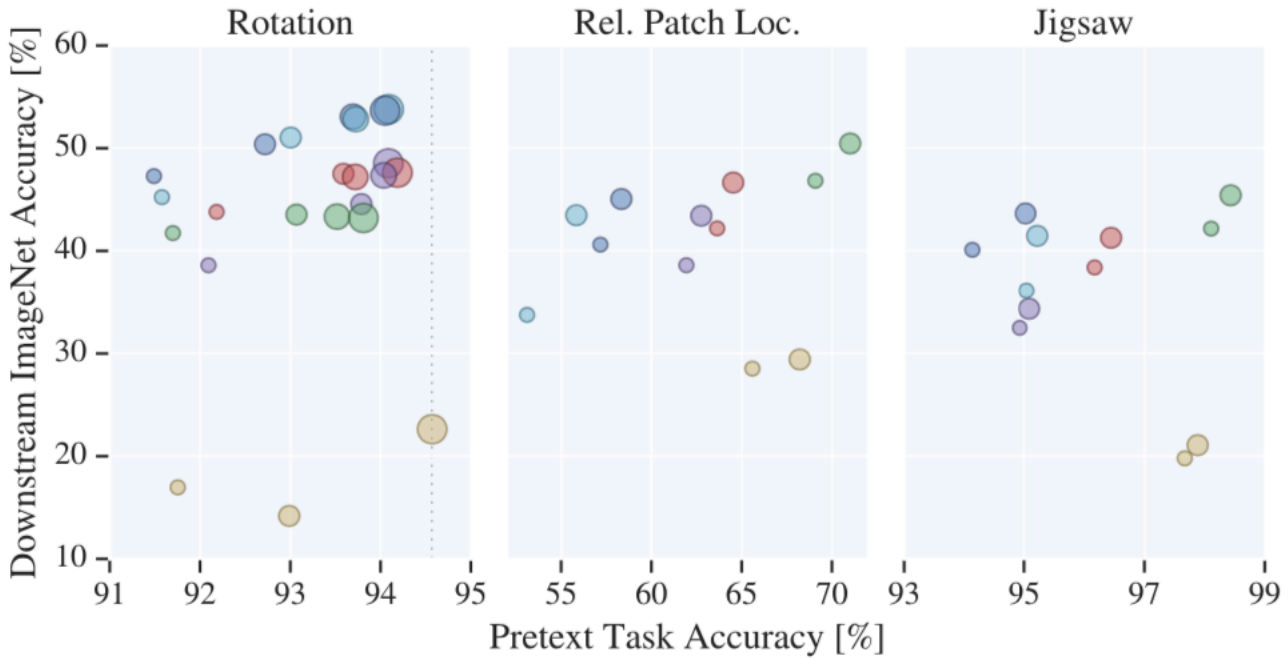


Figure 4. A look at how predictive pretext performance is to eventual downstream performance. Colors correspond to the architectures in Figure 3 and circle size to the widening factor k. Within an architecture, pretext performance is somewhat predictive, but it is not so across architectures. For instance, according to pretext accuracy, the widest *VGG* model is the best one for *Rotation*, but it performs poorly on the downstream task.

- Increasing the number of channels in CNN models improves performance of self-supervised models.

Experiments and Results

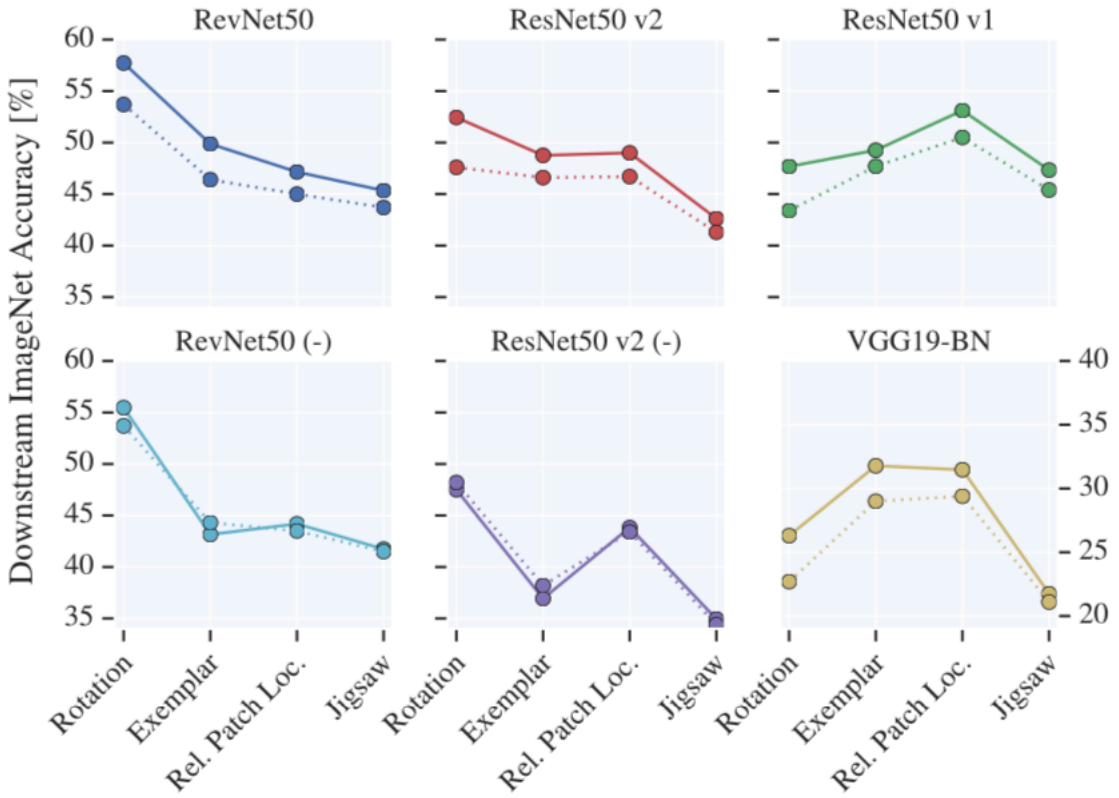


Figure 3. Comparing linear evaluation (·····) of the representations to non-linear (——) evaluation, *i.e.* training a multi-layer perceptron instead of a linear model. Linear evaluation is not limiting: conclusions drawn from it carry over to the non-linear evaluation.

- A linear model is adequate for evaluation.

Experiments and Results

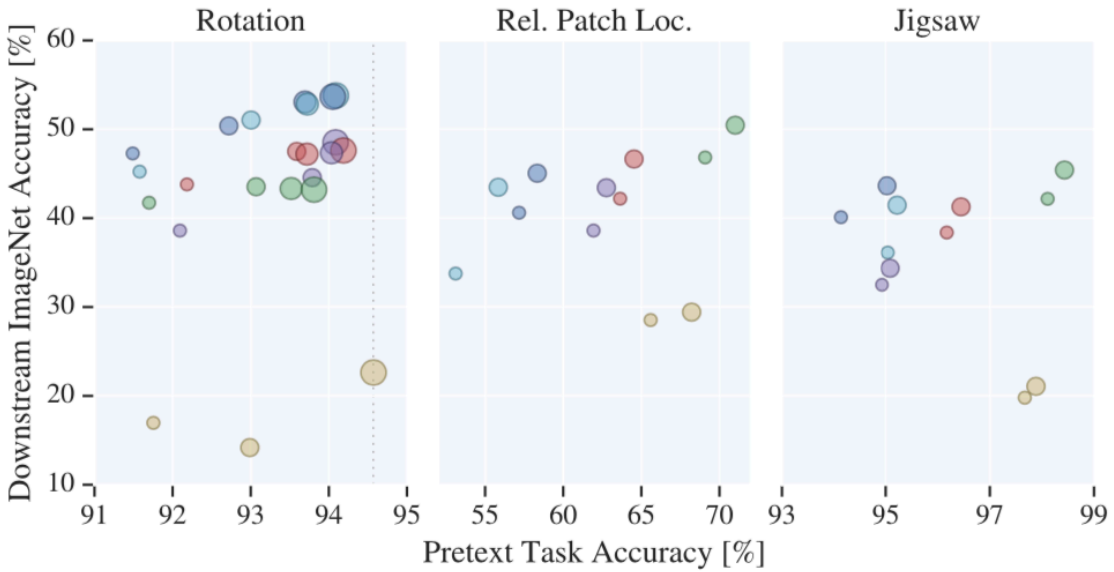


Figure 4. A look at how predictive pretext performance is to eventual downstream performance. Colors correspond to the architectures in Figure 3 and circle size to the widening factor k. Within an architecture, pretext performance is somewhat predictive, but it is not so across architectures. For instance, according to pretext accuracy, the widest *VGG* model is the best one for *Rotation*, but it performs poorly on the downstream task.

- Better performance on the pretext task does not always translate to better representations.

Experiments and Results

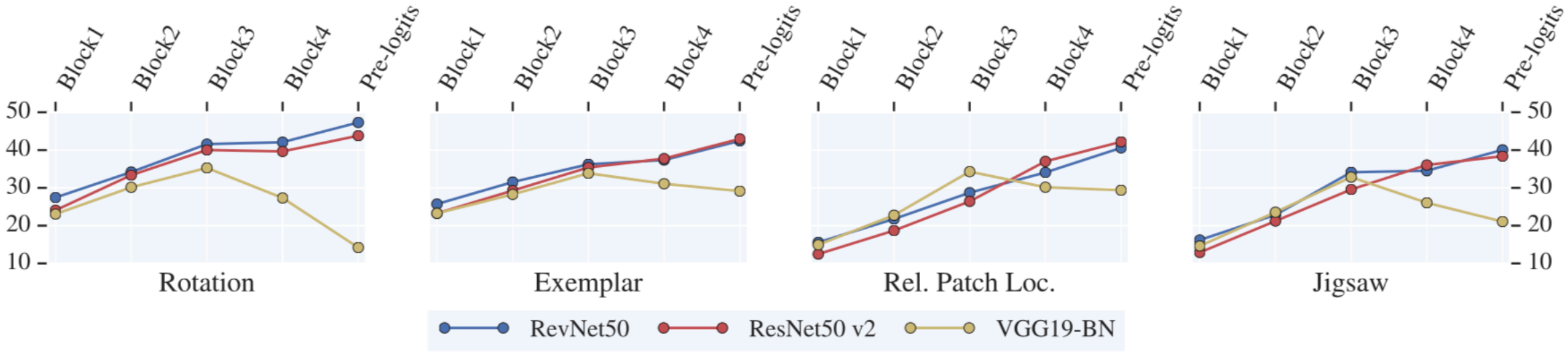


Figure 5. Evaluating the representation from various depths within the network. The vertical axis corresponds to downstream ImageNet performance in percent. For residual architectures, the *pre-logits* are always best.

- Skip-connections prevent degradation of representation quality towards the end of CNNs.

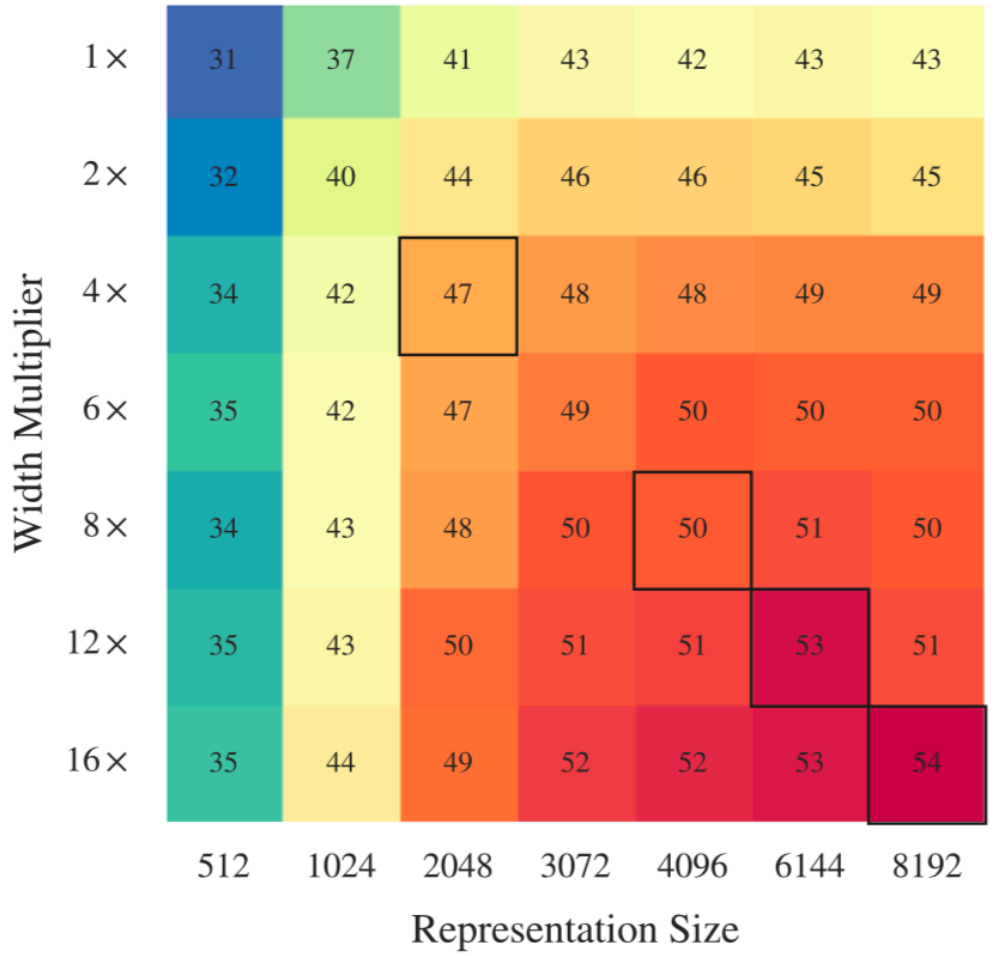


Figure 6. Disentangling the performance contribution of network widening factor versus representation size. Both matter independently, and larger is always better. Scores are accuracies of logistic regression on *ImageNet*. Black squares mark models which are also present in Table 1.

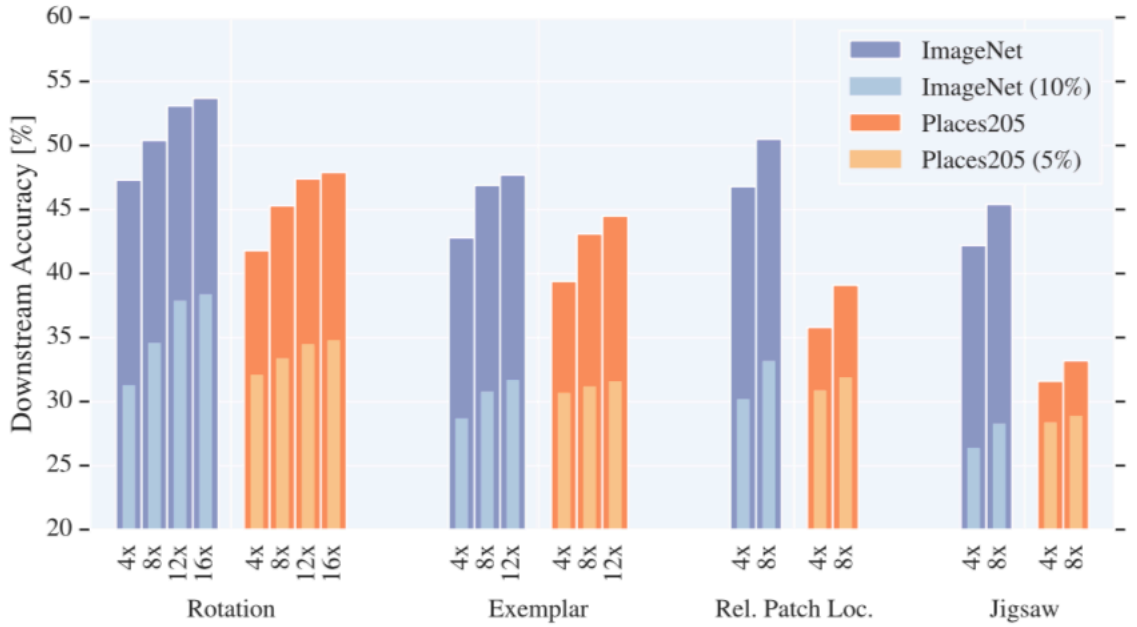


Figure 7. Performance of the best models evaluated using all data as well as a subset of the data. The trend is clear: increased widening factor increases performance across the board.

- Model width and representation size strongly influence the representation quality.

Experiments and Results

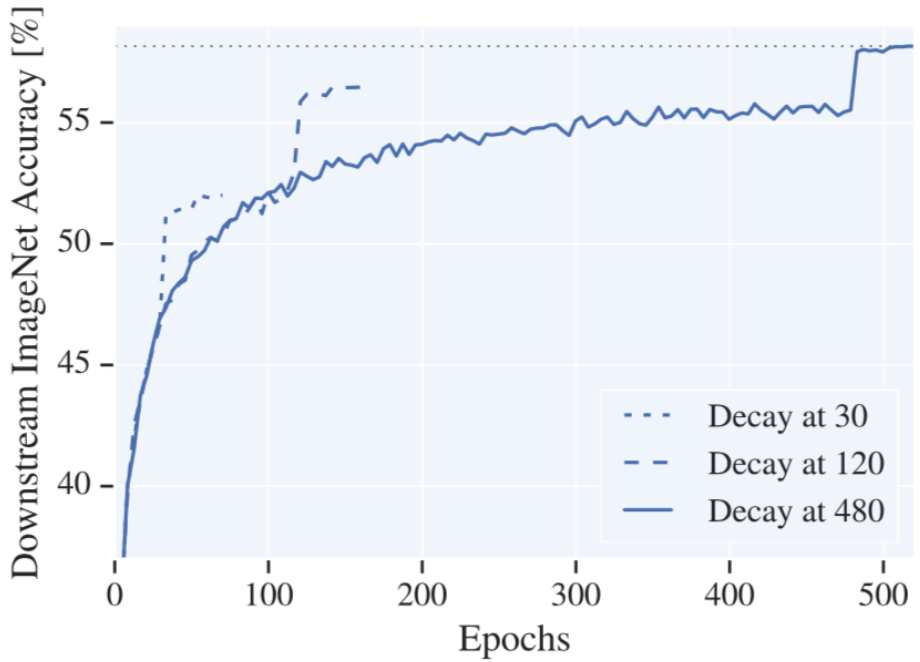


Figure 8. Downstream task accuracy curve of the linear evaluation model trained with SGD on representations from the *Rotation* task. The first learning rate decay starts after 30, 120 and 480 epochs. We observe that accuracy on the downstream task improves even after very large number of epochs.

- SGD for training linear model takes long time to converge

Conclusion

1. Lessons from architecture design in the fully supervised setting do not necessarily translate to the self- supervised setting
2. Contrary to previously popular architectures like AlexNet, in residual architectures, the final *pre- logits* layer consistently results in the best performance
3. The widening factor of CNNs has a drastic effect on performance of self-supervised techniques
4. SGD training of linear logistic regression may require very long time to converge
5. Pretext tasks for self-supervised learning should not be considered in isolation