We thank the reviewers for their detailed and thoughtful comments. Before addressing their remarks, we highlight some of the key results from the submission. These are not new and have been presented thoroughly in the submitted paper.

	Technique	**Linear Classification** (Tab. 1, 2)	**Downstream** ***vs.*****ImageNet supervised** (Tab. 2)	**Semi-supervised** (Tab. 1)
Previous Best	Contrastive	MoCo-v2 or SimCLR	Better on detection. Worse on linear classification.	SimCLR
Ours	Clustering	Δ top-1 $+\mathbf{4.2}\%$ ImageNet, $+\mathbf{4.6}\%$ Places, $+\mathbf{9.7}\%$ iNat18	Better on both detection and linear classification	$+\mathbf{5.6}\%$ top-1 ImageNet

High-level remarks

R1Q1: *"focuses too much on not using momentum and memory bank"* We thank **R1** for this feedback and will rewrite to de-emphasize the fact that we only use a single network. Our intention was not to challenge the momentum mechanism. Combining SwAV with a momentum encoder and/or a large memory bank are indeed interesting follow-ups.

R2Q1 + R4: *"fair comparison"* Comparisons to prior work are complicated as each work uses a different bag of tricks. In Tab.5, we make a best effort fair comparison (same data augmentation, num. epochs, batchsizes, etc). We observe in Tab.5 that clustering brings $+2\%$ compared to SimCLR and that multi-crop particularly improves clustering approaches.

R1Q1: *"with small batch, [MoCo is] go-to solution"* With batches of 256, SwAV reaches higher performance in half the time needed by MoCo: 72.0% after 102h (200ep) while MoCov2 reaches 71.1% after 212h (800ep). One epoch of MoCo is faster in wall clock time than one of SwAV, but MoCo needs more epochs for good downstream performance.

R1Q7: *"larger models"* Our paper shows large improvement over previous state of the art for all considered architectures, which suggests that SwAV does not overfit to R50 and can readily be applied to different models. The fact that BYOL (parallel work not available at submission time) has slightly better performance on large models does not imply that we suffer from poor scalability. Interestingly, our results follow a similar trend as supervised pretraining (Fig.2).

R1Q7: *"wall clock time for larger models"* We will add a "performance versus time" plot, similar to Fig.2 of the supplementary material, for large models. With R50w4, based on our implementation with 64 GPUs, SwAV gives 77.9% after 74.3h and 400ep while SimCLR reaches 76.8% after 130h and 1000ep (see Fig.7 of SimCLR paper).

R1Q7: *"lack of generality"* In Fig.4 we evaluate SwAV on random, uncurated images that have different properties from ImageNet and show that both our online clustering scheme and multi-crop augmentation work out of the box.

R1Q1: *"create artificial problems"* Although curation is a solution, training directly on uncurated data is an important research question that Fig.4 tries to address. Our intent was not to compare with MoCo so we will remove it from Fig.4.

DeepCluster-v2 (DCv2)

R2Q2: *"how it is improved over initial version?"* We introduce explicit comparisons to k-means centroids, which increased stability, and leverage the training improvements from SimCLR. Full details are in supp. D. One goal of improving and using DCv2 as a baseline was to show the strong performance of clustering-based techniques. We train DCv2 in SwAV best setting (800 epochs - 8 crops) and obtain 75.2% top-1 accuracy on ImageNet.

R1Q3 + R2Q2: *"advantage of SwAV given DC-v2 is stronger?"* DCv2 performs comparably to SwAV. However, unlike SwAV, DCv2 is not online which makes it impractical for extremely large datasets. For billion scale trainings, as in MoCo, a single pass on the dataset is usually performed. DCv2 cannot be trained for only one epoch since it works by performing several passes on the dataset to regularly update centroids and cluster assignments for each image.

R1Q3: *"swapping mechanism does not seem important"* We respectfully disagree with this conclusion. DCv2 can be interpreted as a special case of our proposed swapping mechanism: swapping is done across epochs rather than within a batch. Given a crop of an image DCv2 predicts the assignment of another crop, which was obtained at the previous epoch. SwAV swaps assignments directly at the batch level and can thus work online.

R1Q4: *"efficiency comparison to DC-v2"* As discussed above, we will clarify the fact that k-means cost (12% of epoch time on ImageNet) is not the reason why DCv2 does not scale well.

More evaluation experiments

R3: *"finetune results"* When finetuning R50 on Places and iNat18 we get 63.5% and 66.8% respectively, which is higher than training both from scratch and from ImageNet supervised model. We thank **R3** for the missing reference.

R3: *"semi-supervised learning with wider architectures"* Top-1 acc. on ImageNet – 1% labels: 56.5% (R50w2) / 58.7% (R50w4) - 10% labels: 72.6% (R50w2) / 74.5% (R50w4). We thank **R3** and will add the results in the paper.

R1Q6 + R2Q3: *"object detection is quite limited"* We agree that our gains on detection are limited and in the same ballpark as prior work. Yet, unlike prior work, our model outperforms supervised pre-training on both classification and detection tasks. As mentioned in supp. A.5, our VOC07 numbers are averaged over 5 runs.

Implementation details and miscellaneous

R1Q2: *"area_range for the random_crop augmentation"* All details for reproducing SwAV trainings, including random_resized_crop parameters, are in supp. A.1 and A.2. Prior works have also tuned the random_crop: for example MoCo uses a scaling range of $(0.2, 1)$. For SwAV, using $(0.14, 1)$ gives $+0.2\%$ compared to $(0.08, 1)$ after 400 epochs.

R1Q5: *"data augmentation for linear probe"* SimCLR, MoCo and other prior works (all methods in Fig.2, Tab.2) use random crop augmentation when training linear probes on ImageNet, Places, iNat18. We follow their linear probe pipeline exactly to ensure our comparisons are fair. We do not use multi-crop for any of our evaluation.

Misc. We appreciate **R4**'s table re-organization suggestion. We agree with **R1** that mentioning RandAugment is not very informative and will remove it. We thank **R2** for their feedback and will clarify the use of the term "clustering": intuitively as the prototypes are used across different batches, SwAV "clusters" multiple instances to prototypes.