Scaling Object Detection up to More Categories

Y. Lu, B. Xiao, Z. Gong, X. Zhang, H. Guo, L. Wen
Bytedance AI Lab

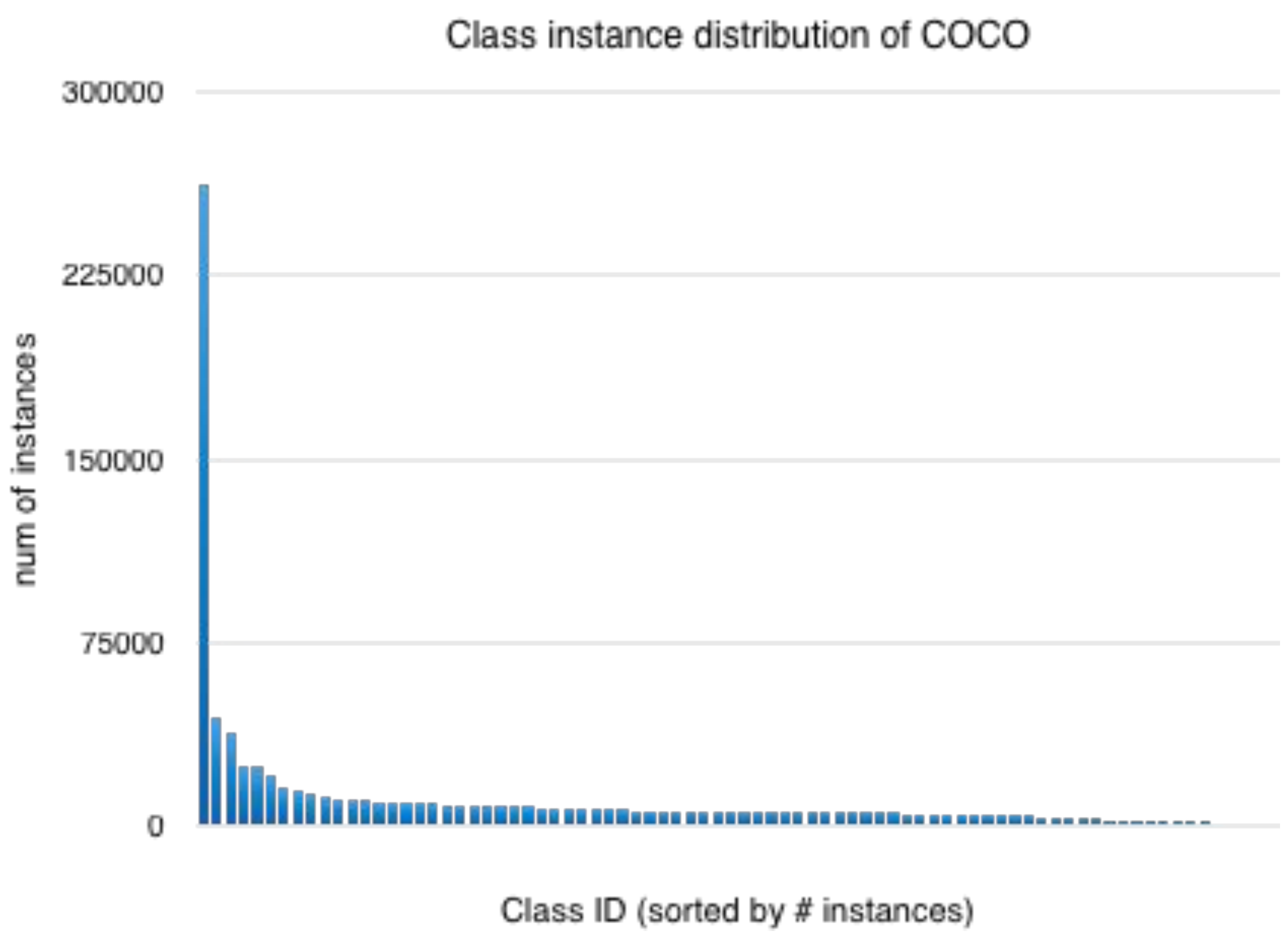
ByteDance

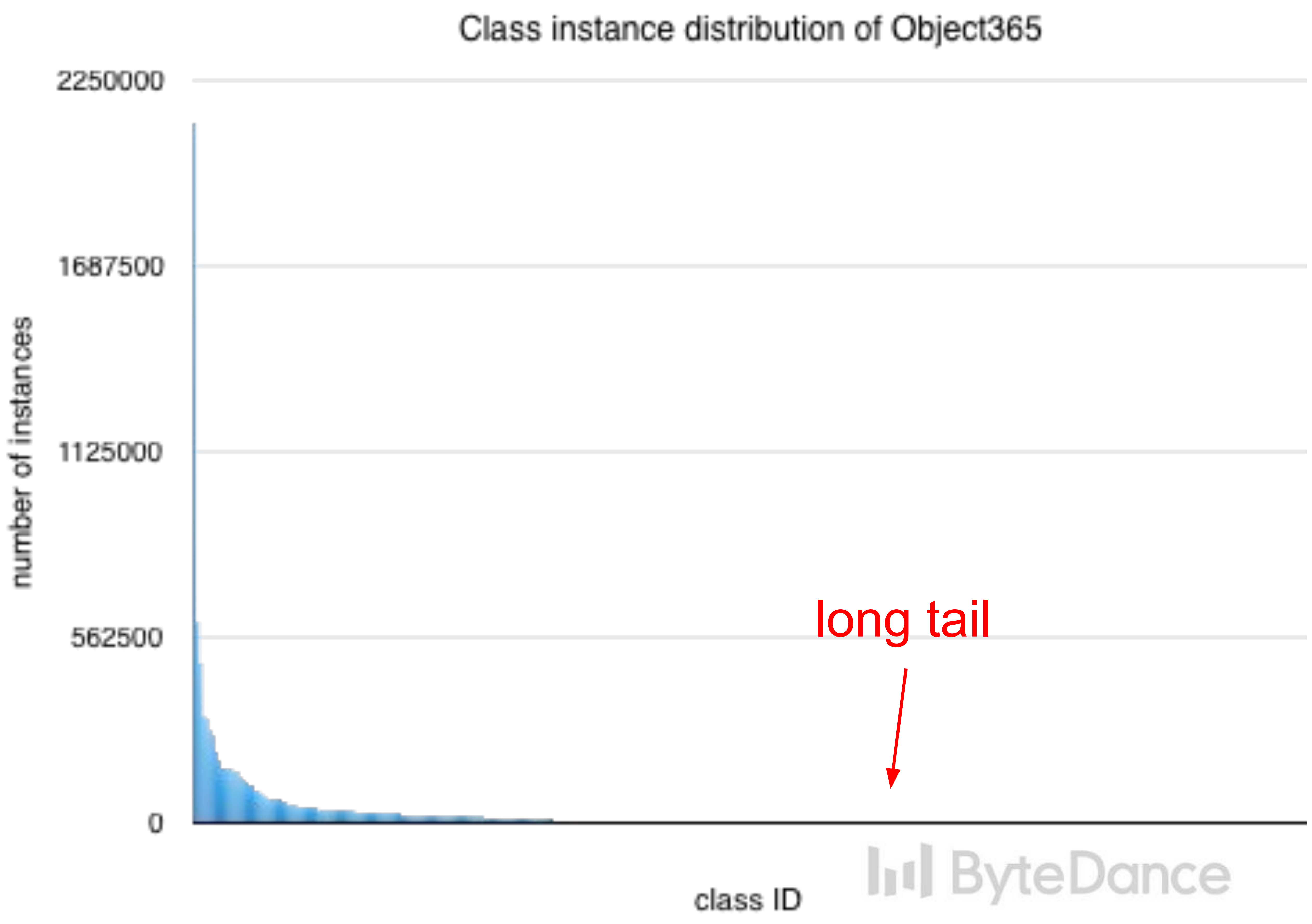
From COCO to Object365

- More object categories: 80 -> 365
- More training images: 11W -> 60W
- More data → more gains
- But...

From COCO to Object365

- Object365 dataset has a **longer tail**





From COCO to Object365

- Class imbalance problem is more severe on Object365

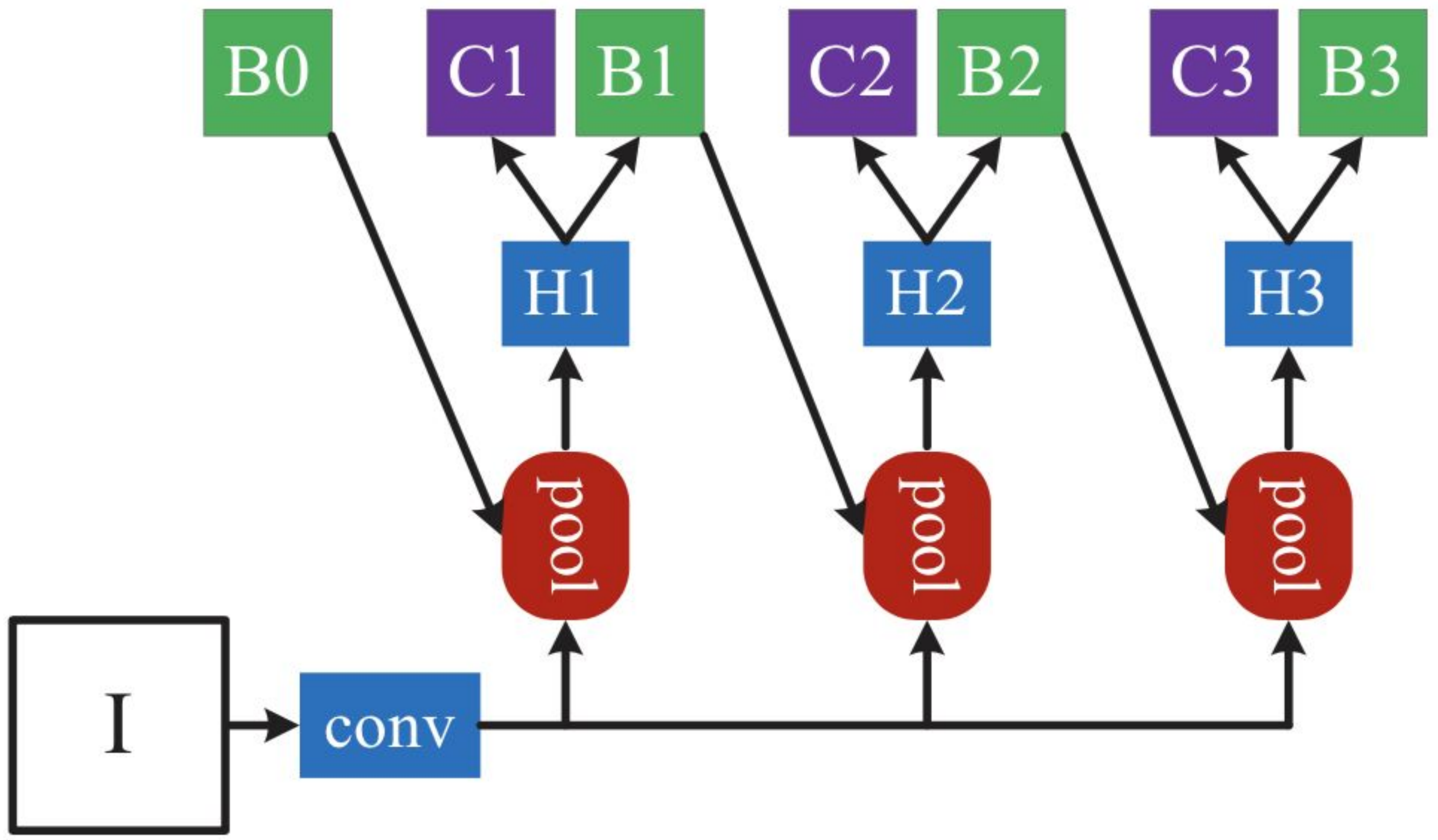
	COCO	Object365
Max #Instance	262465	2120895
Min #Instance	198	28
Max / Min	1326	75746

From COCO to Object365

- More object classes: 80 -> 365
- More training images: 11W -> 60W
- But longer tail and more imbalance data
- What if we simply apply COCO models onto 365 classes?

From COCO to Object365

- Start from Cascade R-CNN [1] with ResNext101 64x4d [2] backbone
  - mAP of **44.7** on COCO
- Achieve only mAP of **29.5** on the validation set of Object365



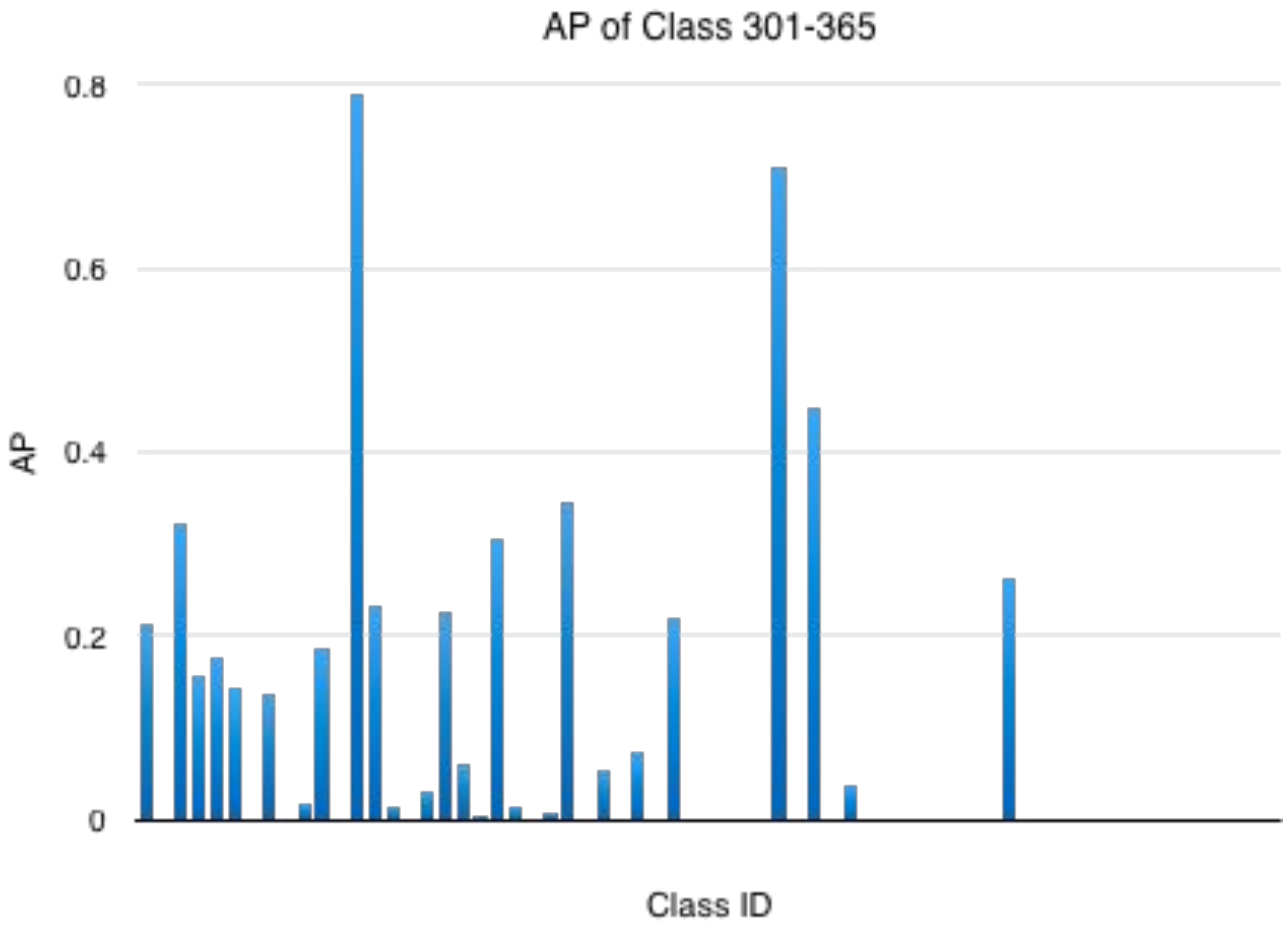
[1] Cai Z, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection. CVPR 2018.
[2] Xie S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks. CVPR 2017.

Class AP distribution on Object365

- The AP is worse for the classes with less instances

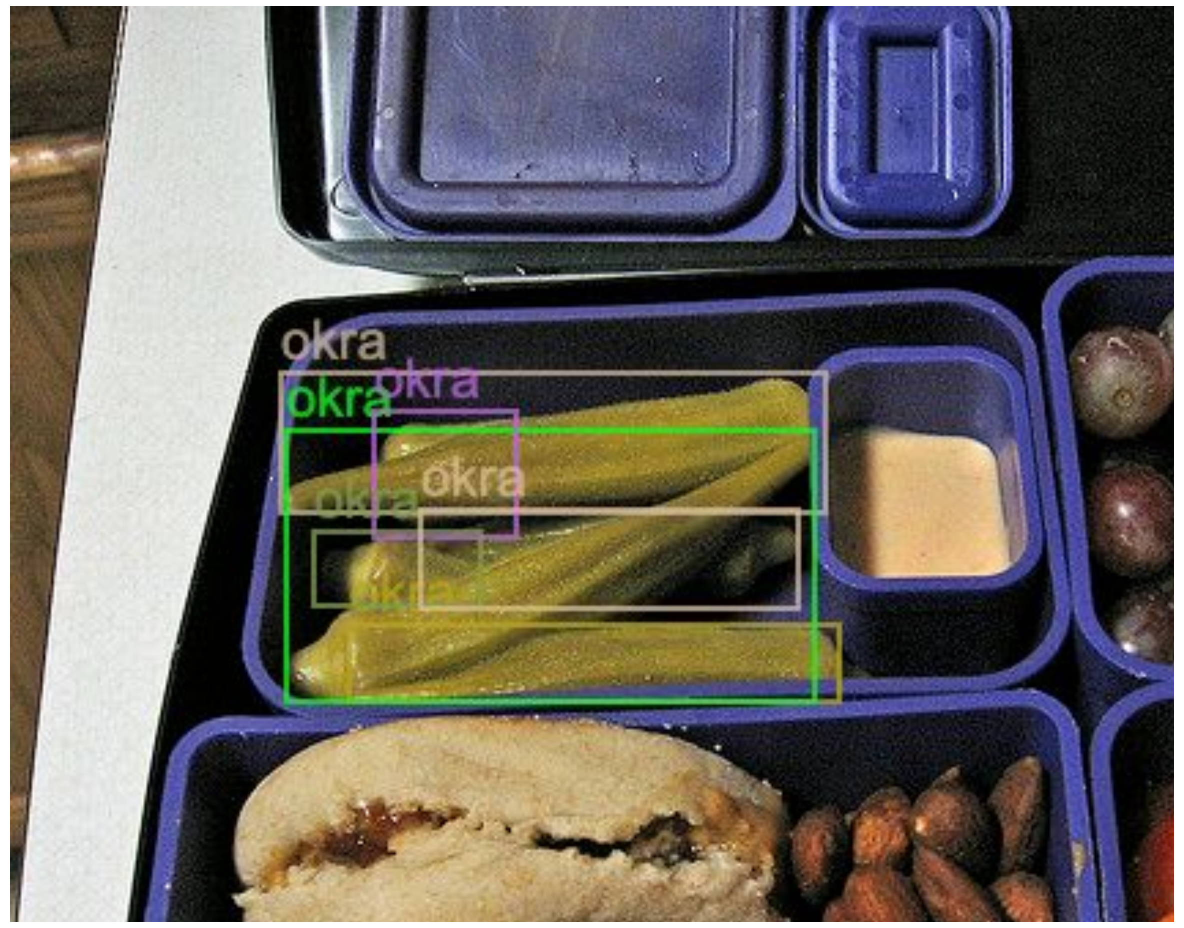
A detailed look on class 301-365

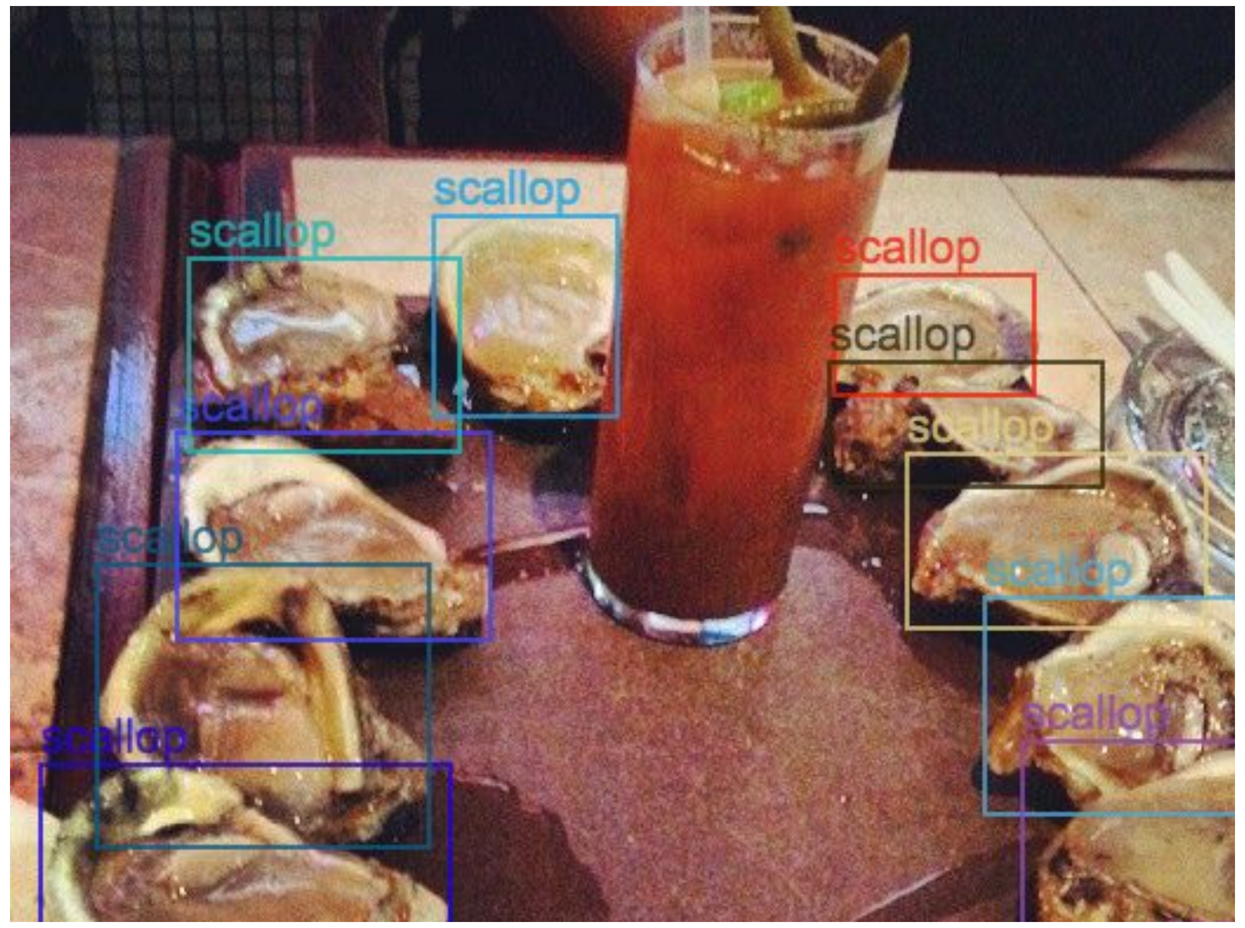
- **39** out of 65 classes has 0 AP！



A detailed look on class 301-365

- **Zero AP classes:** okra, scallop, pitaya

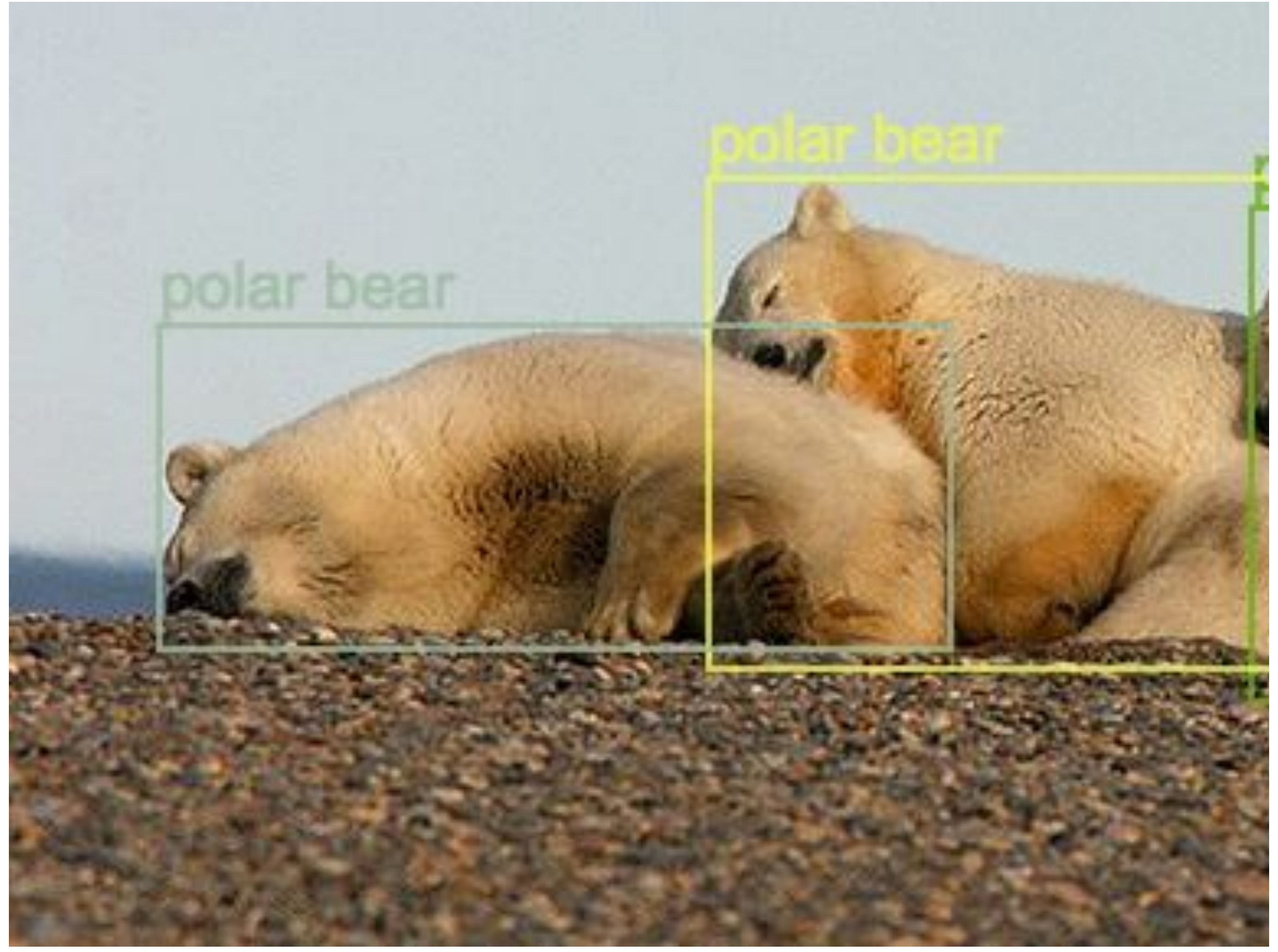




Most small things with heavy clustering

A detailed look on class 301-365

- **High AP classes:** donkey, polar bear, seal



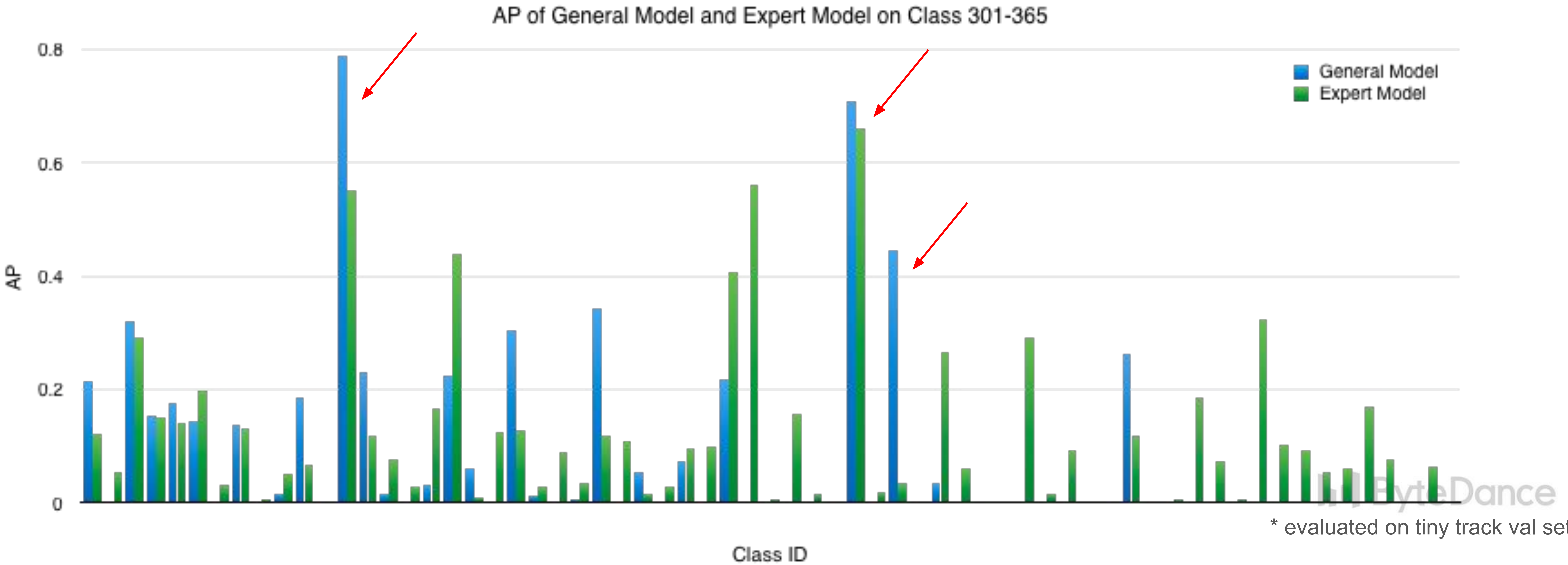
Most animals, with large scales and simple appearance

Possible solutions

- Expert models
- Data distribution resampling

Expert models

- Fine-tuning the full classes model on class 301-365
- mAP on Class 301-365: 18.4 → **29.5***
  - APs of **46** classes increase



* evaluated on tiny track val set

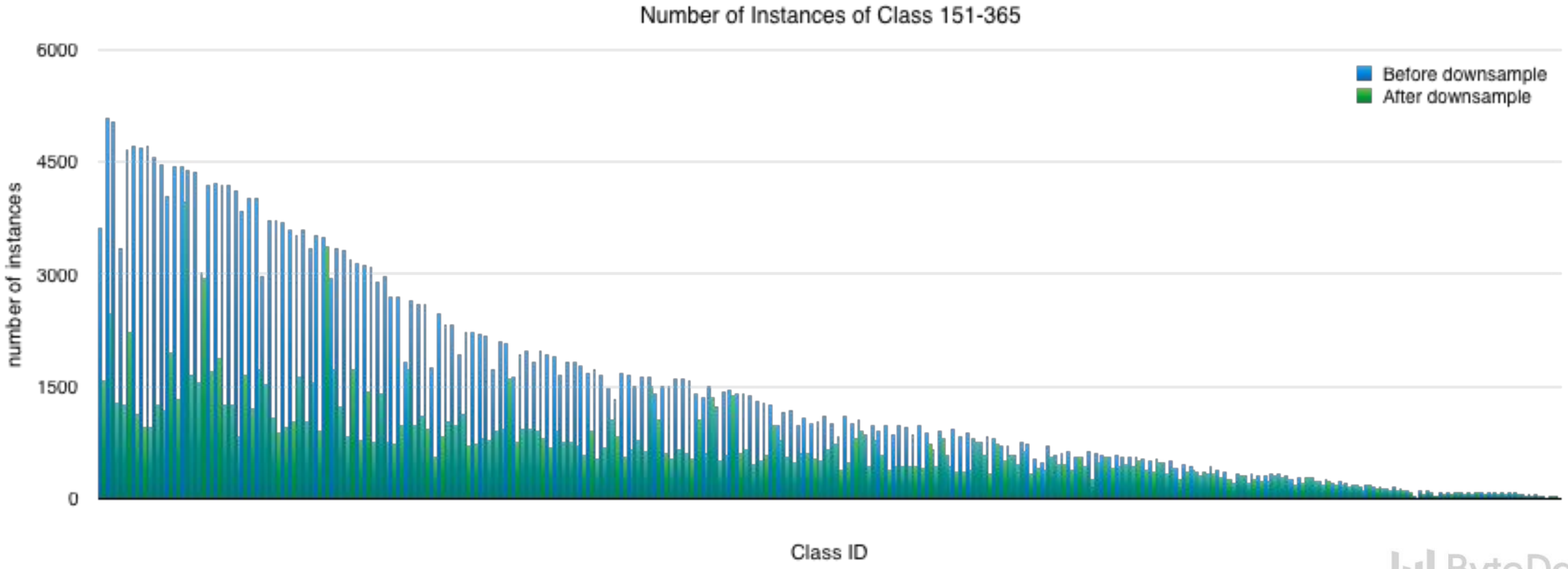
Expert models

- Introducing expert models improves overall mAP by 1.1
  - Expert 1: 301-365 classes
  - Expert 2: 151-300 classes

Model	mAP
General model	29.6
General + Expert 1	29.9
General + Expert 1 + Expert 2	30.7

Data distribution resampling

- Down-sample classes with huge number of instances



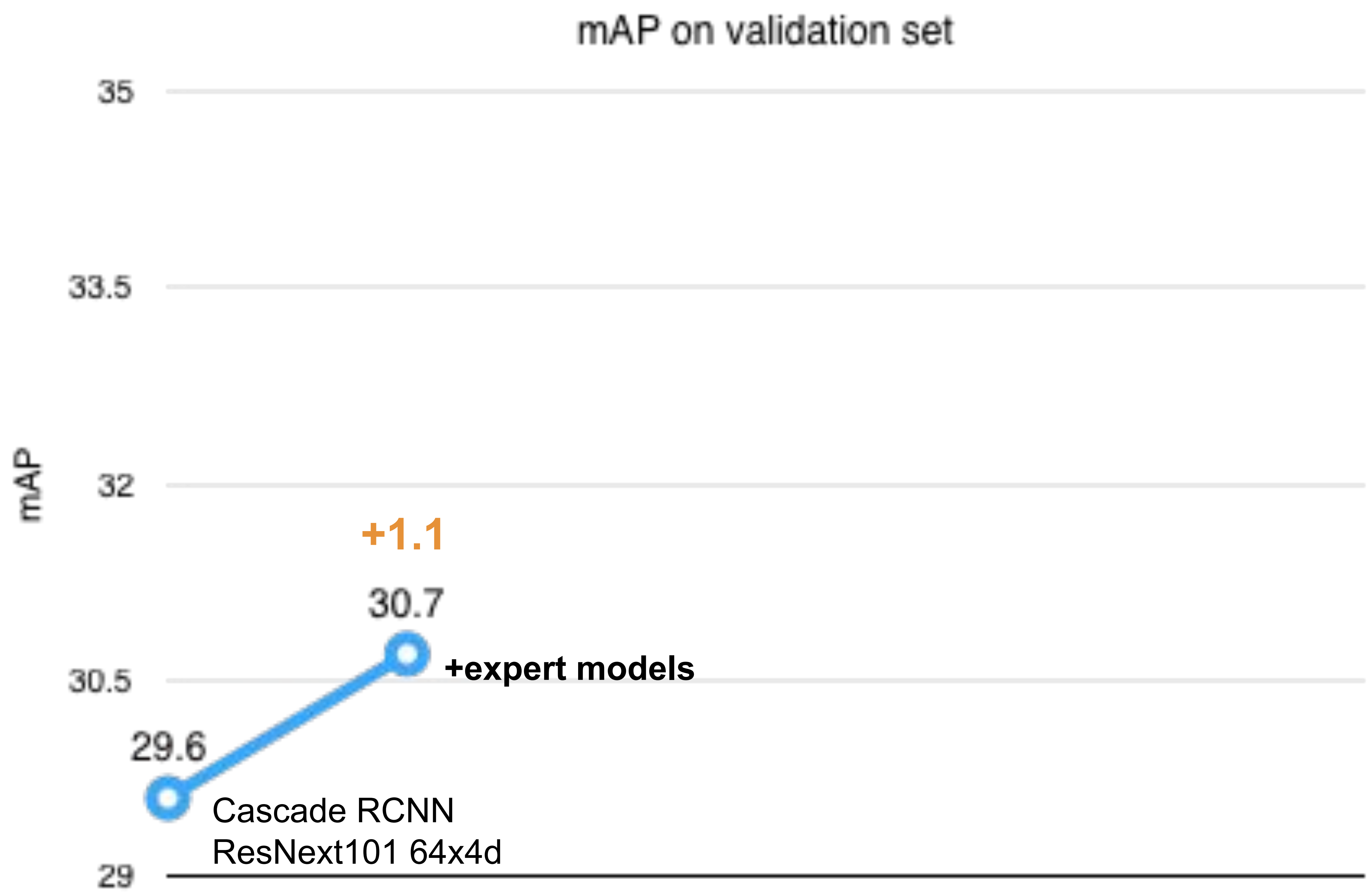
Data distribution resampling

- Down-sample classes with huge number of instances
  - mAP of Class 301-365: 18.4 -> **23.3***
  - overall mAP: 31.3 -> 31.0
- No gain on overall mAP

ByteDance

* evaluated on tiny track val set

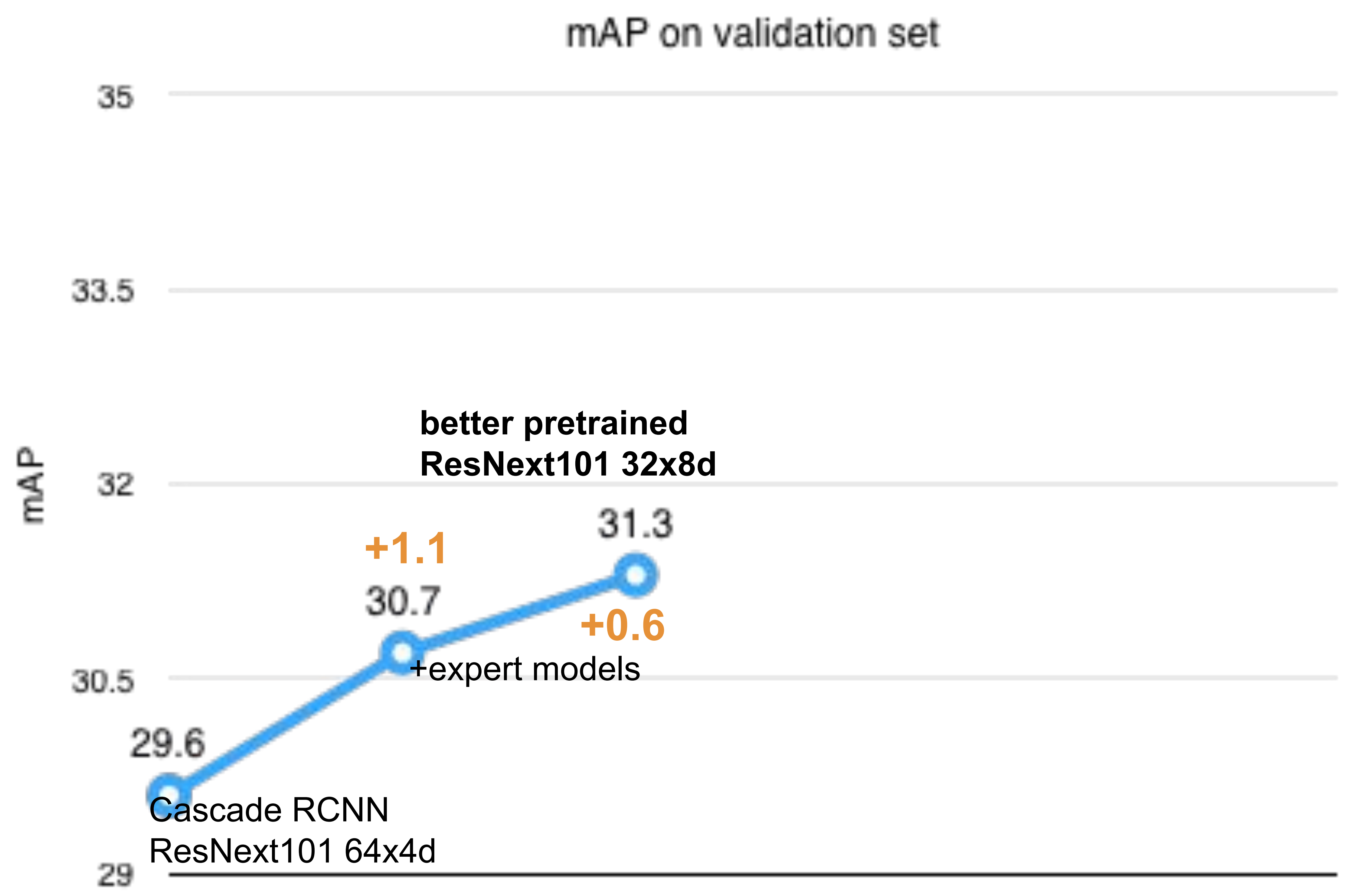
Further improvement



ByteDance

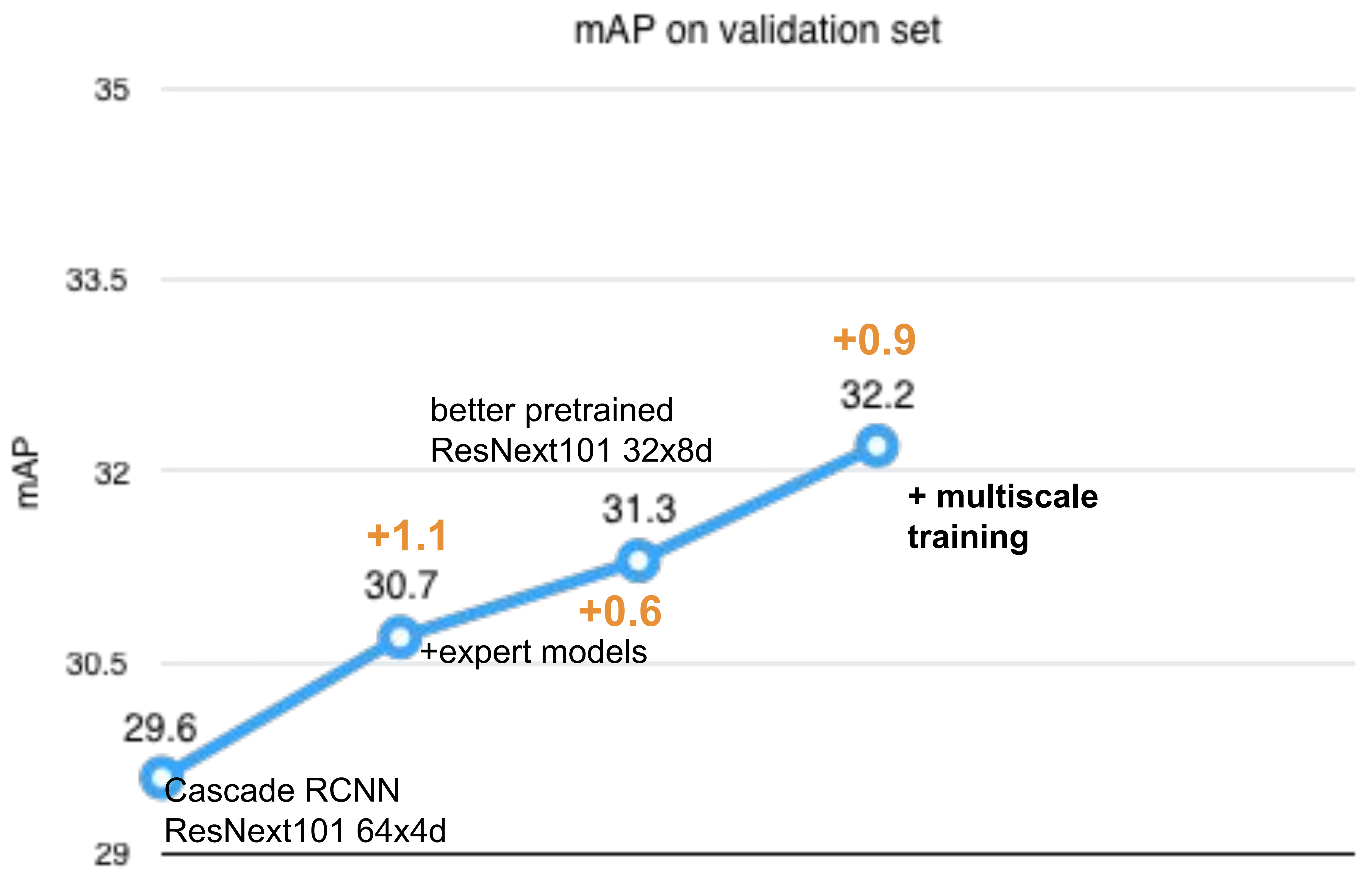
Further improvement

- A better pretrained backbone improves mAP by 0.6



Further improvement

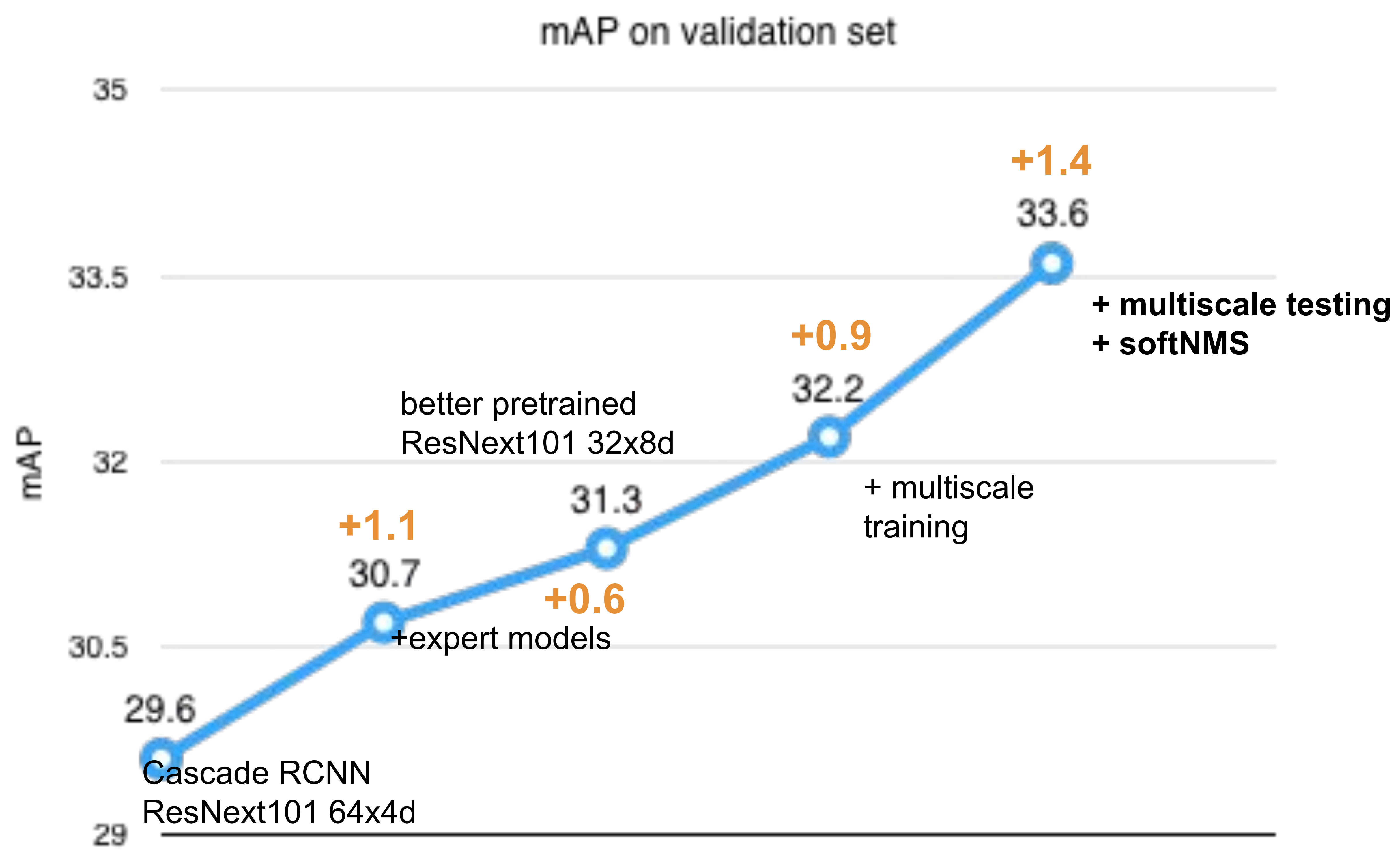
- Multi-scale training improves mAP by 0.9



ByteDance

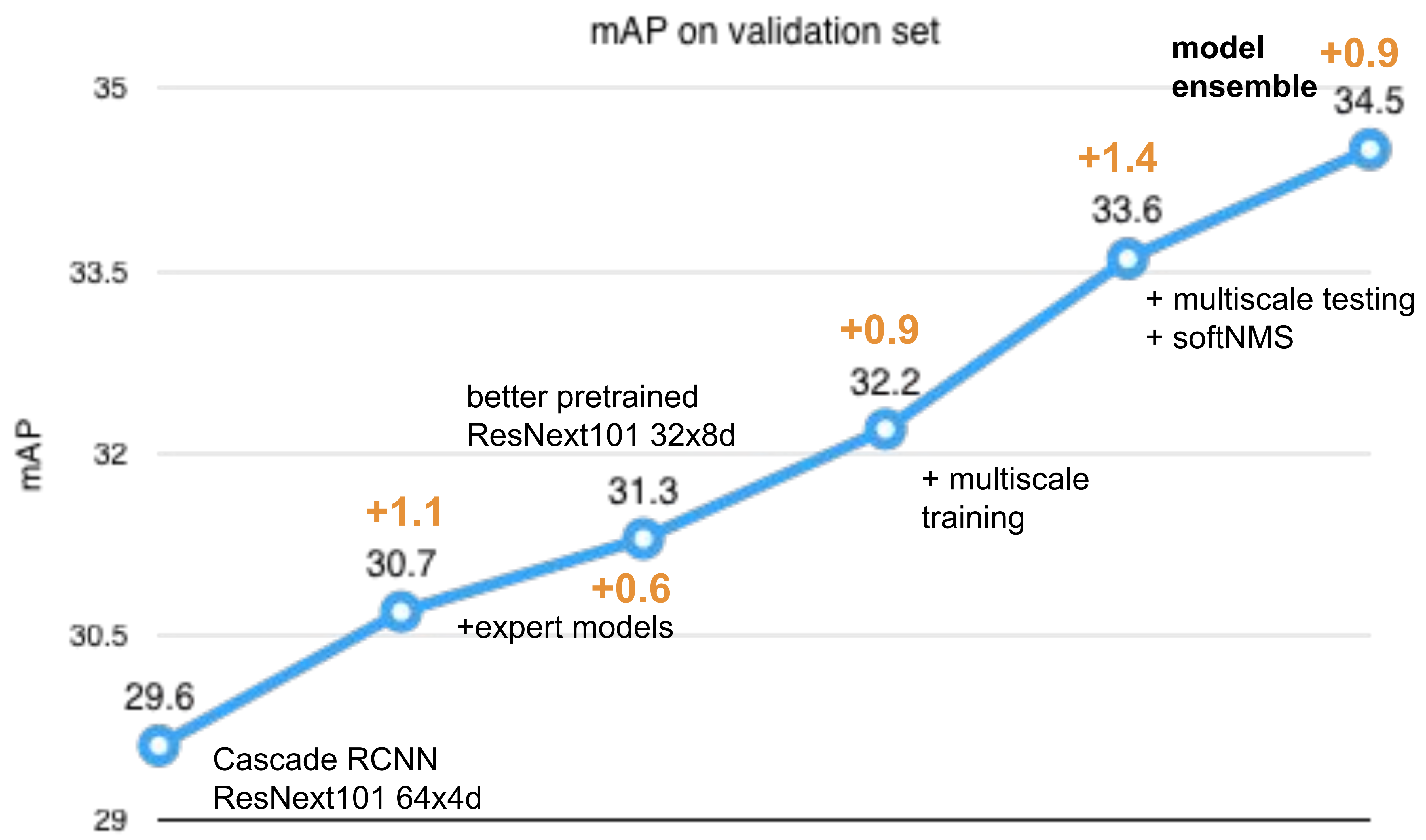
Further improvement

- Multi-scale testing and soft NMS improve mAP by 1.4



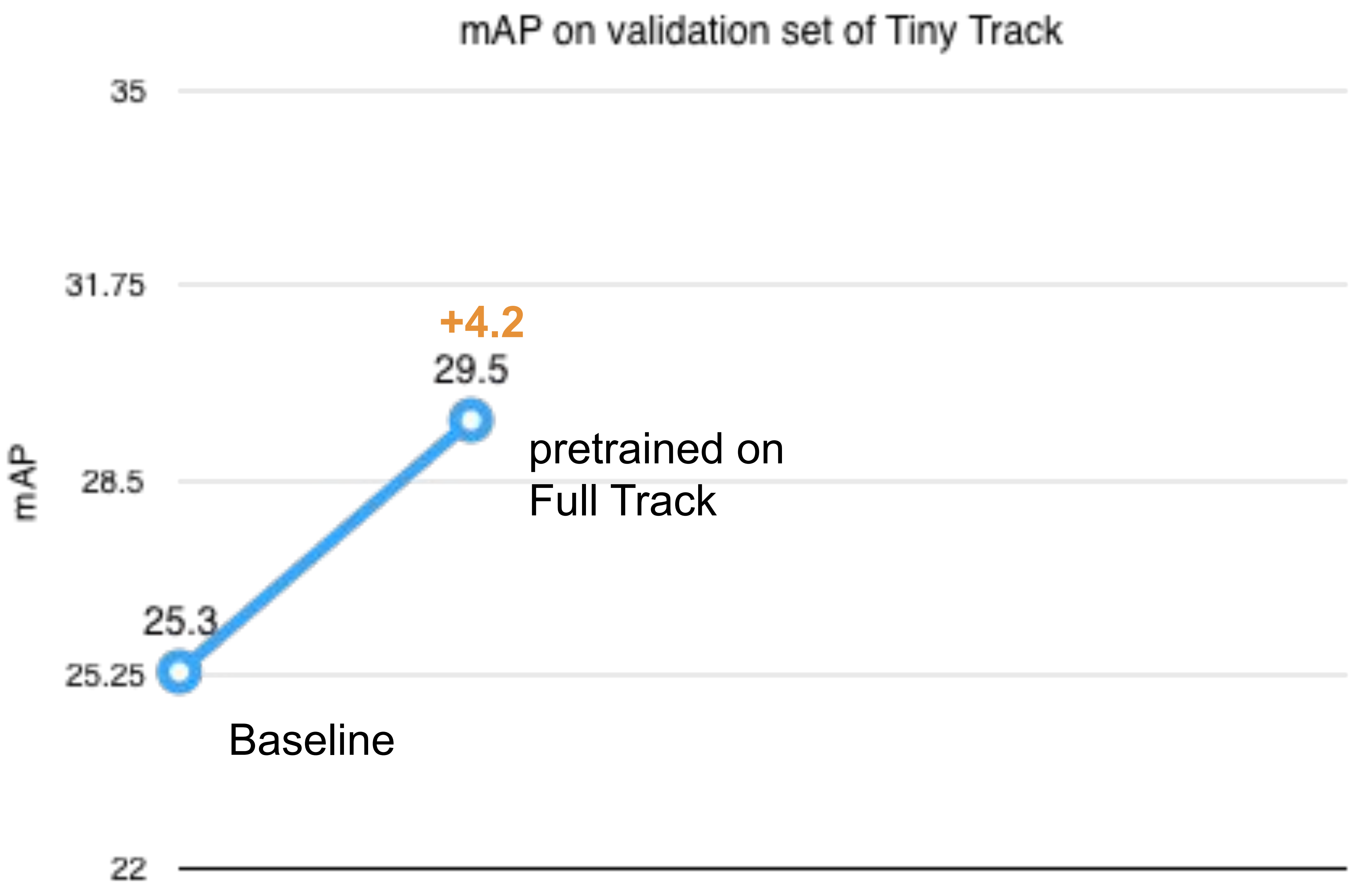
Further improvement

- Model ensemble improves mAP by 0.9



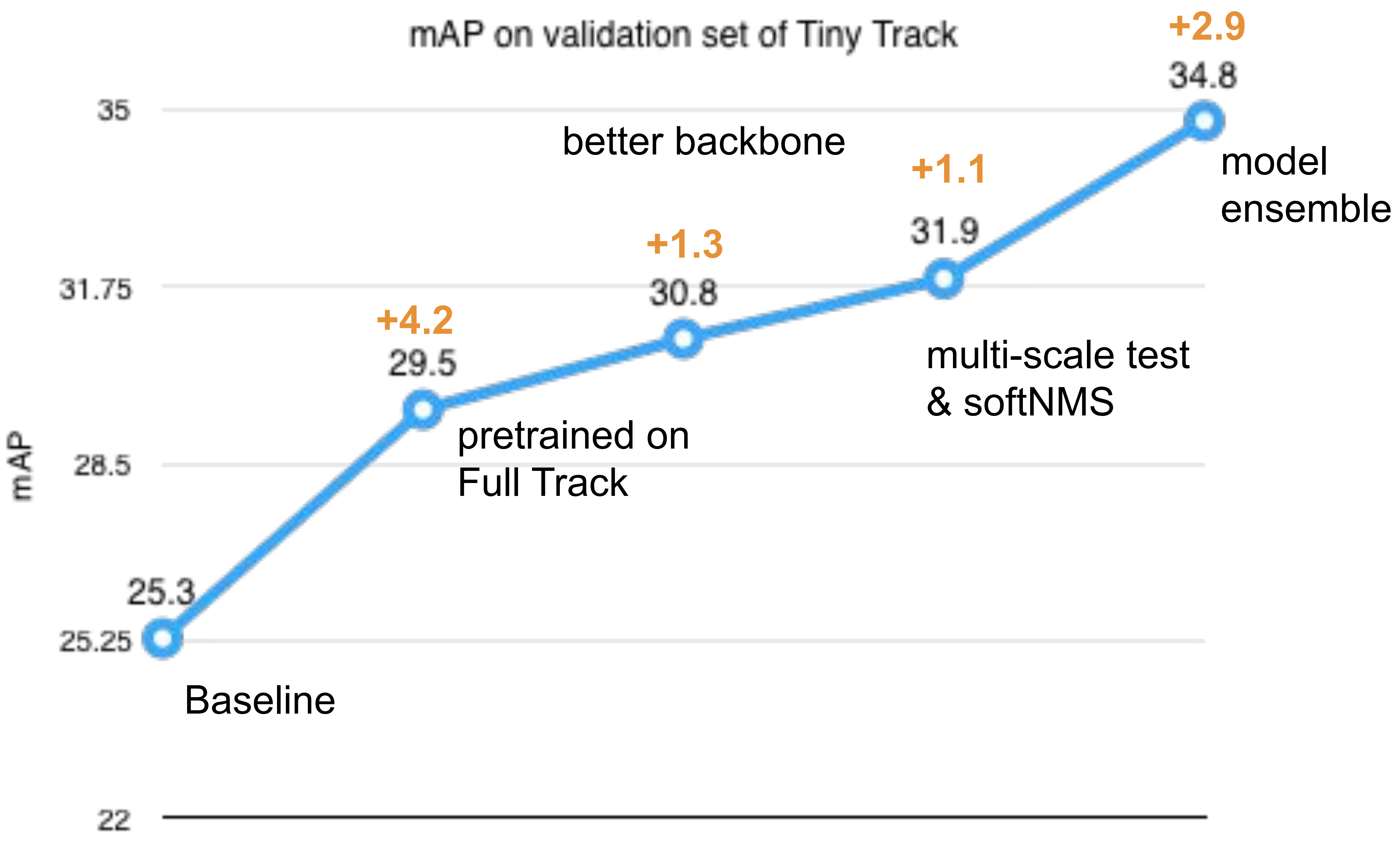
Tiny track experiments

- Baseline: Cascade R-CNN with ResNext101 64x4d pretrained on COCO
- Pretraining on Full Track dataset improves mAP by 4.2



Tiny track experiments

- Other tricks improve mAP by 5.3



ByteDance

Our final results

	mAP
Validation set (Full track)	34.5
Test set (Full track)	31.1
Validation set (Tiny track)	34.8
Test set (Tiny track)	27.4

Experiment details

- **Basic setting**
  - Cascade R-CNN with 3 stages
  - FPN
  - Deformable convolution
- **Backbones**
  - ResNeXt 101 64x4d / 32x8d
  - SENet154
  - Resnet152
- **Training Pipeline and settings**
  - ImageNet pre-train → COCO pre-train for 12 epochs
  - Full Track: training for 20 epochs (lr 0.1 for 6 epochs, 0.01 for 10 epochs, 0.001 for 4 epochs)
  - Tiny Track: fine-tuning for 10 epochs (lr 0.1 for 4 epochs, 0.01 for 6 epochs)
  - Batch size: 80 (2 imgs/GPU * 40 GPUs)

Conclusion

- **Data distribution matters**
  - Long tail distribution greatly degrades the overall performance
- **Expert helps general model**
  - Expert model can improve APs for long tail classes
- **General model also helps expert**
  - Large data pre-training helps the learning of long tail classes
- **Long tail problem for object detection has not been solved**

We are hiring!

ByteDance AI Lab
字节跳动人工智能实验室

We are hiring research scientists, software engineers, and interns in following areas (@Beijing, Shanghai, Shenzhen):

Machine learning, natural language processing, computer vision, speech recognition and synthesis, and distributed systems.

Email：lab-hr@bytedance.com

THANKS.

ByteDance