keywords: self-supervised; BERT-style pretraining; computer vision; convolutional networks (CNN)

Designing BERT for Convolutional Networks: Sparse and Hierarchical Masked Modeling

Keyu Tian[1],
Chen Lin[3],
[1]Peking University

Yi Jiang[2]*,
Liwei Wang[1]*,
[2]Bytedance Inc.

Qishuai Diao[2],
Zehuan Yuan[2]
[3]University of Oxford

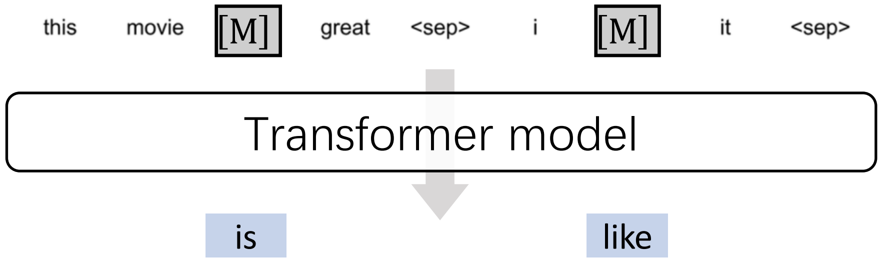
codes & models

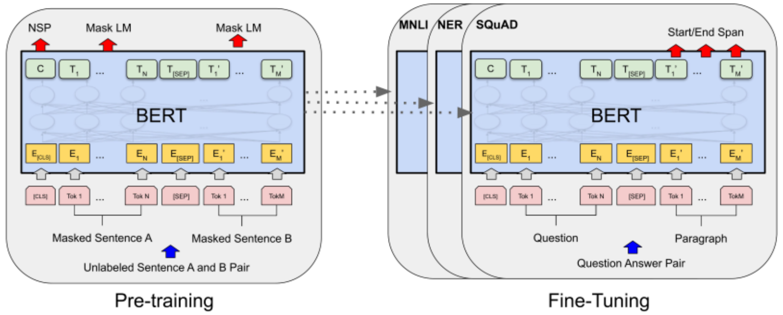
github.com/keyu-tian/SparK

✉ tiankeyu.00@gmail.com

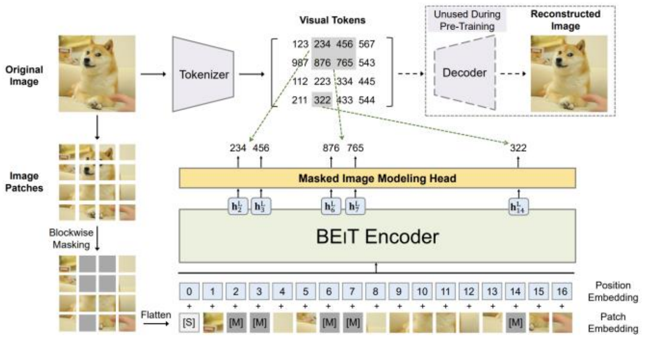
Background: BERT-style self-supervised pretraining

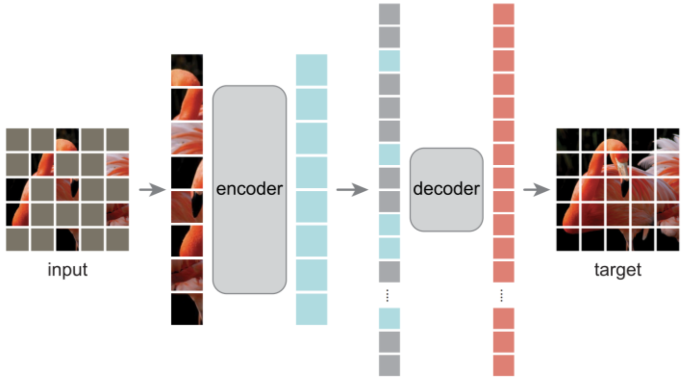
- ➢ BERT-style pretraining (aka. masked modeling/autoencoding) in NLP, mainly on Transformers





- ➢ BERT (MAE) Has been successfully applied on Vision Transformers (BEiT [2], MAE [3])





[1] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[2] Bao, Hangbo, et al. "Beit: Bert pre-training of image transformers." arXiv preprint arXiv:2106.08254 (2021).

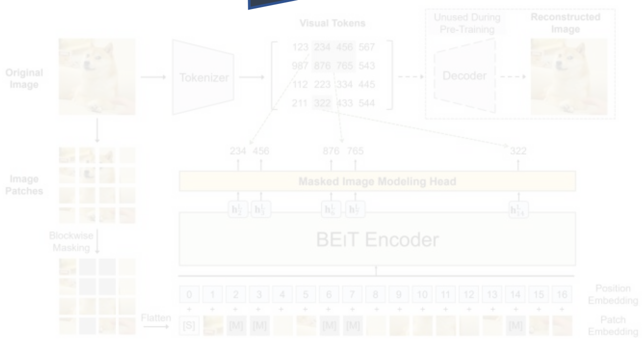
[3] He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

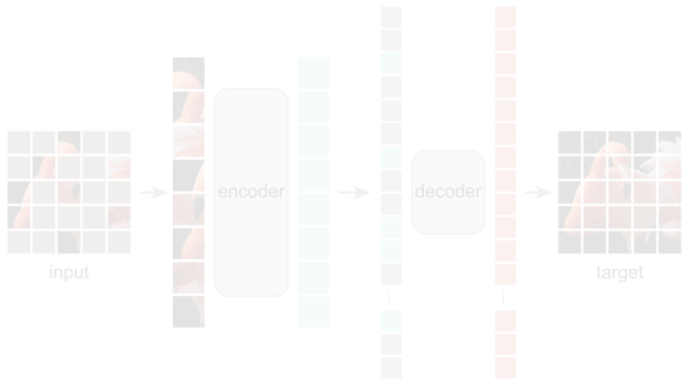
Background: BERT-style self-supervised pretraining

- BERT-style pretraining (aka. masked modeling/autoencoding) in NLP, mainly on Transformers

Can we apply BERT/MAE-style pretraining on convolutional networks (convnets) ?

- BERT (M AE) Has bee pplied on Vision Transformers (BEiT [2], MAE [3])





[1] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[2] Bao, Hangbo, et al. "Beit: Bert pre-training of image transformers." arXiv preprint arXiv:2106.08254 (2021).

[3] He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

Unfortunately, BERT for convnet is still an open problem

- Early works [4,5]: pioneering, but may not successful

pretraining	cls.	det.	seg.
supervised	79.9 (-0.0)	56.8 (-0.0)	48.0 (-0.0)
[1] CVPR’16	56.5 (-23.4)	44.5 (-12.3)	29.7 (-18.3)
[2] CVPR’17	67.1 (-12.8)	46.7 (-10.1)	36.0 (-12.0)

- Replacing the ViT in MAE to a convnet: still failed (useless pretraining)

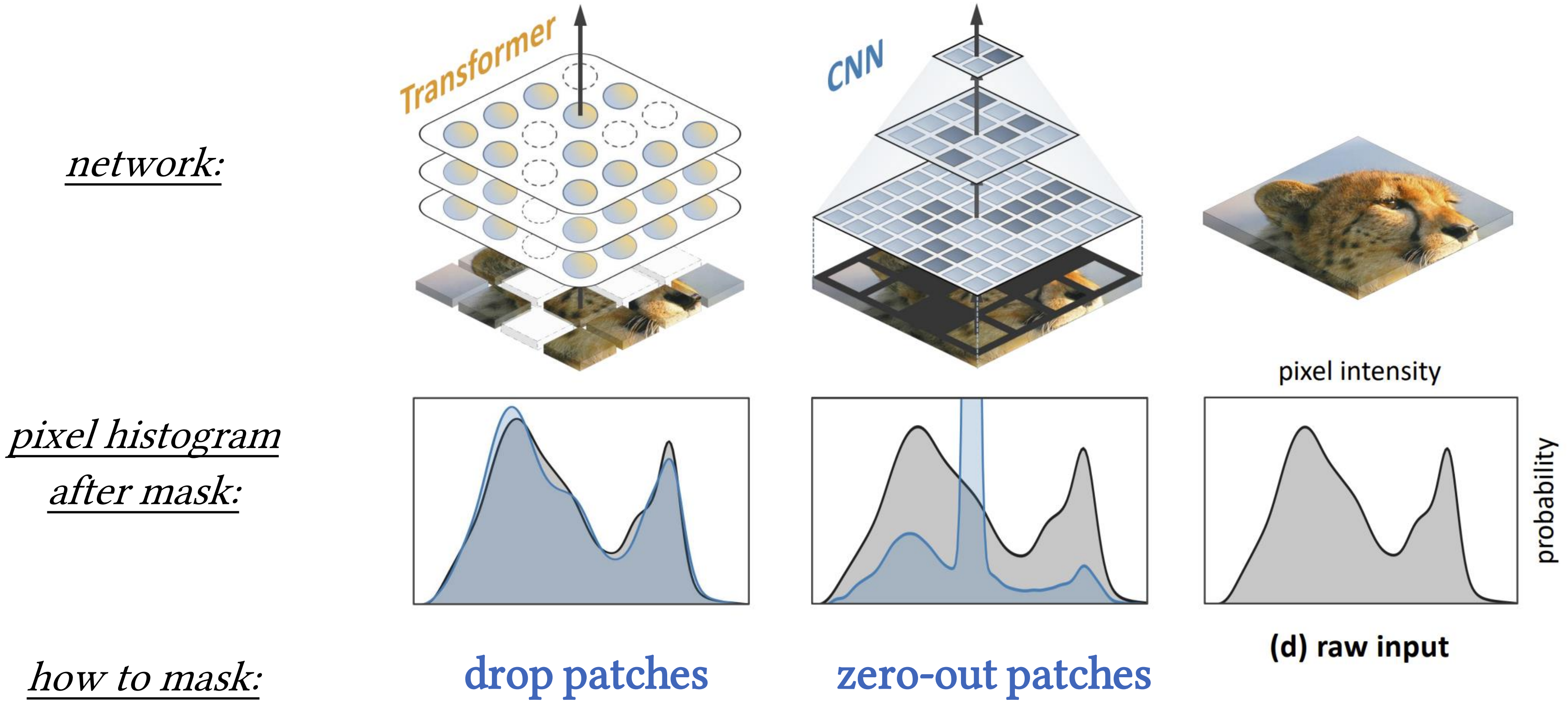
pretraining method	cls. acc.
not pretrained	83.1 (-0.0)
MAE (ViT → ConvNeXt)	83.2 (+0.1)
SparK (ours, introduced next)	84.1 (+1.0)

[4] Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[5] Zhang, Richard, et al. "Split-brain autoencoders: Unsupervised learning by cross-channel prediction." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

We found some issues when applying BERT on convnets

issue 1: pixel intensity distribution changed a lot after “masking” (zero outing)



network:

pixel histogram after mask:

how to mask:

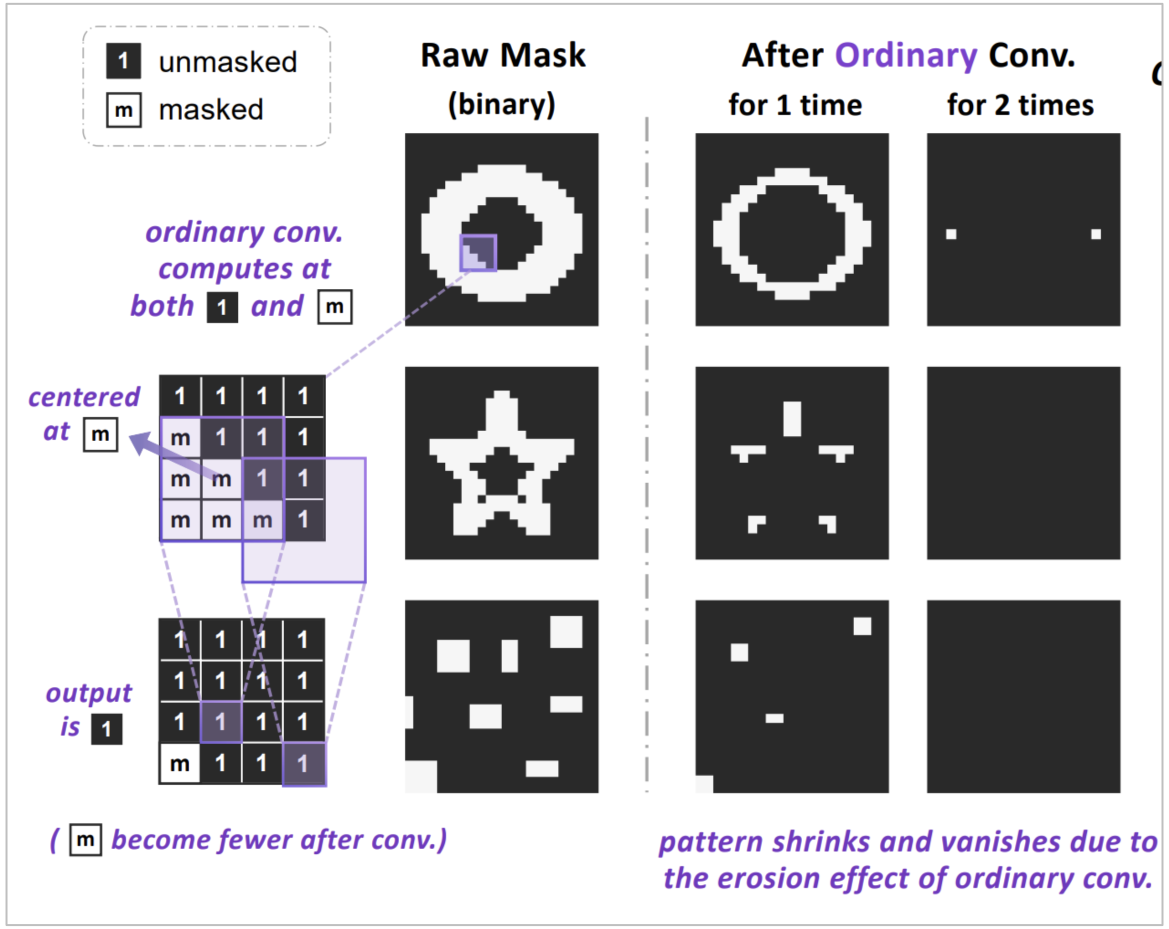
drop patches

zero-out patches

We found some issues when applying BERT on convnets

issue 1: pixel intensity distribution changed a lot after “masking” (zero outing)

issue 2: the pattern of binary masks will vanish after several convolutions



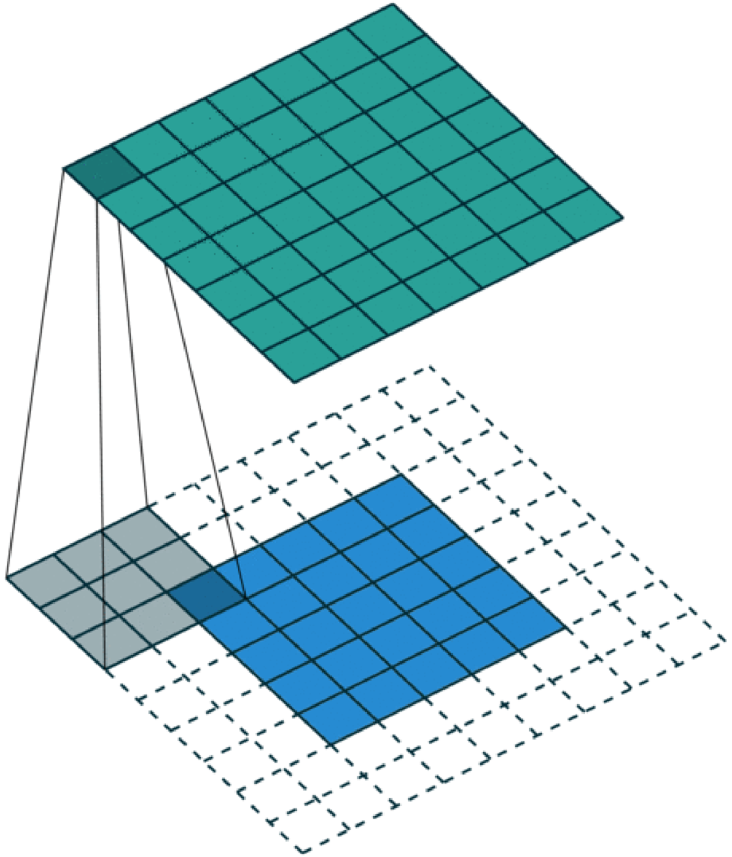
ordinary conv. computes at both 1 and m

1	1	1	1
m	1	1	1
m	m	1	1
m	m	m	1

1	1	1	1
1	1	1	1
1	1	1	1
m	1	1	1

(m become fewer after conv.)

pattern shrinks and vanishes due to the erosion effect of ordinary conv.

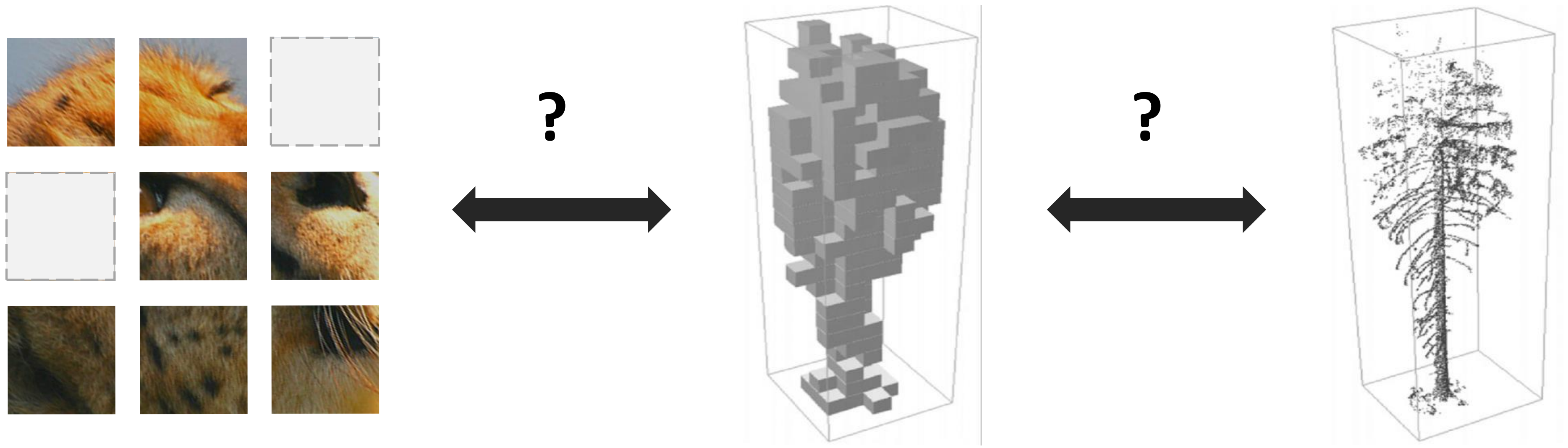


We found some issues and propose the solutions

issues 1 & 2: pixel intensity distribution shift & mask pattern vanishing

our motivation: the sparse nature of masked image coincides with sparse 3D point cloud

our solution: treat masked images as flatten point clouds, and use sparse convolutions [6]



masked 2D images

sparse 3D voxels

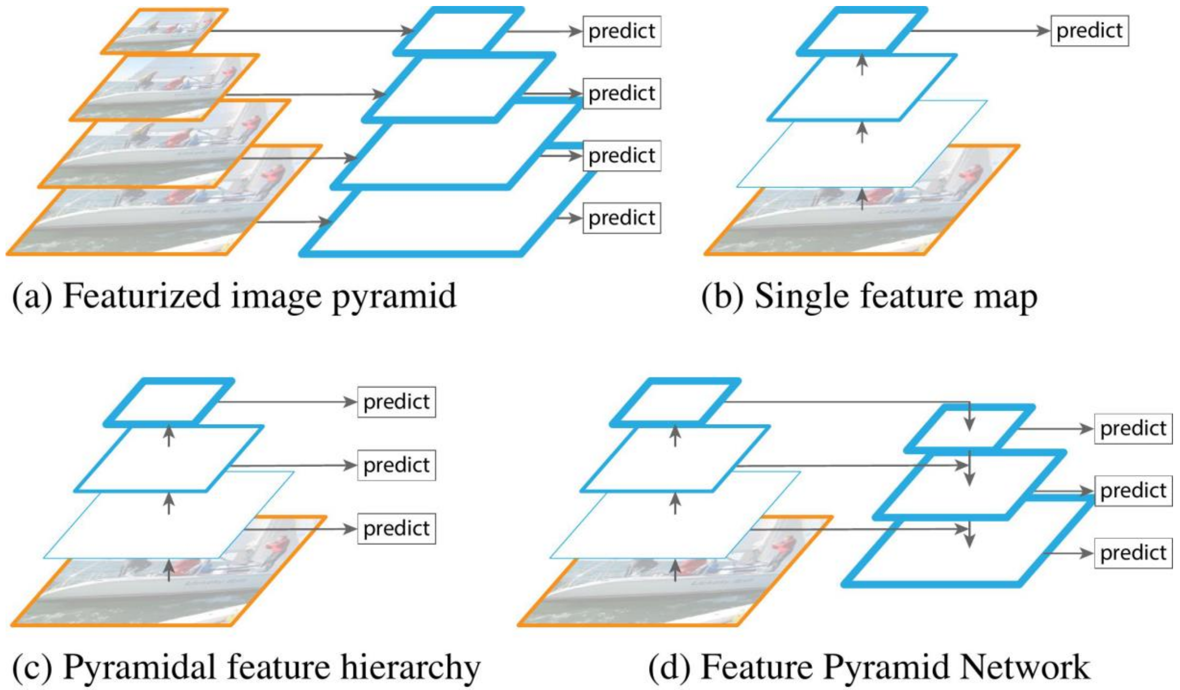
sparse 3D point cloud

[6] Graham, Benjamin, and Laurens Van der Maaten. "Submanifold sparse convolutional networks." arXiv preprint arXiv:1706.01307 (2017).

We found some issues and propose the solutions

issues 3: the single scale nature of BERT-pretraining is inconsistent with convnet’s multi scale

our solution: using a UNet-style encoder-decoder to learn multi scale (hierarchical) features



(a) Featurized image pyramid

(b) Single feature map

(c) Pyramidal feature hierarchy

(d) Feature Pyramid Network

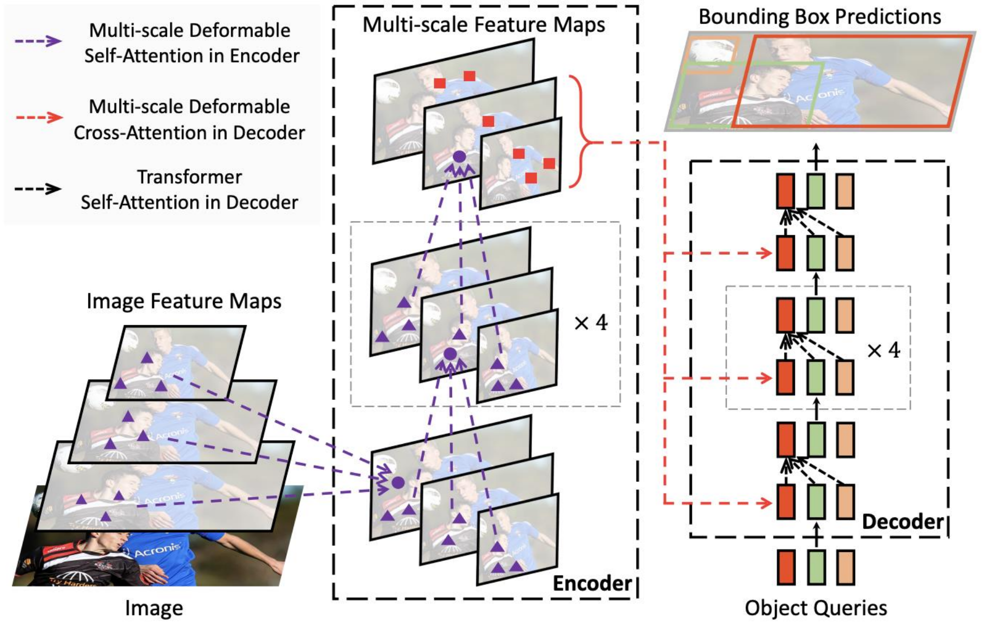
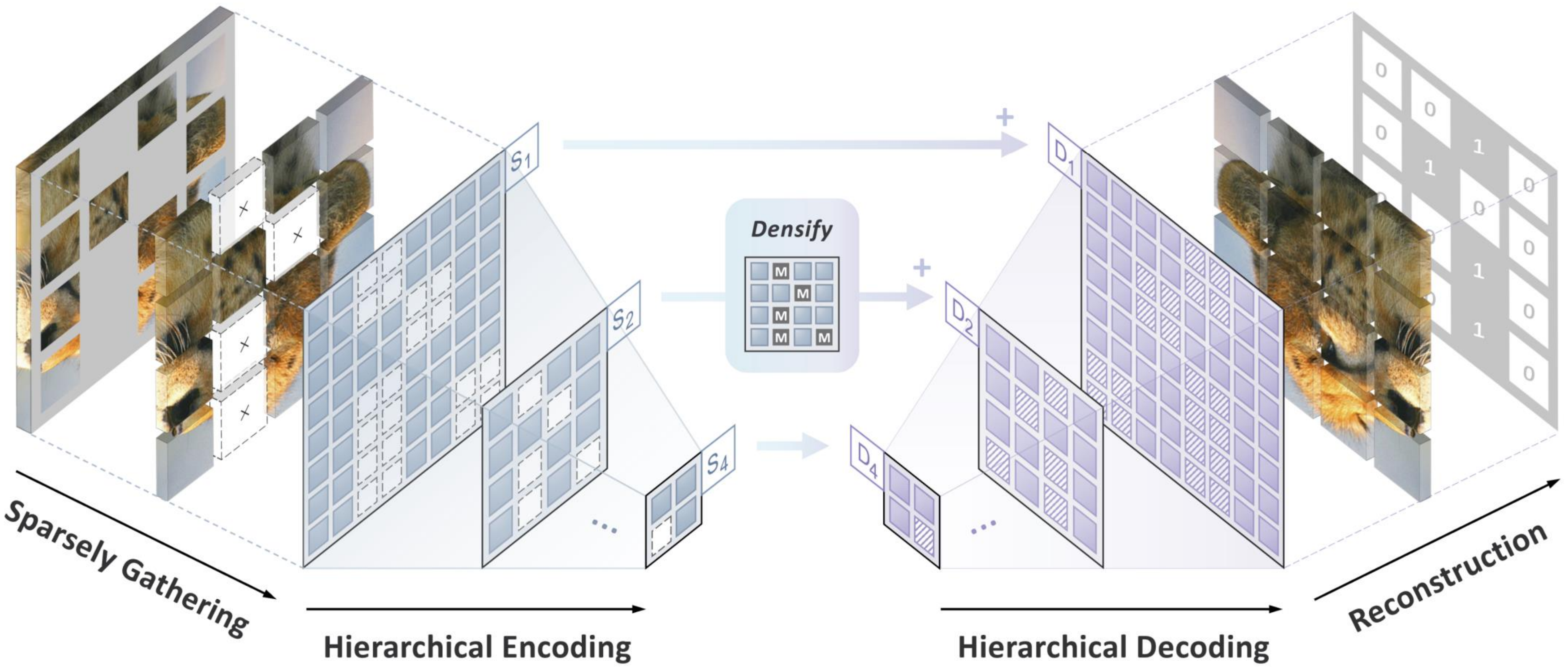

Multi-scale Deformable Self-Attention in Encoder
Multi-scale Deformable Cross-Attention in Decoder
Transformer Self-Attention in Decoder
Multi-scale Feature Maps
Bounding Box Predictions
× 4
× 4
Image Feature Maps
Image
Encoder
Decoder
Object Queries

Figure 1: Illustration of the proposed Deformable DETR object detector.

multi-scale matters in computer vision (left: FPN, right: def. DETR)

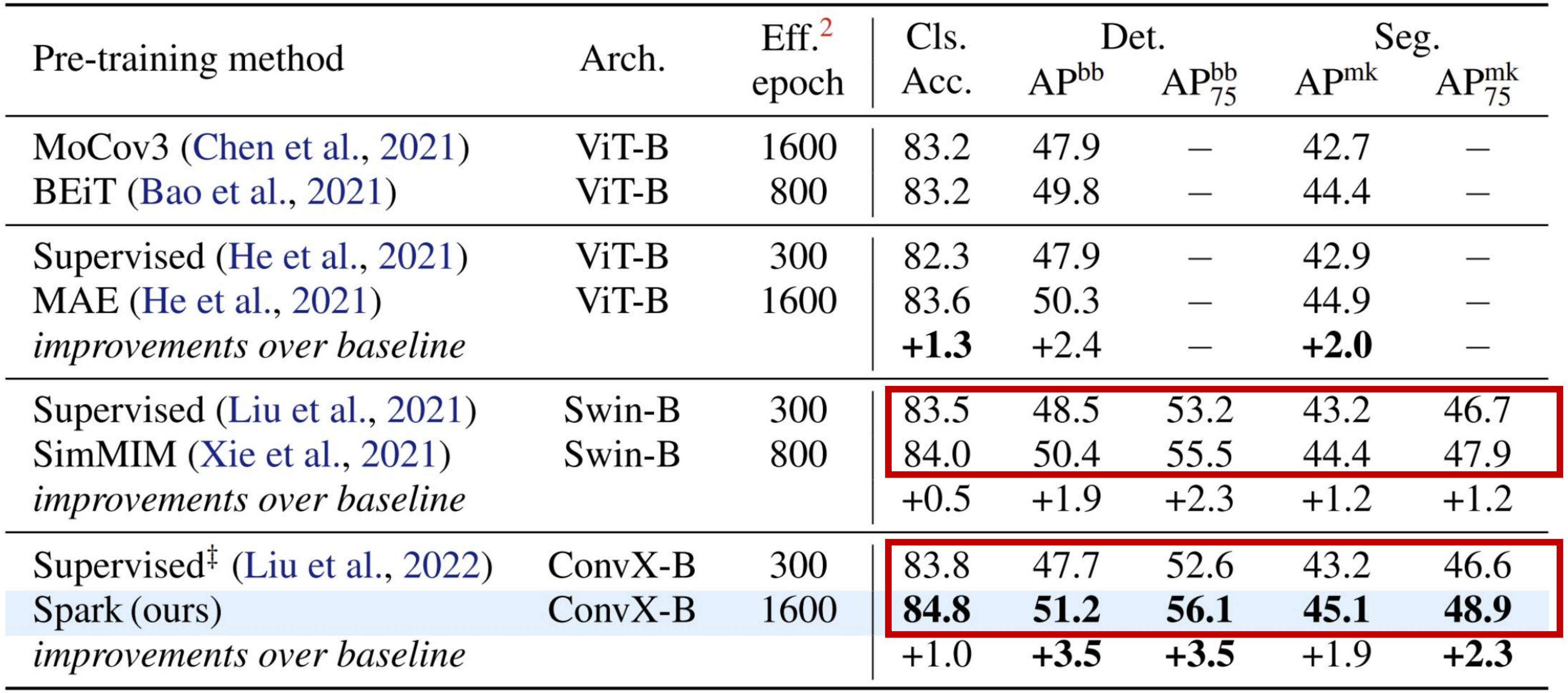
An overview of SparK



SparK: Sparse and hierarchical masKed modeling

Experimental results

- SparK pretrained convnet beat pretrained Swin-Transformer

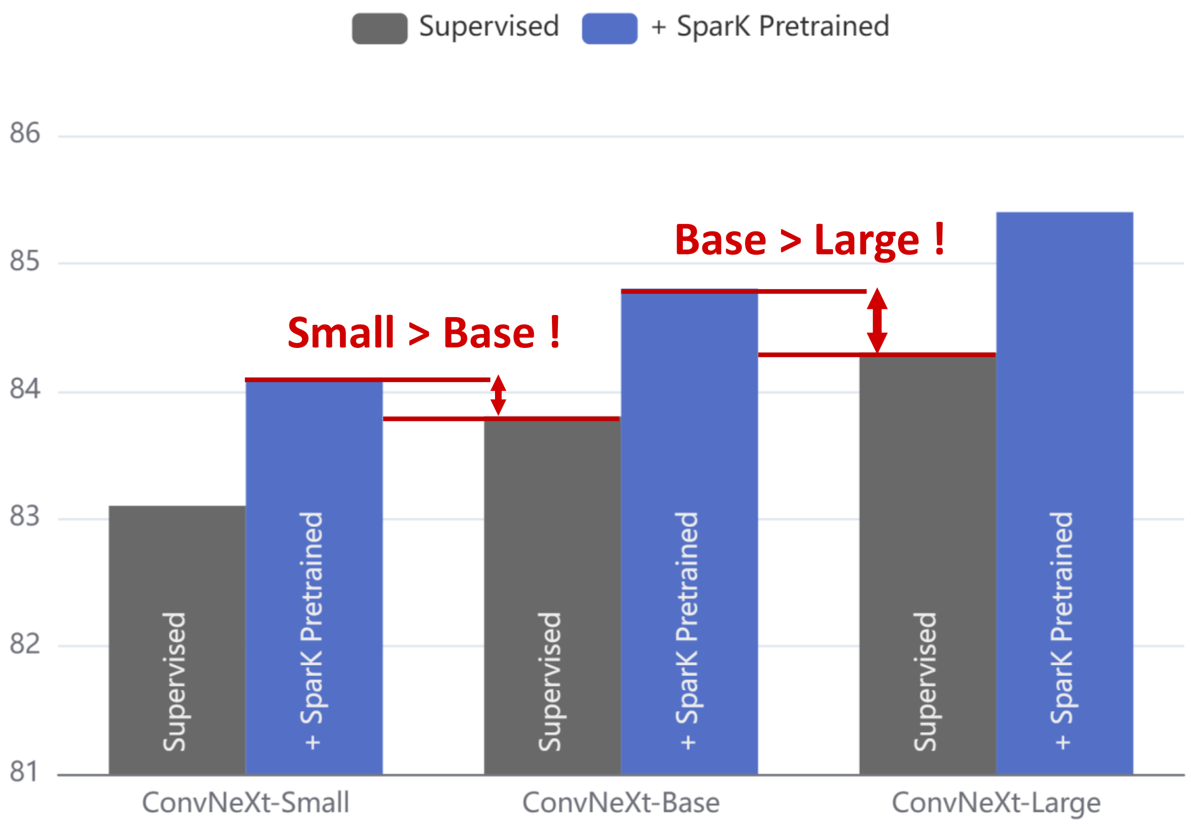


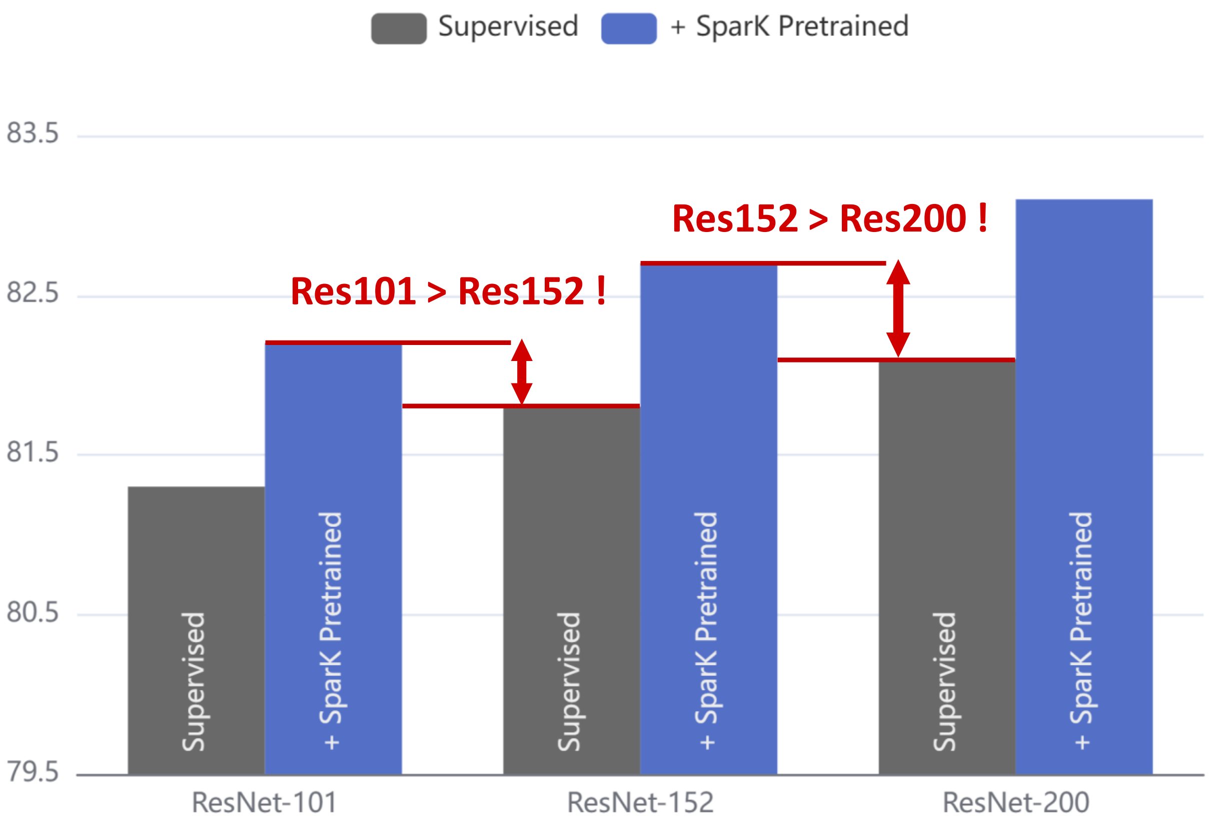
Pre-training method	Arch.	Eff.[2] epoch	Cls. Acc.	Det. AP^{bb}	AP^{bb}_{75}	Seg. AP^{mk}	AP^{mk}_{75}
MoCov3 (Chen et al., 2021)	ViT-B	1600	83.2	47.9	–	42.7	–
BEiT (Bao et al., 2021)	ViT-B	800	83.2	49.8	–	44.4	–
Supervised (He et al., 2021)	ViT-B	300	82.3	47.9	–	42.9	–
MAE (He et al., 2021)	ViT-B	1600	83.6	50.3	–	44.9	–
improvements over baseline			**+1.3**	+2.4	–	**+2.0**	–
Supervised (Liu et al., 2021)	Swin-B	300	83.5	48.5	53.2	43.2	46.7
SimMIM (Xie et al., 2021)	Swin-B	800	84.0	50.4	55.5	44.4	47.9
improvements over baseline			+0.5	+1.9	+2.3	+1.2	+1.2
Supervised‡ (Liu et al., 2022)	ConvX-B	300	83.8	47.7	52.6	43.2	46.6
Spark (ours)	ConvX-B	1600	**84.8**	**51.2**	**56.1**	**45.1**	**48.9**
improvements over baseline			+1.0	**+3.5**	**+3.5**	+1.9	**+2.3**

- without pretraining, Swin-B and ConvNeXt-B perform similarly
- with pretraining (SimMIM/SparK), ConvNeXt-B outperforms Swin-B by **large margin (≈+0.8)**
- this **strongly** verified the effectiveness of SparK pretraining

Experimental results

- ➢ 🔥 with SparK, smaller models can beat larger models! (cnx-small > cnx-base; Res101 > Res152)

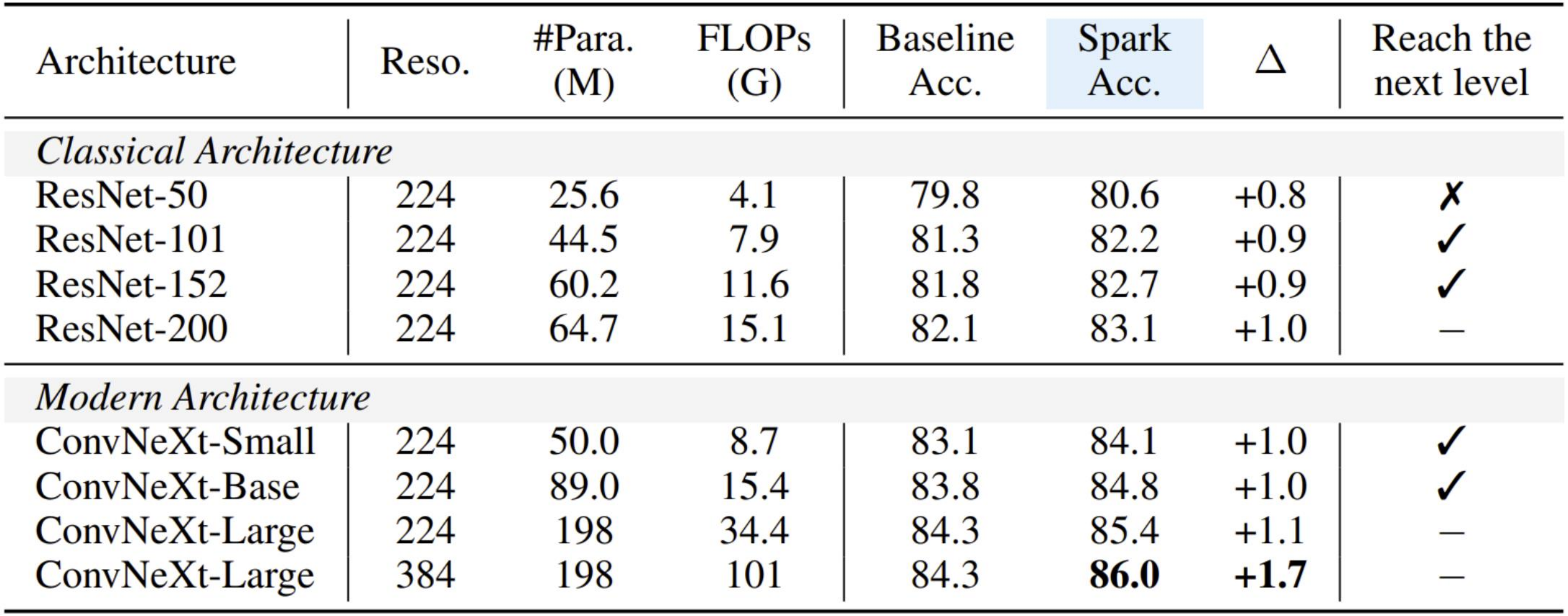




- ➢ another solid proof of SparK's effectiveness

Experimental results

- Good scaling behavior; SparK works on all convnets (ResNets, ConvNeXts, etc…)

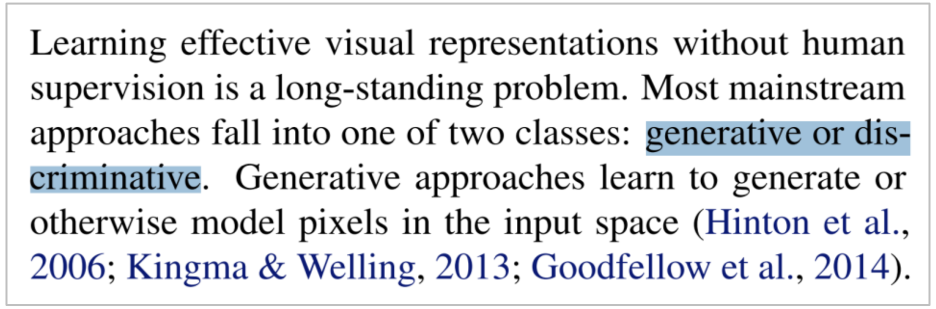


Architecture	Reso.	#Para. (M)	FLOPs (G)	Baseline Acc.	Spark Acc.	Δ	Reach the next level
Classical Architecture							
ResNet-50	224	25.6	4.1	79.8	80.6	+0.8	✗
ResNet-101	224	44.5	7.9	81.3	82.2	+0.9	✓
ResNet-152	224	60.2	11.6	81.8	82.7	+0.9	✓
ResNet-200	224	64.7	15.1	82.1	83.1	+1.0	–
Modern Architecture							
ConvNeXt-Small	224	50.0	8.7	83.1	84.1	+1.0	✓
ConvNeXt-Base	224	89.0	15.4	83.8	84.8	+1.0	✓
ConvNeXt-Large	224	198	34.4	84.3	85.4	+1.1	–
ConvNeXt-Large	384	198	101	84.3	**86.0**	**+1.7**	–

Experimental results

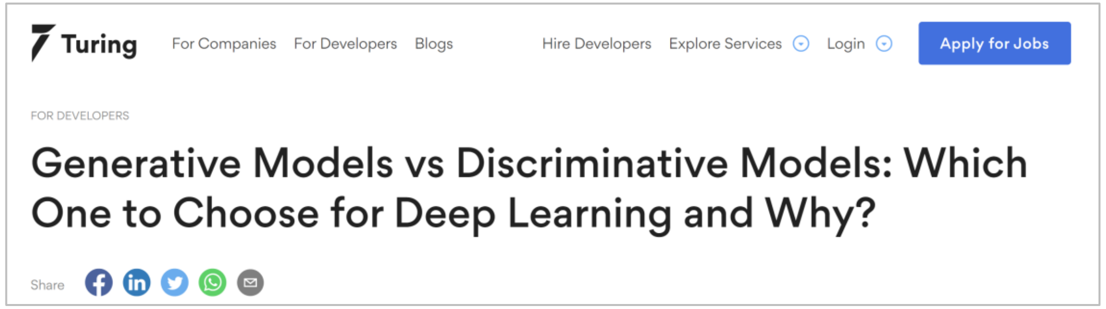
- On ResNets, Generative pre-training surpasses Contrastive Learning for the first time

Pre-training (on ResNet-50)	Pre-train task	Eff. epoch	Cls. (Acc.)	1× Schedule		2× Schedule	
				AP^{bb}	AP^{mk}	AP^{bb}	AP^{mk}
Supervised	–	–	79.8	38.9	35.4	41.3	37.3
SimSiam (Chen & He, 2021)	Contrastive	800	79.1	–	–	–	–
MoCo (He et al., 2020)	Contrastive	800	–	38.5	35.1	40.8	36.9
MoCov2 (Chen et al., 2020b)	Contrastive	1600	79.8	40.4	36.4	41.7	37.6
SimCLR (Chen et al., 2020a)	Contrastive	4000	80.0	–	–	–	–
InfoMin (Tian et al., 2020)	Contrastive	800	–	40.6	36.7	42.5	38.4
BYOL (Grill et al., 2020)	Contrastive	1600	80.0	40.4	37.2	42.3	38.3
SwAV (Caron et al., 2020)	Contrastive	1200	80.1	–	–	42.3	38.2
SparK (ours)	Generative	1600	**80.6**	**41.6**	**37.7**	**43.4**	**39.4**



Learning effective visual representations without human supervision is a long-standing problem. Most mainstream approaches fall into one of two classes: generative or discriminative. Generative approaches learn to generate or otherwise model pixels in the input space (Hinton et al., 2006; Kingma & Welling, 2013; Goodfellow et al., 2014).

source: SimCLR



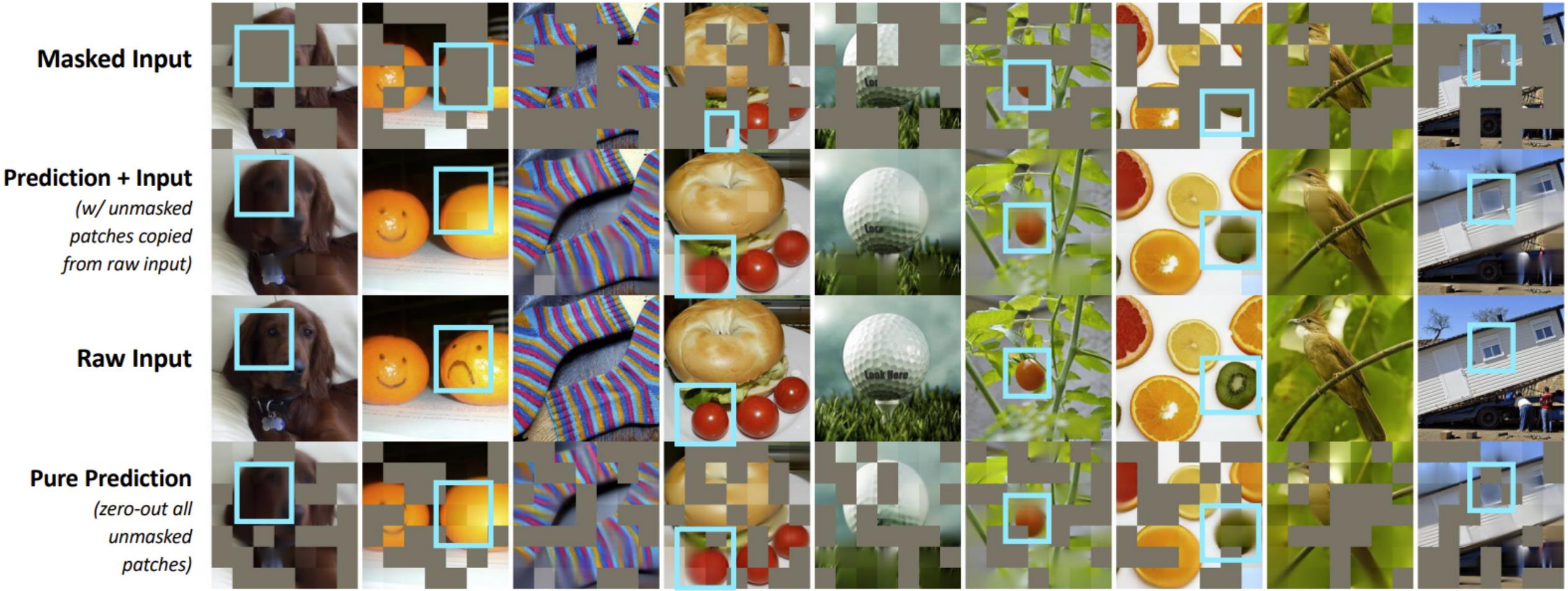
FOR DEVELOPERS

Generative Models vs Discriminative Models: Which One to Choose for Deep Learning and Why?

source: https://www.turing.com/kb/generative-models-vs-discriminative-models-for-deep-learning

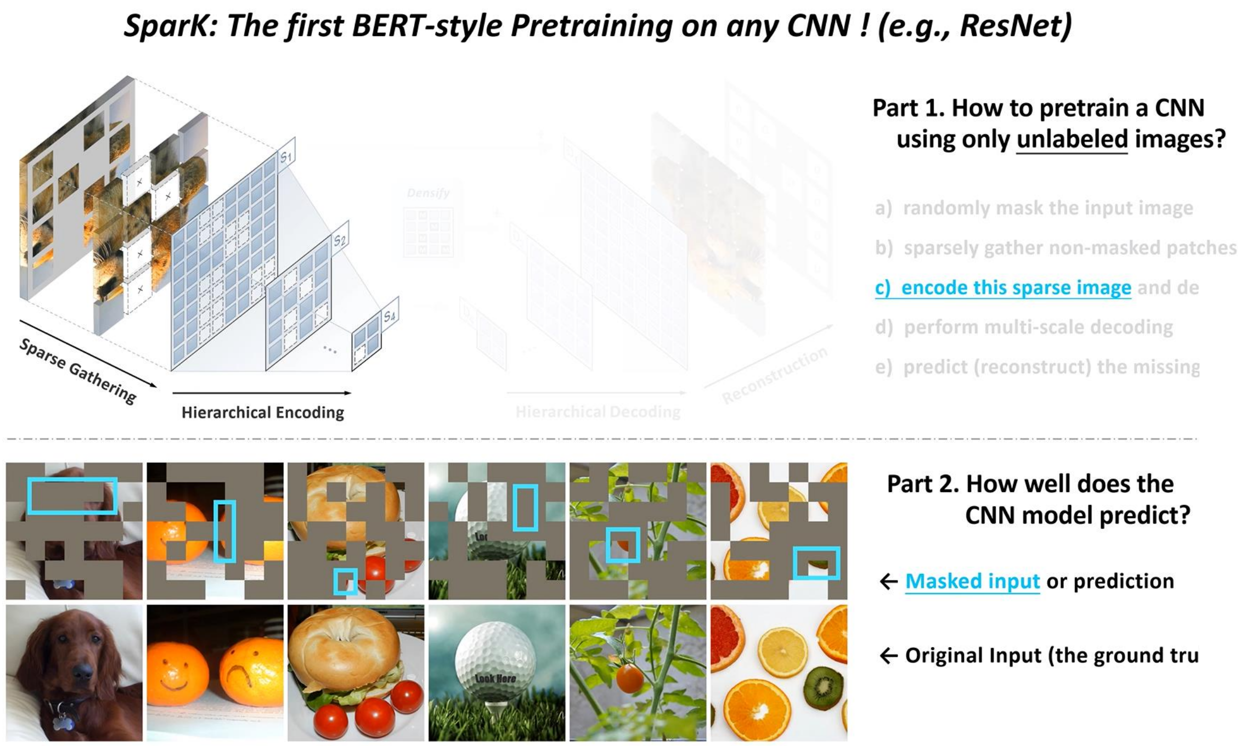
Visualizations

- The model can make different but reasonable predictions


Masked Input
Prediction + Input
(w/ unmasked patches copied from raw input)
Raw Input
Pure Prediction
(zero-out all unmasked patches)

Visualizations

- 🔥 We’ve also uploaded an animated video, and some visualization demos on GitHub



SparK: The first BERT-style Pretraining on any CNN ! (e.g., ResNet)

Part 1. How to pretrain a CNN using only unlabeled images?

a) randomly mask the input image

b) sparsely gather non-masked patches

c) encode this sparse image and de

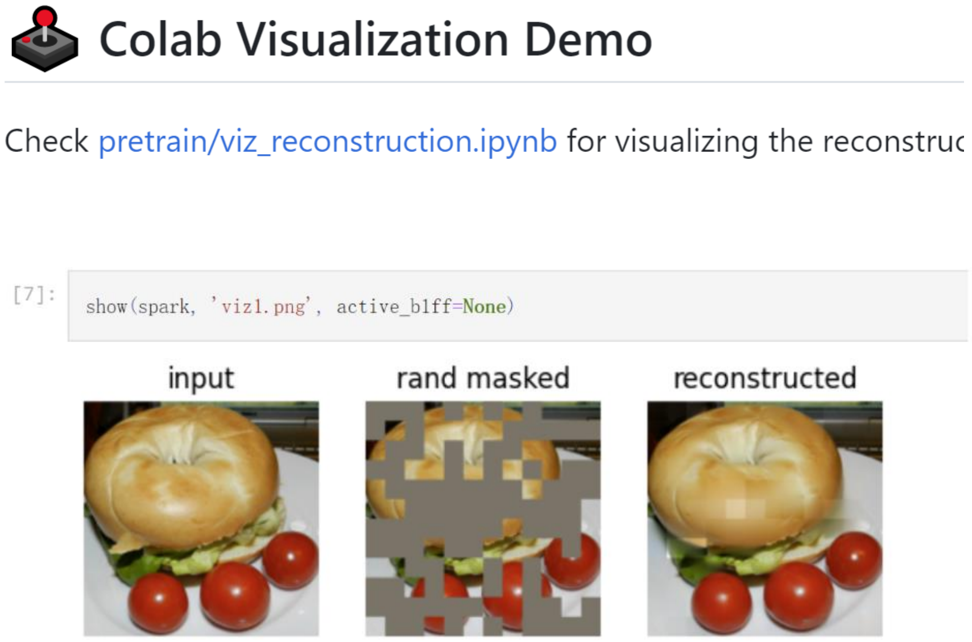
d) perform multi-scale decoding

e) predict (reconstruct) the missing

Part 2. How well does the CNN model predict?

← Masked input or prediction

← Original Input (the ground tru



Colab Visualization Demo

Check pretrain/viz_reconstruction.ipynb for visualizing the reconstruc

```
[7]: show(spark, 'viz1.png', active_b1ff=None)
```

github.com/keyu-tian/SparK

Recap & Takeaways

- we tried to analyze the fundamental challenges in applying BERT to convnets
- we proposed SparK, a BERT-style self-supervised pretraining for any convnet
- SparK shows a lot of inspiring experimental results
- codes, visualization playgrounds, and pretrained models are all released on GitHub
- Come to our poster or GitHub playground!

codes & models

Thanks!

github.com/keyu-tian/SparK

✉ tiankeyu.00@gmail.com