ICLR 2023 Spotlight

DESIGNING BERT FOR CONVOLUTIONAL NETWORKS: SPARSE AND HIERARCHICAL MASKED MODELING

Keyu Tian[1,2,3], **Yi Jiang**[2*], **Qishuai Diao**[2],
Chen Lin[4], **Liwei Wang**[1*], **Zehuan Yuan**[2]
[1]Center for Data Science, Peking University [2]Bytedance Inc
[3]Pazhou Lab (Huangpu) [4]University of Oxford
keyutian@stu.pku.edu.cn, {jiangyi.enjoy,diaoqishuai}@bytedance.com,
chen.lin@eng.ox.ac.uk, wanglw@pku.edu.cn, yuanzehuan@bytedance.com

Pre-train before fine-tuning

Chinese proverb: "Half of success is choosing a good starting point"

Outline

Background

What is BERT style pretraining ?

BERT pretraining in CV

BERT from NLP Transformers → Vision Transformers **Succussed**

BERT from NLP/Vision Transformers → CNN **Failed**

Why Failed?

Issue 1 : Pixel Intensity Distribution Shift

Issue 2 : Mask Pattern Vanishing

Issue 3 : A gap between CV&NLP in data processing

Solution to Issues 1&2 : Use **sparse convolution**

Solution to Issues 3 : Use **hierarchical encoder-decoder**

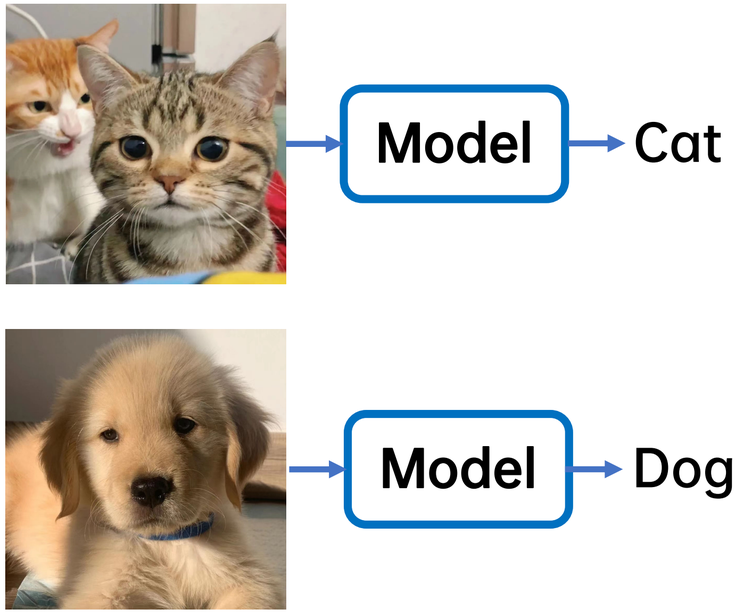
Results

Compared to MAE/ConvNeXt-V2

Background: What is BERT style pretraining ?

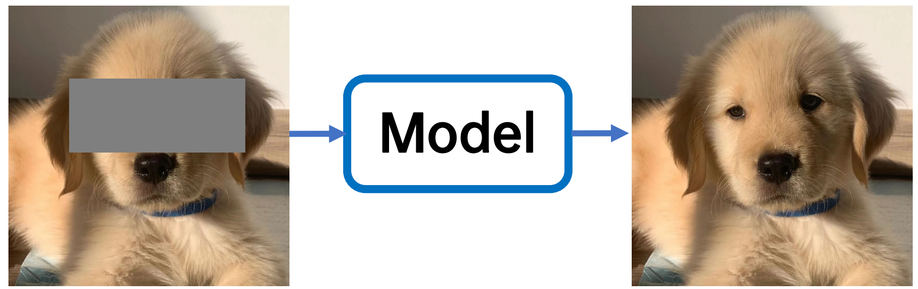
It's not supervised pretraining

Label dataset: $x \rightarrow y$



It's unsupervised/self-supervised pretraining

Unlabeled dataset: $\tilde{x} \rightarrow x$



Background: What is BERT style pretraining ?

It's not supervised pretraining

It's unsupervised/self-supervised pretraining

NLP self-supervised learning: BERT（Cloze test） & GPT（Answer）

BERT-style Masked Modeling[1]

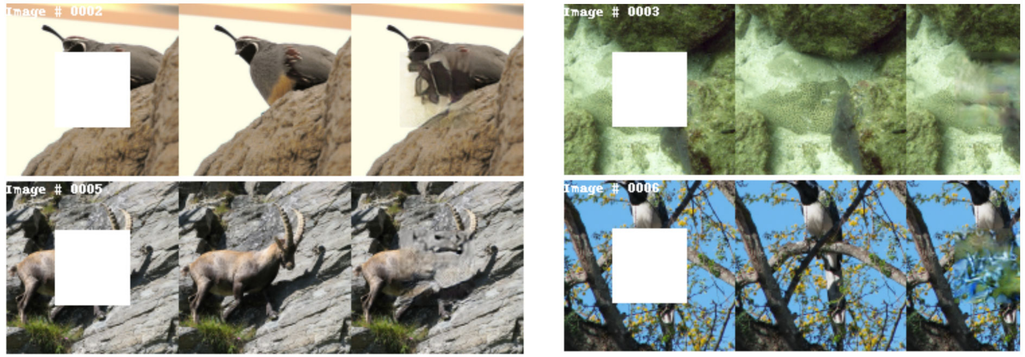
帮我__到老默，告诉__，我想__ __了
风浪越大，鱼__ __
什么__ __，和我用一样的

GPT-style Language Modeling

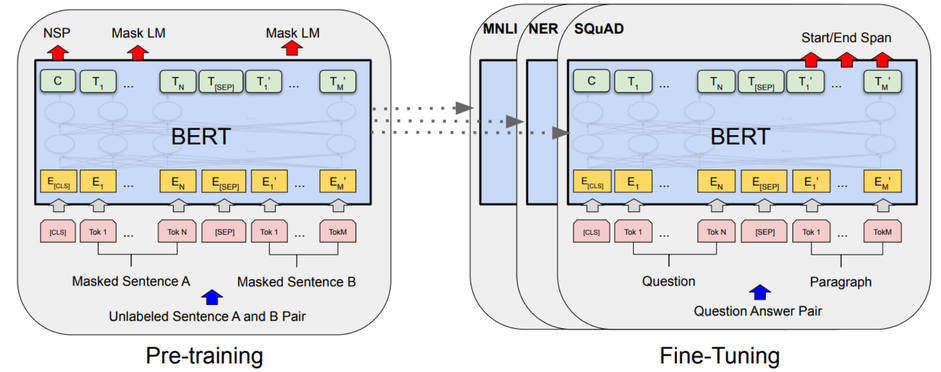
帮我找到老默，告诉他，我...
风浪越大，鱼...
什么档次，和我...

[1] credit:《狂飙》

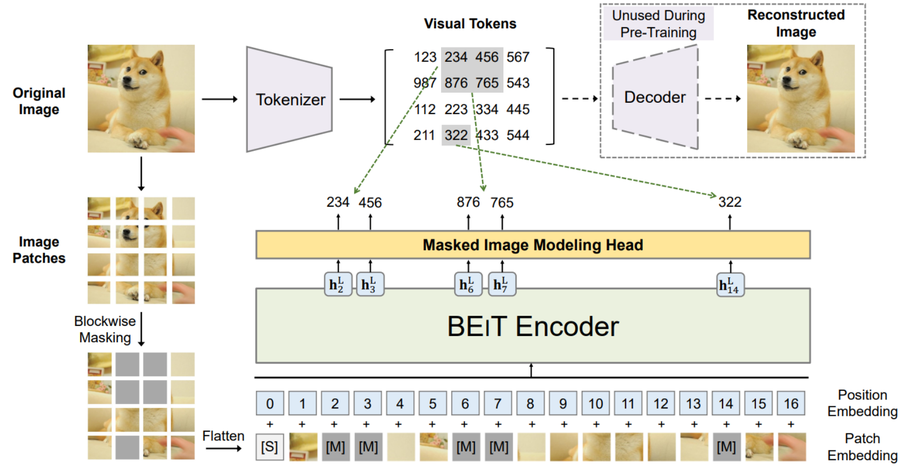
Background: BERT pretraining in CV



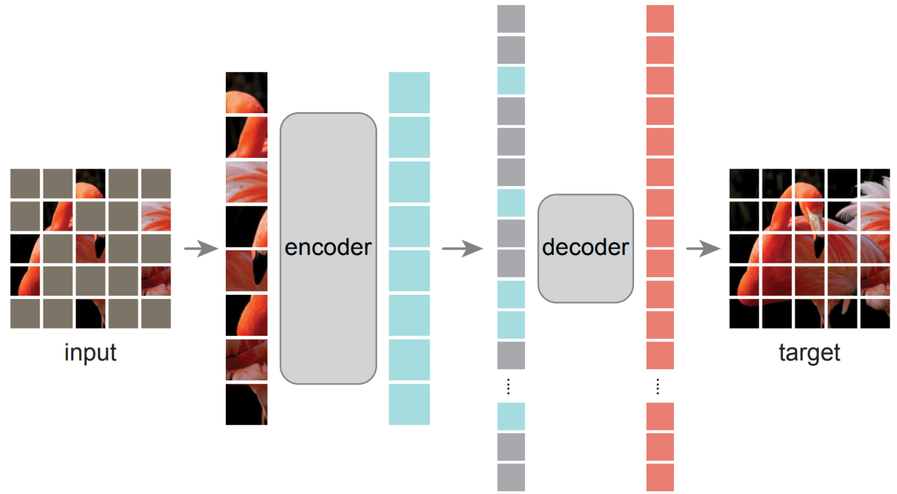
2016 Context encoders [1]



2018 Bert [2]



2021 Beit [3]



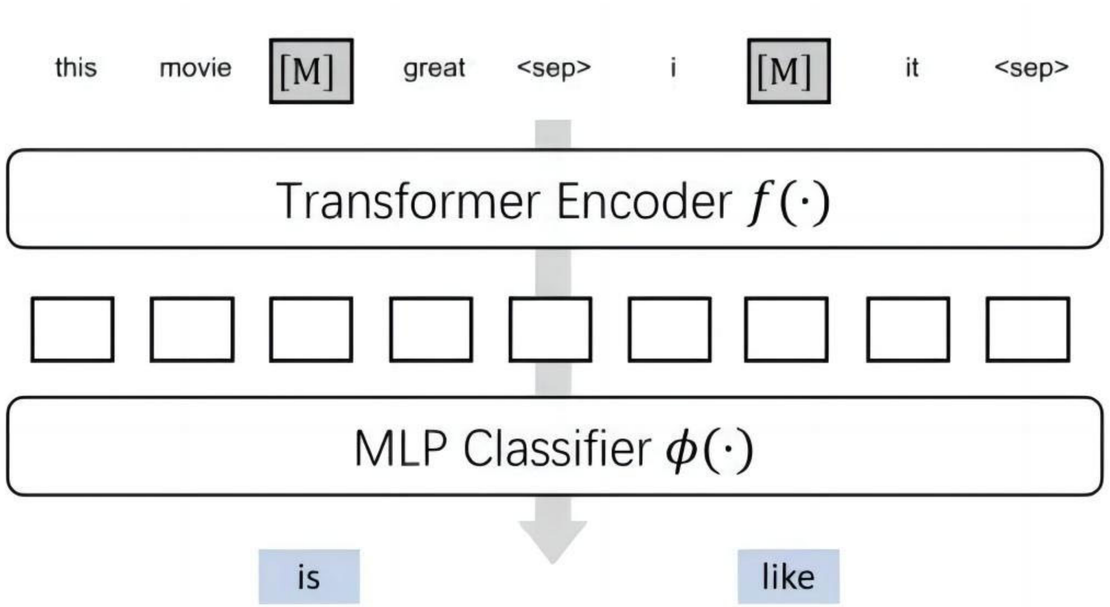
2021 MAE [4]

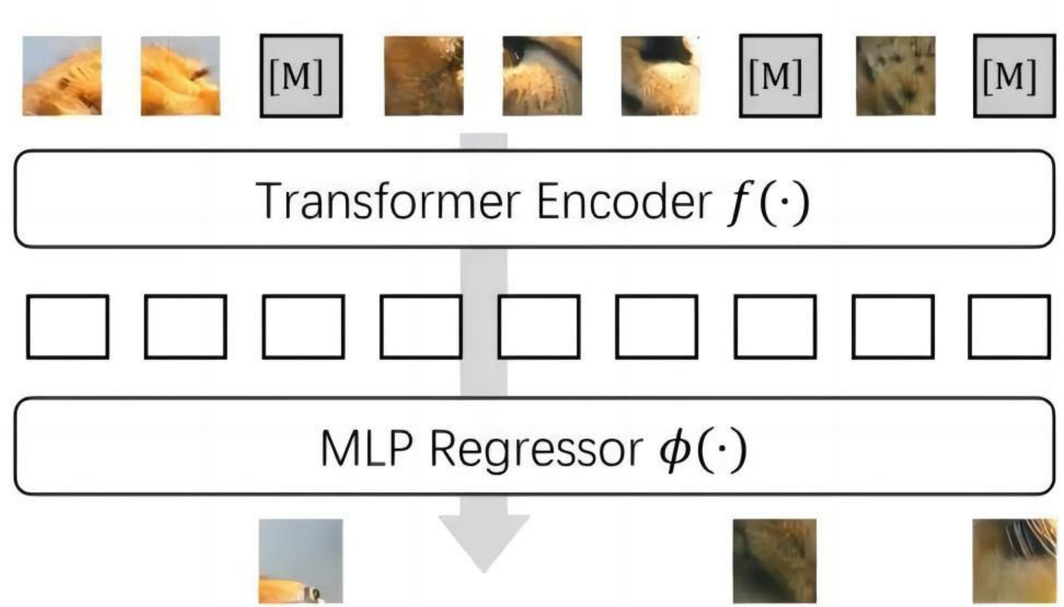
[1] Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[2] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
[3] Bao, Hangbo, et al. "Beit: Bert pre-training of image transformers." arXiv preprint arXiv:2106.08254 (2021).
[4] 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 from NLP Transformers → Vision Transformers

This success can be relatively straightforward to achieve because:

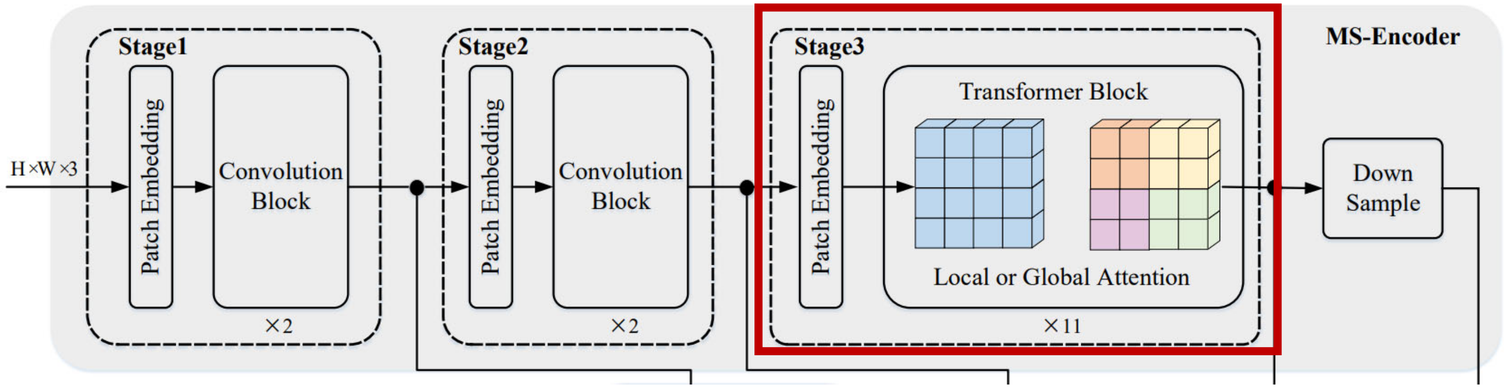
1. After word/patch embedding, there's no actual difference between NLP's T and ViT
2. Transformers are born to process variable-length token sequences
3. so "masking tokens" or "deleting tokens " can be very straightforward in this context





Background: BERT from NLP/Vision Transformers → CNN Failed

ConvMAE [1] still compromised by using not-fully-convolution model



Model	$[L_1, L_2, L_3]$
ConvMAE-S	[2, 2, 11]
ConvMAE-B	[2, 2, 11]
ConvMAE-B*	[2, 2, 11]
ConvMAE-L	[2, 2, 23]
ConvMAE-H	[2, 2, 31]

4 Conv blocks + 11 Transformer blocks

directly replacing the ViT to ConvNet (in MAE [2]) would result in a useless pretraining

	Method	Masking	Hierarchy	APE	Loss	Epoch	Acc.	Δ	std.
1	Not pretrained						83.1	-1.0	
2	SparK (ours)	sparse	✓	✗	masked only	1600	84.1	0.0	0.07
3	zero-outing	zero-outing	✓	✗	masked only	1600	83.2	-0.9	0.06

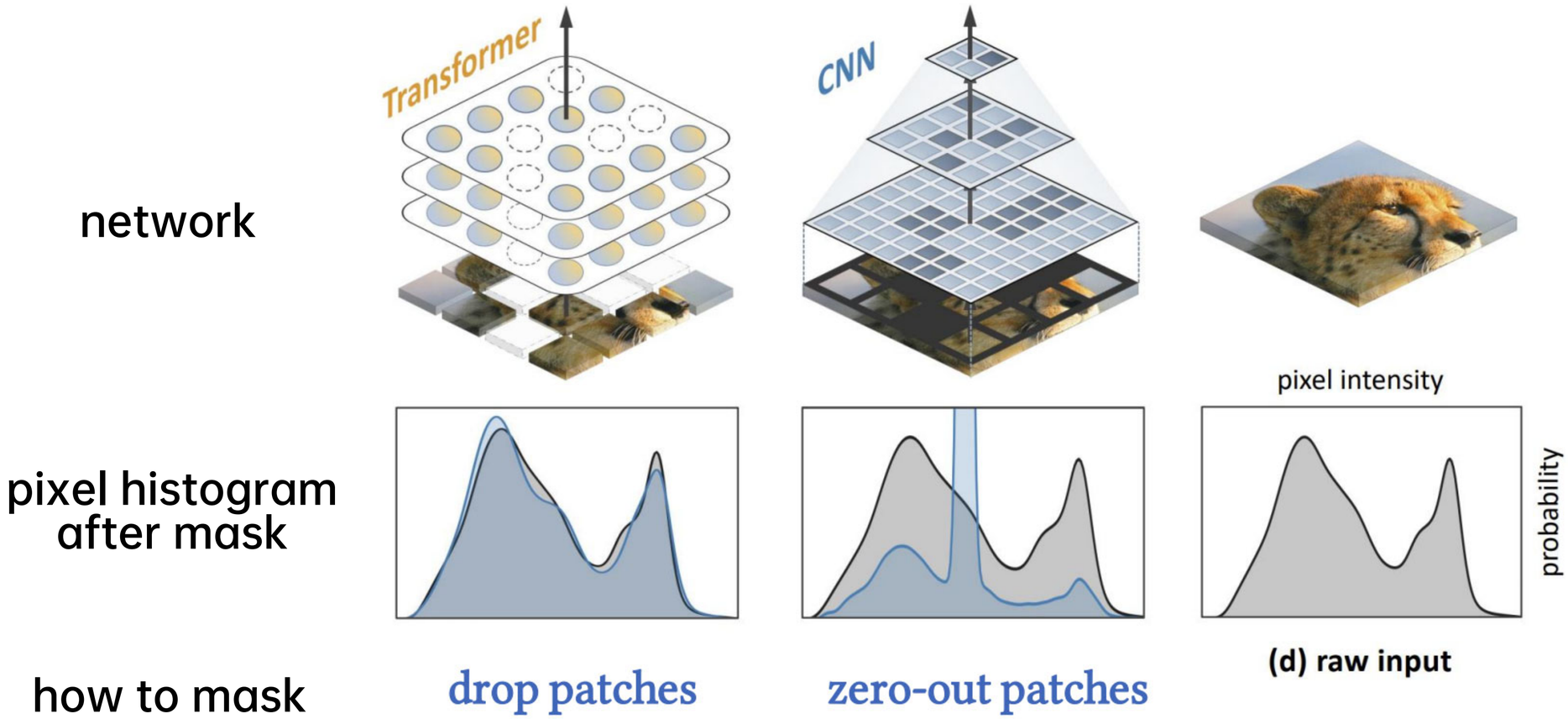
MAE (ViT → ConvNeXt)

[1] Gao, Peng, et al. "Convmae: Masked convolution meets masked autoencoders." arXiv 2022.
[2] He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

Why Failed? Issue 1 : Pixel Intensity Distribution Shift

Why Failed?

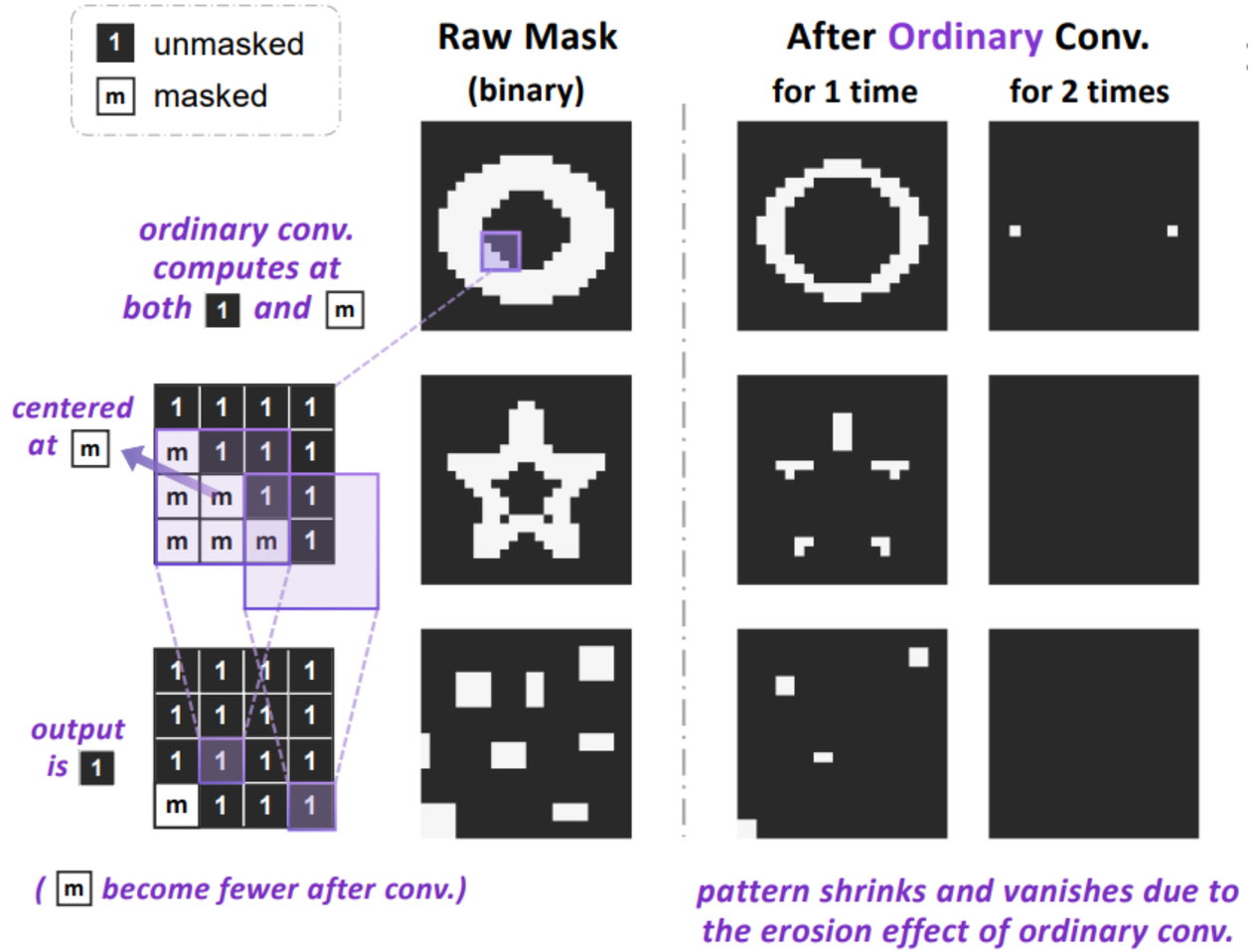
Transformer can drop patches, but CNN has to "zero-out" patch pixels

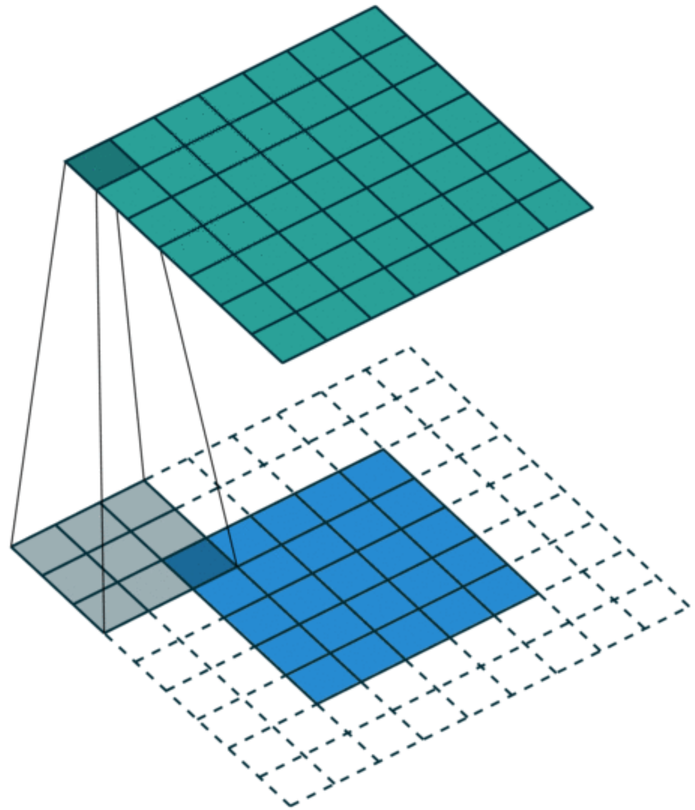


Why Failed? Issue 2 : Mask Pattern Vanishing

Doing convolution on a zero-outed image will make zero values (black pixels) fewer

This property of convolution makes those “mask patterns” vanish



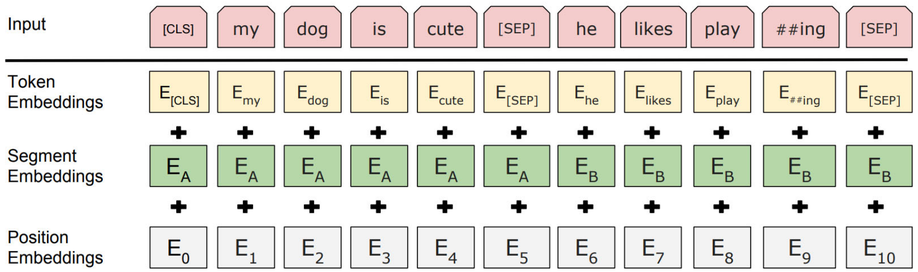


Why Failed? Issue 3 : A gap between CV&NLP in data processing

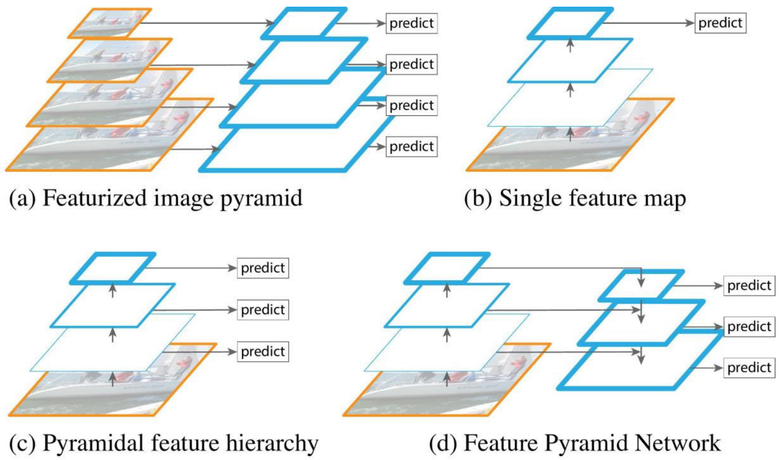
Words are semantic units in NLP; But there's no such "units" for image data.

Any visual processing system has to recognize objects (as "units") at different scales

Using multi-scale (hierarchical) structure is really a principle of many classical models in CV



BERT input representation[1]



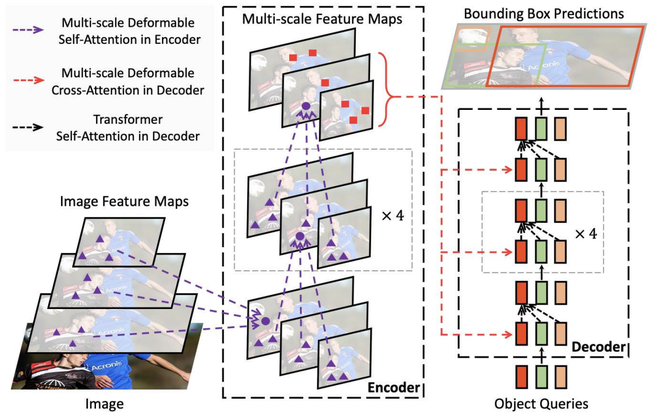


Figure 1: Illustration of the proposed Deformable DETR object detector.

Countless usages of feature pyramid (FPN[2], DETR[3], ...)

The BERT algorithm, from NLP, is naturally single-scale; But CNNs are always multi-scale

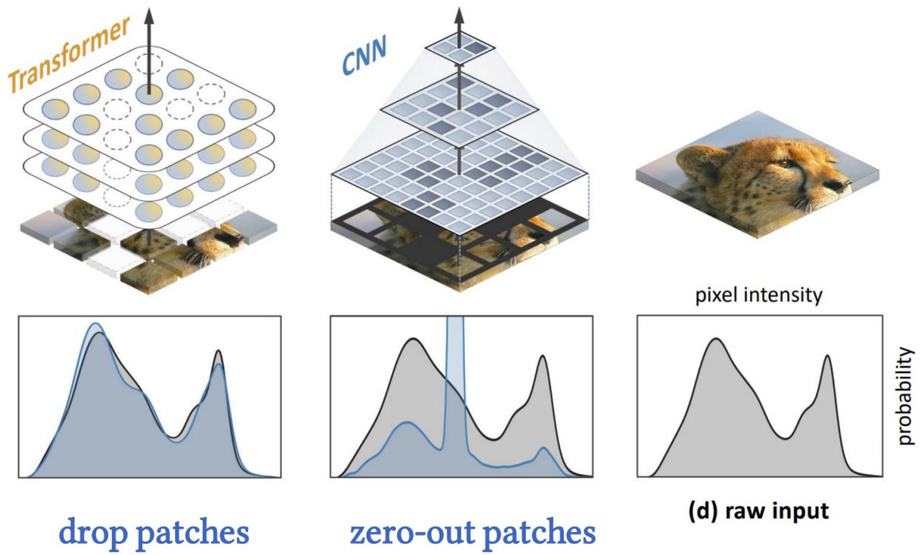
[1] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

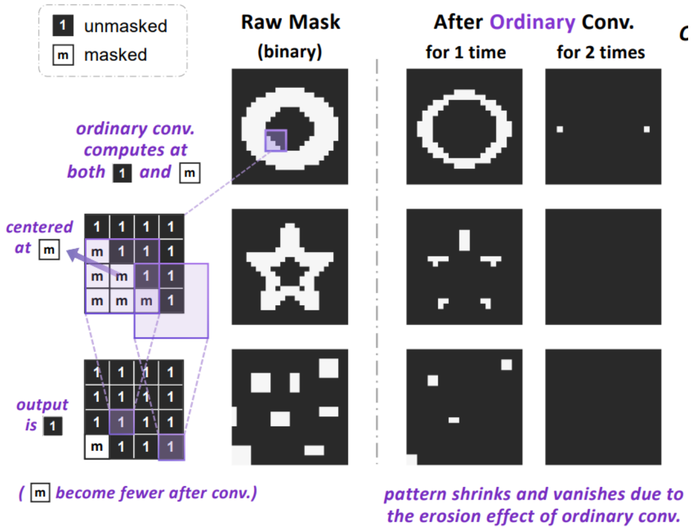
[2]Tsung-Yi Lin, et al. "Feature Pyramid Networks for Object Detection." 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017

[3] Xizhou Zhu, et al. "Deformable DETR: Deformable Transformers for End-to-End Object Detection." 9th International Conference on Learning Representations, ICLR 2021

Solution to Issues 1&2 : Use sparse convolution

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





ViT drops patches , so the pixel distribution does not shift

CNN has to zero-out patch pixels, so the distribution shifts

doing conv on a zero-outed image will make zero values fewer and fewer, thus break the sparsity

Root problem: CNNs can not handle irregular, randomly masked images (but ViT can)

Solution to Issues 1&2 : Use sparse convolution

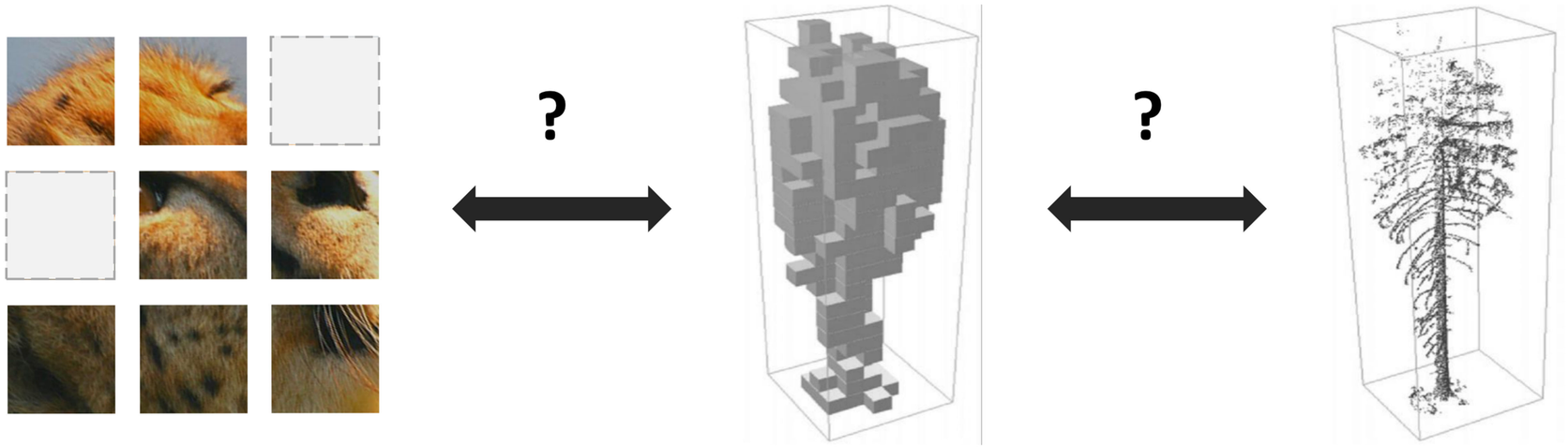
recall issues 1 & 2: pixel intensity distribution shift & mask pattern vanishing

Root problem: CNNs cannot handle irregular, randomly masked images (but ViT can)

Solution: use sparseconv[1] to allow CNNs to handle irregular, randomly masked images

Motivation:

The sparse nature of 3D point clouds coincides with those unmasked pixels

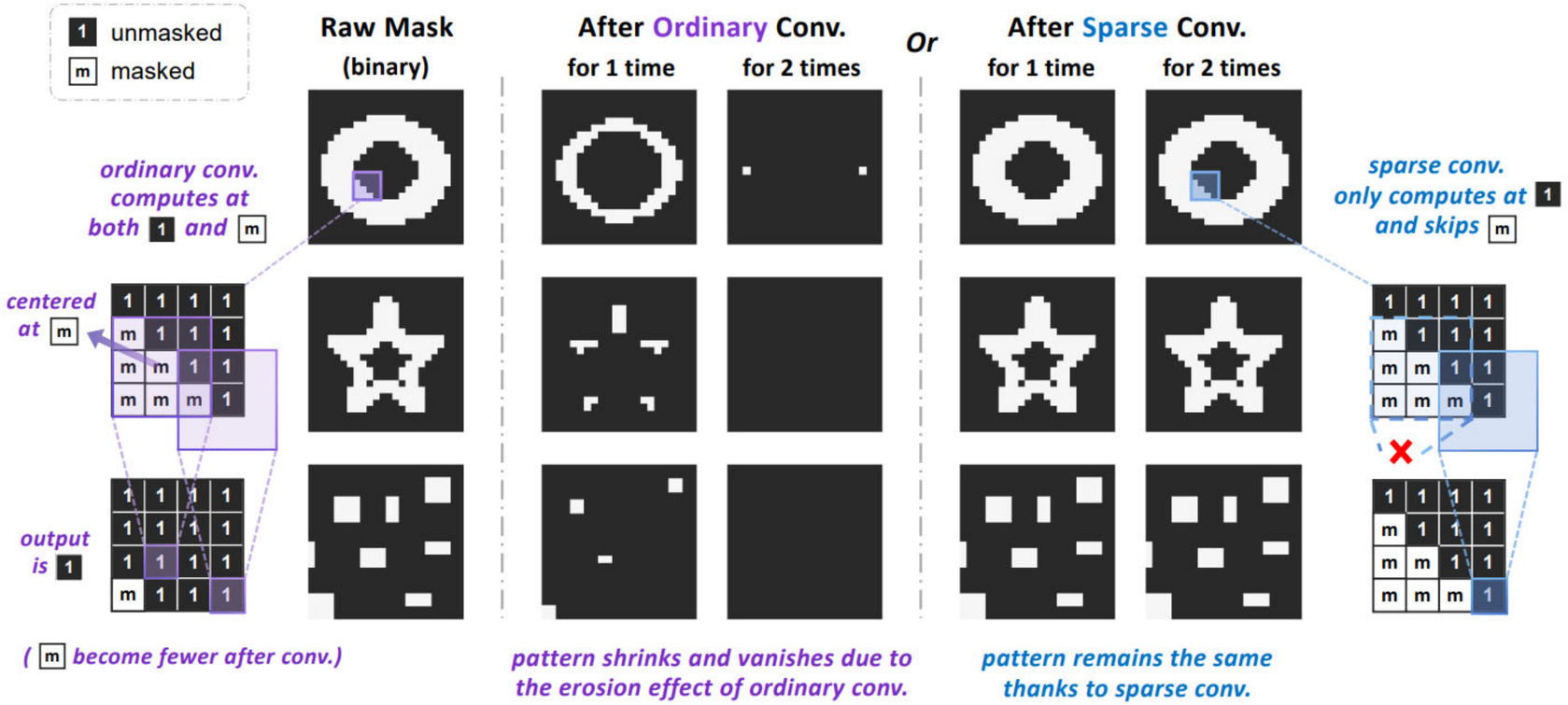


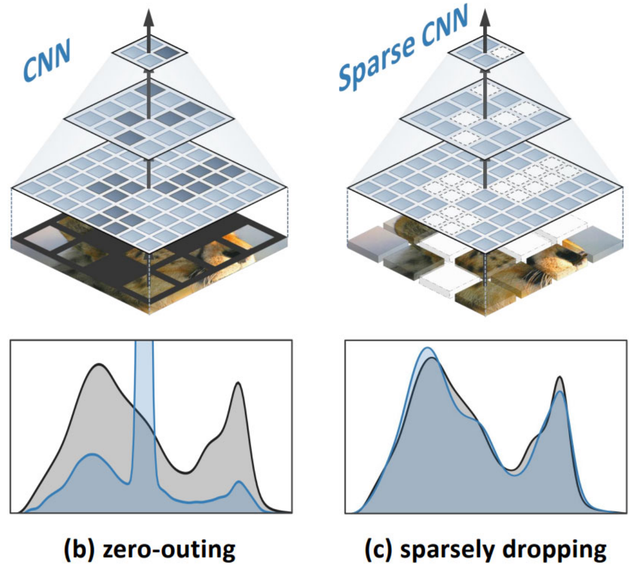
masked 2D images　　sparse 3D voxels　　sparse 3D point cloud

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

Solution to Issues 1&2 : Use sparse convolution

Solution: use sparseconv to allow CNNs to handle irregular, randomly masked images





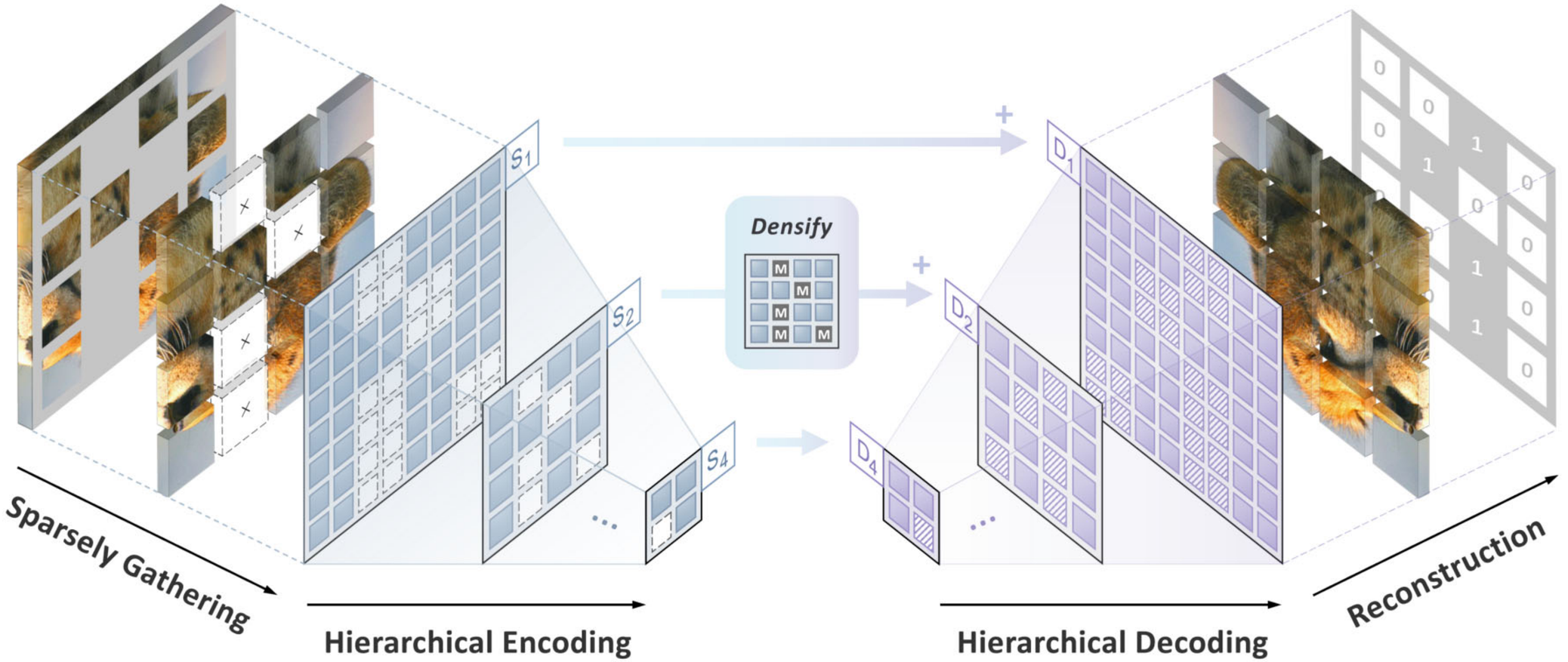
sparseconv will skip all "empty/masked/zero"positions

1. masked positions don't become fewer after sparseconv (solving "mask pattern vanishing")
2. no need to "zero-out pixels" to "simulate" the dropping operation(solving "pixel distribution shift")

Solution to Issues 3 : Use hierarchical encoder-decoder

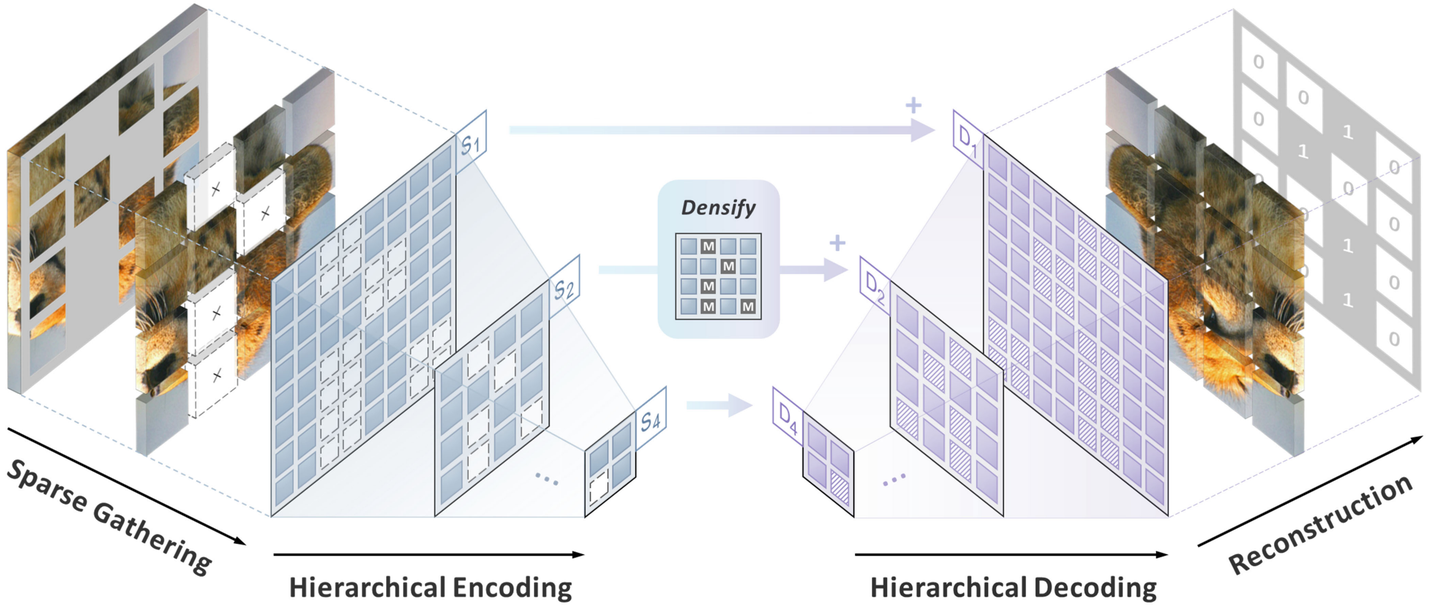
recall issues 3: BERT from NLP is single-scale, but CNNs are always multi-scale

make BERT multi-scale by using a multi-scale encoder-decoder (UNet-style)



the sparse feature S, will be "densify" by filling in [mask] tokens, and then fed into decoder to get D

SparK: Sparse and hierarchical masKed modeling



Part 1. How does our pretraining algorithm work?

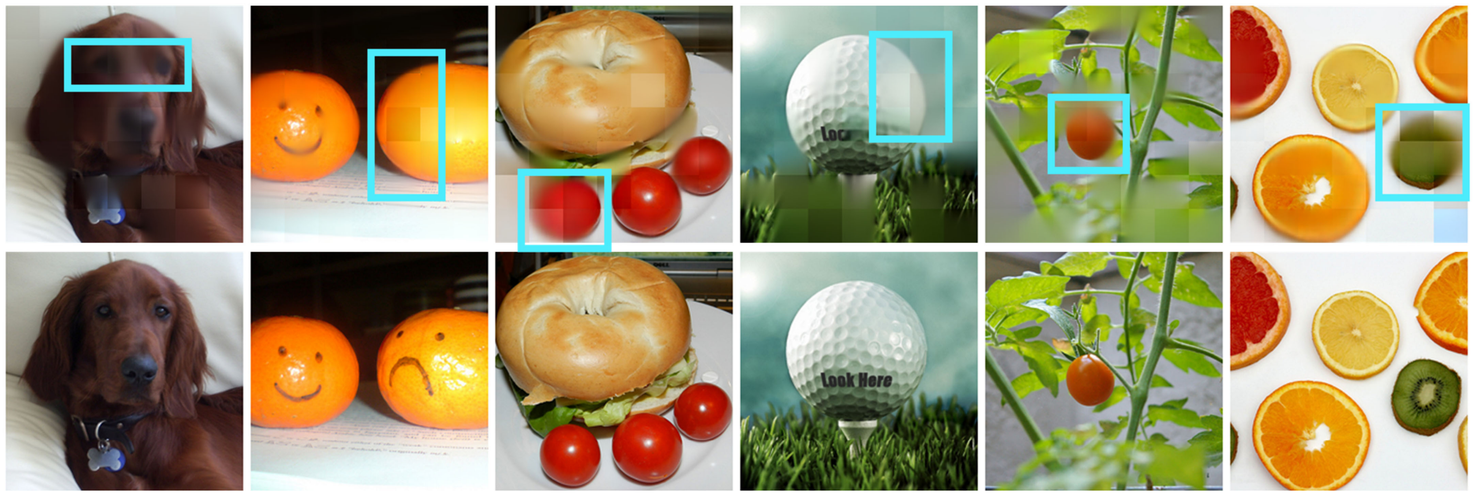
a) randomly mask the input image

b) sparsely gather non-masked patches

c) encode this sparse image and decode

d) perform multi-scale decoding

e) predict (reconstruct) the missing parts



Part 2. How well does the model predict?

(Masked input or model prediction)

(Original Input; the ground truth)

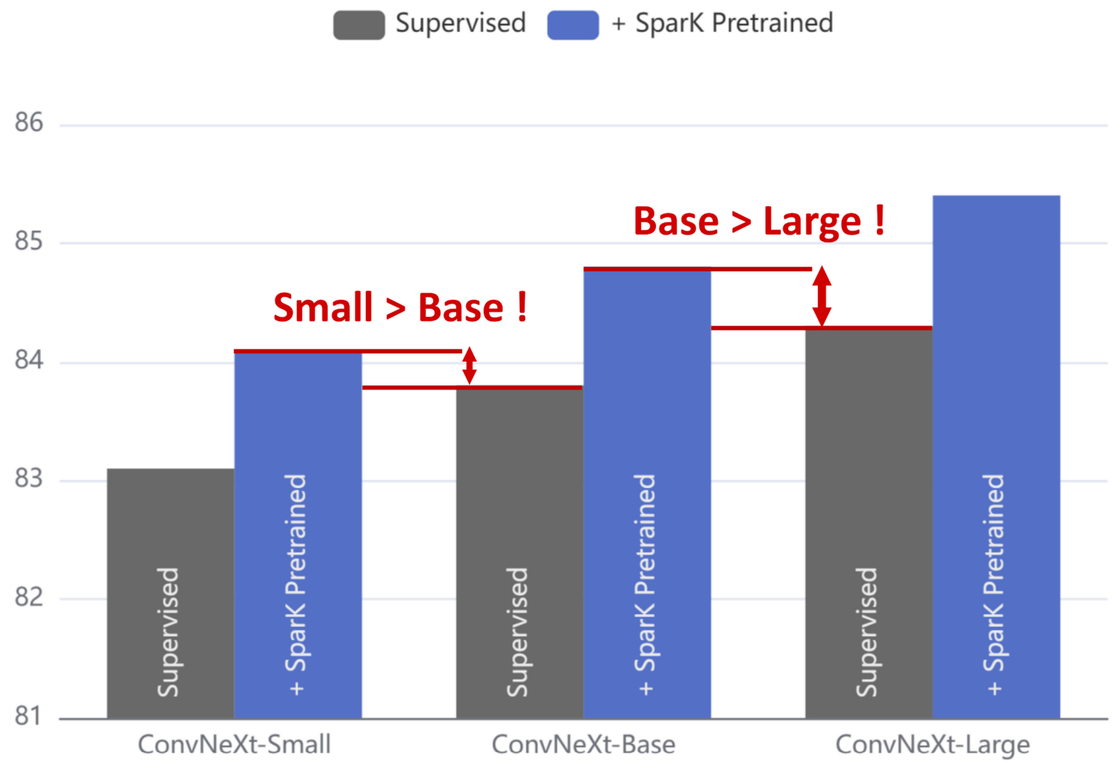
Results: pretrained CNNs beat pretrained transformers

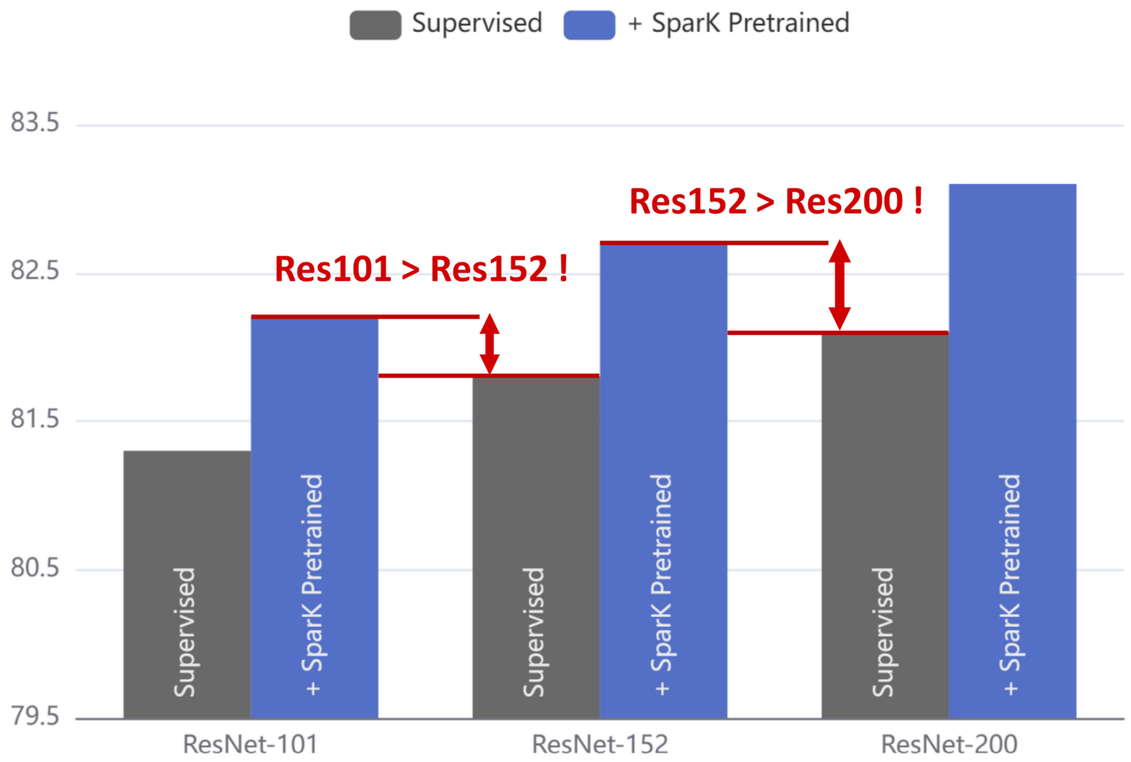
Pre-training method	Arch.	Eff.[2] epoch	Cls. Acc.	Det. AP^{bb}	Det. AP^{bb}_{75}	Seg. AP^{mk}	Seg. AP^{mk}_{75}
MoCov3 (Chen et al., 2021)	ViT-B	1600	83.2	47.9	–	42.7	–
BEiT (Bao et al., 2021)	ViT-B	800	83.2	49.8	–	44.4	–
Supervised (He et al., 2021)	ViT-B	300	82.3	47.9	–	42.9	–
MAE (He et al., 2021)	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 or Spark), ConvNeXt-B outperforms Swin-B by large margin (≈+0.8)

Results: smaller CNNs, with SparK, can beat larger ones

SparK-pretrained smaller models can beat non-pretrained larger models





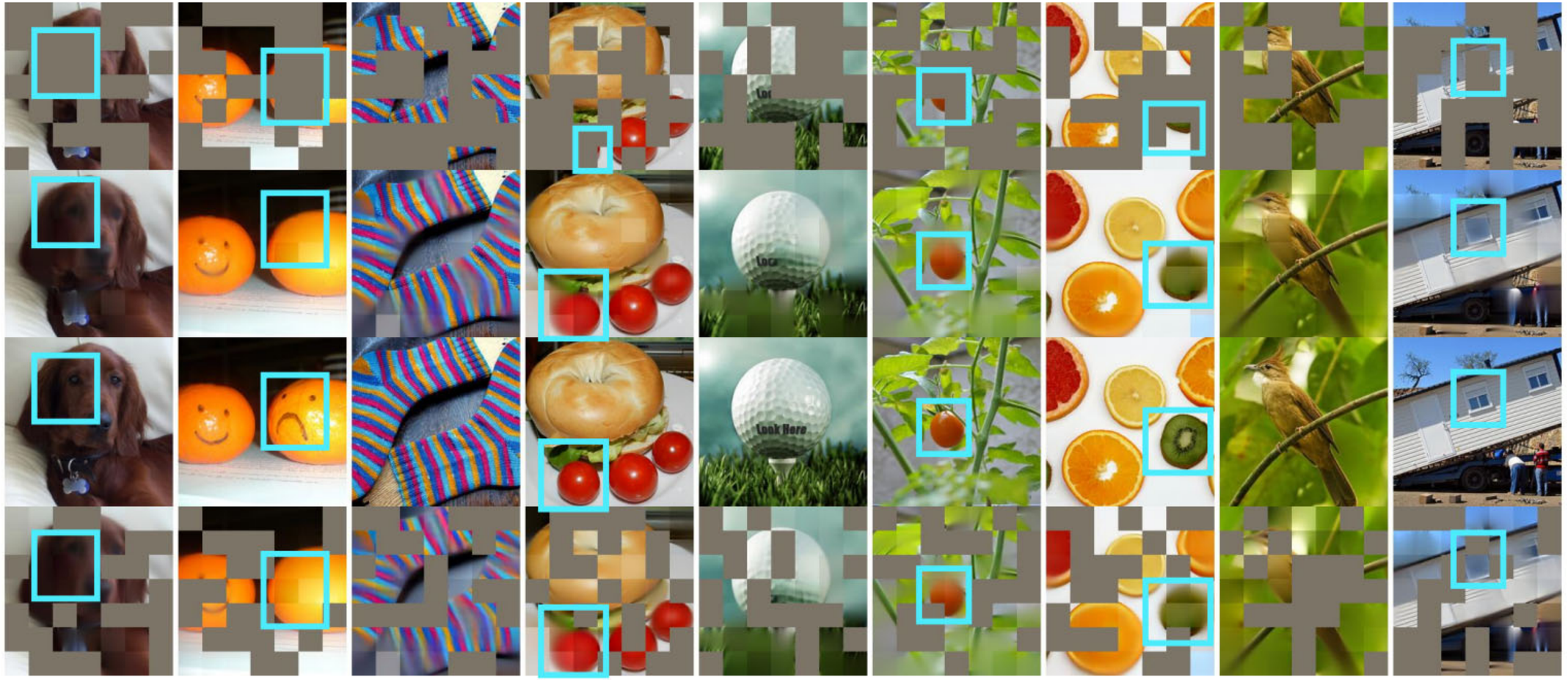
Results: generative self-supervised learning beats contrastive learning

SparK, as a generative pre-training, surpasses state-of-the-art contrastive learning

Table 3: **ResNet-50 results on downstream tasks.** SparK is compared to state-of-the-art contrastive learning algorithms. For ImageNet, the same training recipe from Wightman et al. (2021) (300-epoch fine-tuning with 224 resolution) is used. For COCO, Mask R-CNN ResNet50-FPN is equally fine-tuned for 12 or 24 epochs (1× or 2×), with average precision on `val2017` reported. SparK is highlighted as the only *generative* method.

Pre-training (on ResNet-50)	Pre-train task	Eff. epoch	Cls. (Acc.)	1× Schedule AP^{bb}	1× Schedule AP^{mk}	2× Schedule AP^{bb}	2× Schedule 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**

Results: Visualization



Results: Ablation studies

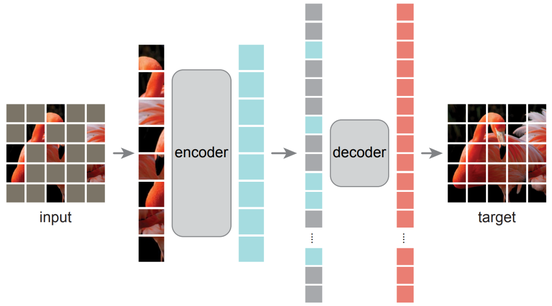
every component matters (sparse masking, hierarchical modeling)
some other choices (e.g., not using APE) are purely performance-driven

Table 5: **The ablation study on the importance of each components in SparK**. Experiments are based on ConvNeXt-Small, with ImageNet validation accuracy reported. Our default setting is in row 2. Differences are highlighted in blue. "APE": absolute positional embedding; "std.": standard deviation of four experiments.

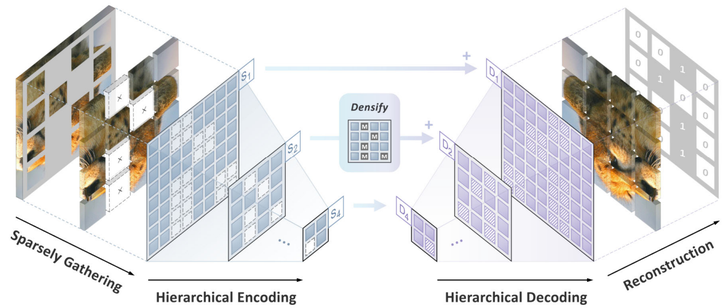
	Method	Masking	Hierarchy	APE	Loss	Epoch	Acc.	Δ	std.
1	Not pretrained						83.1	-1.0	
2	SparK (ours)	sparse	✓	✗	masked only	1600	84.1	0.0	0.07
3	zero-outing	zero-outing	✓	✗	masked only	1600	83.2	-0.9	0.06
4	w/o hierarchy	sparse	✗	✗	masked only	1600	83.6	-0.5	0.04
5	w/ APE	sparse	✓	✓	masked only	1600	83.9	-0.2	0.10
6	w/ more loss	sparse	✓	✗	all	1600	83.3	-0.8	0.12
7	pre-train less	sparse	✓	✗	masked only	800	83.7	-0.4	0.05

Compared to MAE/ConvNeXt-V2

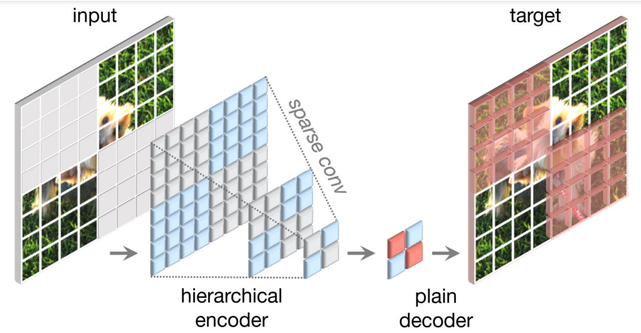
MAE (2021/11 @ arxiv)



SparK (2022/10 @ openreview)



ConvNextV2 (2023/01 @ arxiv)



Method	MAE[1] (2021/11)	Spark[2] (2022/10)	ConvNextV2[3] (2023/01)
can be used on any convnet	✗ on VITs	✓ ResNets, ConvNexts..	✗ on modified ConvNext
Downstream codebases provided (besides ImageNet)	✗ ImageNet only	✓ COCO det, COCO seg	✗ ImageNet only
tested on very light models	✗ base+ only	✗ small+ and ResNet50+	✓ some nano models

[1] He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
[2] Keyu Tian, et al. "Designing BERT for Convolutional Networks: Sparse and Hierarchical Masked Modeling." The Eleventh International Conference on Learning Representations, ICLR 2023.
[3] J Woo, Sanghyun, et ol."ConvNeXt V2: Co-designing and Scaling ConyNets with Mosked Autoencoders,"arXiv preprint arXiv:2301.00808 (2023).

Conclusion

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

Solution: use sparseconv to allow CNNs to handle irregular, randomly masked images

Issues 3: BERT from NLP is single-scale, but CNNs are always multi-scale

Solution: make BERT multi-scale by using a multi-scale encoder-decoder (UNet-style)

1.Thinking about the fundamental difference between language and vision would help.

2.Some "old but "golden"principles in classics of computer vision still make a lot of sense.

3.Convnets would live for a long time; Trying to exploit its more potential by BERT/Spark?

4.Future researches: GPT-style pretraining?

Code : github.com/keyu-tian/SparK