KAIST Meta AI NYU

Arxiv: https://arxiv.org/abs/2301.00808
Code: https://github.com/facebookresearch/ConvNeXt-V2

@ WED-PM-360

ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders

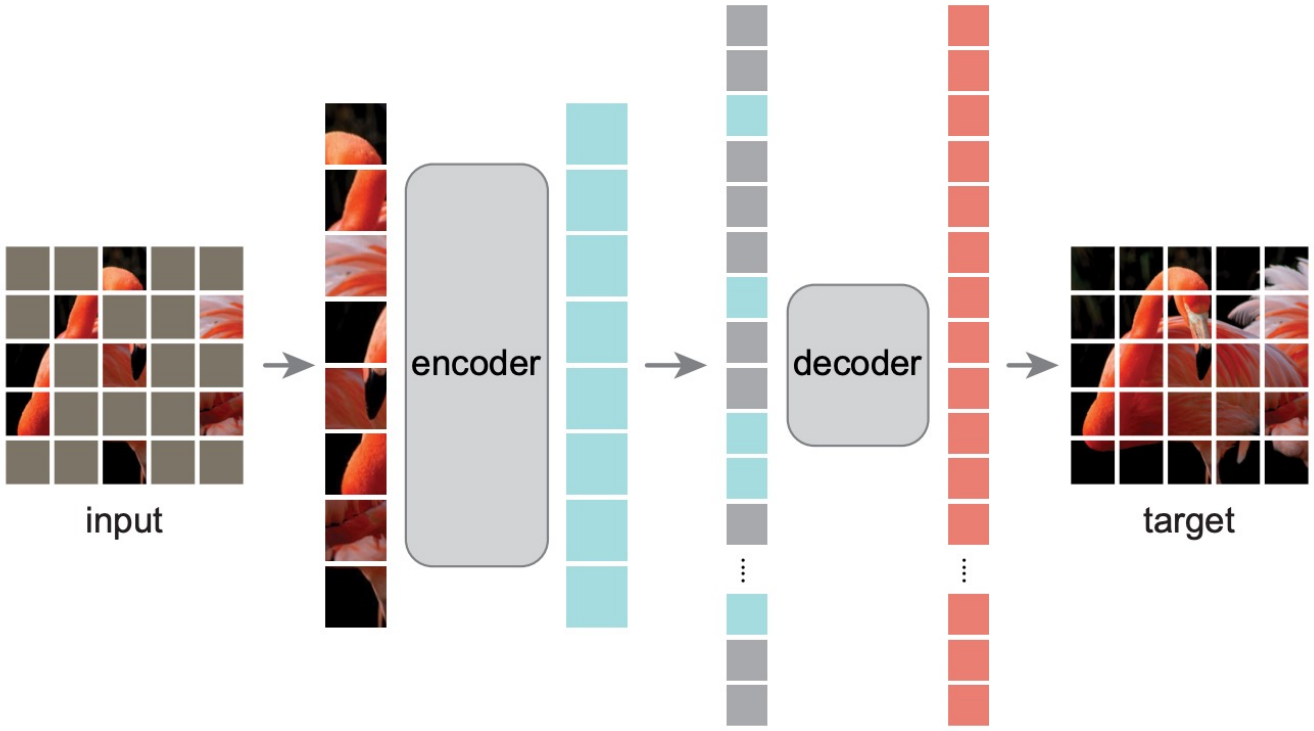
Sanghyun Woo[1], Shoubhik Debnath[2], Ronghang Hu[2],
Xinlei Chen[2], Zhuang Liu[2], In So Kweon[1], Saining Xie[3]

[1]KAIST, [2]Meta AI, [3]New York University

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

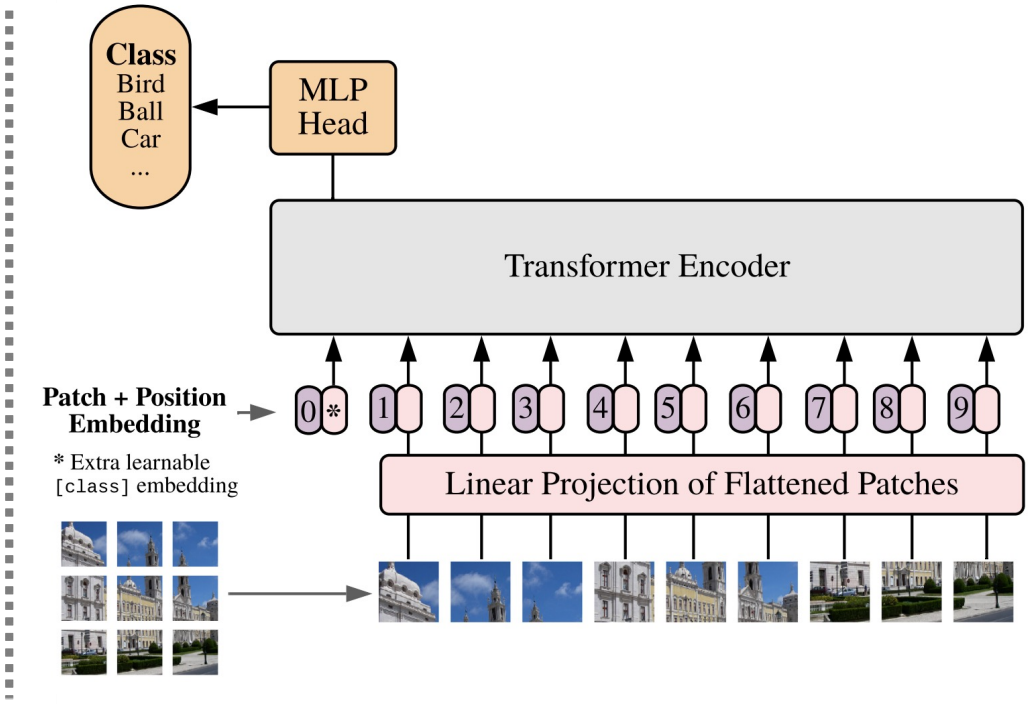
Independent design of objective or architecture

[Objective Design]
MAE: Masked Autoencoder



[MAE, He et.al. 2022]

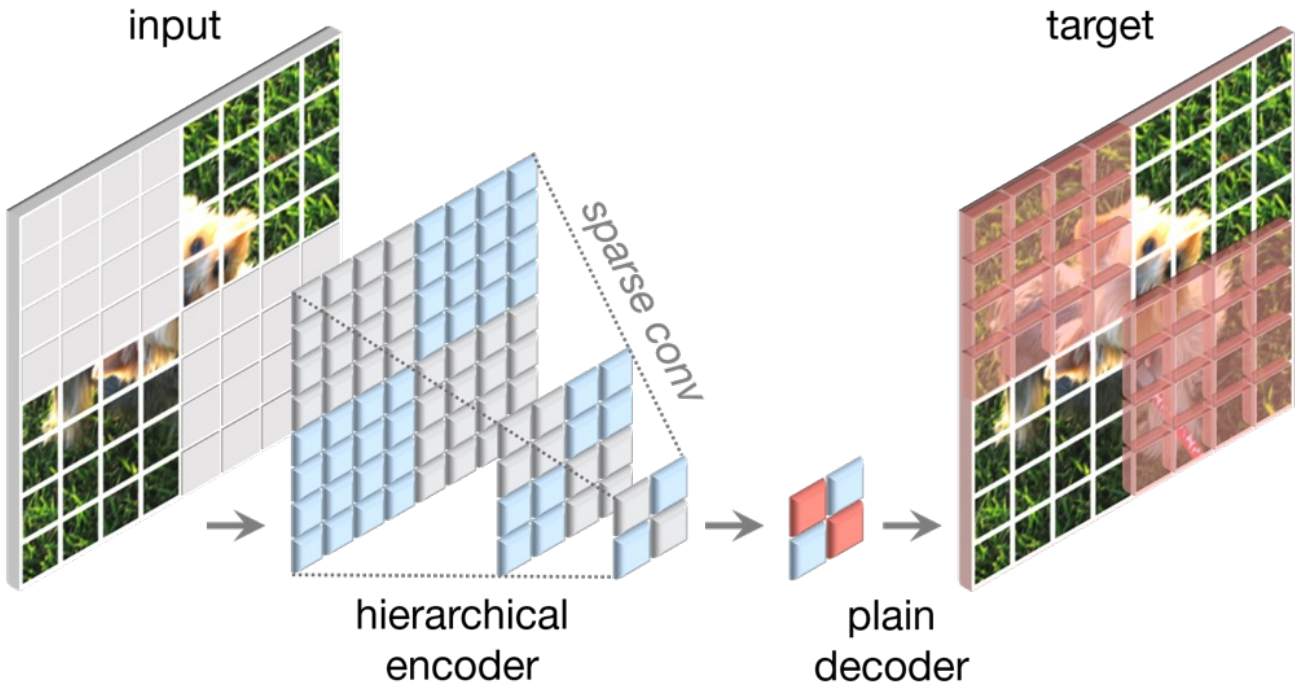
[Architecture Design]
ViT: Vision Transformer



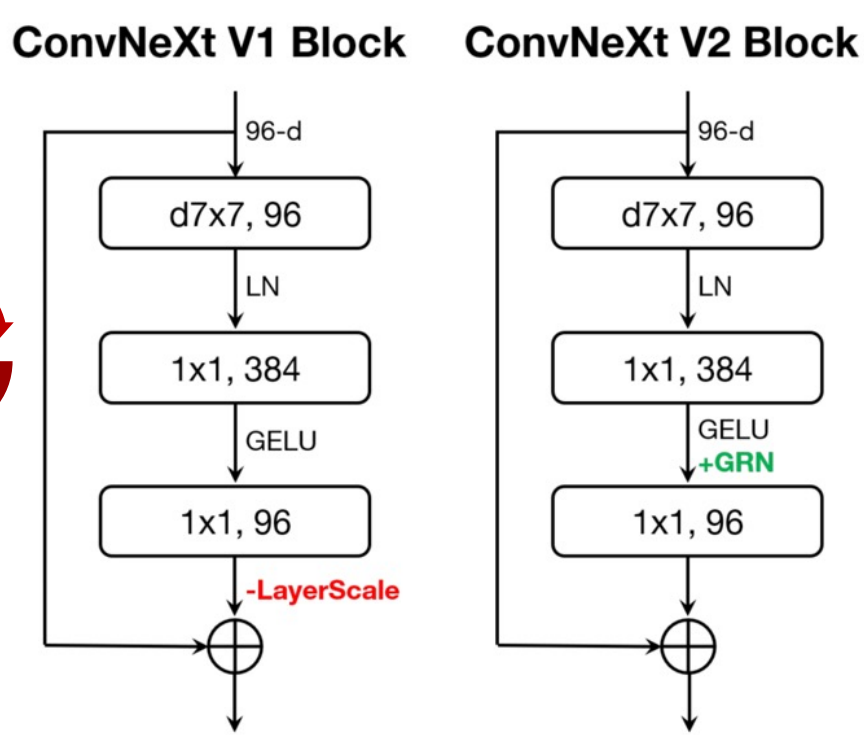
[ViT, Dosovitskiy et.al. 2021]

ConvNeXt V2, Co-Design Matters

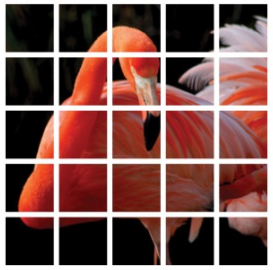
[Objective Design]
FCMAE: Fully Convolutional Masked Autoencoder



[Architecture Design]
GRN: Global Response Normalization

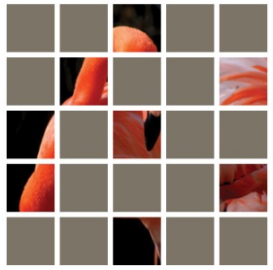


Masked Modeling in Vision



- Patch as a Word.

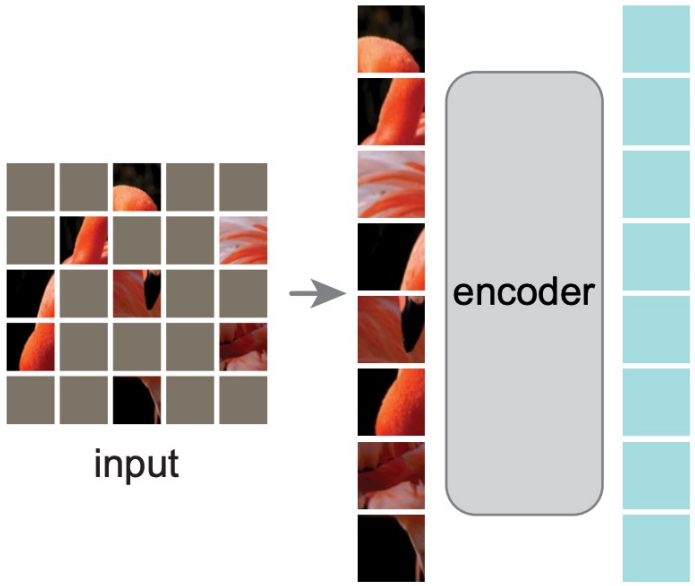
Masked Modeling in Vision



input

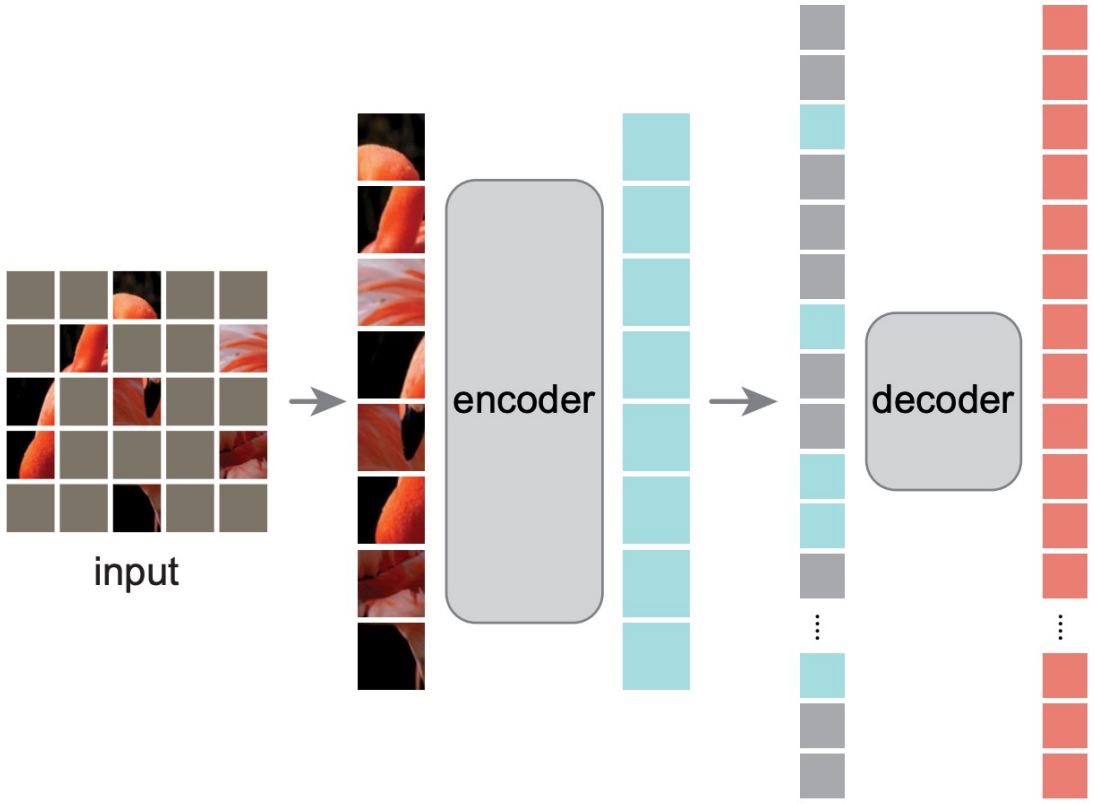
- Patch as a Word.
- Random patch dropout.

Masked Modeling in Vision



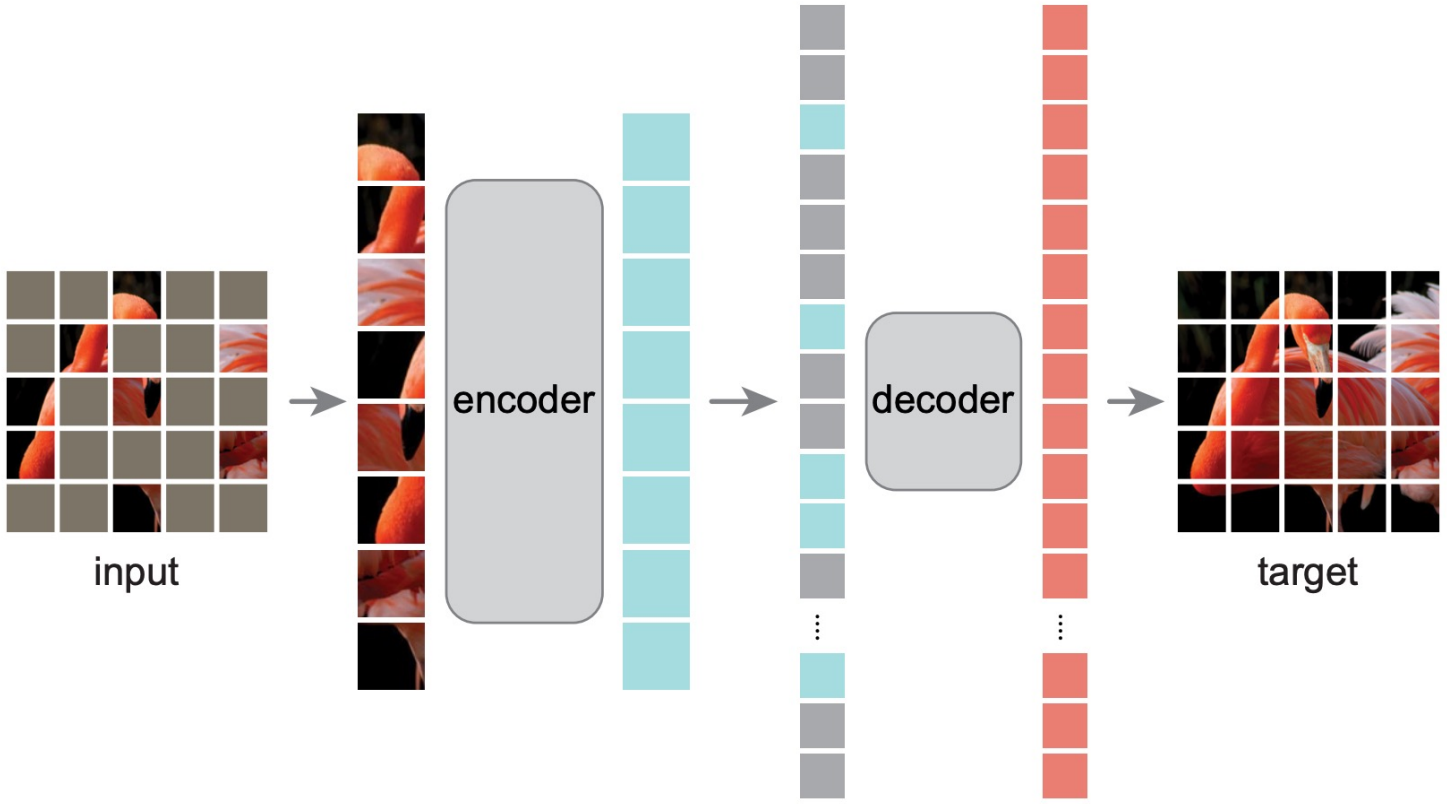
- Patch as a Word.
- Random patch dropout.
- Encoder only sees the visible patches,

Masked Modeling in Vision



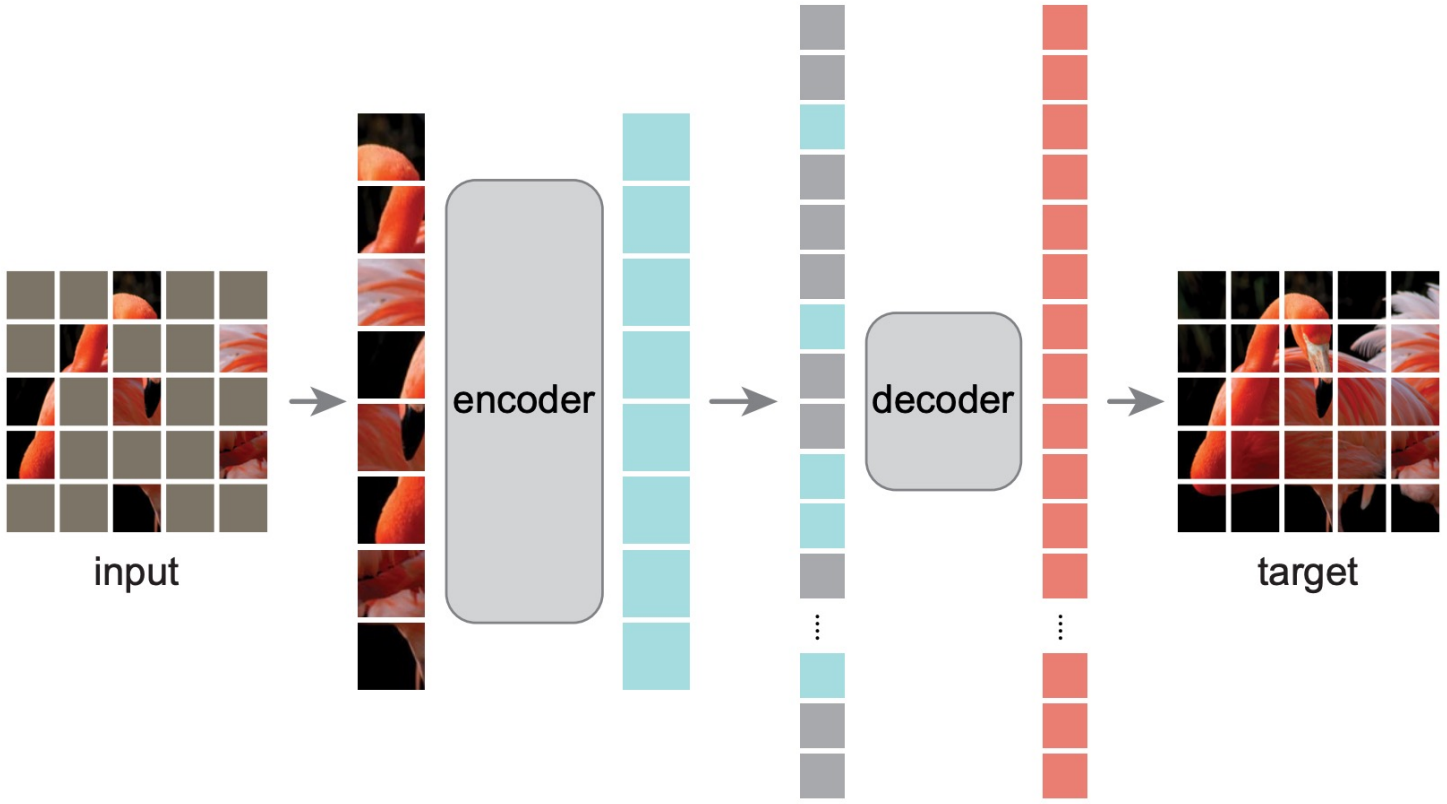
- Patch as a Word.
- Random patch dropout.
- Encoder only sees the visible patches, Decoder comprehends all.

Masked Modeling in Vision



- Patch as a Word.
- Random patch dropout.
- Encoder only sees the visible patches, Decoder comprehends all.
- Reconstructs masked patches.

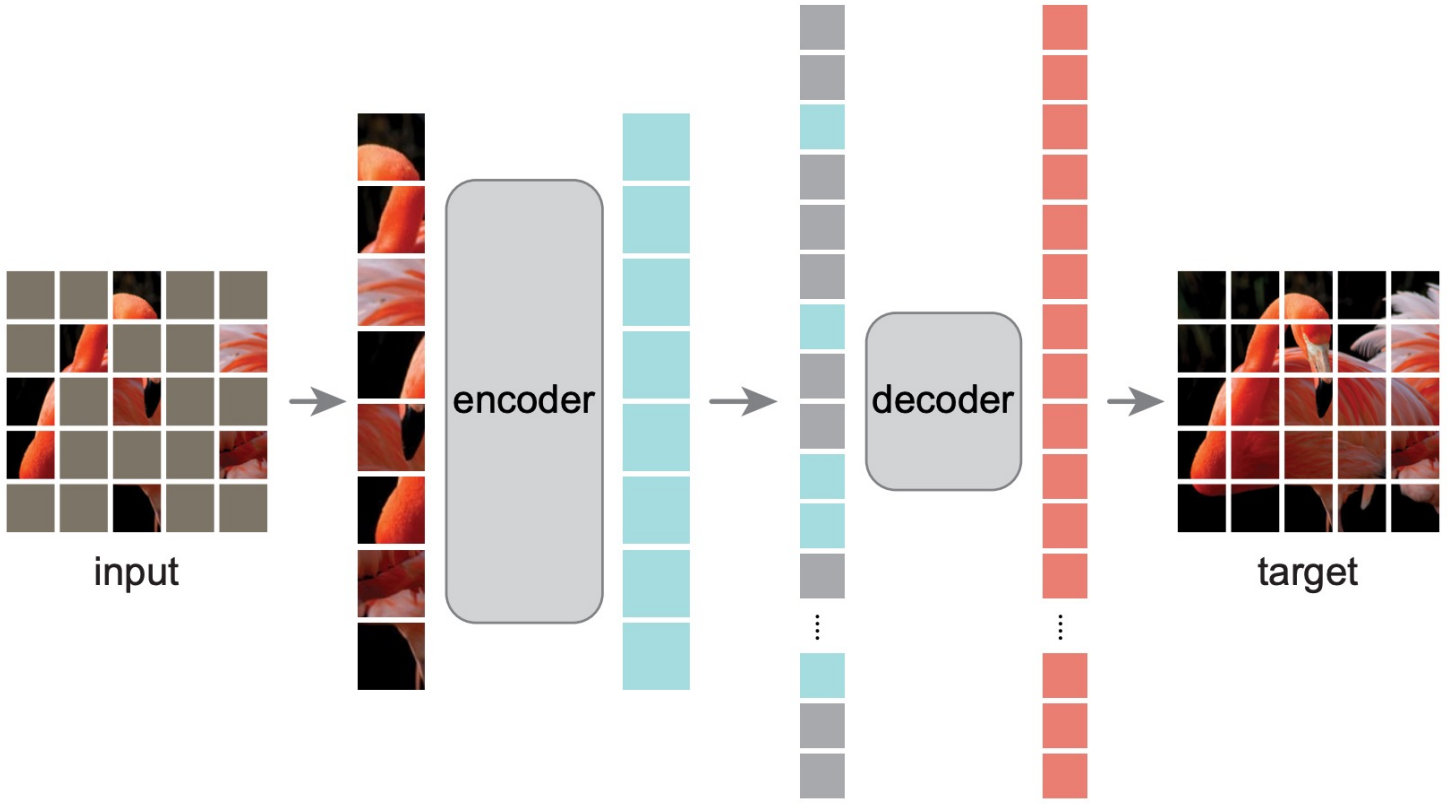
Masked Modeling for Vision Transformer



- Patch as a Word.
- Random patch dropout.
- Encoder only sees the visible patches, Decoder comprehends all.
- Reconstructs masked patches.

[MAE, He et.al. 2022]

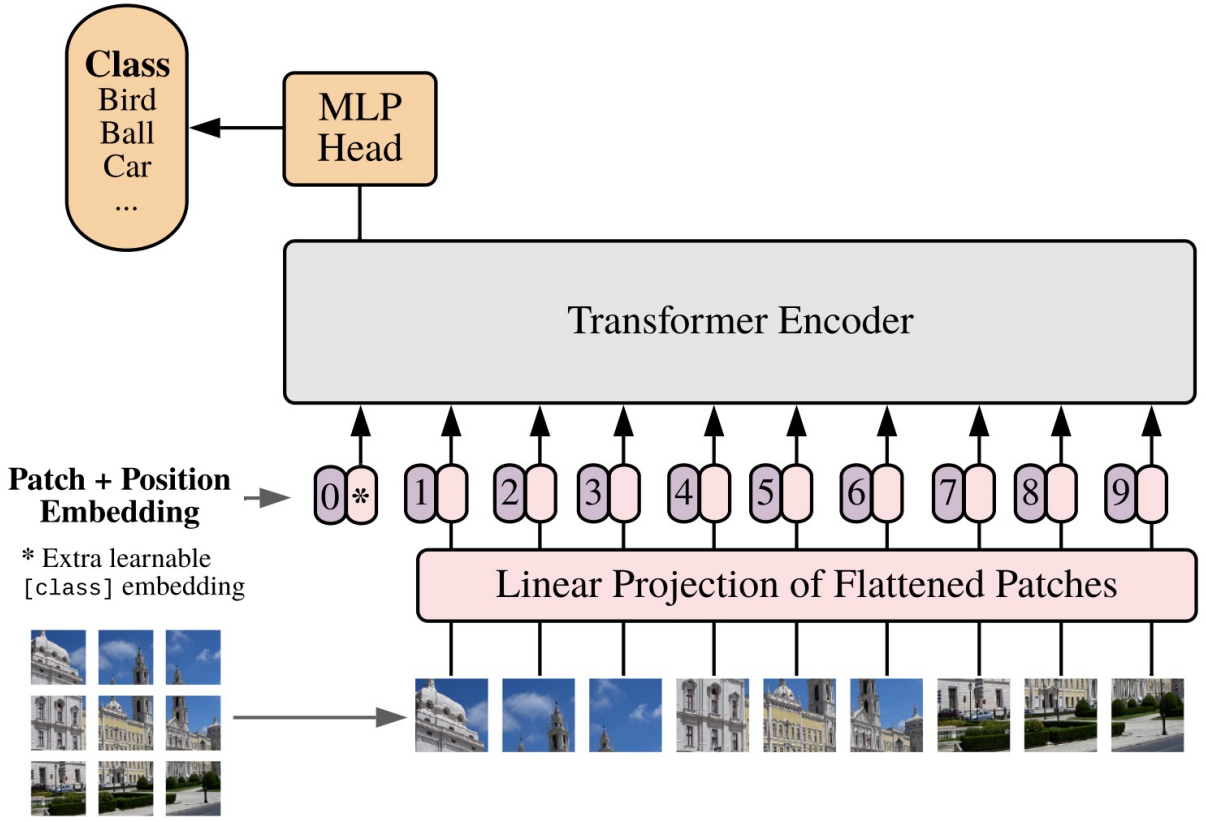
Masked Modeling for ConvNets?



- Patch as a Word.
- Random patch dropout.
- **Encoder only sees the visible patches, Decoder comprehends all.** 🤔
- Reconstructs masked patches.

Transformers: Image as sequence of tokens

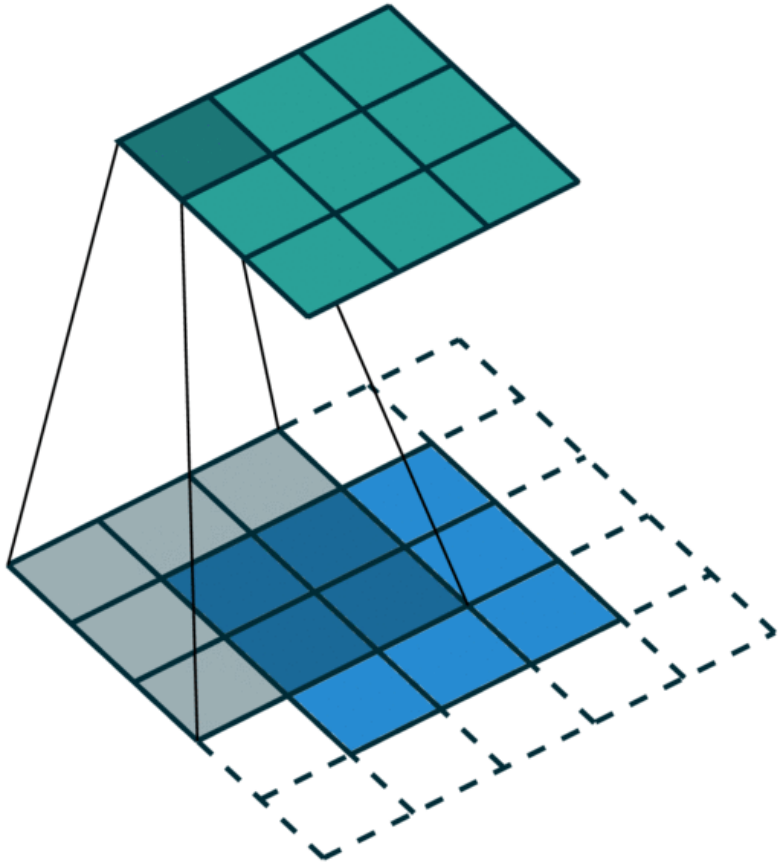
By design, Transformers can easily manage patch dropout.



[ViT, Dosovitskiy et.al. 2021]

ConvNets: Image as 2D grids of pixels

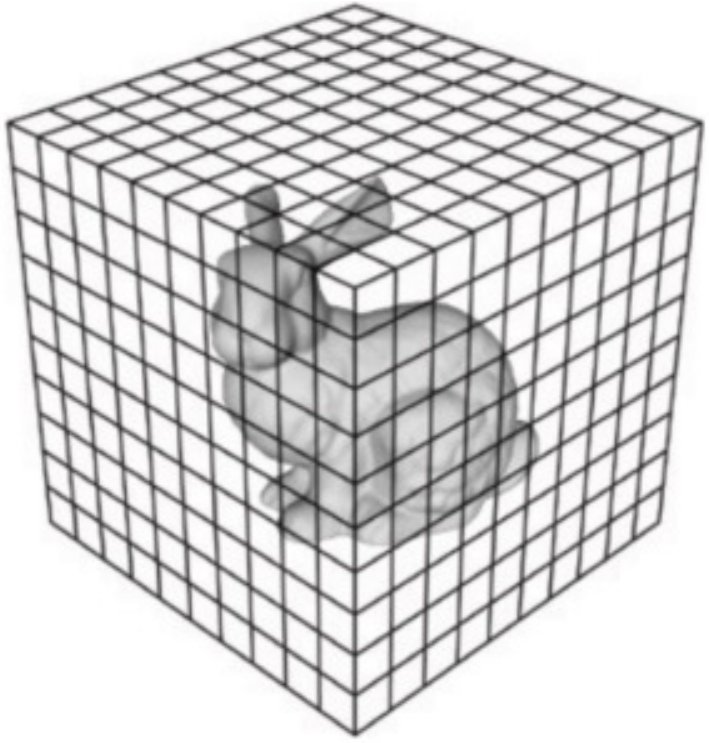
However, this poses a challenge for ConvNets, which use dense sliding windows.

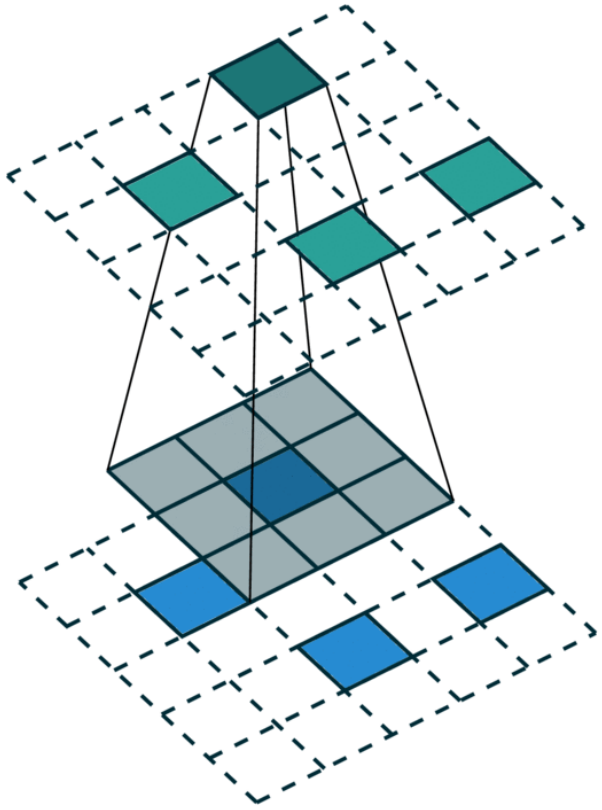


[MinkowskiEngine, Choy et.al. 2019]

Sparse Convolution for 3D point cloud processing

A special convolutional operation exists to efficiently handle sparse point clouds.

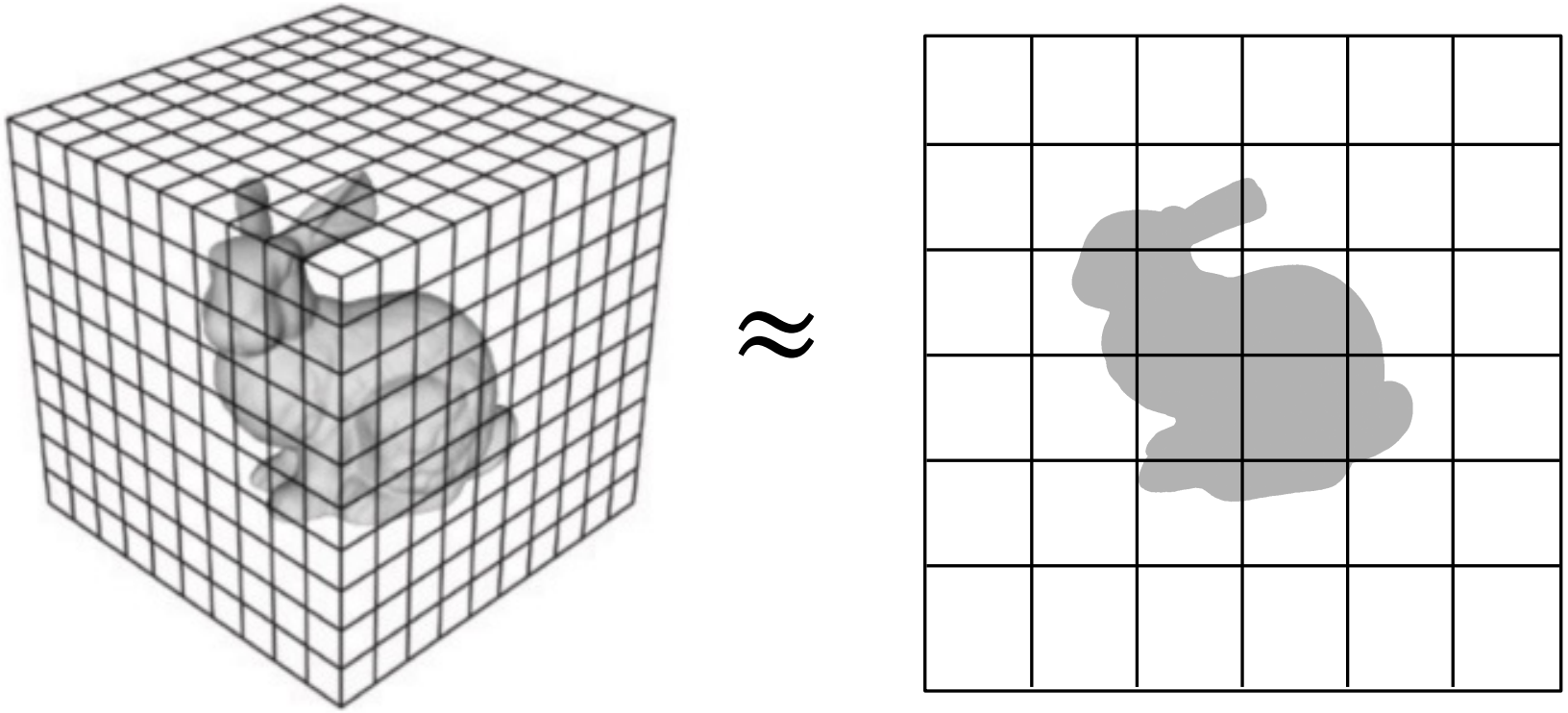




[MinkowskiEngine, Choy et.al. 2019]

From Sparse 3D point clouds to Masked 2D images

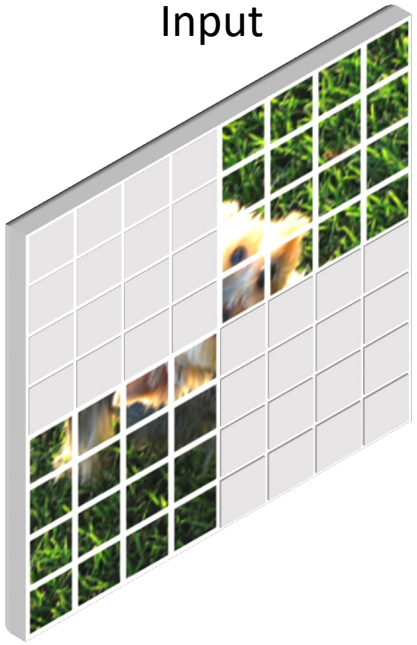
We view a masked images as sparse patch sets, like 3D point clouds
– a novel perspective enabling masked modeling for ConvNets



FCMAE: Fully Convolutional Masked Autoencoder

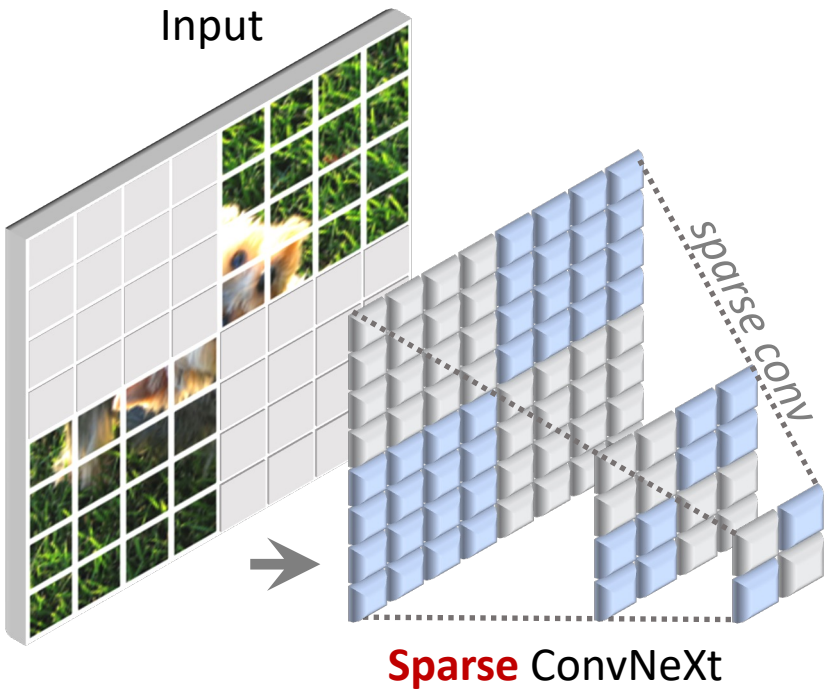
Random patch dropout

Input

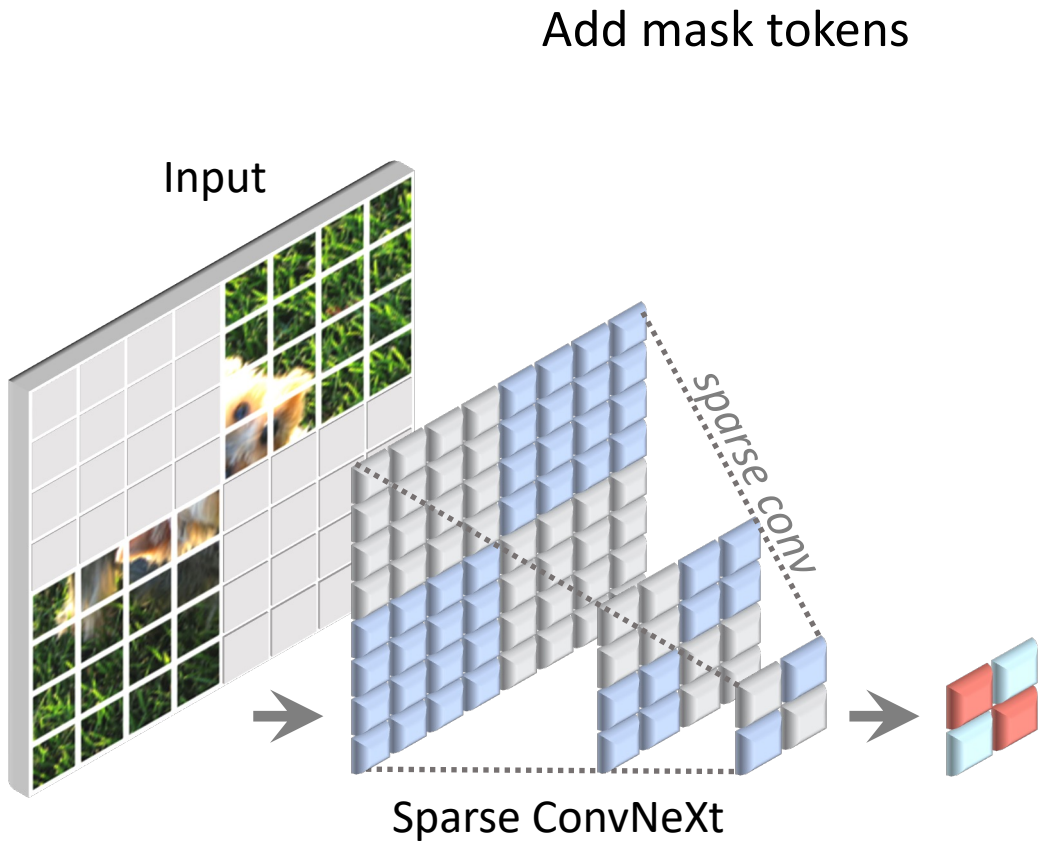


FCMAE: Fully Convolutional Masked Autoencoder

Encode visible patches using **sparse convolutions**

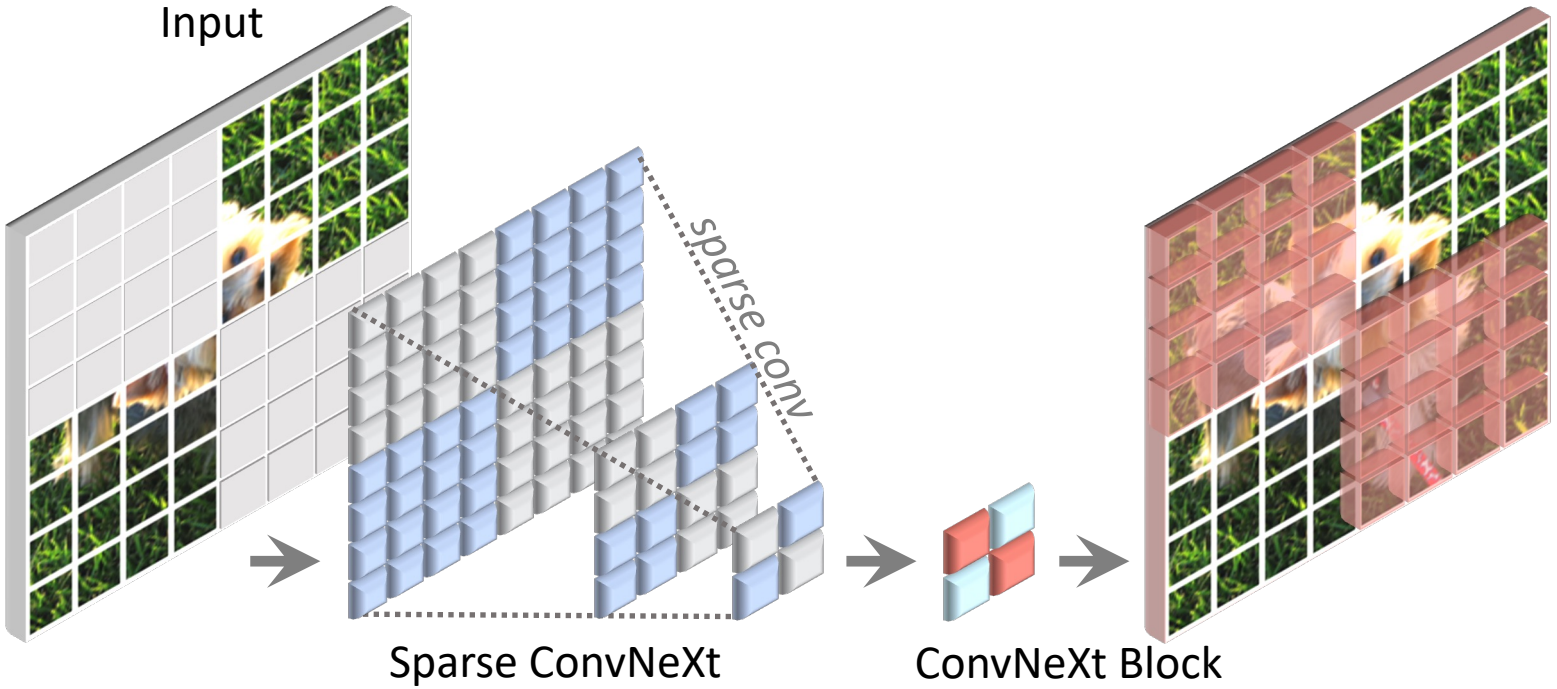


FCMAE: Fully Convolutional Masked Autoencoder



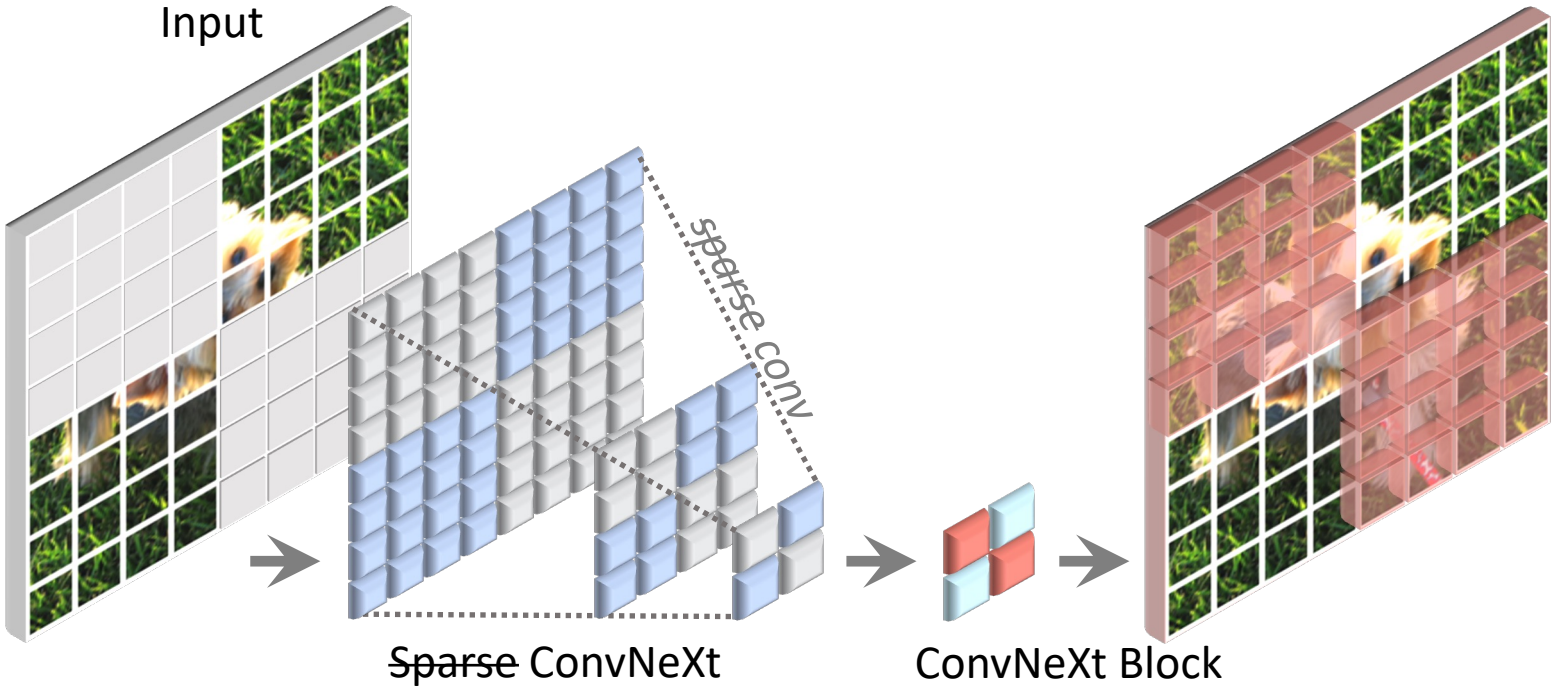
FCMAE: Fully Convolutional Masked Autoencoder

Reconstruct masked patches



FCMAE: Fully Convolutional Masked Autoencoder

After pre-training, the weights are converted back to standard layers without requiring special handling.



The impact of FCMAE

Train & Test Protocol

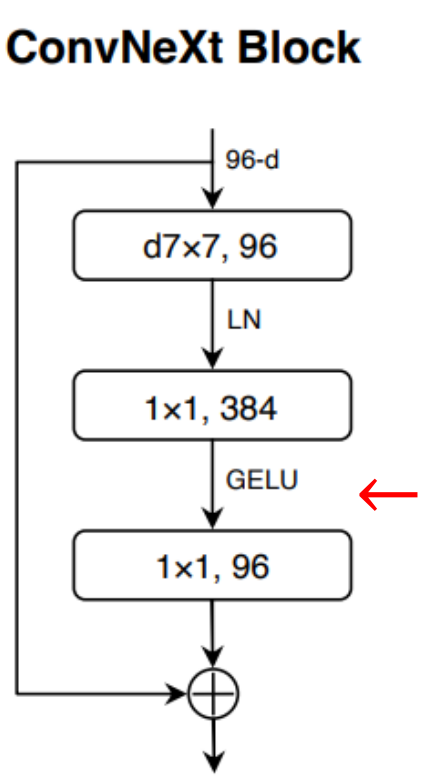
- 800ep Pre-training on ImageNet-1K *train*
- 100ep Fine-tuning on ImageNet-1K *train*
- Test on ImageNet-1K *val*

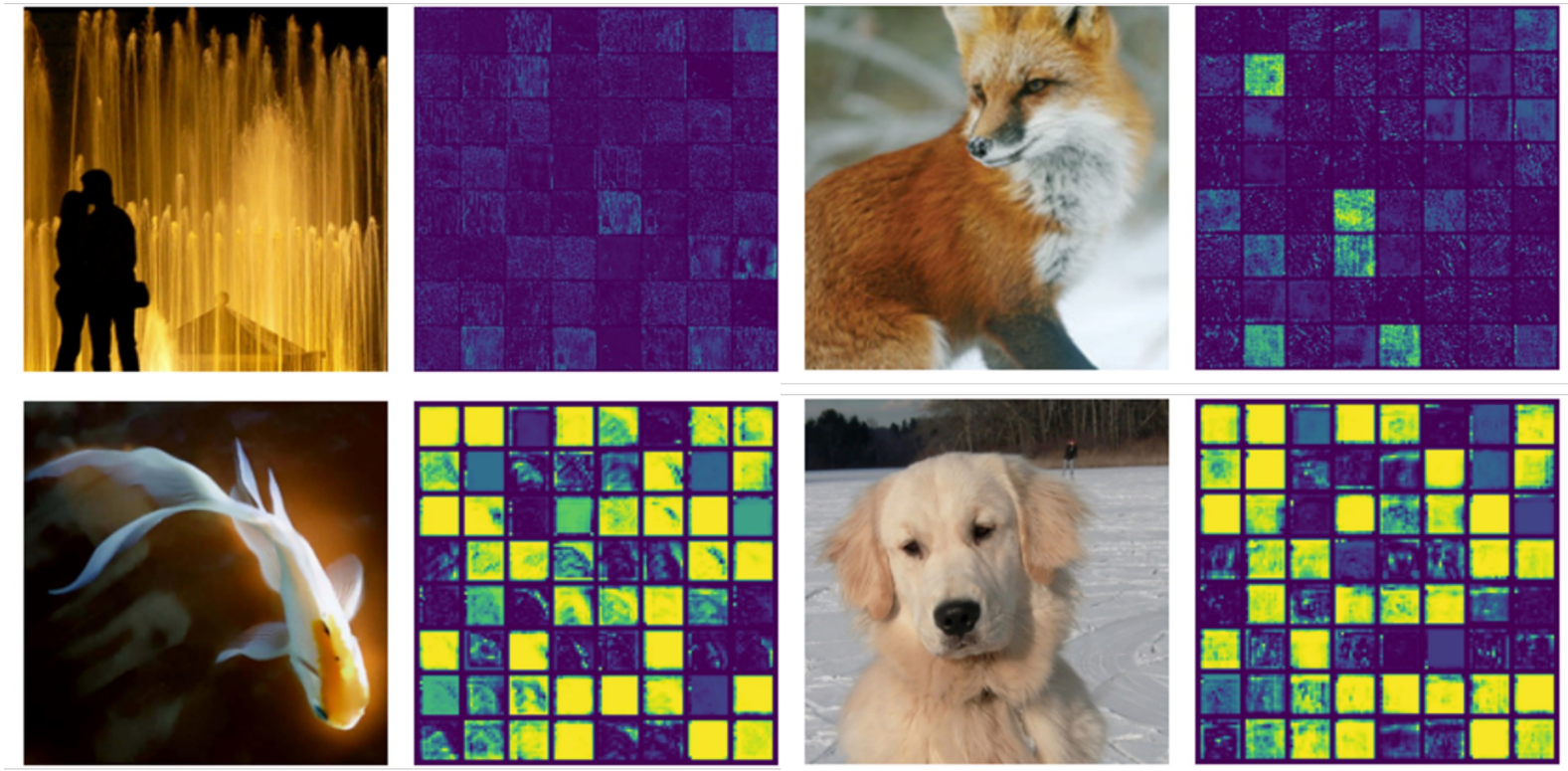
Sup, 100ep	Sup, 300ep. [33]	FCMAE
82.7	83.8	83.7

FCMAE pre-trained ConvNeXt surpasses random initialization but lags behind the original supervised setup.

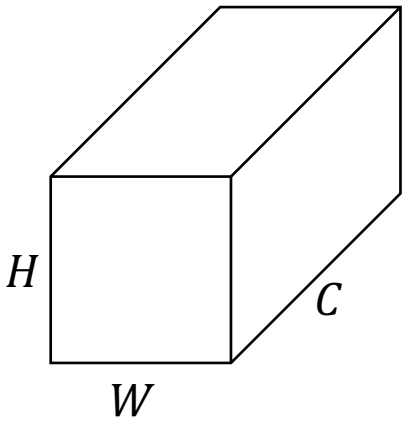
Feature Collapse with Dead or Saturated Neurons

Feature collapse detected in the dimension-expansion MLP layers in FCMAE pre-trained ConvNeXtV1 blocks (more analyses on appendix!).





GRN: Global Response Normalization

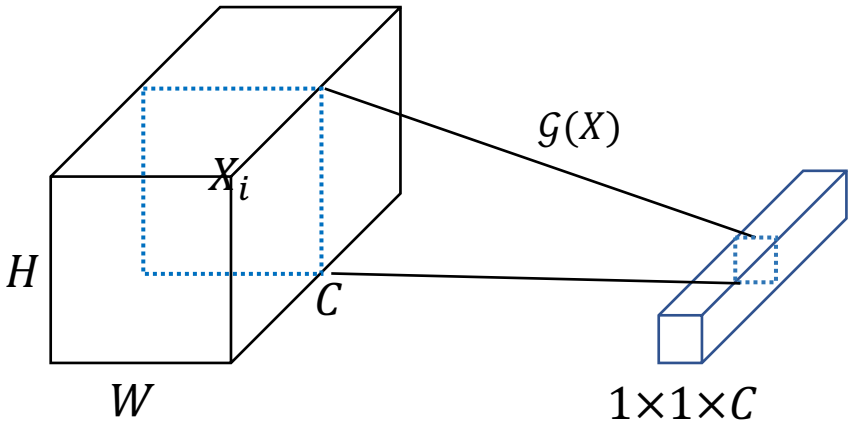


GRN: Global Response Normalization

1) Global feature aggregation

case	ft
g.avg.	83.7
L1	84.3
L2	**84.6**

(a) **Global aggregation** $G(\cdot)$. L2 Norm-based aggregation function produces the best result.

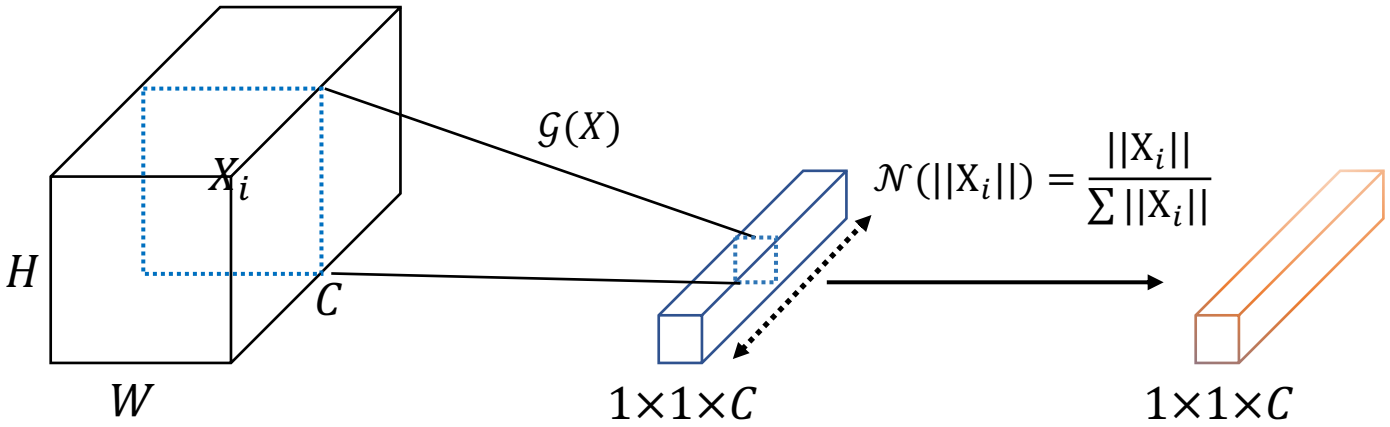


GRN: Global Response Normalization

1) Global feature aggregation
2) Feature normalization

case	ft
$(\|\|X_i\|\| - \mu)/\sigma$	84.5
$1/\sum \|\|X_i\|\|$	83.8
$\|\|X_i\|\|/\sum \|\|X_i\|\|$	**84.6**

(b) **Normalization operator,** $N(\cdot)$. Divisive normalization is an effective channel importance calibrator.



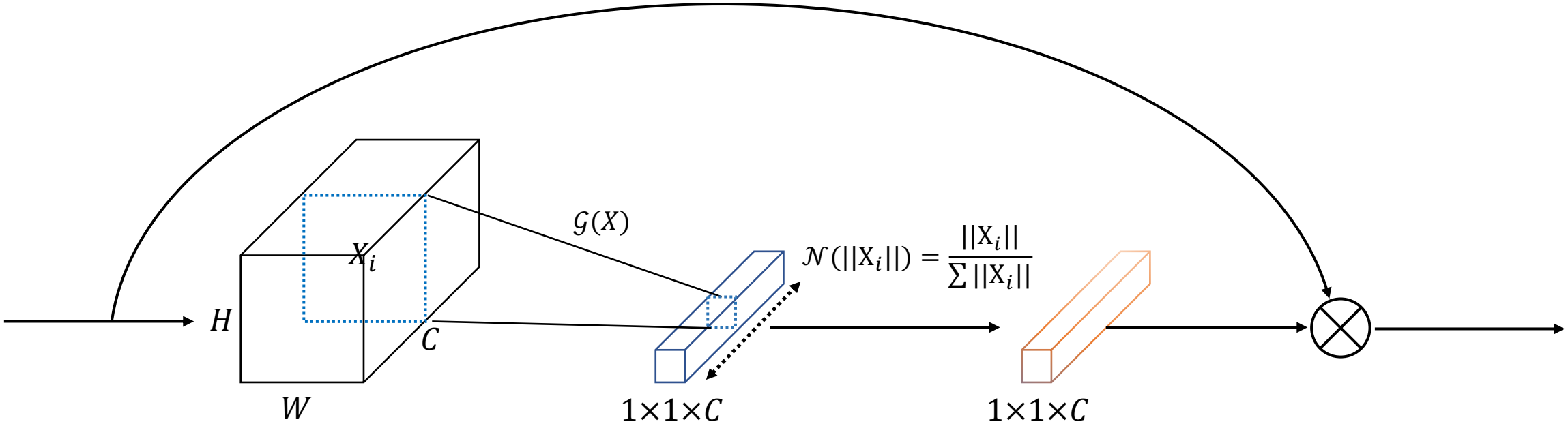
GRN: Global Response Normalization

1) Global feature aggregation
2) Feature normalization
3) Feature calibration

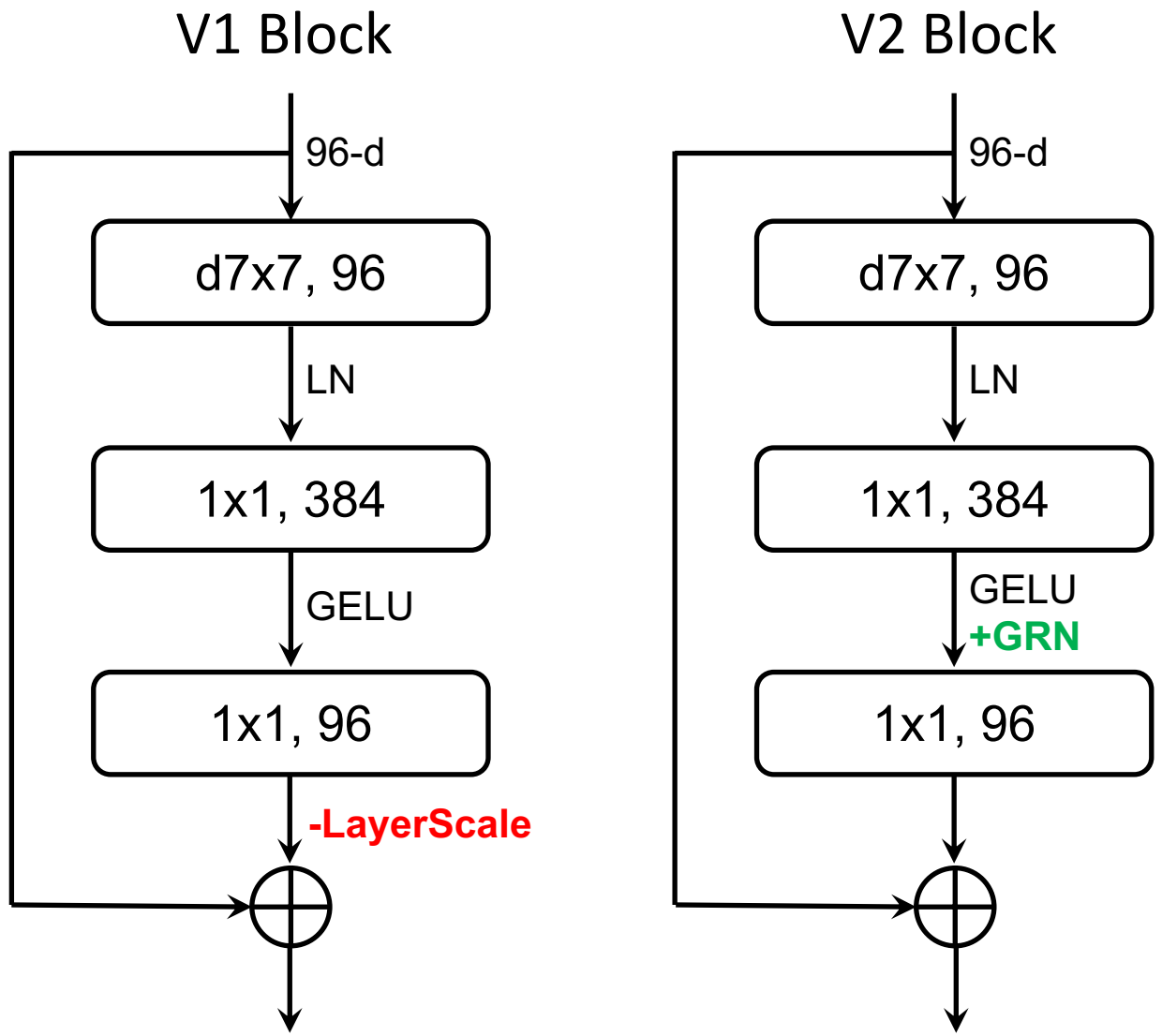
Two lines of code are all you need!

Algorithm 1 Pseudocode of GRN in a PyTorch-like style.

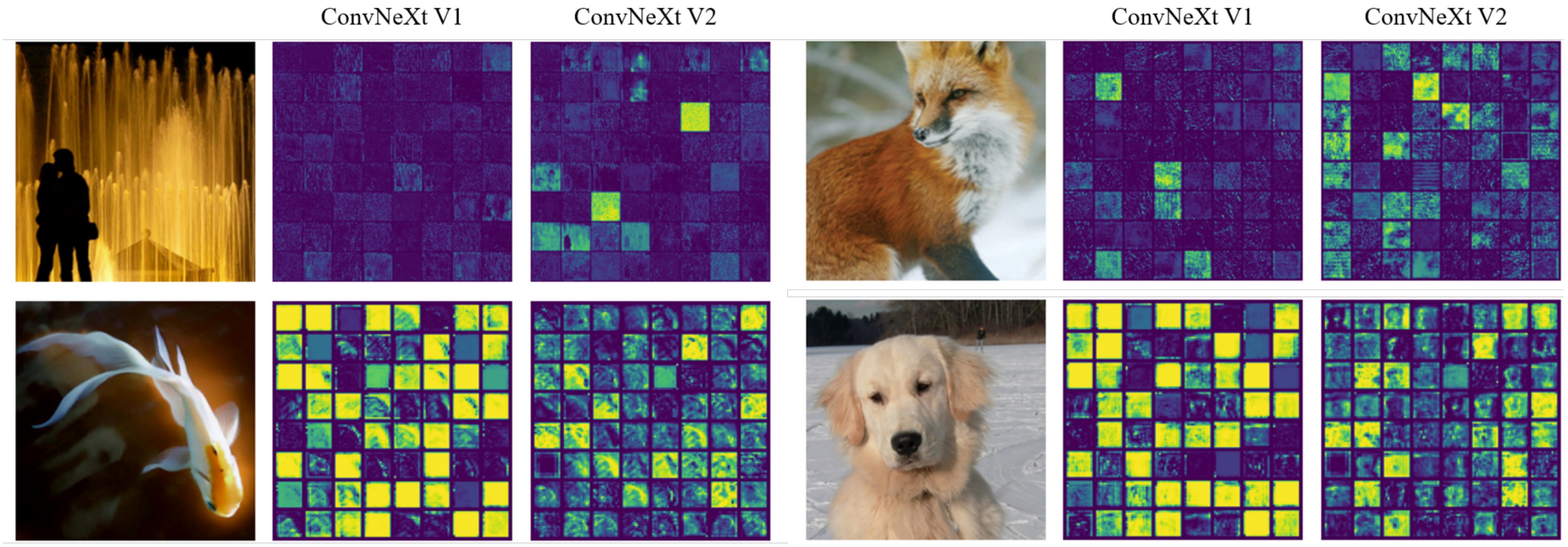
```
# gamma, beta: learnable affine transform parameters
# X: input of shape (N,H,W,C)

gx = torch.norm(X, p=2, dim=(1,2), keepdim=True)
nx = gx / (gx.mean(dim=-1, keepdim=True)+1e-6)
return gamma * (X * nx) + beta + X
```



ConvNeXt V2


V1 Block
96-d
d7x7, 96
LN
1x1, 384
GELU
1x1, 96
-LayerScale
V2 Block
96-d
d7x7, 96
LN
1x1, 384
GELU
+GRN
1x1, 96

Impact of GRN

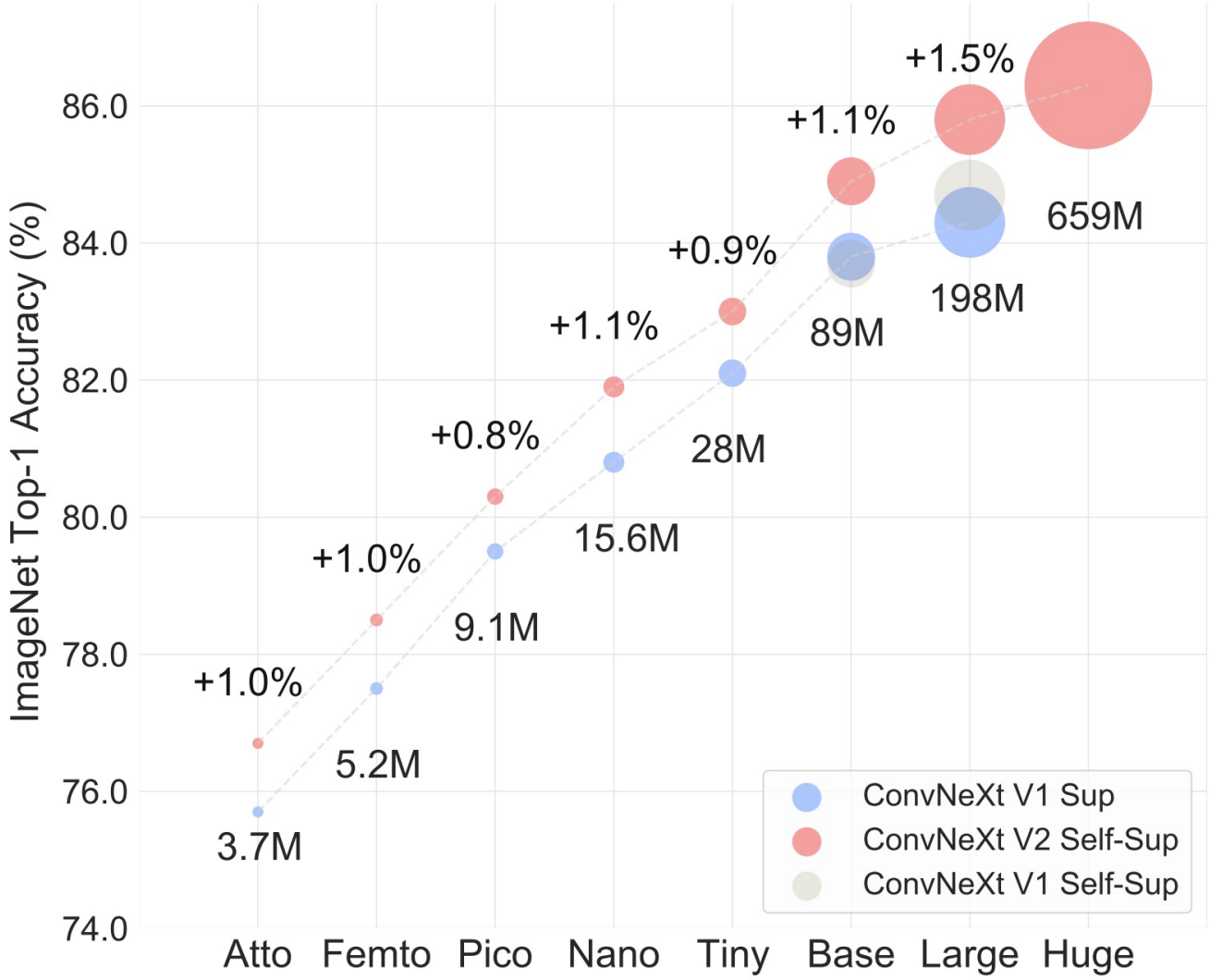


Co-design Matters

Backbone	Method	#param	FLOPs	Val acc.
ConvNeXt V1-B	Supervised	89M	15.4G	83.8
ConvNeXt V1-B	FCMAE	89M	15.4G	83.7
ConvNeXt V2-B	Supervised	89M	15.4G	84.3 (+0.5)
ConvNeXt V2-B	FCMAE	89M	15.4G	**84.6** (+**0.8**)
ConvNeXt V1-L	Supervised	198M	34.4G	84.3
ConvNeXt V1-L	FCMAE	198M	34.4G	84.4
ConvNeXt V2-L	Supervised	198M	34.4G	84.5 (+0.2)
ConvNeXt V2-L	FCMAE	198M	34.4G	**85.6** (+**1.3**)

When the architecture and the objective are *co-designed* and used together, masked modeling becomes effective for ConvNeXt.

Model Scaling



State-of-the-art Classification Accuracy

Type	Backbone	size	#param	FLOPS	Val acc.
Conv	Efficient V2-XL	480^2	208M	94.0G	87.3
	ConvNeXt V1-XL	384^2	350M	179.0G	87.8
Hybrid	CoAtNet-4	512^2	275M	360.9G	88.1
	MaxViT-XL	384^2	475M	293.7G	88.5
	MaxViT-XL	512^2	475M	535.2G	88.7
Trans	MViTV2-H	384^2	667M	388.5G	88.6
	MViTV2-H	512^2	667M	763.5G	88.8
Conv	ConvNeXt V2-H	384^2	659M	337.9G	88.7
	ConvNeXt V2-H	512^2	659M	600.7G	**88.9**

ImageNet-1K fine-tuning results using ImageNet-21K labels

Transfer Learning

Backbone	Method	FLOPS	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
ConvNeXt V1-B	Supervised	486G	50.3	71.6	56.1	44.9	68.5	48.8
ConvNeXt V2-B	Supervised	486G	51.0	72.4	56.6	45.6	69.5	49.7
Swin-B	SimMIM	497G	52.3	–	–	–	–	–
ConvNeXt V2-B	FCMAE	486G	**52.9**	72.6	58.9	**46.6**	70.0	51.1
ConvNeXt V1-L	Supervised	875G	50.6	71.5	56.3	45.1	68.7	49.2
ConvNeXt V2-L	Supervised	875G	51.5	72.5	57.3	45.8	69.4	49.9
Swin-L	SimMIM	904G	53.8	–	–	–	–	–
ConvNeXt V2-L	FCMAE	875G	**54.4**	73.9	60.4	**47.7**	71.4	52.3
Swin V2-H	SimMIM	–	54.4	–	–	–	–	–
ConvNeXt V2-H	FCMAE	2525G	**55.7**	75.2	61.8	**48.9**	72.8	53.6

COCO detection

Backbone	Method	input	mIoU	#param	FLOPS
ConvNeXt V1-B	Supervised	512^2	49.9	122M	1170G
ConvNeXt V2-B	Supervised	512^2	50.5	122M	1170G
Swin-B	SimMIM	512^2	**52.8**	121M	1181G
ConvNeXt V2-B	FCMAE	512^2	52.1	122M	1170G
ConvNeXt V1-L	Supervised	512^2	50.5	235M	1573G
ConvNeXt V2-L	Supervised	512^2	51.6	235M	1573G
Swin-L	SimMIM	512^2	53.5	234M	1601G
ConvNeXt V2-L	FCMAE	512^2	**53.7**	235M	1573G
Swin V2-H	SimMIM	512^2	54.2	–	–
ConvNeXt V2-H	FCMAE	512^2	**55.0**	707M	3272G
ConvNeXt V2-H	FCMAE, *22K ft*	640^2	**57.0**	707M	5113G

ADE20k segmentation

Conclusion

- ConvNeXt V2: *Co-design* of objective (FCMAE) and architecture (GRN)
- Broader complexity models and better performance
- Strong scalability

Poster Session: @WED-PM-360

https://github.com/facebookresearch/ConvNeXt-V2