Green Hierarchical Vision Transformer for Masked Image Modeling

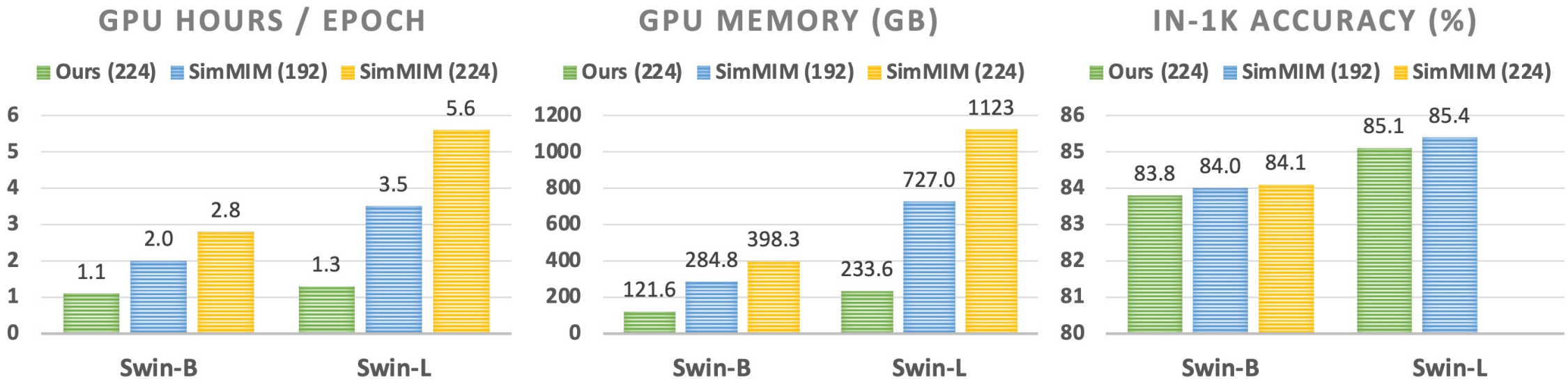
Lang Huang[1], Shan You[2], Mingkai Zheng[3],
Fei Wang[2], Chen Qian[2], Toshihiko Yamasaki[1]

[1]The University of Tokyo; [2]SenseTime Research; [3]The University of Sydney

GreenMIM

A green approach for Masked Image Modeling (MIM) with hierarchical Vision Transformers

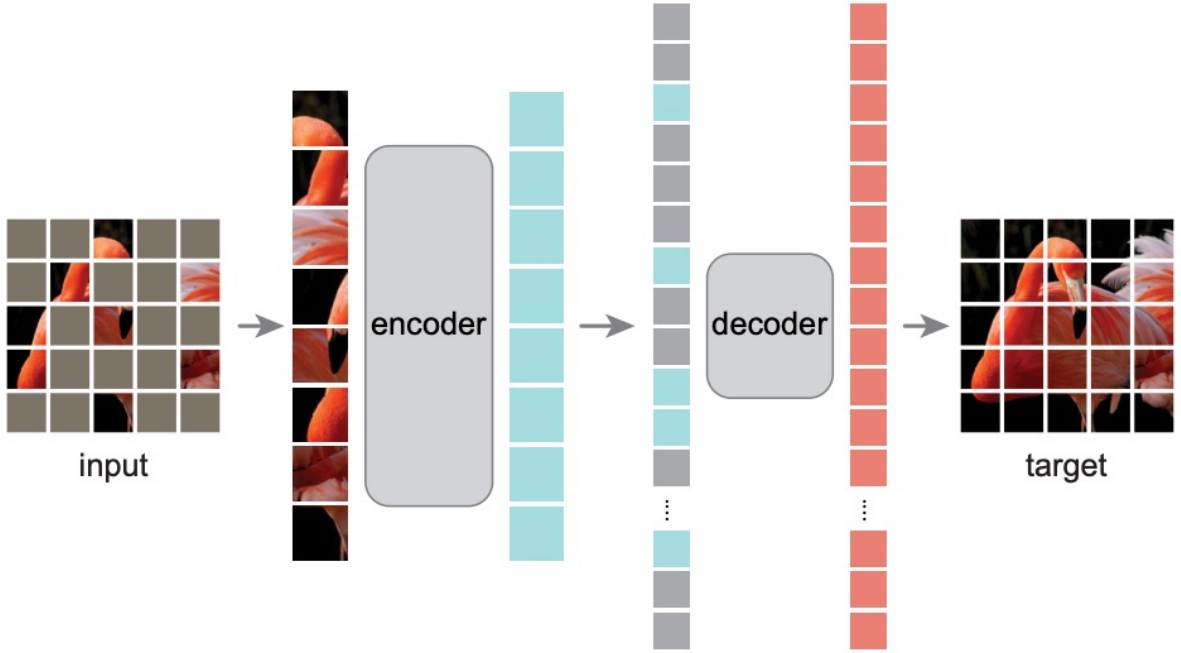
- Up to 2.7x speedup and 70% GPU memory reduction
- Competitive performance



Background

Masked Image Modeling.

- Predict masked patches from visible patches
- Representative work: Masked Autoencoders (MAE)

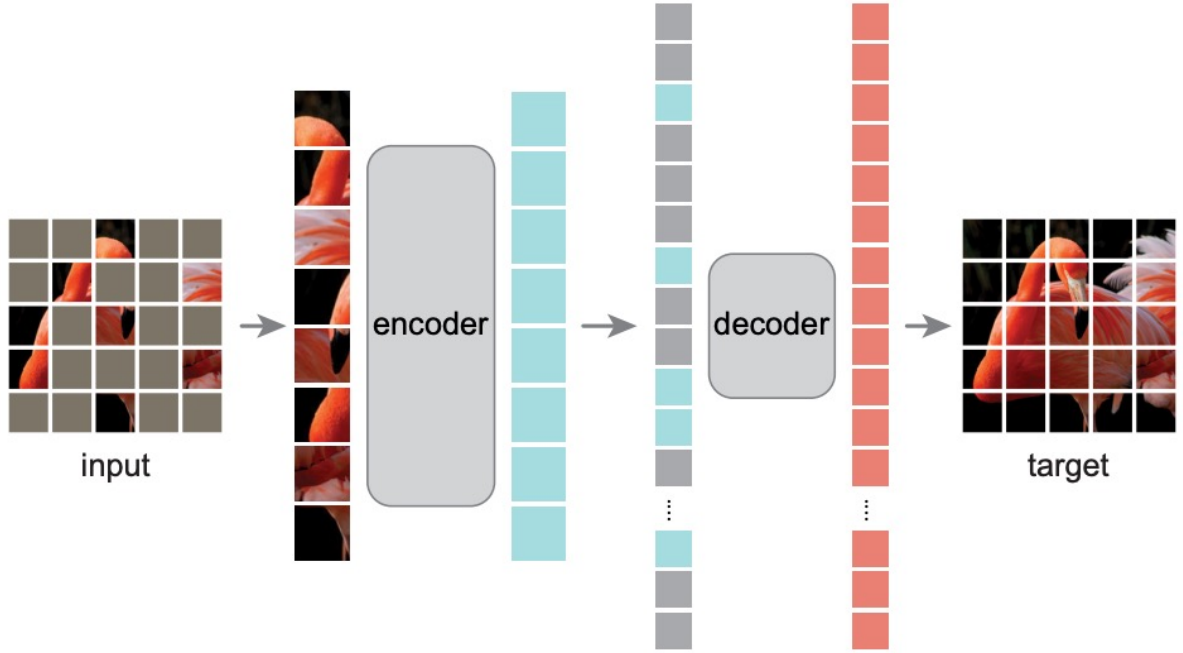


He et al., "Masked Autoencoders Are Scalable Vision Learners", *CVPR2022*.

Background

Masked Autoencoders

- Pro: Highly efficient because it discards masked patches
- Con: Only support isotropic Vision Transformers (ViTs)

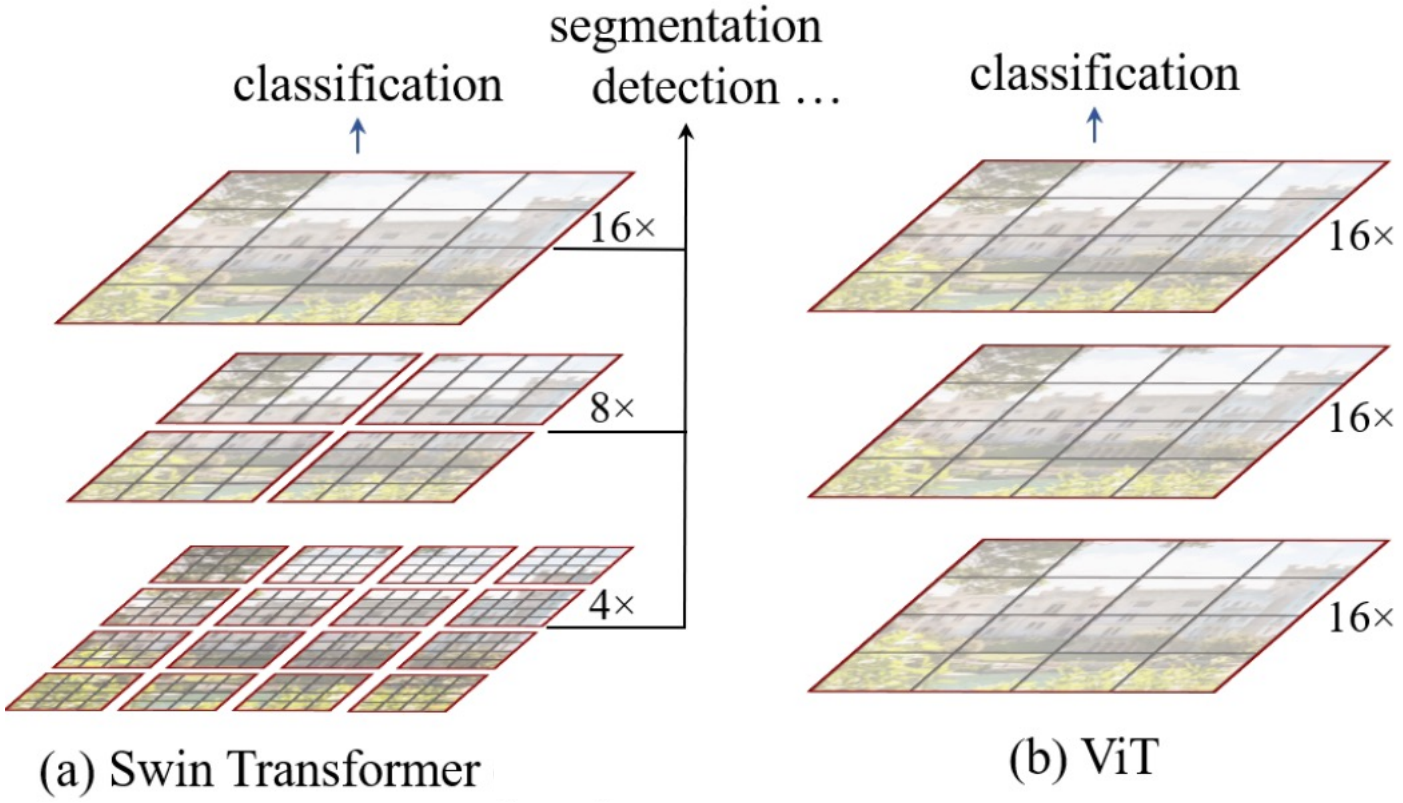


He et al., "Masked Autoencoders Are Scalable Vision Learners", *CVPR2022*.

Motivations

Hierarchical ViTs vs. Isotropic ViTs

- Hierarchical ones are more suitable for vision tasks



Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", *ICCV2021*.

Motivations

How to translate the efficiency of MAE to hierarchical ViTs?

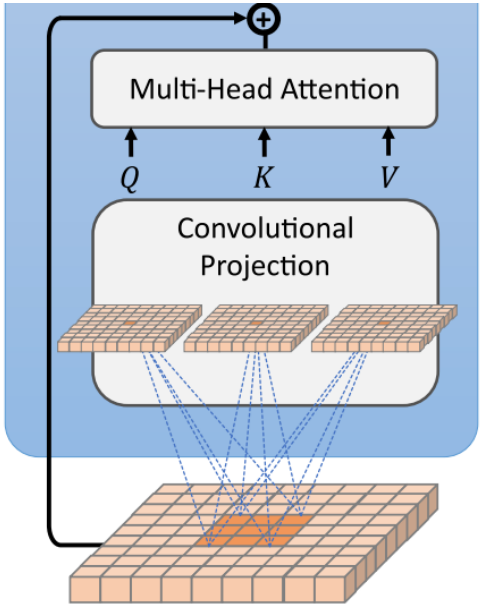
- Need to handle the local ops in hierarchical ViTs
  a) Non-overlapped window based ops, e.g., window attention
  b) Overlapped window base ops, e.g., convolution and pooling

a)

A local window to perform self-attention

A patch

b)



Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", *ICCV2021*.
Wu et al. "CvT: Introducing Convolutions to Vision Transformers", ICCV2021.

Approach

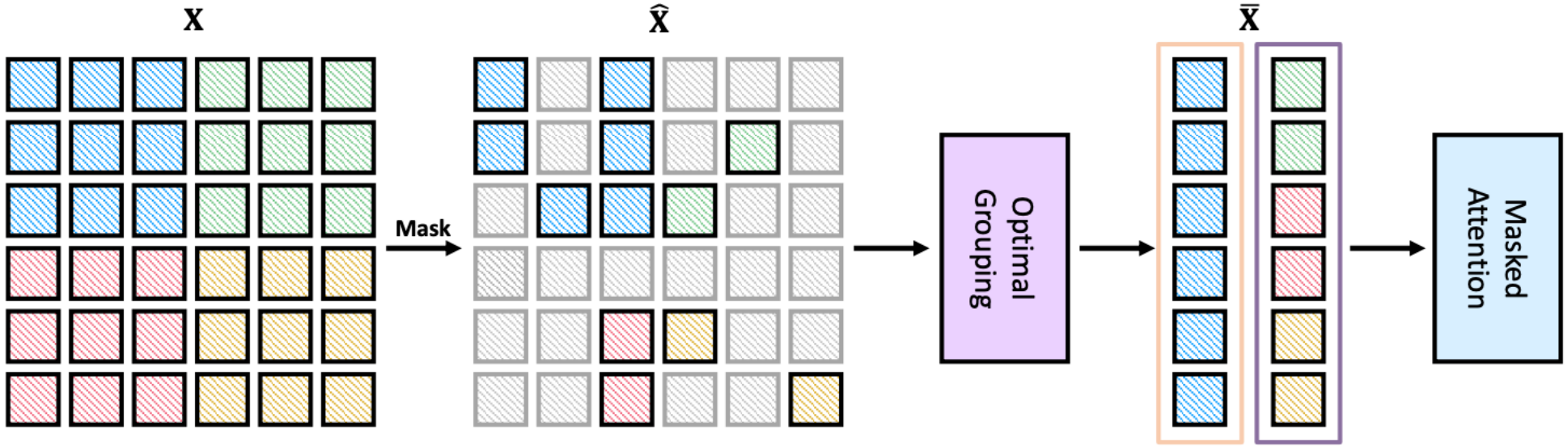
GreenMIM

- Group Window Attention with optimal grouping
- Incorporating Sparse Convolution

Approach: Group Window Attention

Divide-and-Conquer

- Partition the patches into groups of an equal size
- Perform masked attention within each group



Approach: Group Window Attention

Optimal Grouping

Algorithm 1 Optimal Grouping

Require: The number of visible patches within each local window $\{w_i\}_{i=0}^{n_w}$,

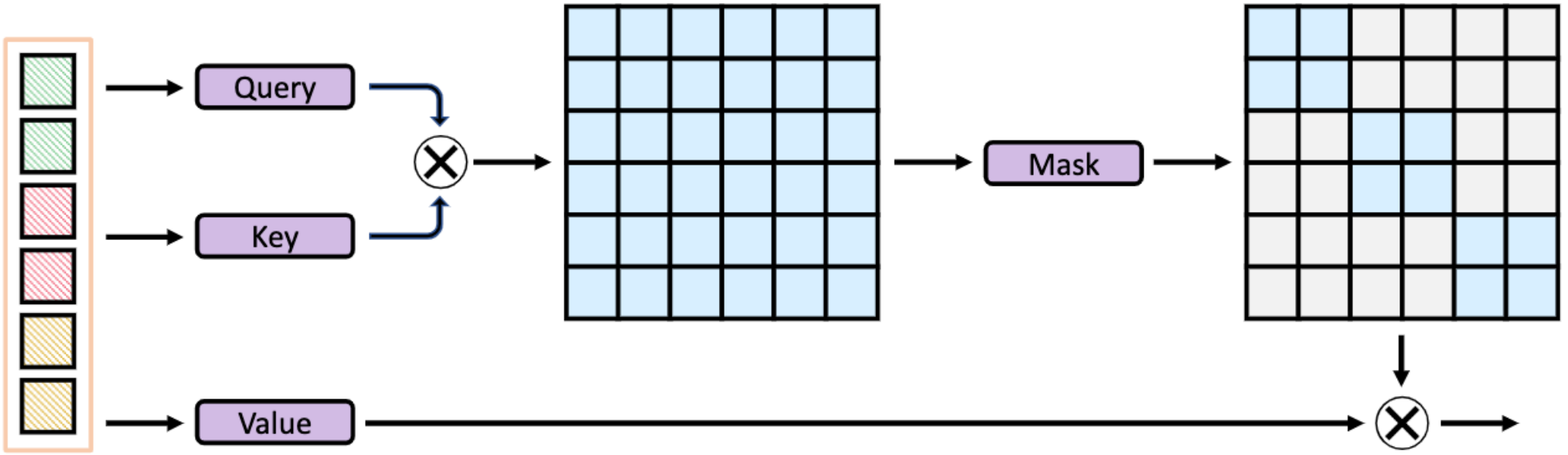
Minimum computational cost $c^* \leftarrow +\infty$
for $g_s = \max_i\{w_i\}_{i=1}^{n_w}$ **to** $\sum_{i=1}^{n_w} w_i$ **do**
  Remaining windows $\Phi \leftarrow \{w_i\}_{i=1}^{n_w}$; partition $\Pi \leftarrow \emptyset$; the number of group $n_g \leftarrow 0$
  repeat
    $\pi_{n_g} \leftarrow \text{Knapsack}(g_s, \Phi)$, as in Equation (7)
    $\Pi \leftarrow \Pi \cup \pi_{n_g}$; $\Phi \leftarrow \Phi \setminus \pi_{n_g}$
    $n_g \leftarrow n_g + 1$
  until $\Phi = \emptyset$
  $c \leftarrow \mathcal{C}(g_s, \Pi)$, as in Equation (8)
  if $c < c^*$ **then**
    $c^* \leftarrow c$; $\Pi^* \leftarrow \Pi$
  end if
end for
return Optimal group partition Π^*

Greedy selection of group size

Optimal grouping with a dynamic programing based knapsack problem solver

Approach: Group Window Attention

Masked Attention



Approach: Sparse Convolution

Sparse Convolution

- Activates only on the visible positions

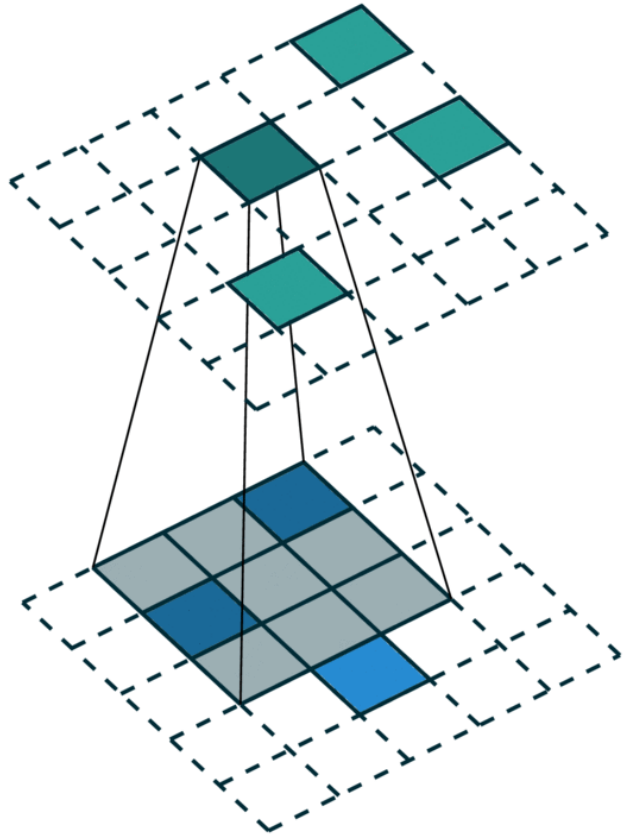


Image source: https://nvidia.github.io/MinkowskiEngine/sparse_tensor_network.html

Experiments

ImageNet Classification

Method	Model	#Params	PT Ep.	Ep. Hours	Total Hours	FT Ep.	Acc. (%)
Training from scratch							
Scratch, DeiT [62]	ViT-B	86M	0	-	-	300	81.8
Scratch, MAE [28]	ViT-B	86M	0	-	-	300	82.3
Scratch, Swin [49]	Swin-B	88M	0	-	-	300	83.5
Scratch, Twins [14]	Twins-L	99M	0	-	-	300	83.7
Supervised Pre-training							
Supervised, SimMIM [73]	Swin-B	88M	300	-	-	100	83.3
Supervised, SimMIM [73]	Swin-L	197M	300	-	-	100	83.5
Pre-training with Contrastive Learning							
MoCov3 [12]	ViT-B	86M	800	-	-	100	83.2
DINO [8]	ViT-B	86M	800	-	-	100	82.8
Pre-training with Masked Image Modeling							
BEiT [3]	ViT-B	86M	800	-	-	100	83.2
MaskFeat [68]	ViT-B	86M	800	-	-	100	84.0
MAE [28]	ViT-B	86M	1600	1.3	2069	100	83.6
SimMIM_{224} [73]	ViT-B	86M	800	4.1	3307	100	83.8
SimMIM_{192} [73]	Swin-B	88M	800	2.0	1609	100	84.0
SimMIM_{192} [73]	Swin-L	197M	800	3.5	2821	100	85.4
Ours	**Swin-B**	**88M**	**800**	**1.1**	**887**	**100**	**83.8**
Ours	**Twins-L**	**99M**	**800**	**0.8**	**676**	**100**	**83.9**
Ours	**Swin-L**	**197M**	**800**	**1.3**	**1067**	**100**	**85.1**

Experiments

MS-COCO Objection Detection

Method	Backbone	PT Ep.	PT Hours	FT Epochs	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Training from scratch										
Benchmarking [39]	ViT-B	0	0	400	48.9	-	-	43.6	-	-
Supervised Pretraining										
Benchmarking [39]	ViT-B	300	992	100	47.9	-	-	42.9	-	-
PVT [60]	PVT-L	300	-	36	44.5	66.0	48.3	40.7	63.4	43.7
Swin [43]	Swin-B	300	840	36	48.5	69.8	53.2	43.2	66.9	46.7
Self-Supervised Pre-training										
MoCov3 [11]	ViT-B	800	-	100	47.9	-	-	42.7	-	-
BEiT [2]	ViT-B	800	-	100	49.8	-	-	44.4	-	-
MAE [22]	ViT-B	1600	2069	25	48.1	-	-	-	-	-
MAE [22]	ViT-B	1600	2069	100	50.3	-	-	44.9	-	-
SimMIM [66]	Swin-B	800	1609	36	50.4	70.9	55.5	44.4	68.2	47.9
Ours	**Swin-B**	**800**	**887**	**36**	**50.0**	**70.7**	**55.4**	**44.1**	**67.9**	**47.5**

Thanks!

GreenMIM-arXiv

GreenMIM-Github