Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture

Mahmoud Assran[1,2,3*] **Quentin Duval**[1] **Ishan Misra**[1] **Piotr Bojanowski**[1]
Pascal Vincent[1] **Michael Rabbat**[1,3] **Yann LeCun**[1,4] **Nicolas Ballas**[1]

[1]Meta AI (FAIR) [2]McGill University [3] Mila, Quebec AI Institute [4]New York University

I-JEPA

Interpreted by: John Tan Chong Min

Do you need to predict everything?

- Some things in input space are not important to understand for your goals

Transformers: Representation via Prediction

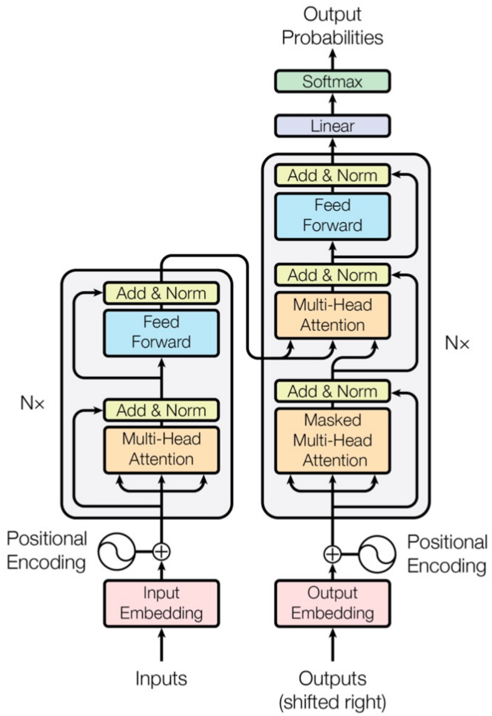


Figure 1: The Transformer - model architecture.

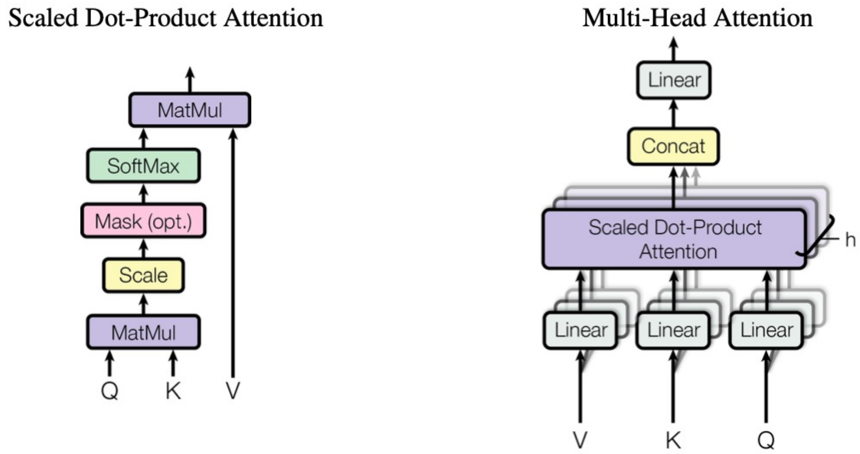
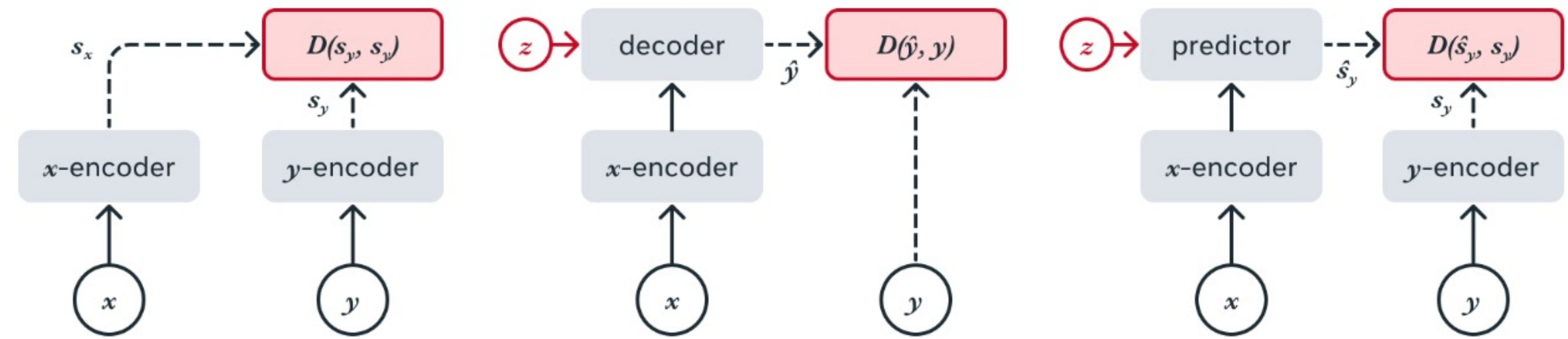


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Taken from: Attention is all you need. Vaswani et al. 2017

Prediction in Latent Space is powerful



(a) Joint-embedding architecture

(b) Generative architecture

(c) Joint-embedding predictive architecture

e.g. Contrastive Language-Image Pre-training (CLIP)

Audio and text embeddings

e.g. Generation of images in pixel space with masked patches

Generation of next token in Transformers

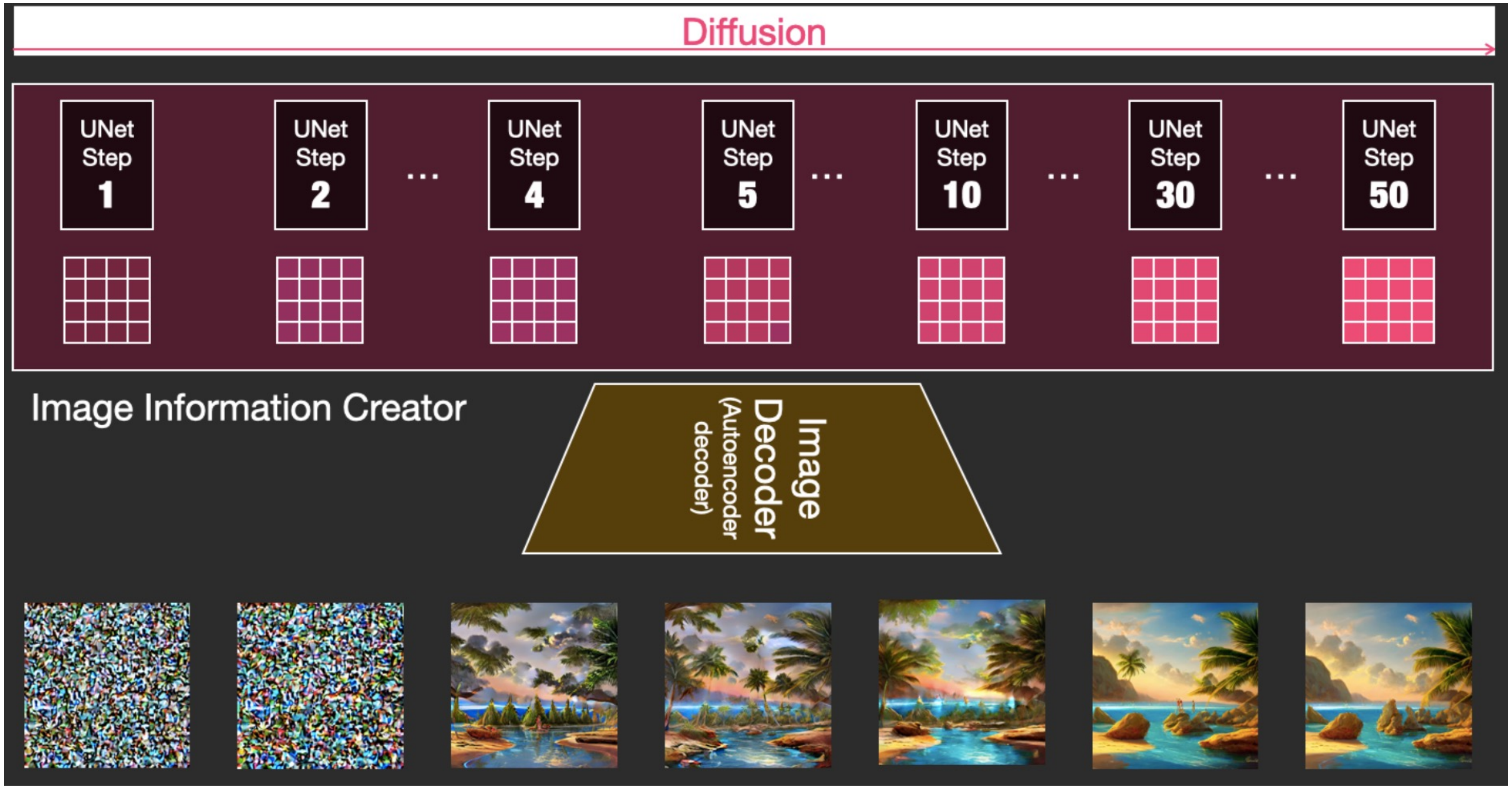
e.g. Predicting patches in latent space

Downstream:
Classification Tasks
Decision making with agents

Preliminaries

Latent Space

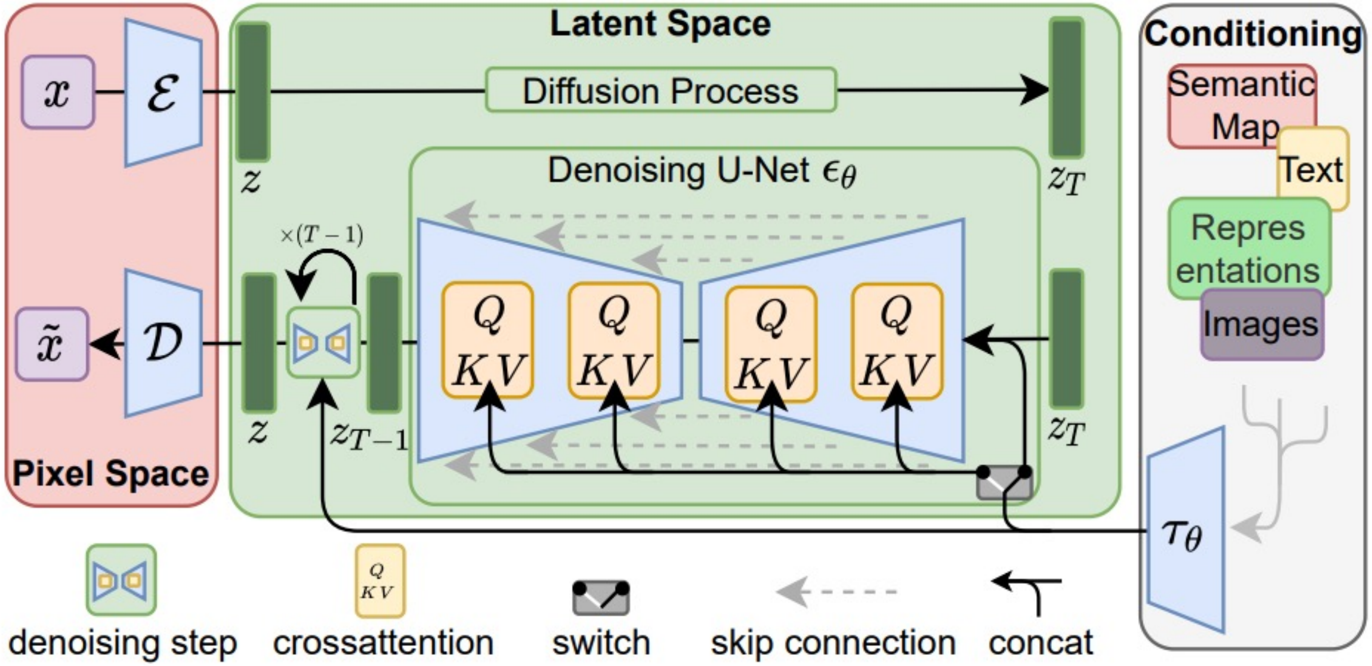
Stable Diffusion: Noise Removal in Latent Space!



https://jalammar.github.io/illustrated-stable-diffusion/

Stable Diffusion: Noise Removal in Latent Space!

- Text gets mapped to same latent space as image
- Image is recursively refined in latent space by removing noise based on text prompt



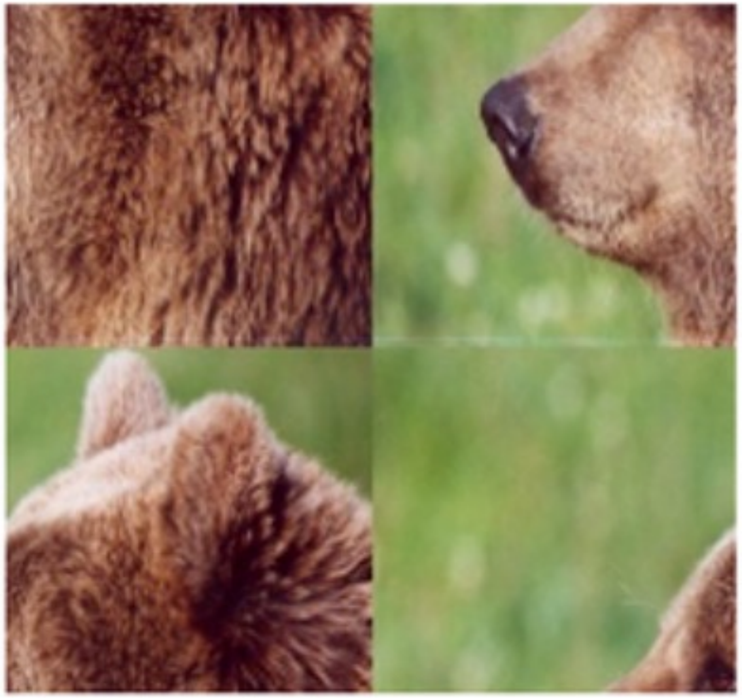
High-Resolution Image Synthesis with Latent Diffusion Models. Rombach et al. 2022.

Vision Models

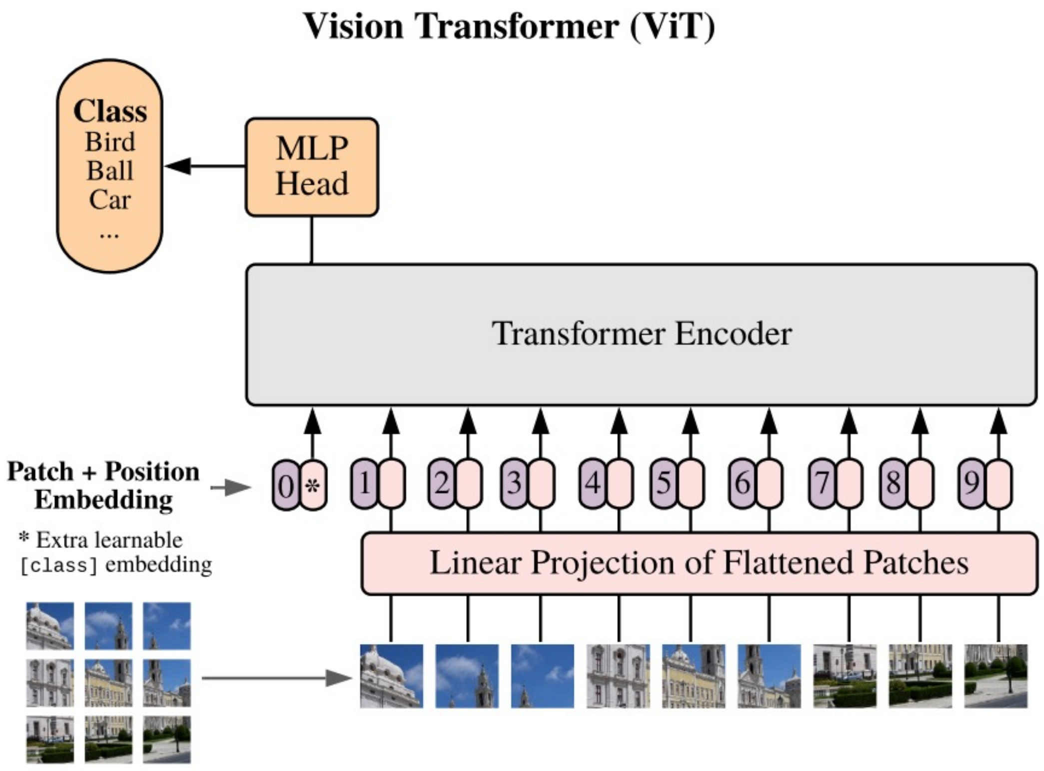
Vision Transformers (ViT)

Swin Transformers

Do these look the same to you?



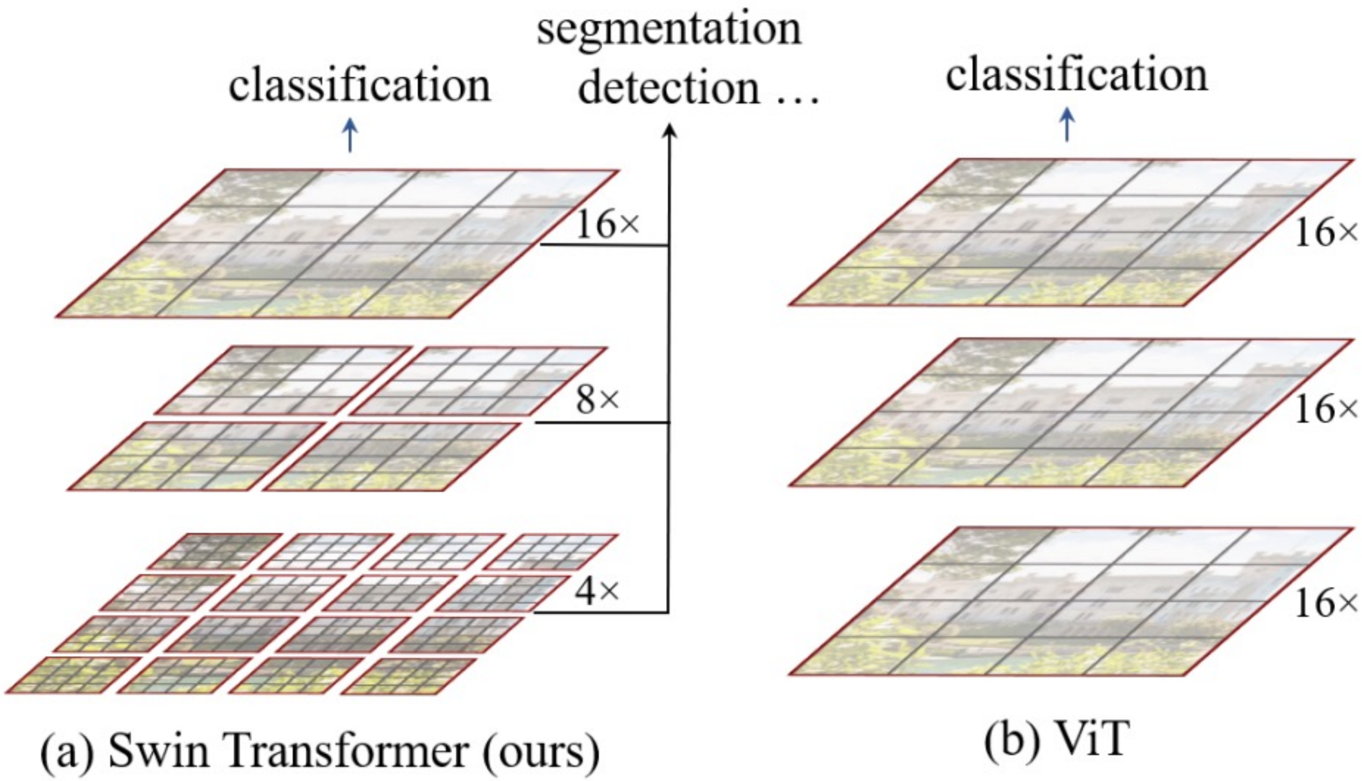
Vision Transformers



- Loosely inspired by Transformers
- Split image into patches
- Patches are arbitrarily cut off and linearly embedded!
- Patches are flattened!
- Loss function is not next-token prediction!
  - Uses a lot of data to learn compared to CNN
- Why are people still using this?

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.
Dosovitskiy et al. 2021

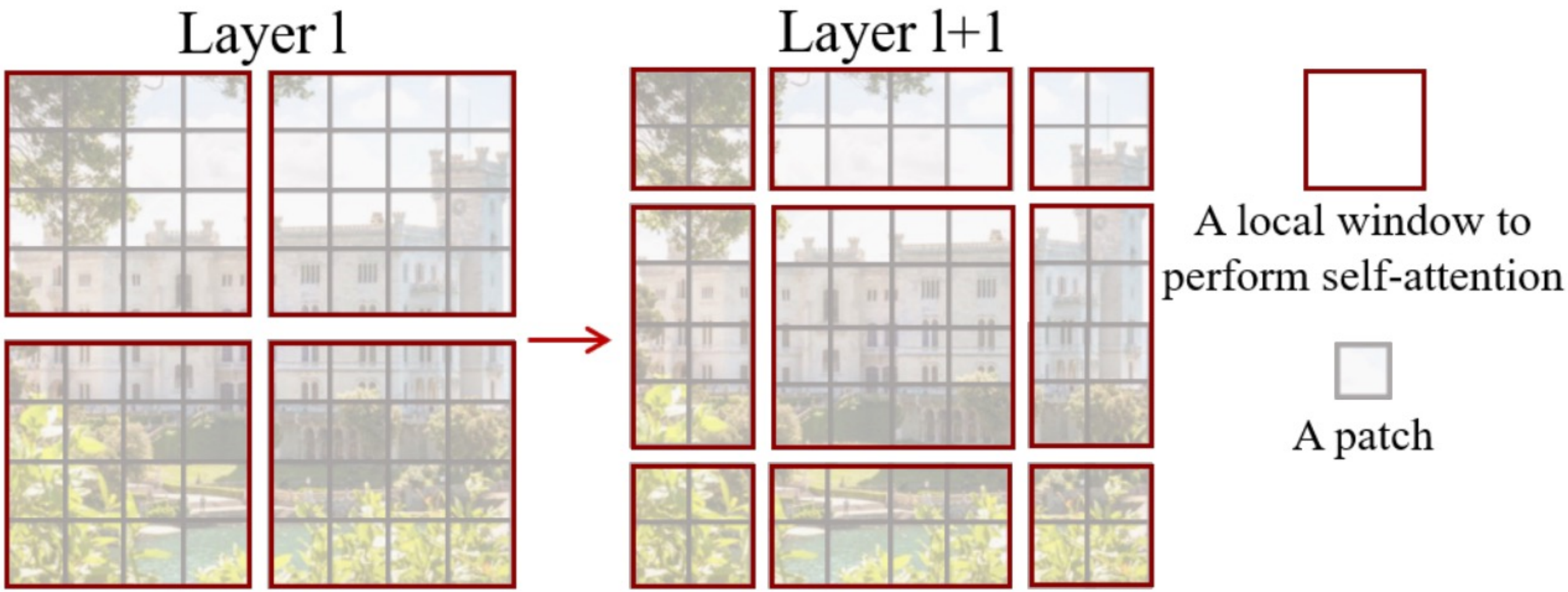
Swin Transformers vs ViT



- Swin Transformer has hierarchical segmentation of patches
- Finer details processed first at bottom, broad details processed last at top
- **My view:** Why not condition from the broad and use it to predict the bottom (specific)?

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. Liu et al. 2021.

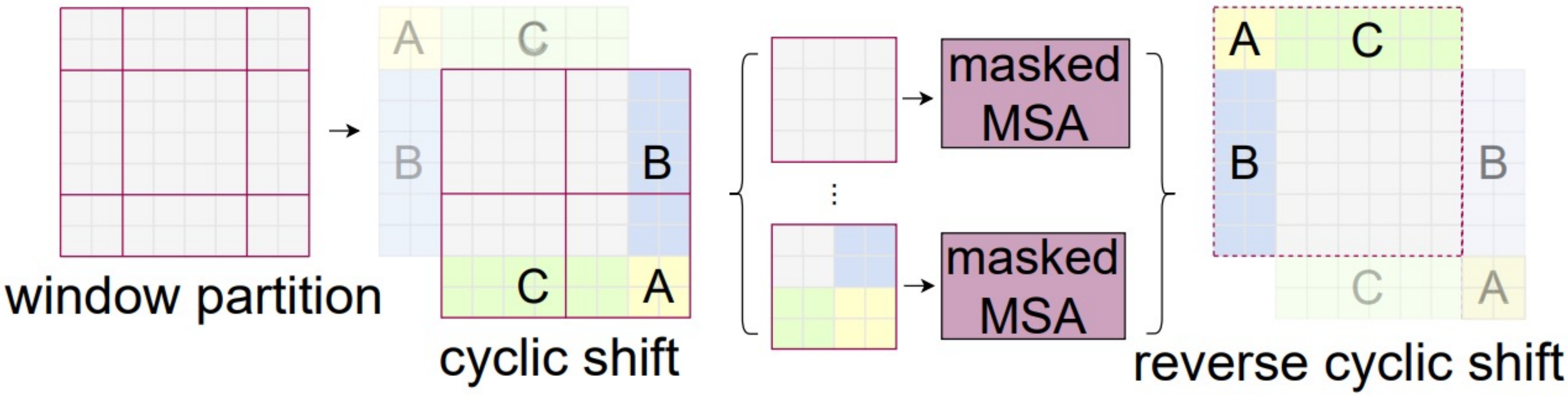
Swin Transformers: Shifted windows to view different combination of patches



Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. Liu et al. 2021.

Swin Transformers: Patch Masking

- Shifting windows help model to pay attention over different combinations of patches
- Patches which don't belong in original positions are masked



Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. Liu et al. 2021.

ViT's positional encoding may not be good!

	ImageNet top-1	ImageNet top-5	COCO AP^{box}	COCO AP^{mask}	ADE20k mIoU
w/o shifting	80.2	95.1	47.7	41.5	43.3
shifted windows	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**
no pos.	80.1	94.9	49.2	42.6	43.8
abs. pos.	80.5	95.2	49.0	42.4	43.2
abs.+rel. pos.	81.3	95.6	50.2	43.4	44.0
rel. pos. w/o app.	79.3	94.7	48.2	41.9	44.1
rel. pos.	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**

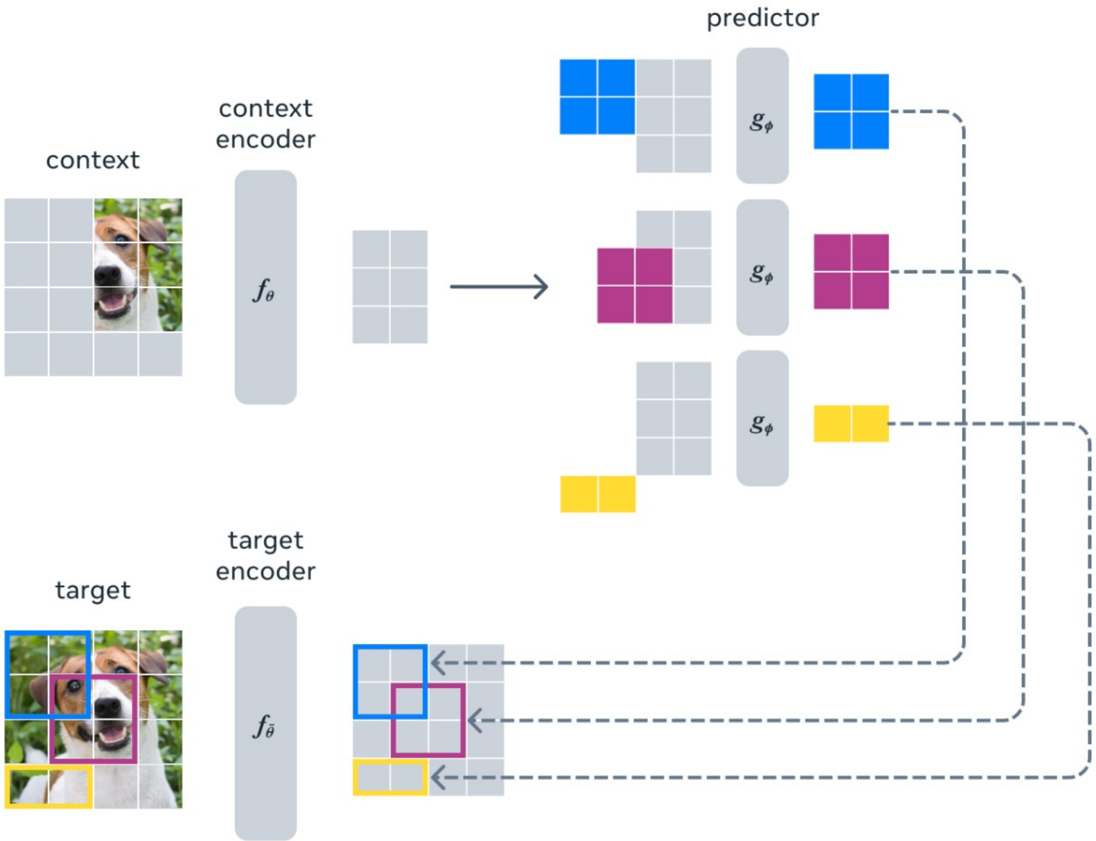
Table 4. Ablation study on the *shifted windows* approach and different position embedding methods on three benchmarks, using the Swin-T architecture. w/o shifting: all self-attention modules adopt regular window partitioning, without *shifting*; abs. pos.: absolute position embedding term of ViT; rel. pos.: the default settings with an additional relative position bias term (see Eq. (4)); app.: the first scaled dot-product term in Eq. (4).

- Swin Transformer has shown that **positional embeddings** in ViT **can largely be ignored** and get almost the same results!
- Inductive biases of translational invariance not present as compared to CNNs

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. Liu et al. 2021.

I-JEPA

I-JEPA: Predicting Image Patches in Latent Space



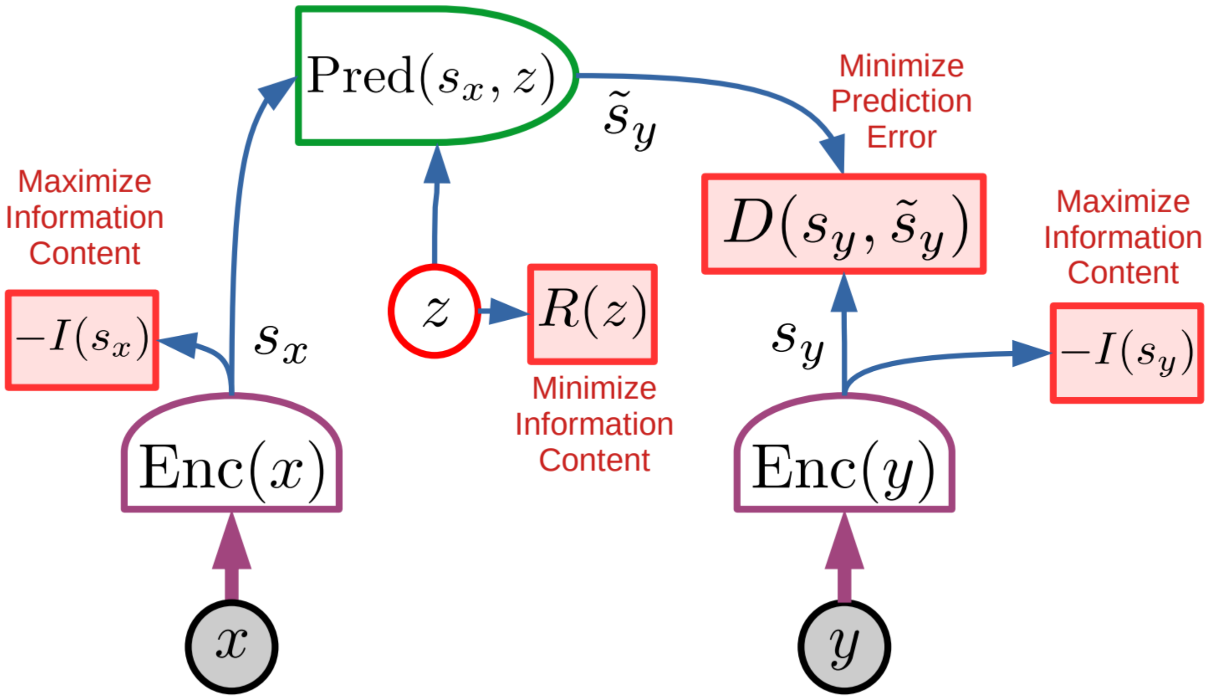
- Simple idea:
  - Mask out some parts of an image
  - Use non-masked parts as context
  - Predict the masked components in **latent space**!
- Pretty similar to masked token prediction in BERT!
- Self-supervised Learning
  - Can learn from unlabelled data

Use context to predict missing details

- Area outside blue box is context and fed as input
- Predict **latent space** representation of blue box
- Generative model trained to provide sketches of latent space

JEPA - Only use whatever is necessary to predict

- Prediction is done in latent space
- I-JEPA does not use information content losses



A Path towards Autonomous Machine Intelligence. Yann LeCun. 2022.

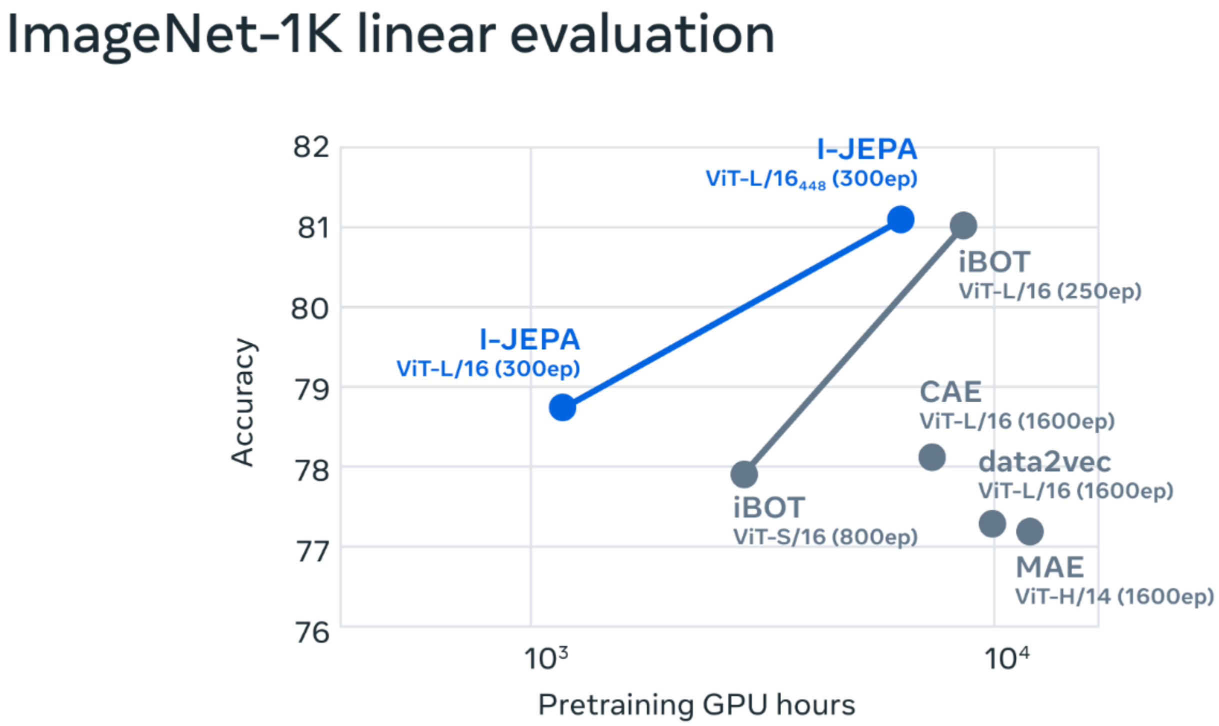
Loss Function

Loss. The loss is simply the average L_2 distance between the predicted patch-level representations $\hat{\boldsymbol{s}}_y(i)$ and the target patch-level representation $\boldsymbol{s}_y(i)$; i.e.,

$$\frac{1}{M}\sum_{i=1}^{M} D\left(\hat{\boldsymbol{s}}_y(i), \boldsymbol{s}_y(i)\right) = \frac{1}{M}\sum_{i=1}^{M}\sum_{j\in B_i} \|\hat{\boldsymbol{s}}_{y_j} - \boldsymbol{s}_{y_j}\|_2^2.$$

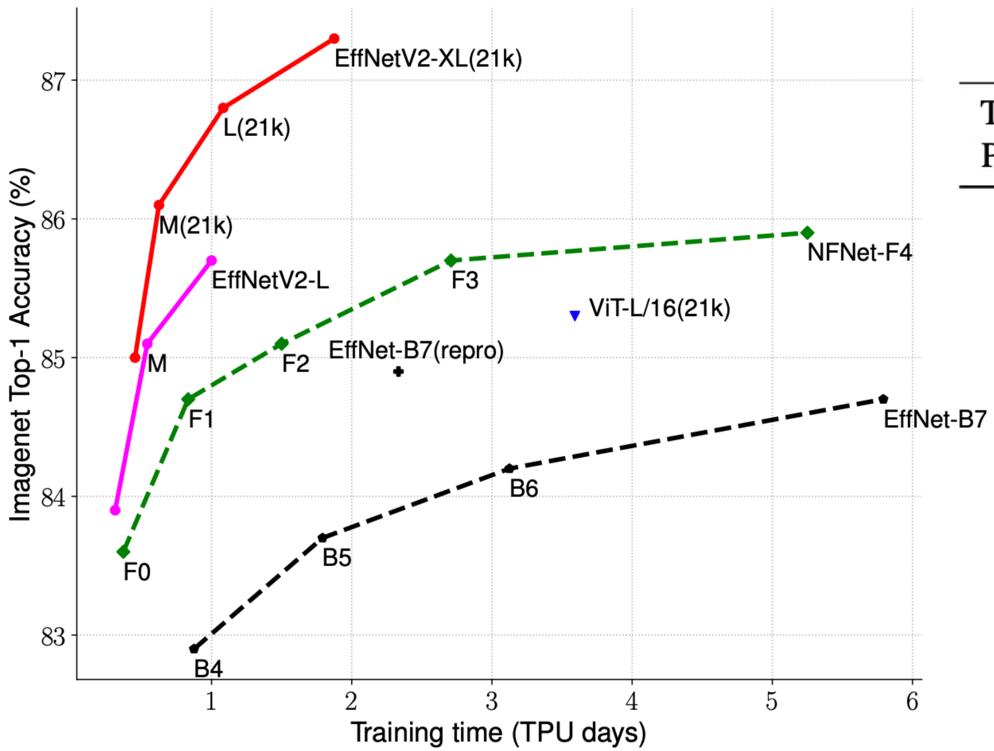
The parameters of the predictor, ϕ, and context encoder, θ, are learned through gradient-based optimization, while the parameters of the target encoder $\bar{\theta}$ are updated via an exponential moving average of the context-encoder parameters.

I-JEPA Pre-training is computationally efficient?



- It is, when compared to ViT
- Not when you compare with CNN-based architectures which can train on ImageNet within a day
- No need data augmentations unlike contrastive methods like BYOL, VICReg

Comparison: EfficientNet V2



	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

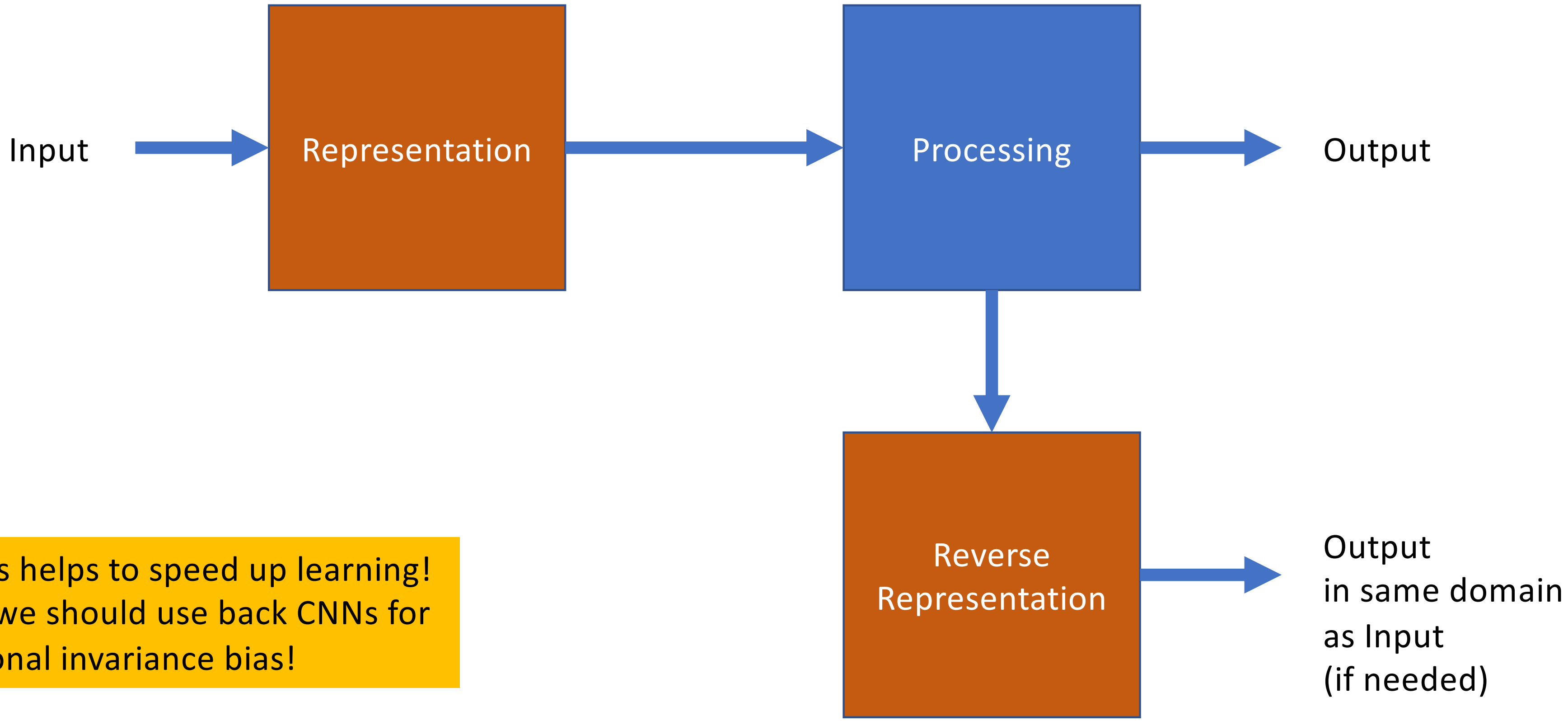
Table 4. EfficientNetV2-S architecture – MBConv and Fused-MBConv blocks are described in Figure 2.

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

EfficientNetV2: Smaller Models and Faster Training. Tan and Le. 2021.

Thoughts

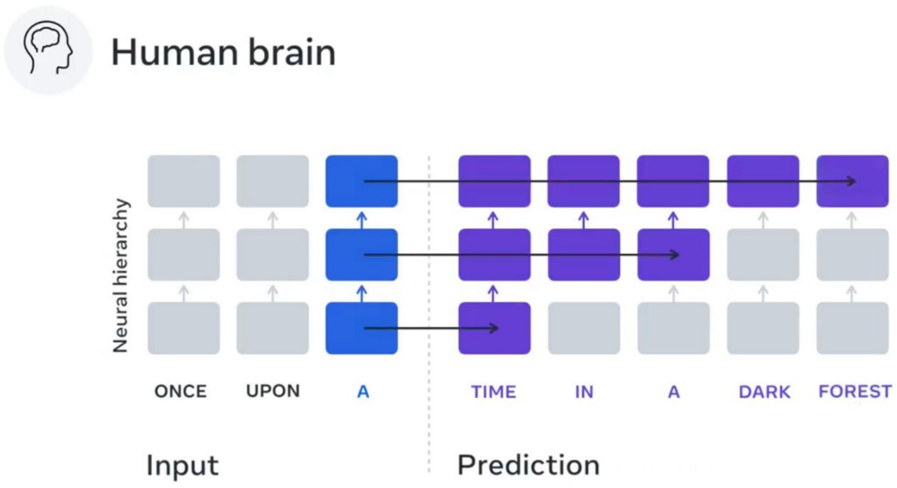
Information Pipeline – Bias for Representation



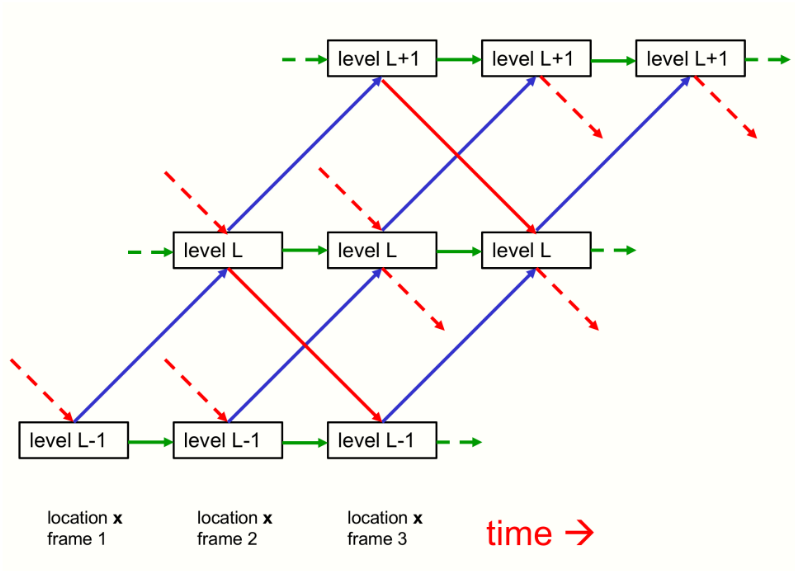
Fixed bias helps to speed up learning!
Perhaps we should use back CNNs for
translational invariance bias!

Hierarchical Prediction is the future

- Hierarchical prediction of more than just next token, but broader prediction at higher levels
- Higher level prediction can be more abstract and less detailed than lower levels



Evidence of a predictive coding hierarchy in the human brain listening to speech. Caucheteux. 2022. Nature Human Behaviour.



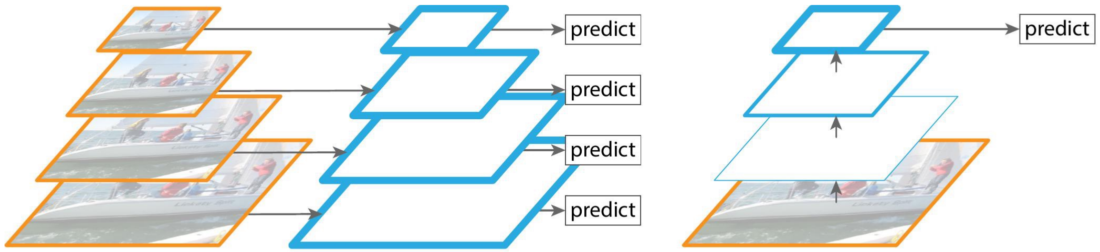
How to represent part-whole hierarchies in a neural network. Hinton. 2021.

Better Grounding

- Perhaps we do in-filling by grounding our generation with some context – high to low level context conditioning at various scales
- Innate Biases – We have certain fixed priors which we use to predict the world
  - Extend lines in a straight way
  - Extrapolate patterns
- Memory – We could use memory of objects/similar scenes in latent space form or text for context to ground the generation of latent representations
- Memory could be the Key, Value for the Transformer architecture, while the present state/latent space is the Query.

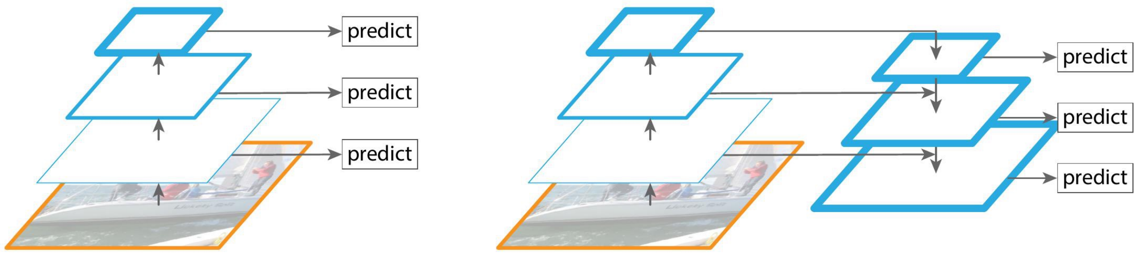
Hierarchical Prediction - Feature Pyramid Network

- Hierarchical prediction from coarse-grained image to fine-grained image
- **My view:** can perhaps use text/latent space in memory for grounding various scales
- **My view:** Condition finer grained prediction on the upper layers of the hierarchy



(a) Featurized image pyramid

(b) Single feature map



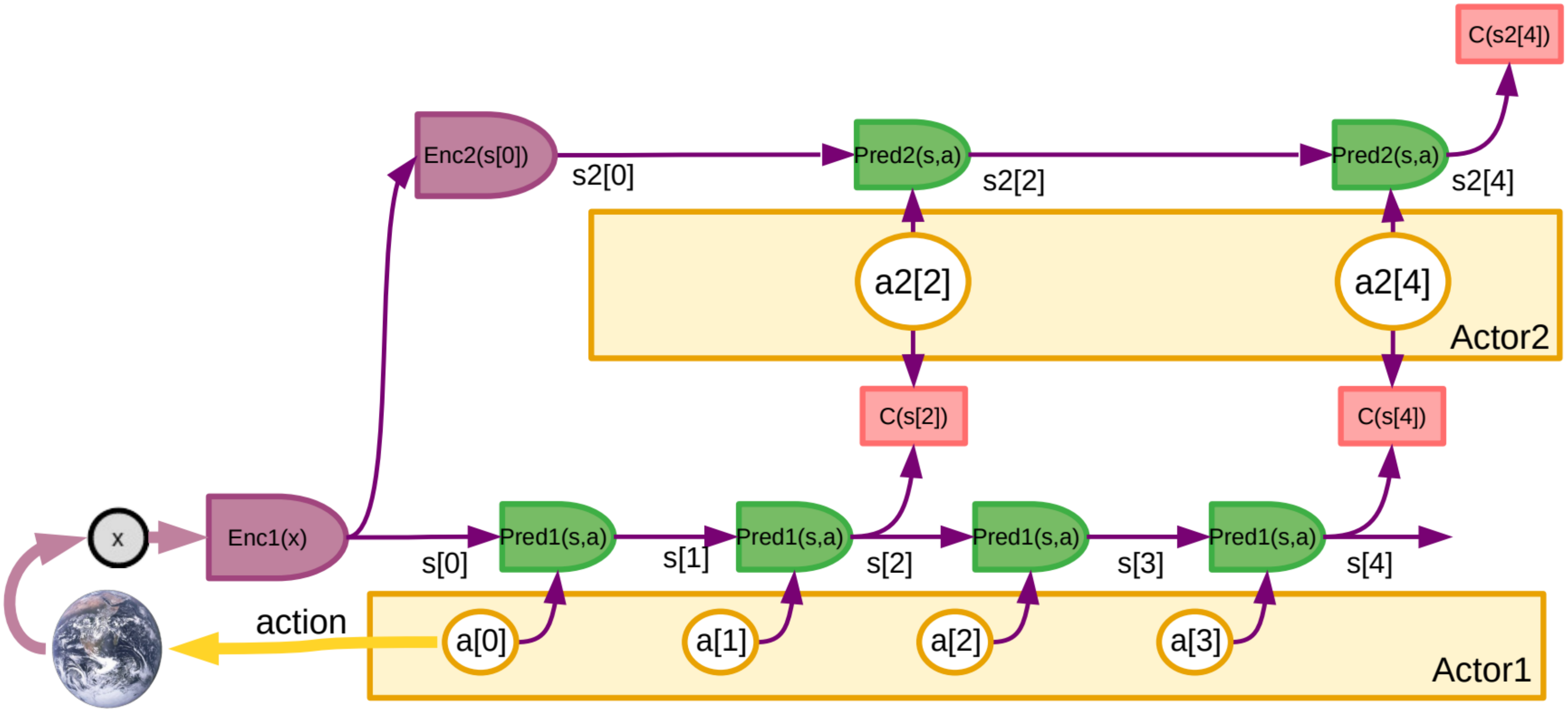
(c) Pyramidal feature hierarchy

(d) Feature Pyramid Network

Feature Pyramid Networks for Object Detection. Lin et al. 2017.

Hierarchical JEPA

- Hierarchical prediction of actions from the highest level action to the lowest level action



A Path towards Autonomous Machine Intelligence. Yann LeCun. 2022.

Questions to Ponder

- Are Vision Transformers (ViT) an effective way to learn? How can we make it better and incorporate the inductive biases of space like in CNNs?
- Is prediction in latent space good? Why can’t we do it in pixel space?
- Is there a way to do self-supervised learning for images better?
- How can we incorporate hierarchy into I-JEPA?
- How can memory be used for prediction?