Topics:

Vision Transformers (ViT, Swin)

Vision Encoders as Foundation Models: DINO → DINOv3

Setting the stage for VLMs

CS 8803-VLM

ZSOLT KIRA

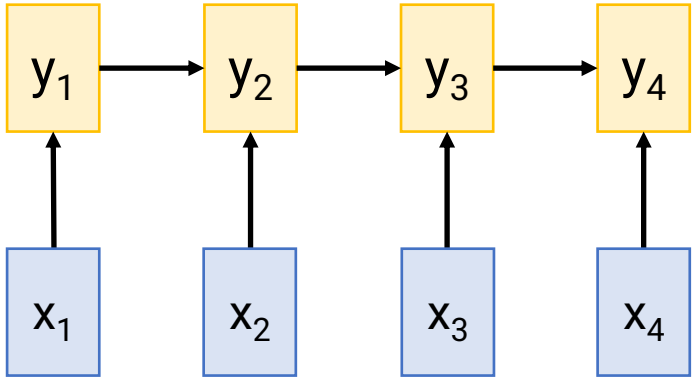
Many slides by Justin Johnson

- Logistics:
- Paper presentation sign-up was due Aug 31 (11:59pm ET)
- Team roster submission due Sep 10 (11:59pm ET)
- Project proposal presentations: Sep 17 (slides due Sep 16)

- Today (L3): vision encoders as foundation models — ViT, Swin, DINO → DINOv3
- Thursday (L4): vision–language alignment — CLIP → SigLIP 2

Three Ways of Processing Sequences

Recurrent Neural Network

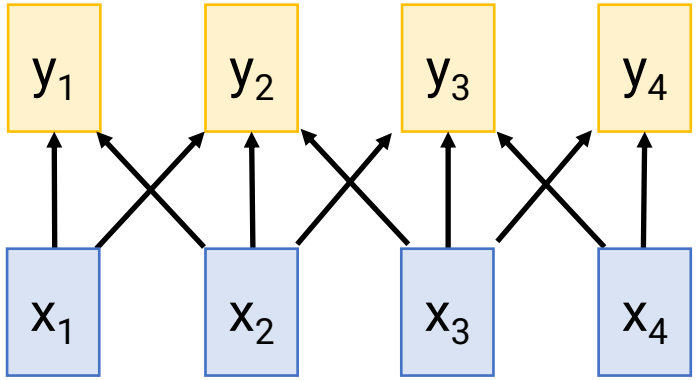


Works on **Ordered Sequences**

(+) Good at long sequences: After one RNN layer, h_T "sees" the whole sequence

(-) Not parallelizable: need to compute hidden states sequentially

1D Convolution

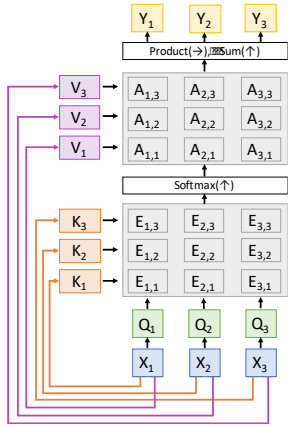


Works on **Multidimensional Grids**

(-) Bad at long sequences: Need to stack many conv layers for outputs to "see" the whole sequence

(+) Highly parallel: Each output can be computed in parallel

Self-Attention



Works on **Sets of Vectors**

(+) Good at long sequences: after one self-attention layer, each output "sees" all inputs!

(+) Highly parallel: Each output can be computed in parallel

(-) Very memory intensive

Slide credit: Justin Johnson

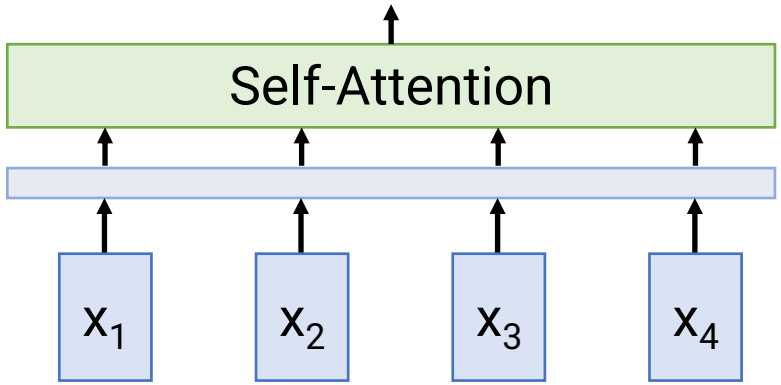
The Transformer

Vaswani et al, “Attention is all you need”, NeurIPS 2017

Slide credit: Justin Johnson

The Transformer

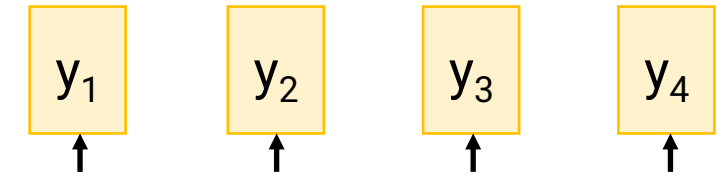
All vectors interact with each other



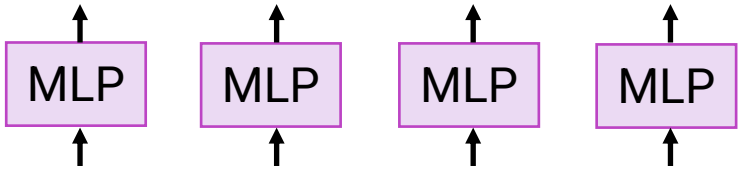
Vaswani et al, “Attention is all you need”, NeurIPS 2017

Slide credit: Justin Johnson

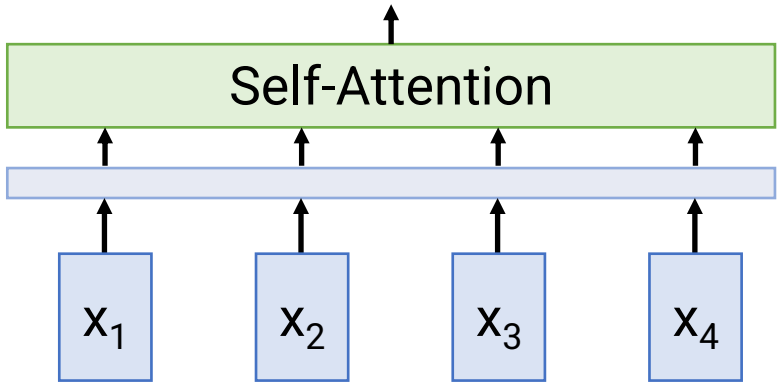
The Transformer



MLP independently on each vector **(weight shared!)**



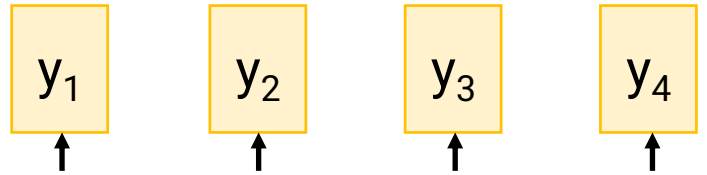
All vectors interact with each other



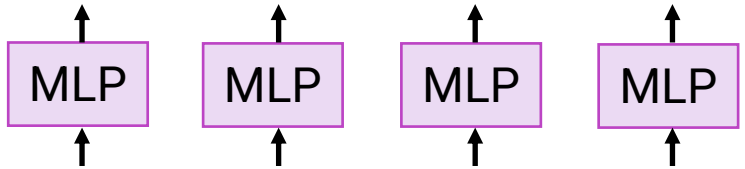
Vaswani et al, “Attention is all you need”, NeurIPS 2017

Slide credit: Justin Johnson

The Transformer

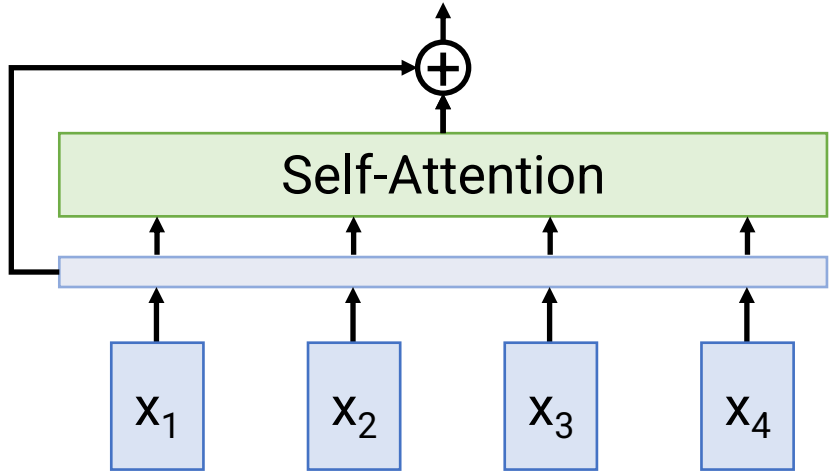


MLP independently on each vector



Residual connection

All vectors interact with each other



Vaswani et al, “Attention is all you need”, NeurIPS 2017

Slide credit: Justin Johnson

The Transformer

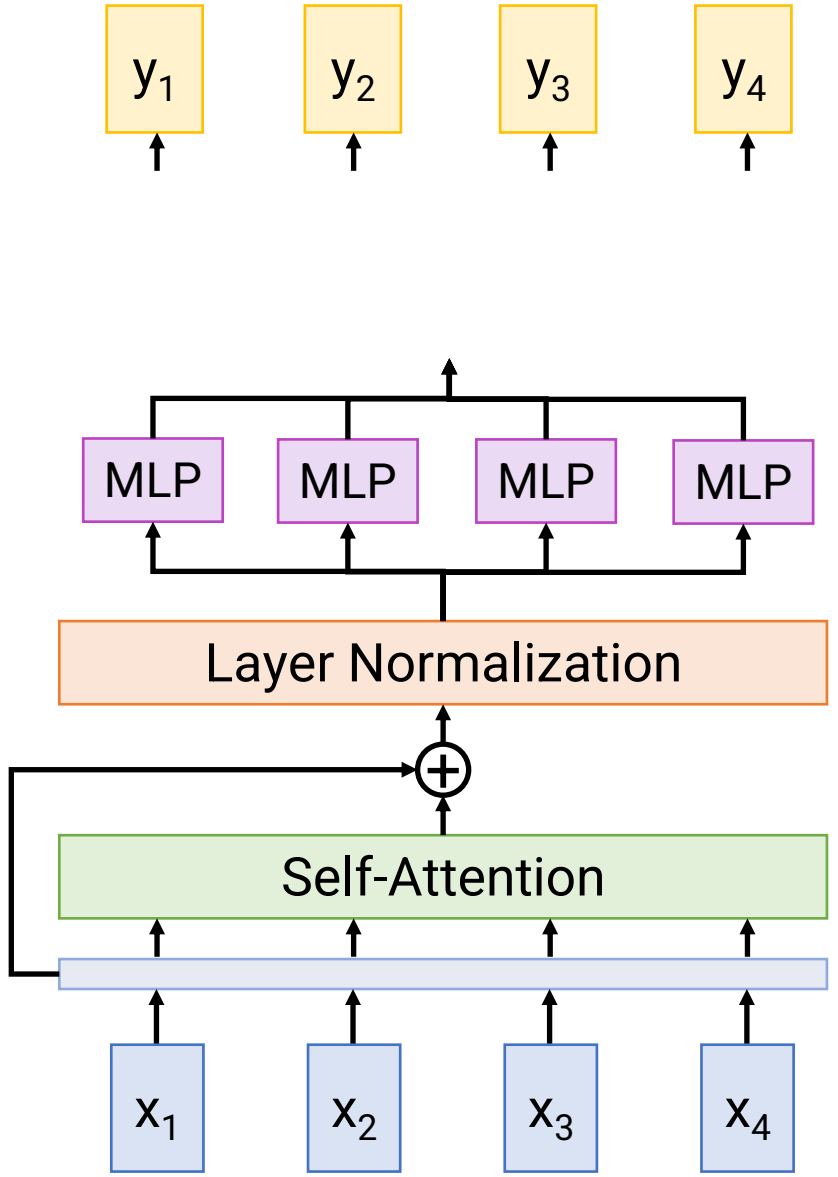
Recall **Layer Normalization**:
Given $h_1, \ldots, h_N$ (Shape: D)
scale: γ (Shape: D)
shift: β (Shape: D)
$\mu_i = (1/D)\sum_j h_{i,j}$ (scalar)
$\sigma_i = (\sum_j (h_{i,j} - \mu_i)^2)^{1/2}$ (scalar)
$z_i = (h_i - \mu_i) / \sigma_i$
$y_i = \gamma * z_i + \beta$

Ba et al, 2016

MLP independently on each vector

Residual connection

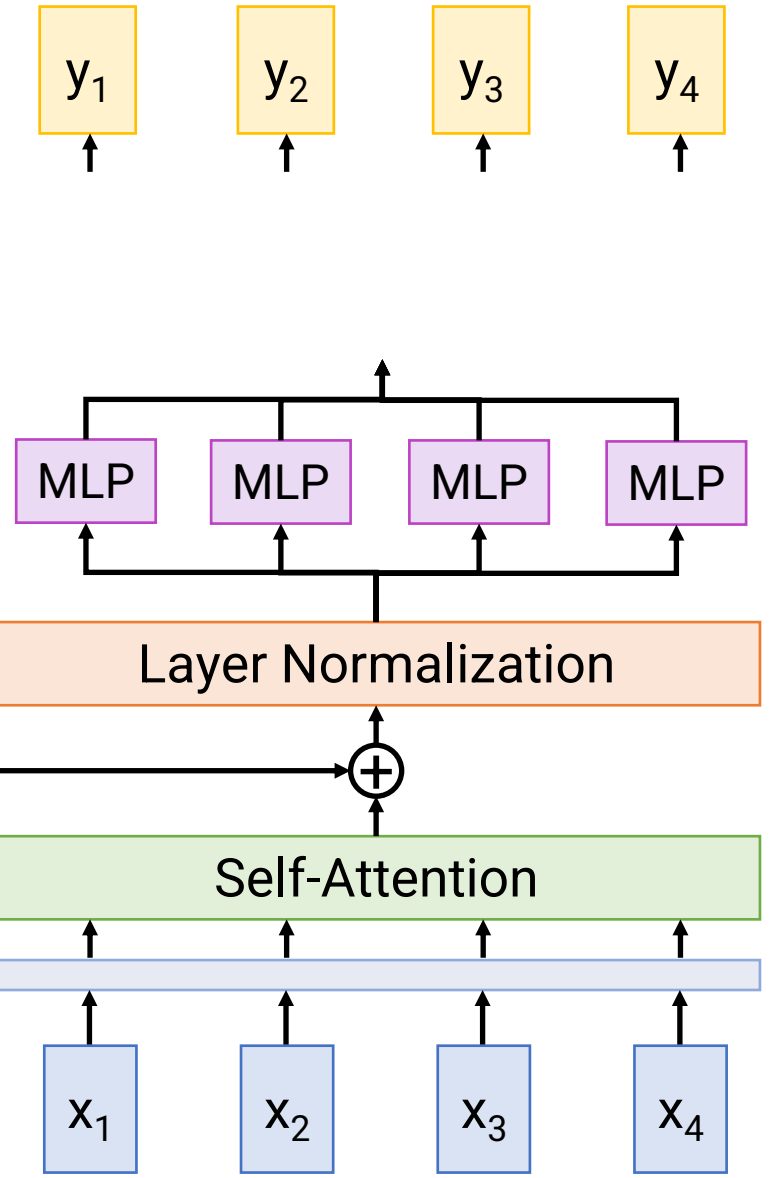
All vectors interact with each other



Vaswani et al, "Attention is all you need", NeurIPS 2017

Slide credit: Justin Johnson

The Transformer



MLP independently on each vector

Residual connection

All vectors interact with each other

Vaswani et al, “Attention is all you need”, NeurIPS 2017

Slide credit: Justin Johnson

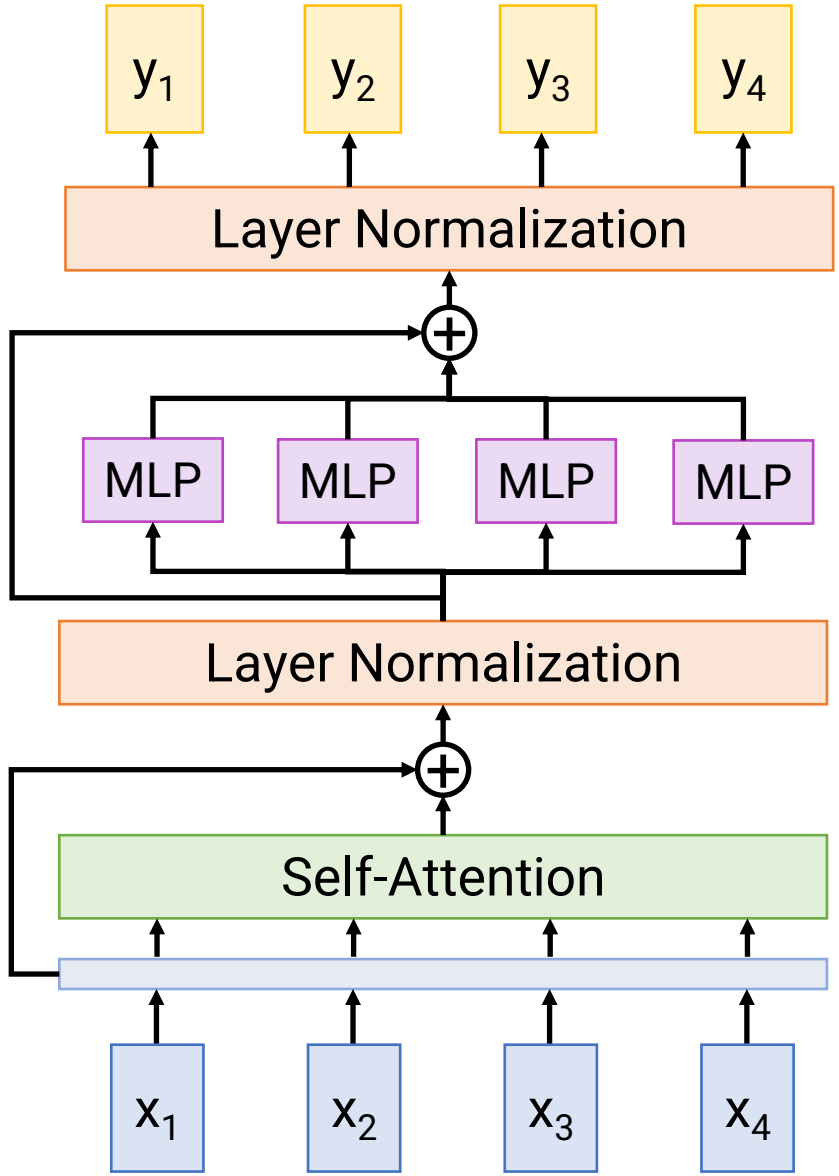
The Transformer

Residual connection

MLP independently on each vector

Residual connection

All vectors interact with each other



Vaswani et al, “Attention is all you need”, NeurIPS 2017

Slide credit: Justin Johnson

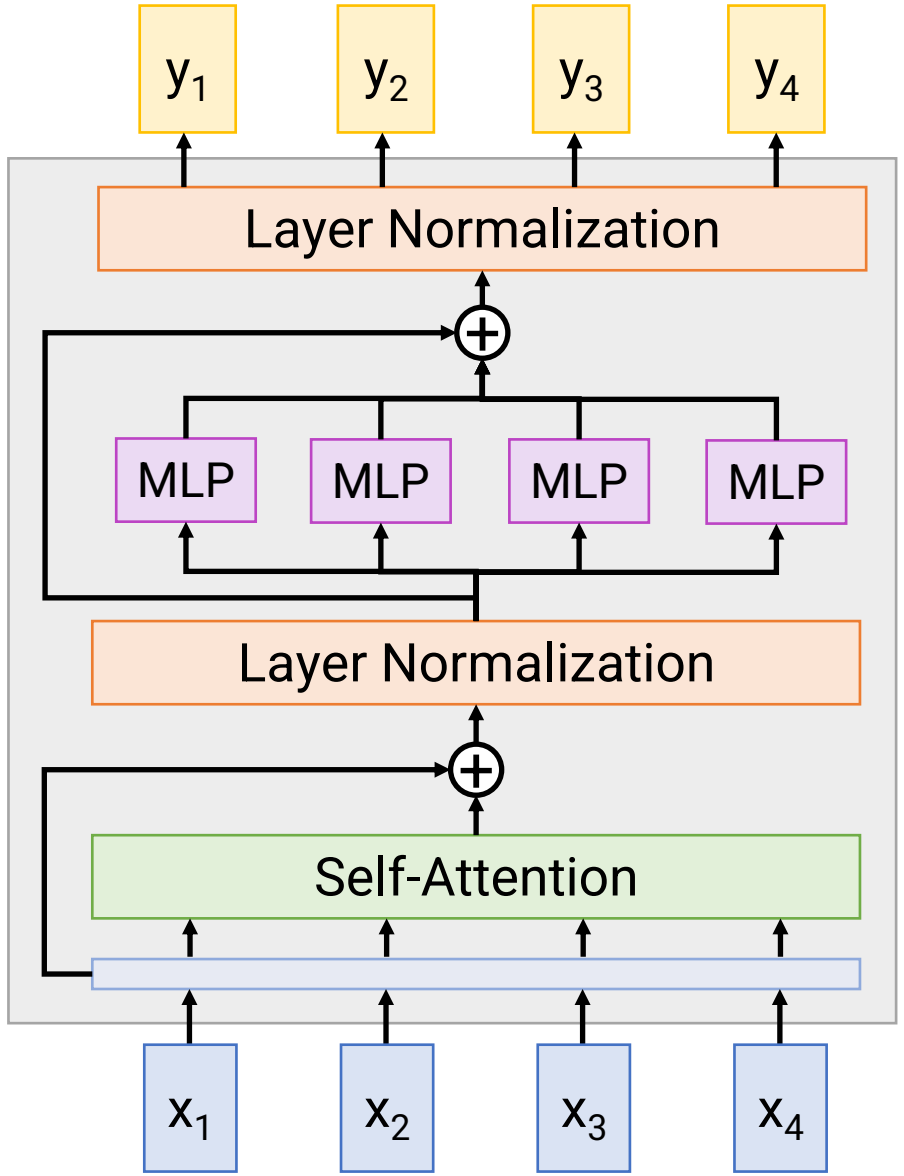
The Transformer

Transformer Block:
Input: Set of vectors x
Output: Set of vectors y

Self-attention is the only interaction between vectors!

Layer norm and MLP work independently per vector

Highly scalable, highly parallelizable



Vaswani et al, "Attention is all you need", NeurIPS 2017

Slide credit: Justin Johnson

The Transformer

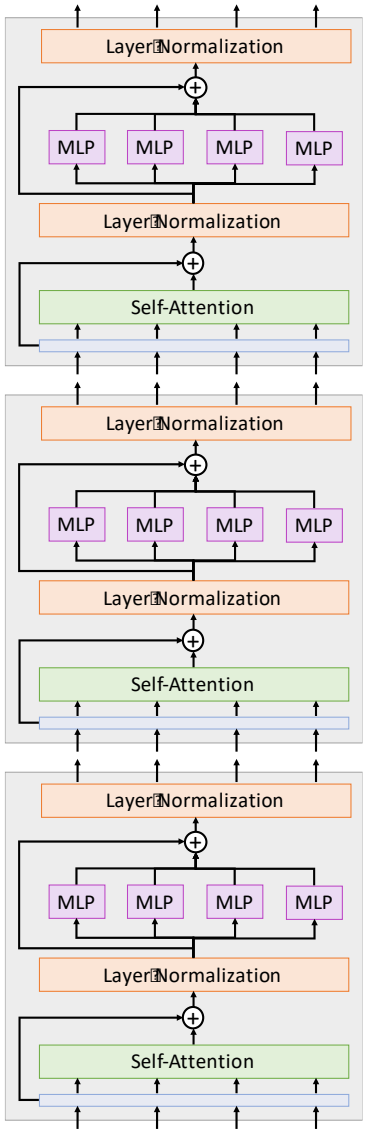
Transformer Block:
Input: Set of vectors x
Output: Set of vectors y

Self-attention is the only interaction between vectors!

Layer norm and MLP work independently per vector

Highly scalable, highly parallelizable

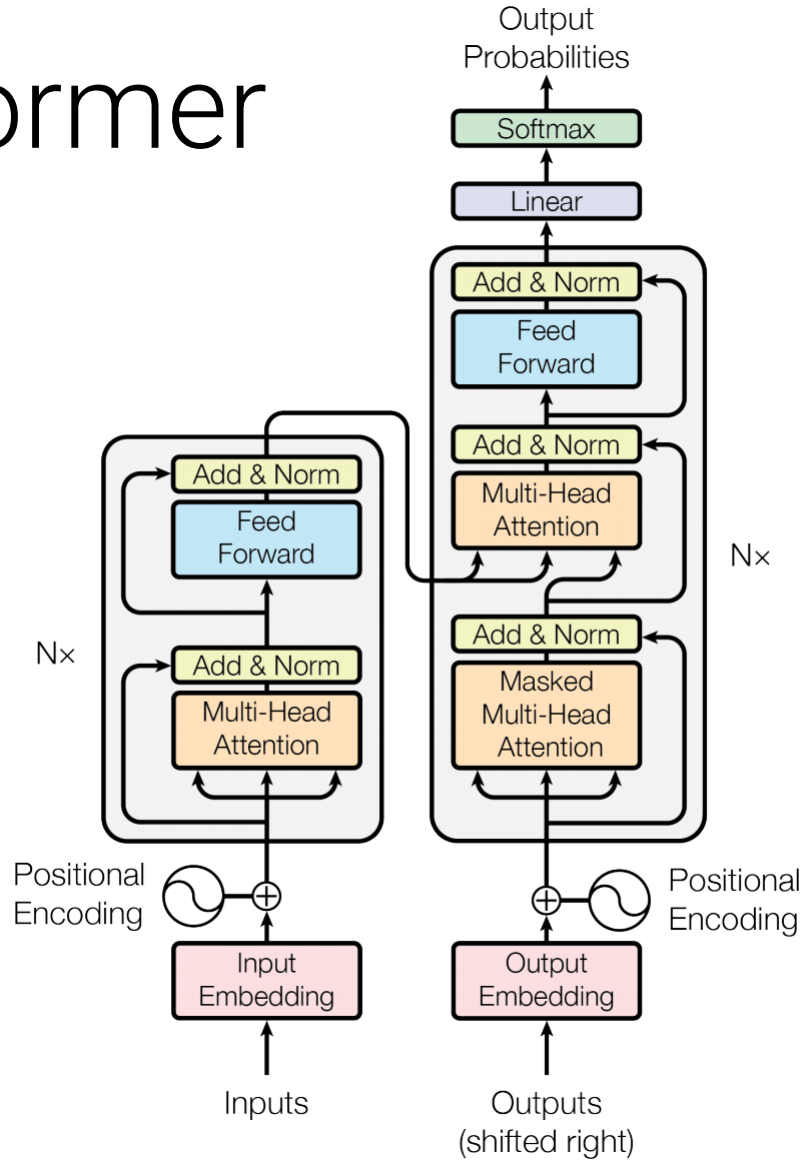
A **Transformer** is a sequence of transformer blocks



Vaswani et al, “Attention is all you need”, NeurIPS 2017

Slide credit: Justin Johnson

The Transformer



Encoder-Decoder

Details:

- Tokenization is messy! Trained chunking mechanism
- Position encoding
  - sin/cos: Normalized, nearby tokens have similar values, etc.
  - Added to input embedding

- When to use decoder-only versus encoder-decoder model is open problem
  - GPT is decoder only!

Vaswani et al, “Attention is all you need”, NeurIPS 2017

Slide credit: Justin Johnson

Model	Layers	Width	Heads	Params	Data	Training
Transformer-Base	12	512	8	65M		8x P100 (12 hours)
Transformer-Large	12	1024	16	213M		8x P100 (3.5 days)
BERT-Base	12	768	12	110M	13 GB	
BERT-Large	24	1024	16	340M	13 GB	
XLNet-Large	24	1024	16	~340M	126 GB	512x TPU-v3 (2.5 days)
RoBERTa	24	1024	16	355M	160 GB	1024x V100 GPU (1 day)
GPT-2	48	1600	?	1.5B	40 GB	
Megatron-LM	72	3072	32	8.3B	174 GB	512x V100 GPU (9 days)
Turing-NLG	78	4256	28	17B	?	256x V100 GPU
GPT-3	96	12,288	96	175B	694GB	?
Gopher	80	16,384	128	280B	10.55 TB	4096x TPUv3 (38 days)

Can Attention/Transformers be used from more than text processing?

ViLBERT: A Visolinguistic Transformer

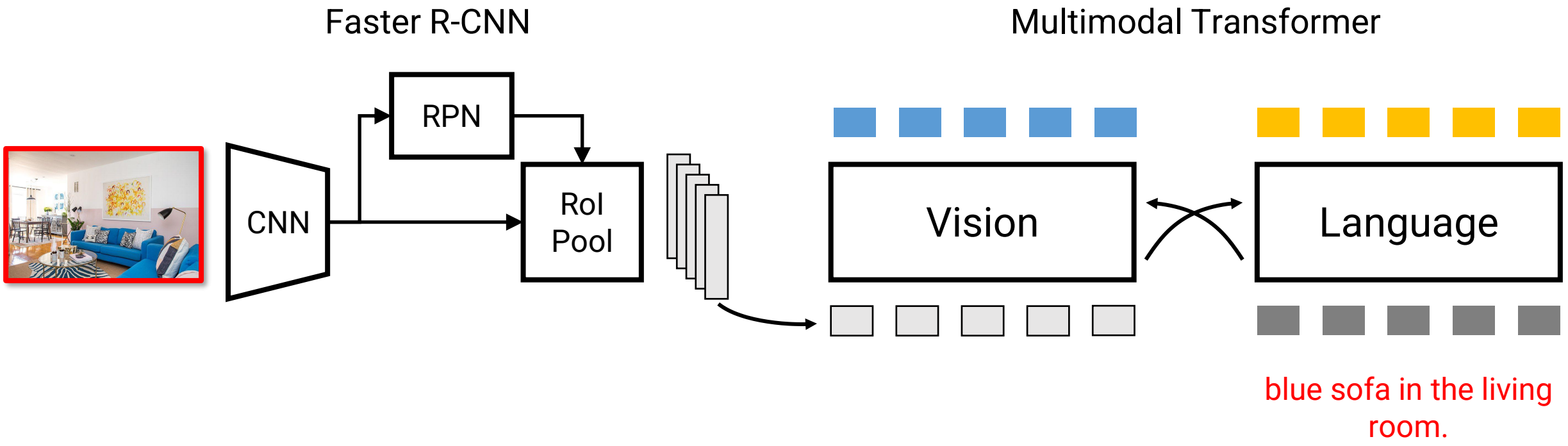
pop artist performs at the festival in a city.

a worker helps to clear the debris.

blue sofa in the living room.

Image and captions from: Sharma, Piyush, et al. "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning." ACL. 2018.

ViLBERT: A Visolinguistic Transformer



Lu et al "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks." *NeurIPS*. 2019.
Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *NeurIPS*. 2015.

What about for just image inputs? Without Convolution?

Preprint. Under review.

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†

*equal technical contribution, †equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

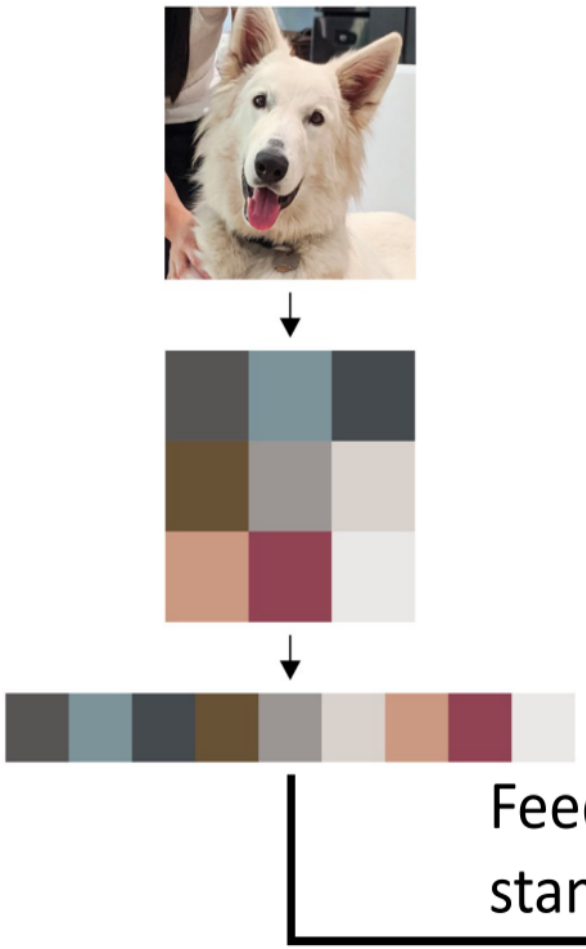
Slide progression inspired by Soheil Feizi

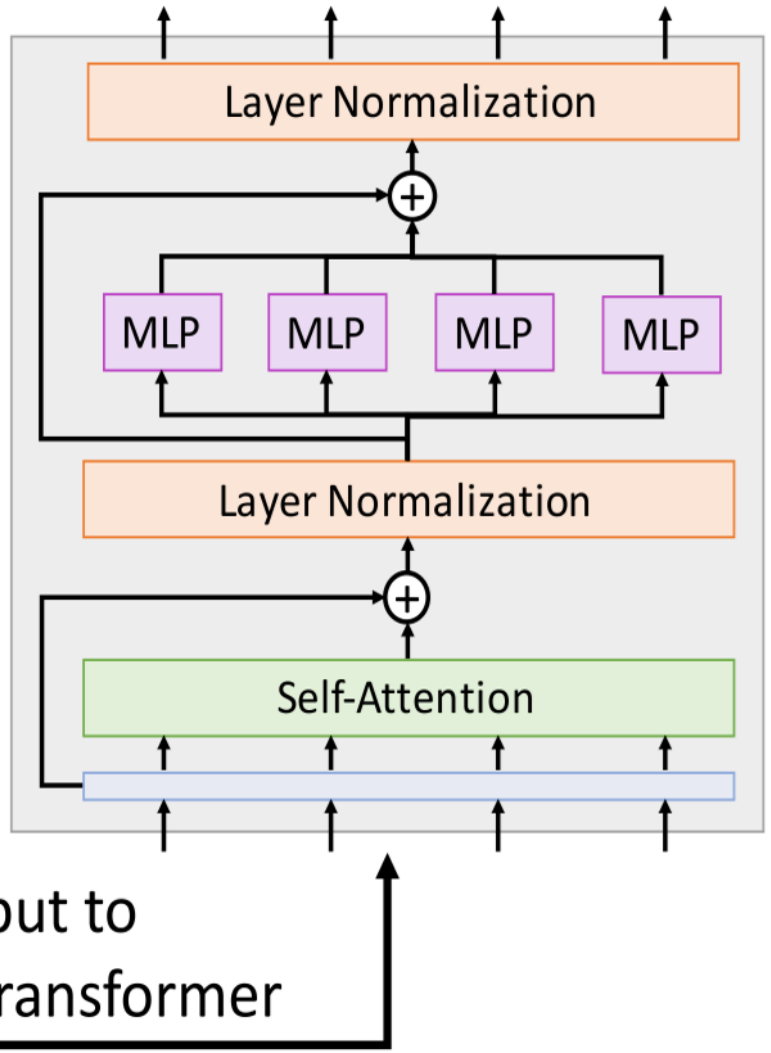
What About Vision with just Self-Attention?

Georgia Tech

Idea : Standard Transformer on Pixels

Treat an image as a set of pixel values





Feed as input to standard Transformer

Problem: Memory use!

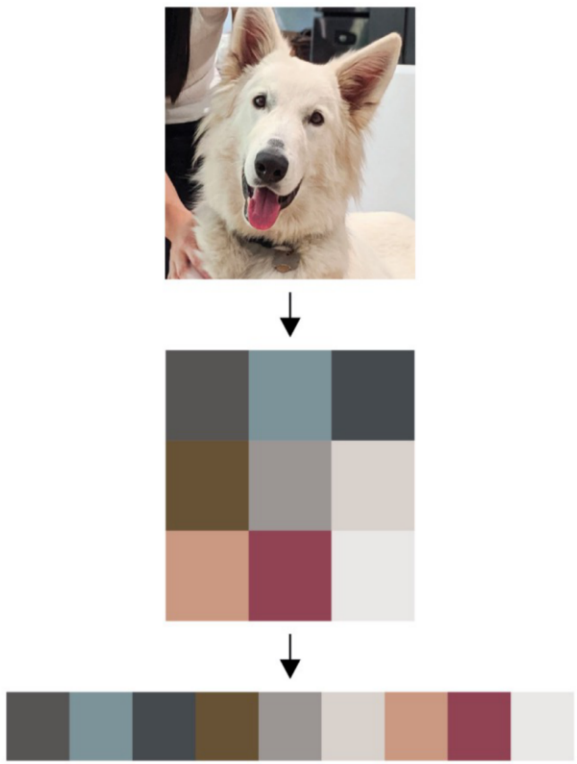
R x R image needs R^4 elements per attention matrix

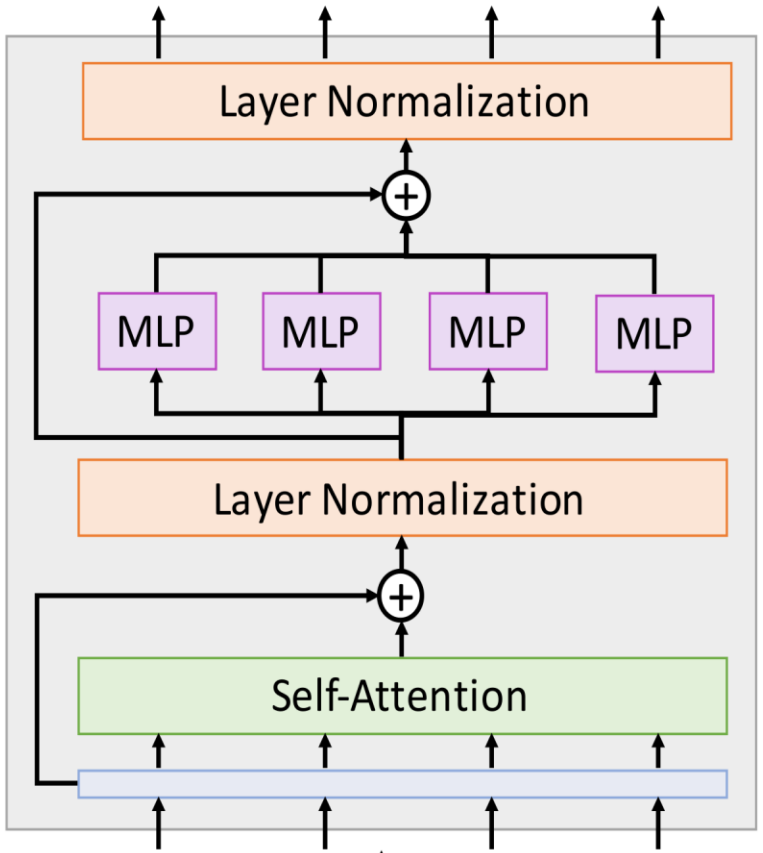
Chen et al, "Generative Pretraining from Pixels", ICML 2020

Slide credit: Justin Johnson

Idea #3: Standard Transformer on Pixels

Treat an image as a set of pixel values





Feed as input to standard Transformer

Problem: Memory use!

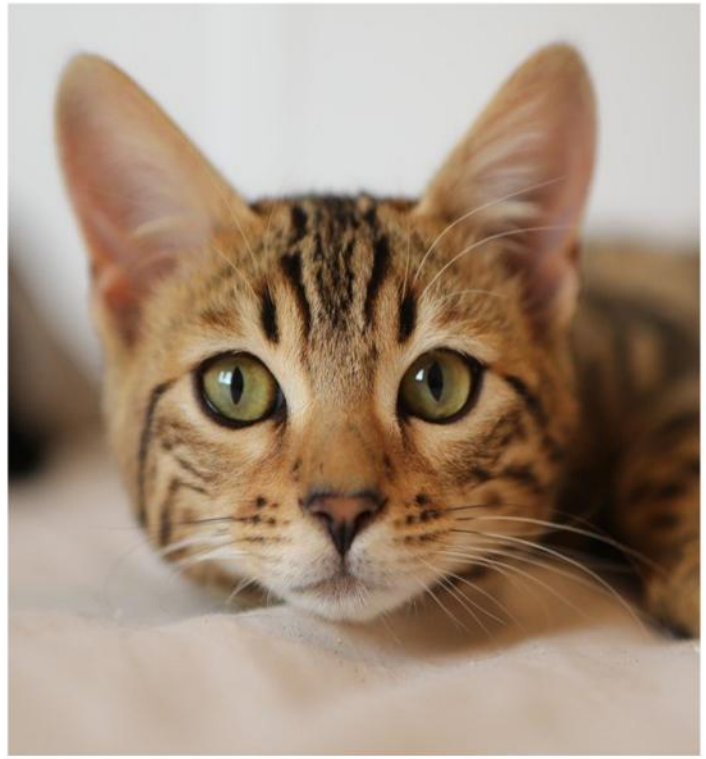
R x R image needs R^4 elements per attention matrix

R=128, 48 layers, 16 heads per layer takes 768GB of memory for attention matrices for a single example…

Chen et al, "Generative Pretraining from Pixels", ICML 2020

Slide credit: Justin Johnson

Idea : Standard Transformer on Patches

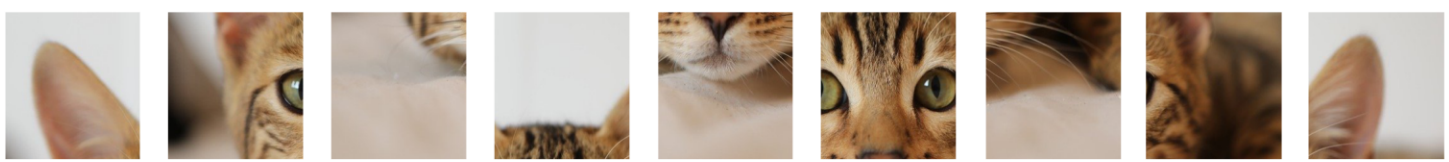


Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR 2021

Slide credit: Justin Johnson

Idea #4: Standard Transformer on Patches

N input patches, each of shape 3x16x16



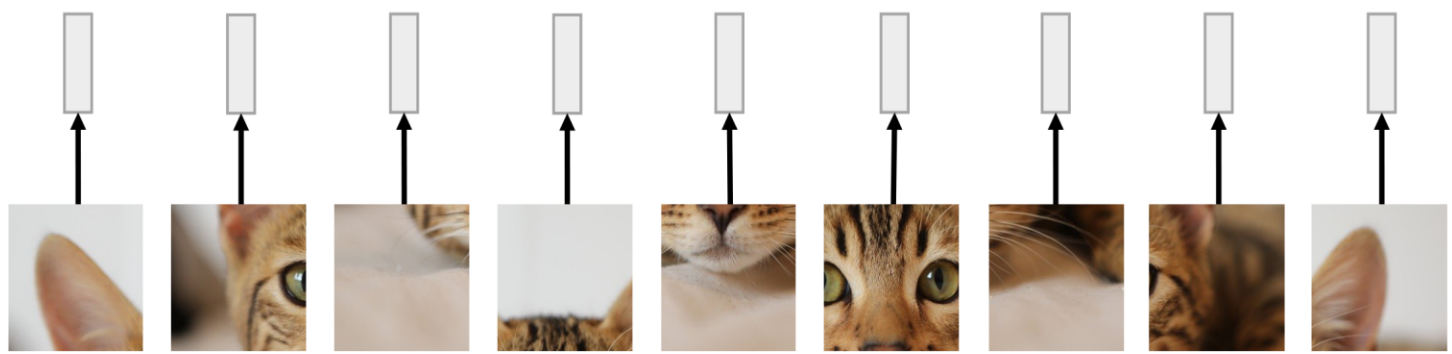
Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR 2021

Slide credit: Justin Johnson

Idea #4: Standard Transformer on Patches

Linear projection to D-dimensional vector

N input patches, each of shape 3x16x16



Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR 2021

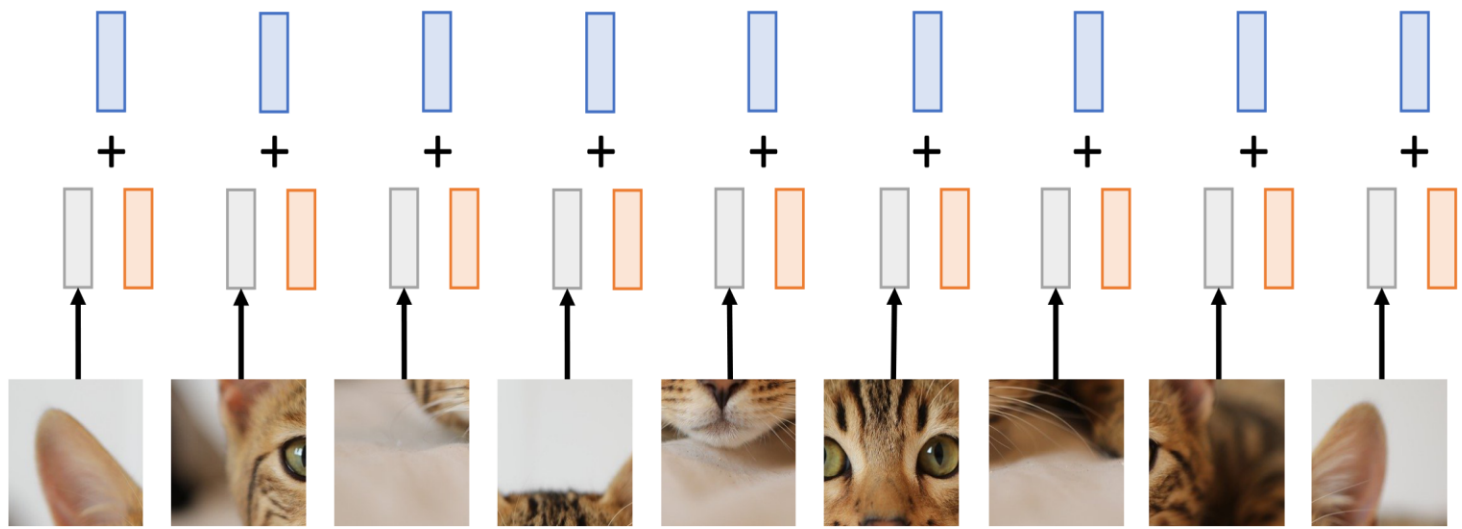
Slide credit: Justin Johnson

Idea #4: Standard Transformer on Patches

Add positional embedding: learned D-dim vector per position

Linear projection to D-dimensional vector

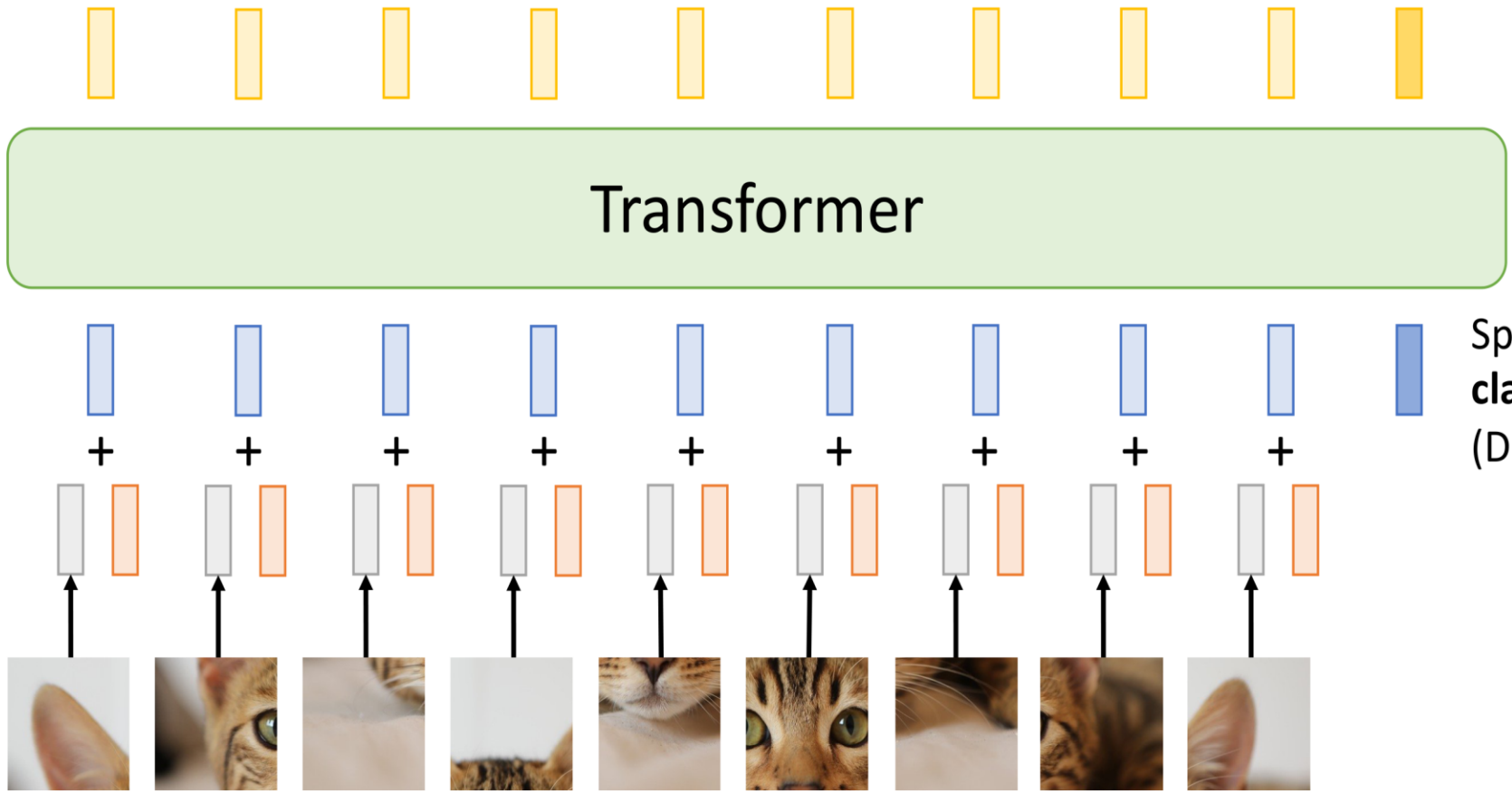
N input patches, each of shape 3x16x16



Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR 2021

Slide credit: Justin Johnson

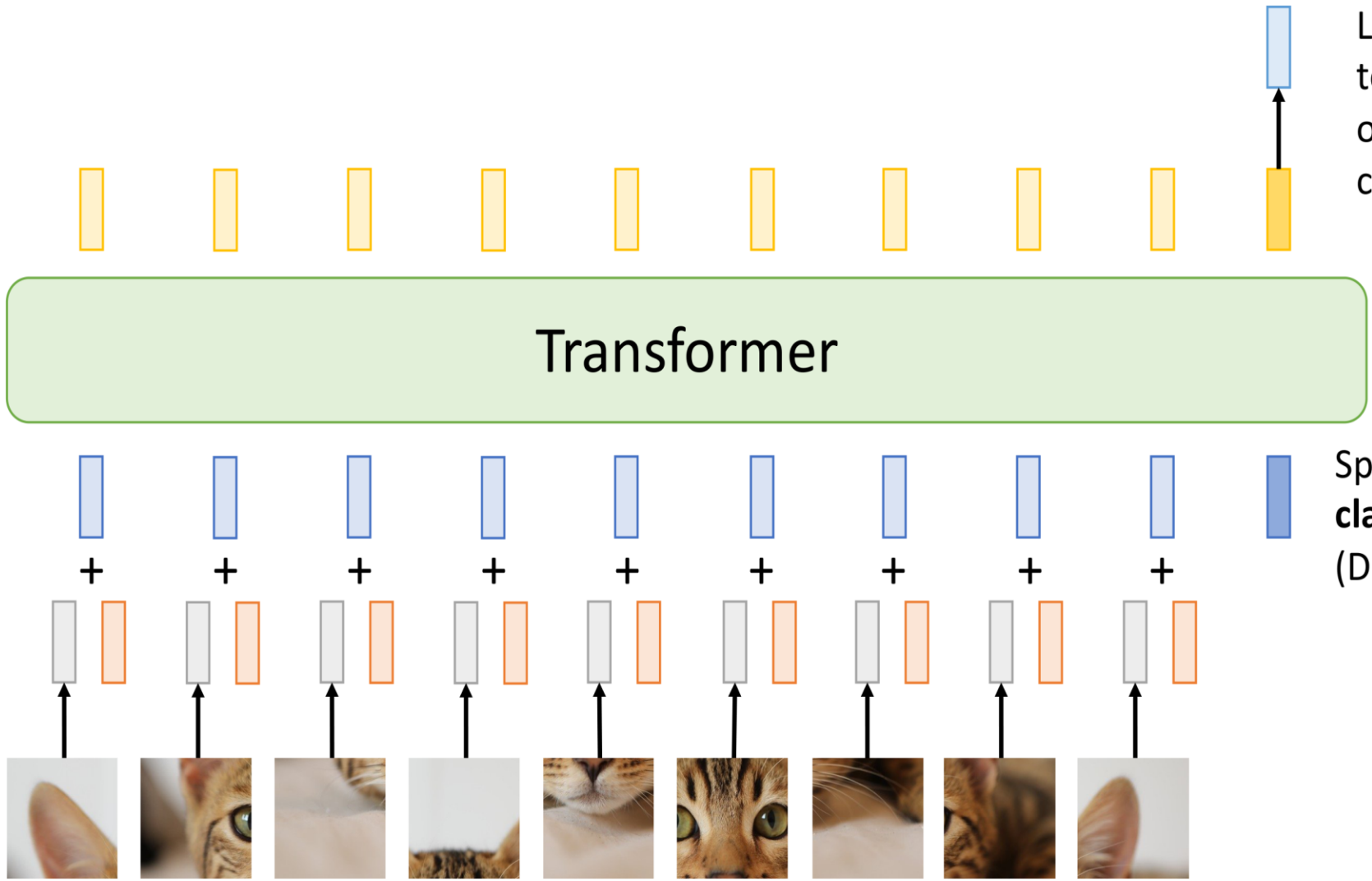
Idea #4: Standard Transformer on Patches



Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR 2021

Slide credit: Justin Johnson

Idea #4: Standard Transformer on Patches



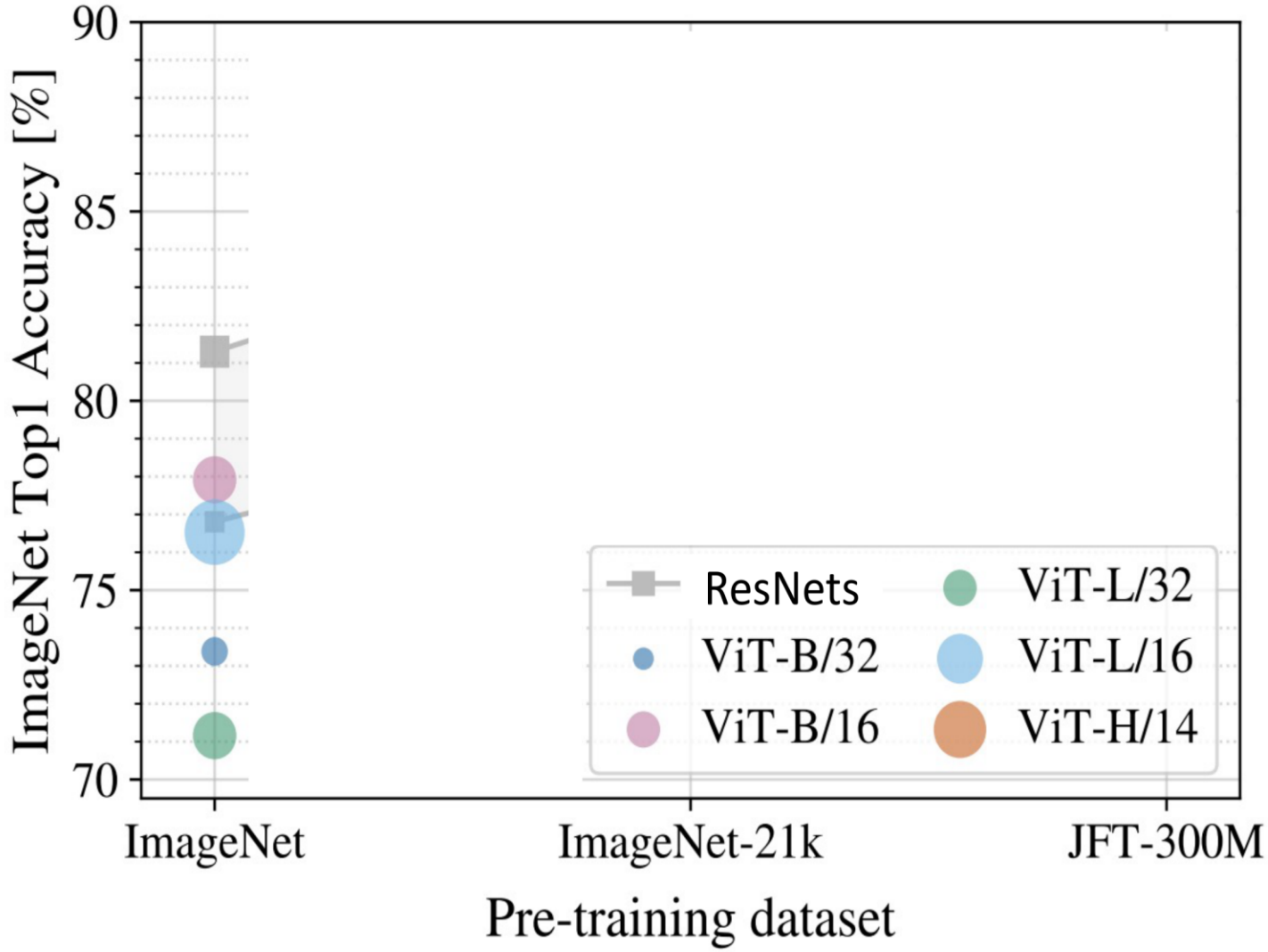
Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR 2021

Cat image is free for commercial use under a Pixabay license

Vision Transformer (ViT) vs ResNets

Recall: ImageNet dataset has 1k categories, 1.2M images

When trained on ImageNet, ViT models perform worse than ResNets



B = Base
L = Large
H = Huge

/32, /16, /14 is patch size; smaller patch size is a bigger model (more patches)

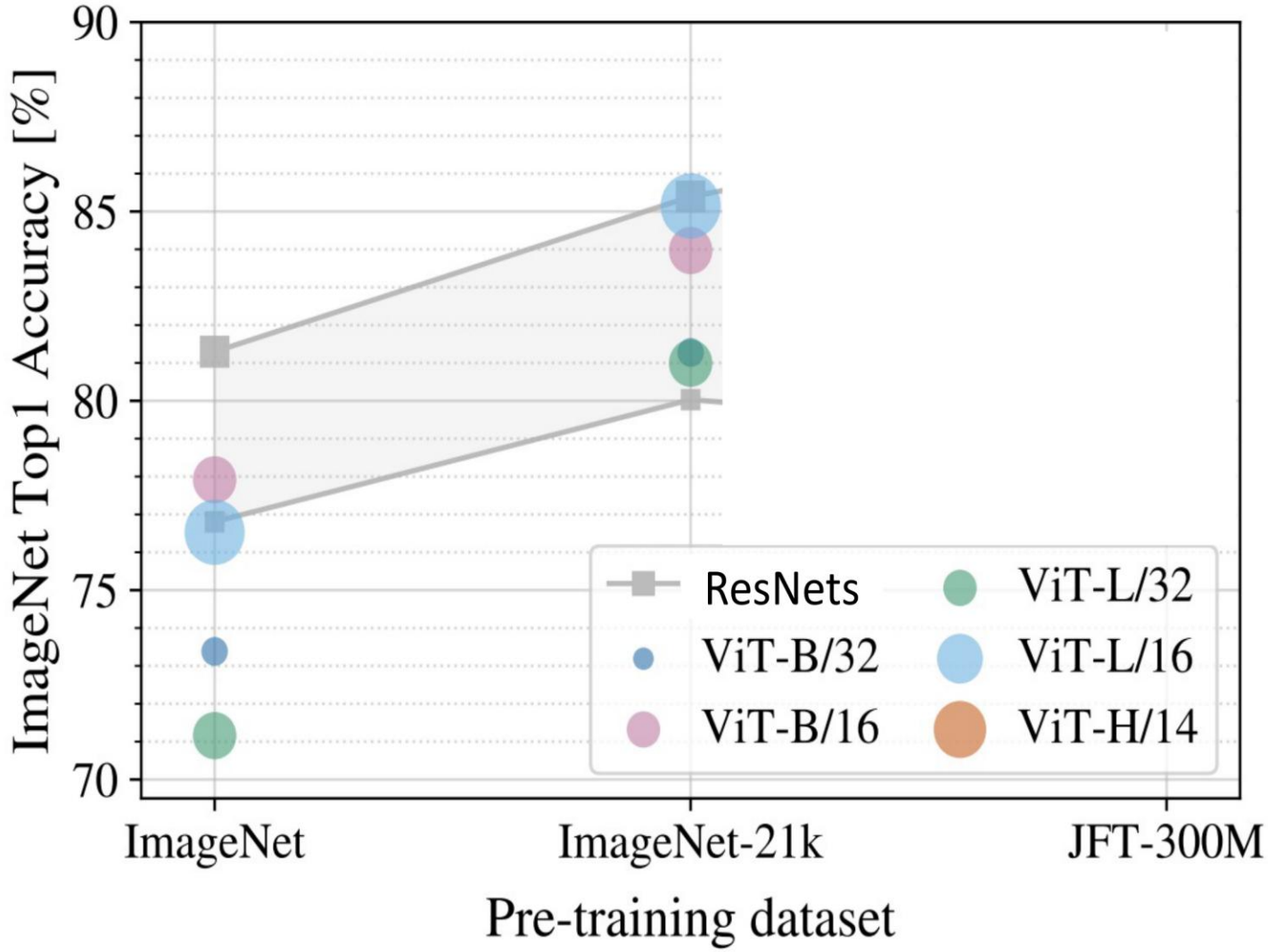
Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR 2021

Slide credit: Justin Johnson

Vision Transformer (ViT) vs ResNets

ImageNet-21k has 14M images with 21k categories

If you pretrain on ImageNet-21k and fine-tune on ImageNet, ViT does better: big ViTs match big ResNets



B = Base
L = Large
H = Huge

/32, /16, /14 is patch size; smaller patch size is a bigger model (more patches)

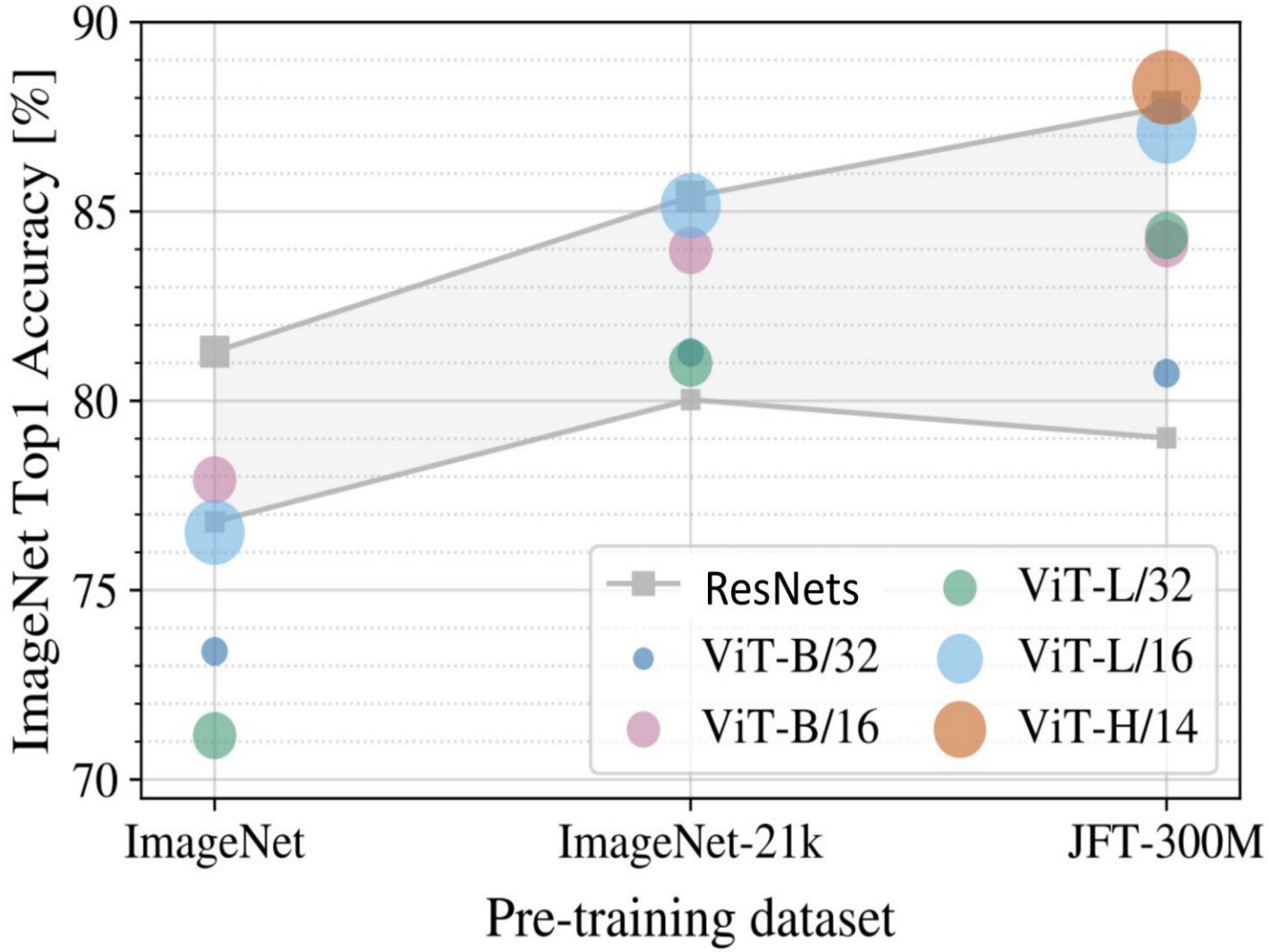
Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR 2021

Slide credit: Justin Johnson

Vision Transformer (ViT) vs ResNets

JFT-300M is an internal Google dataset with 300M labeled images

If you pretrain on JFT and finetune on ImageNet, large ViTs outperform large ResNets



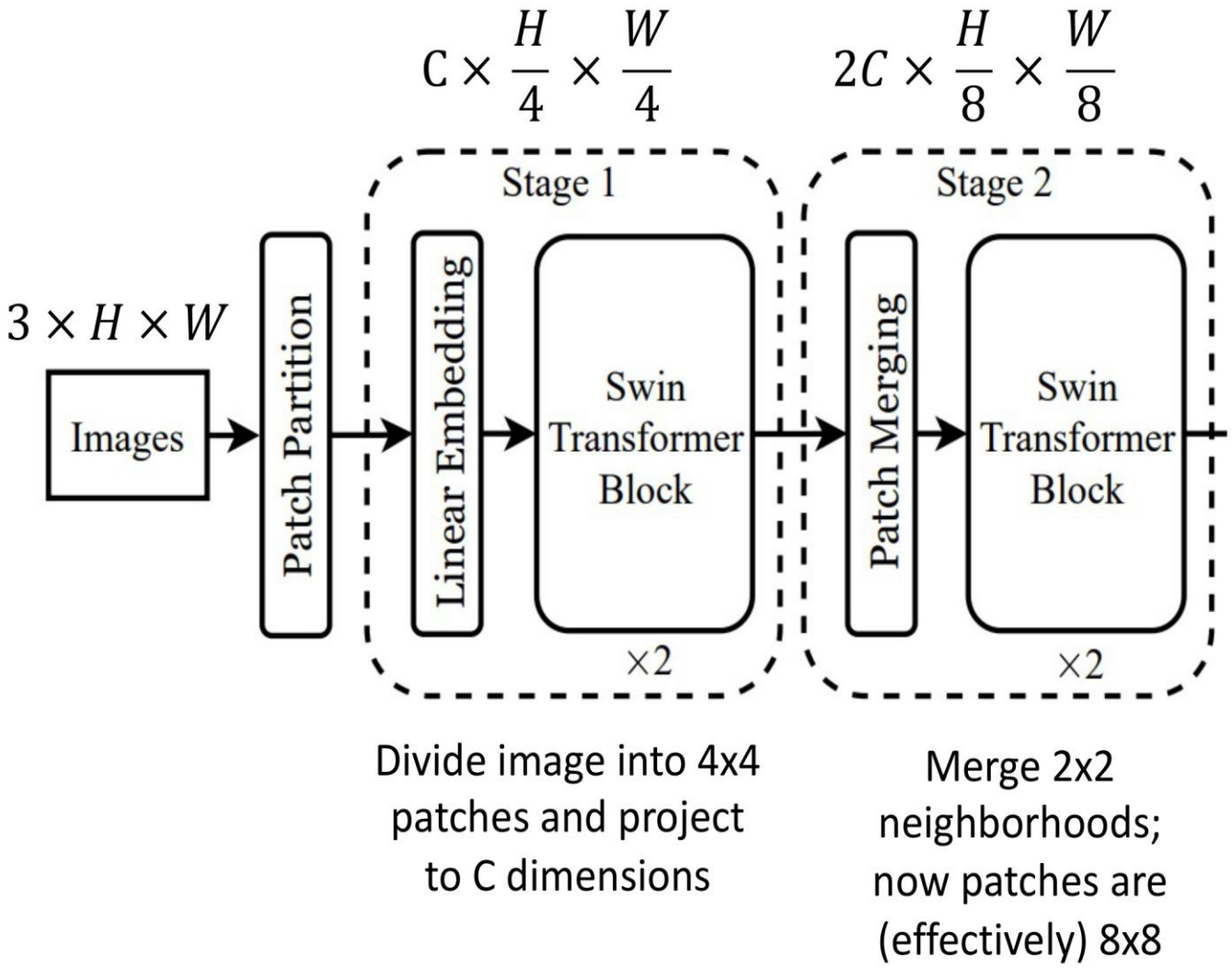
B = Base
L = Large
H = Huge

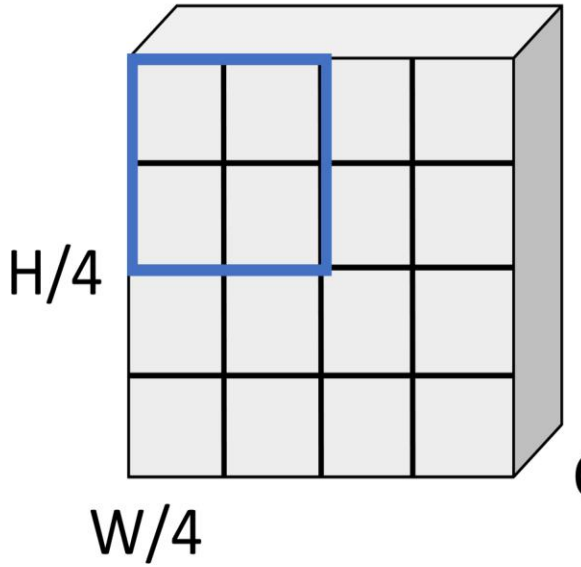
/32, /16, /14 is patch size; smaller patch size is a bigger model (more patches)

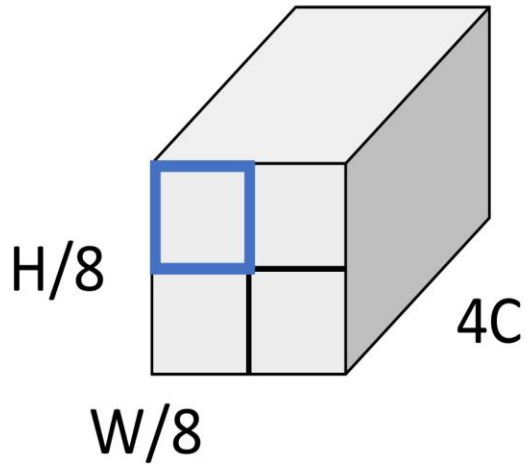
Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR 2021

Slide credit: Justin Johnson

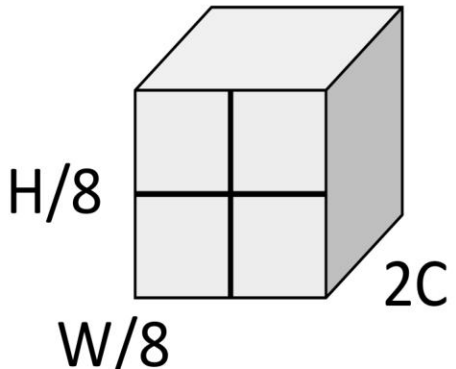
Hierarchical ViT: Swin Transformer







Concatenate groups of 2x2 features

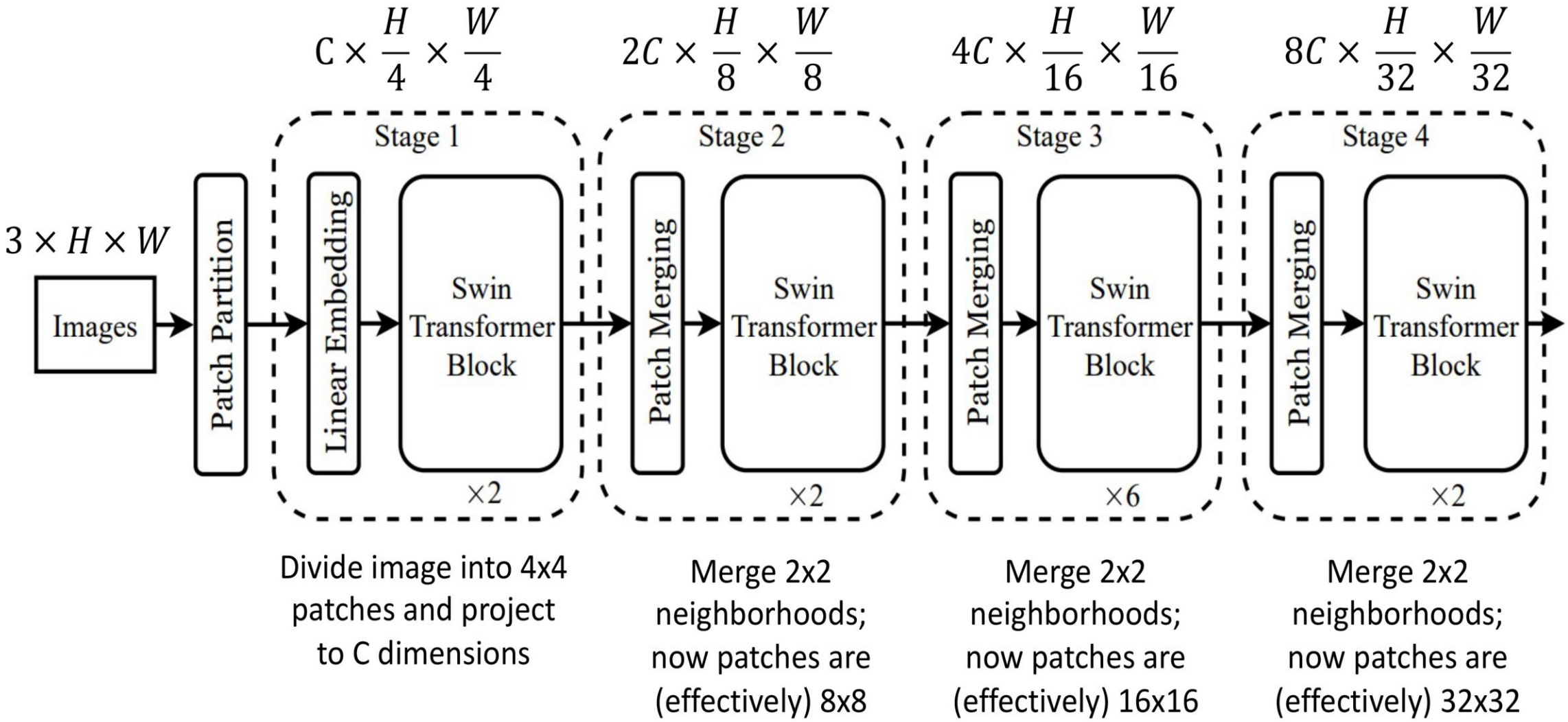


Linear projection from 4C to 2C channels (1x1 conv)

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

Slide credit: Justin Johnson

Hierarchical ViT: Swin Transformer

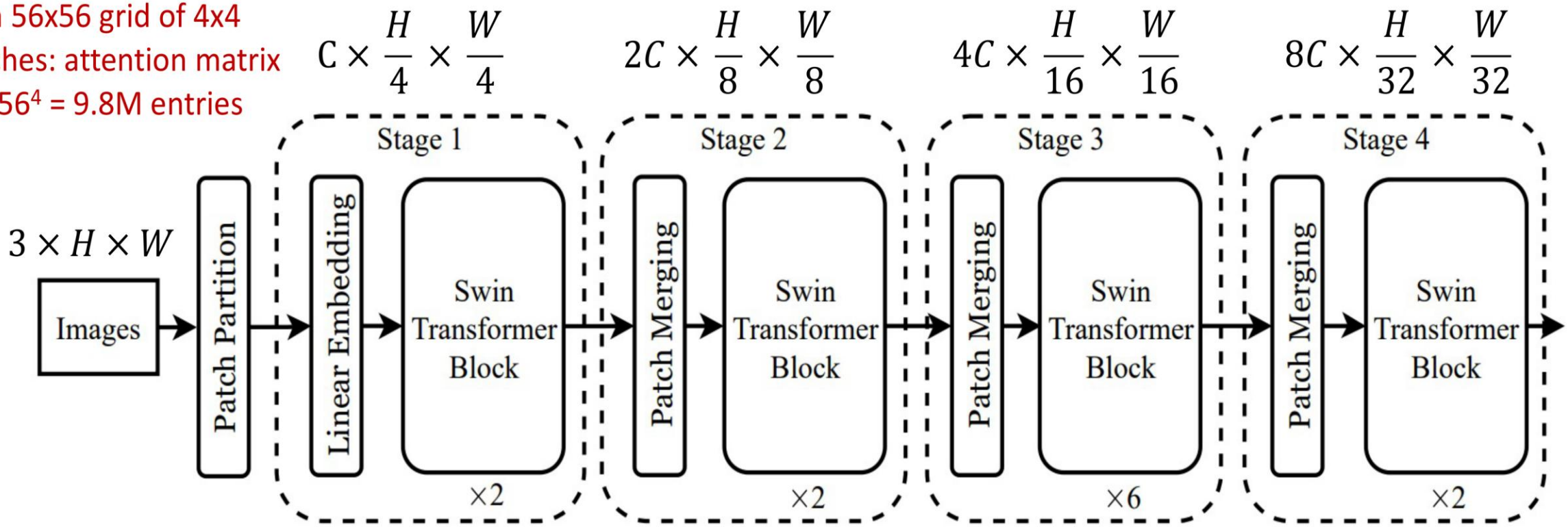


Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", CVPR 2021

Slide credit: Justin Johnson

Hierarchical ViT: Swin Transformer

Problem: 224x224 image with 56x56 grid of 4x4 patches: attention matrix has 56^4 = 9.8M entries



Solution: don't use full attention, instead use attention over patches

Divide image into 4x4 patches and project to C dimensions

Merge 2x2 neighborhoods; now patches are (effectively) 8x8

Merge 2x2 neighborhoods; now patches are (effectively) 16x16

Merge 2x2 neighborhoods; now patches are (effectively) 32x32

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

Slide credit: Justin Johnson

Swin Transformer: Window Attention

With H x W grid of **tokens**, each attention matrix is H^2W^2 – **quadratic** in image size

Rather than allowing each **token** to attend to all other tokens, instead divide into **windows** of M x M tokens (here M=4); only compute attention within each window

Total size of all attention matrices is now:
$M^4(H/M)(W/M) = M^2HW$

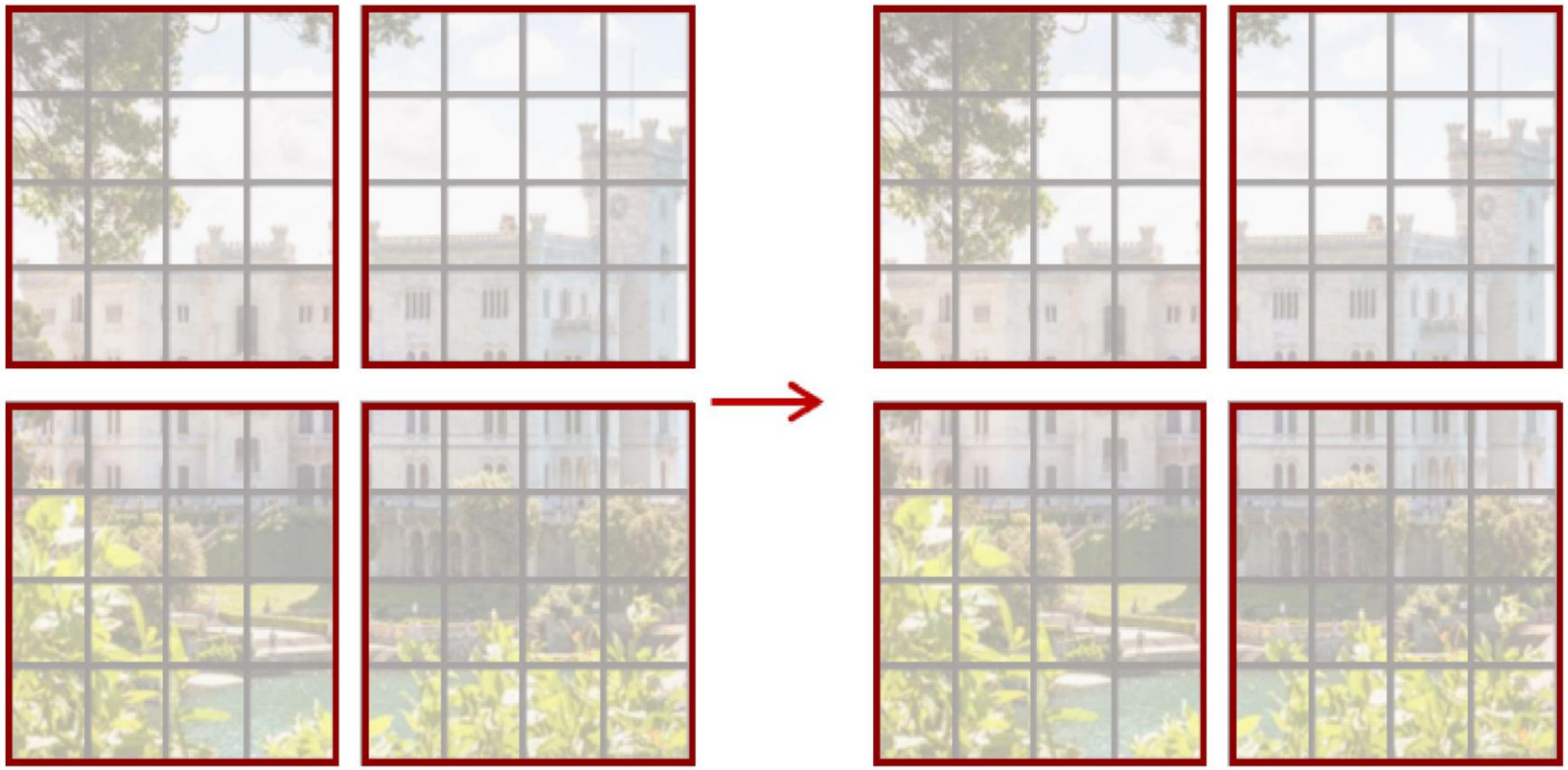
Linear in image size for fixed M!
Swin uses M=7 throughout the network

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

Slide credit: Justin Johnson

Swin Transformer: Window Attention

Problem: tokens only interact with other tokens within the same window; no communication across windows

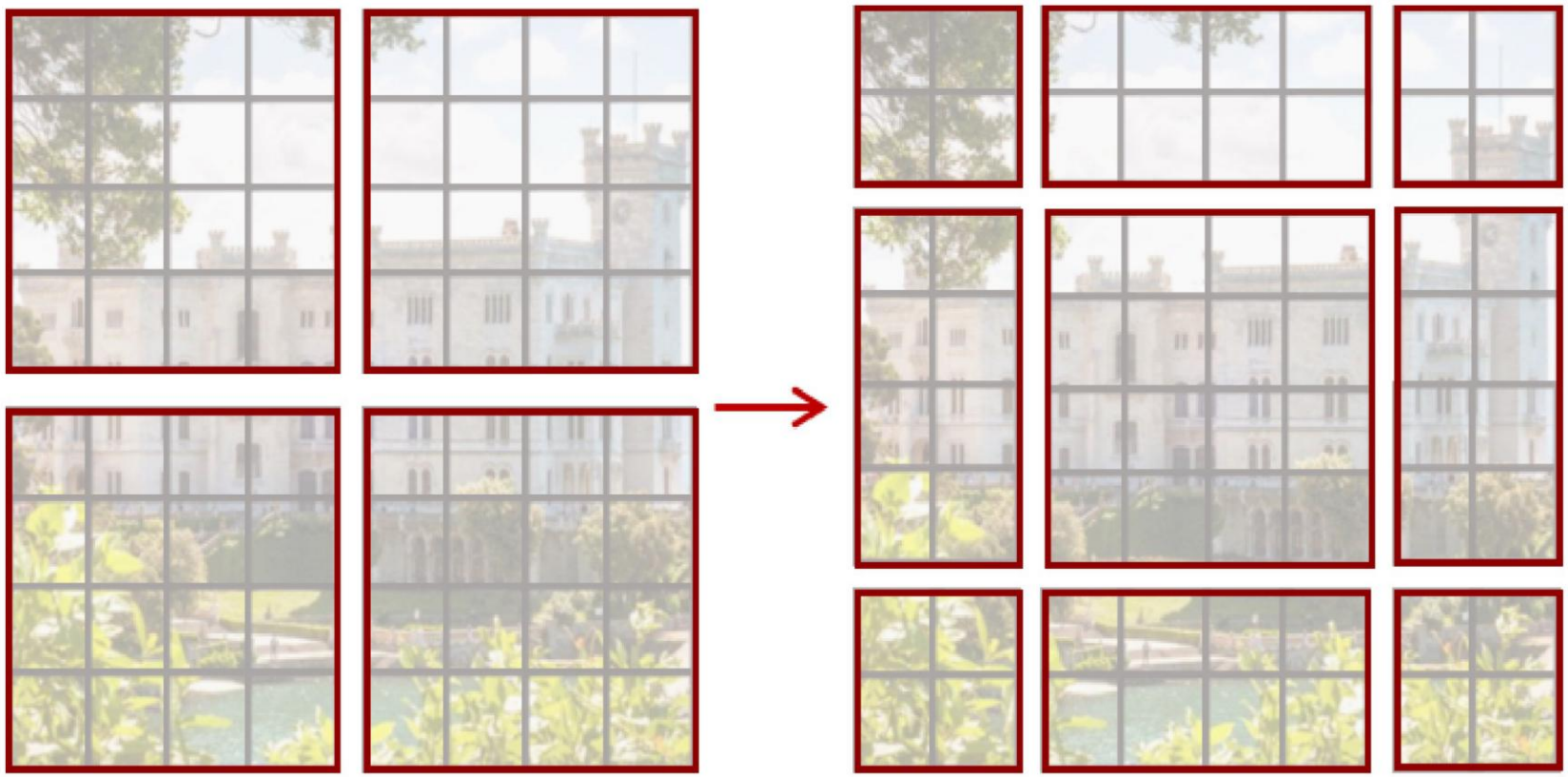


Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", CVPR 2021

Slide credit: Justin Johnson

Swin Transformer: Shifted Window Attention

Solution: Alternate between normal windows and shifted windows in successive Transformer blocks



Block L: Normal windows

Block L+1: Shifted Windows

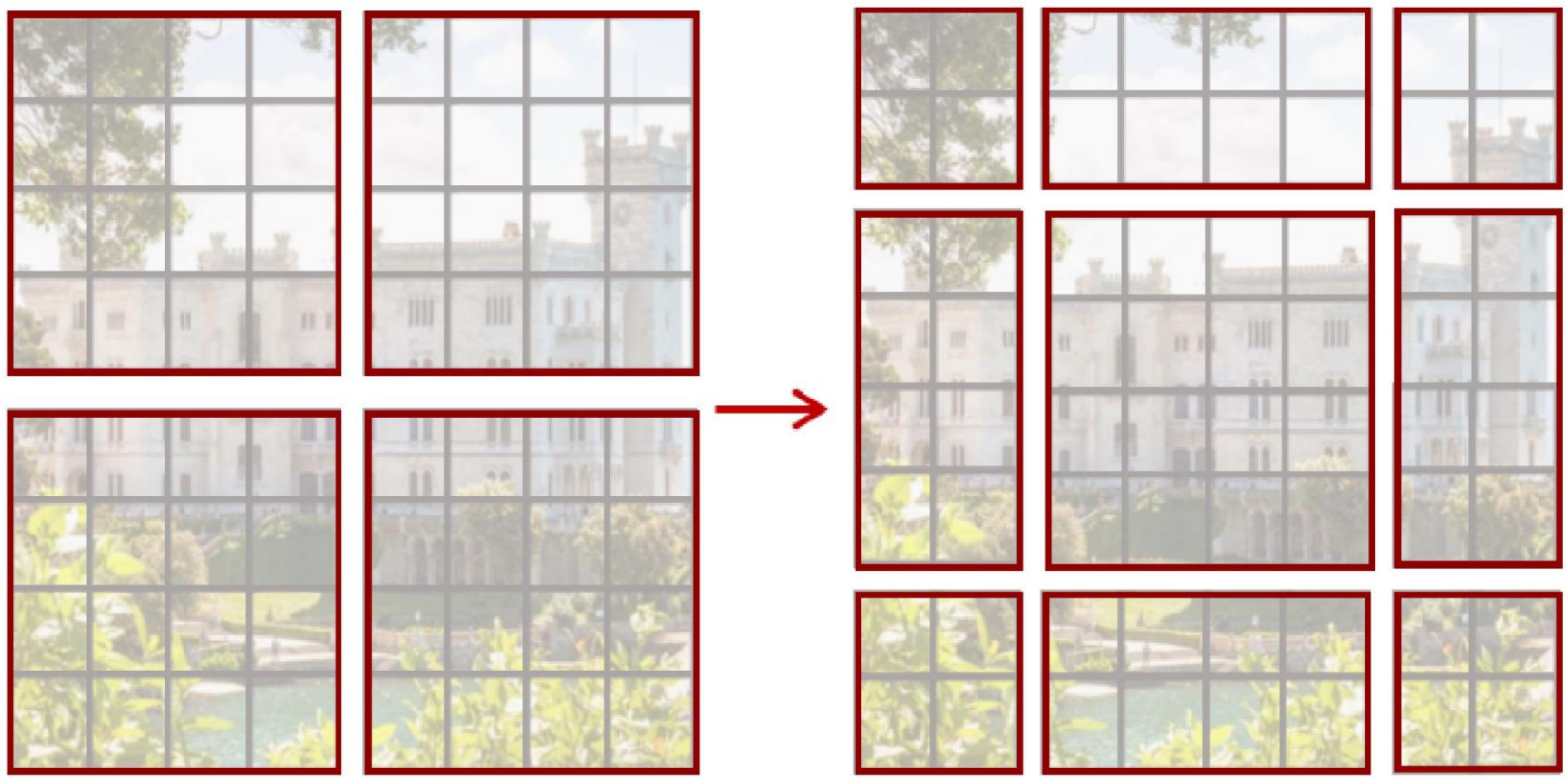
Ugly detail: Non-square windows at edges and corners

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

Slide credit: Justin Johnson

Swin Transformer: Shifted Window Attention

Solution: Alternate between normal windows and shifted windows in successive Transformer blocks



Block L: Normal windows

Block L+1: Shifted Windows

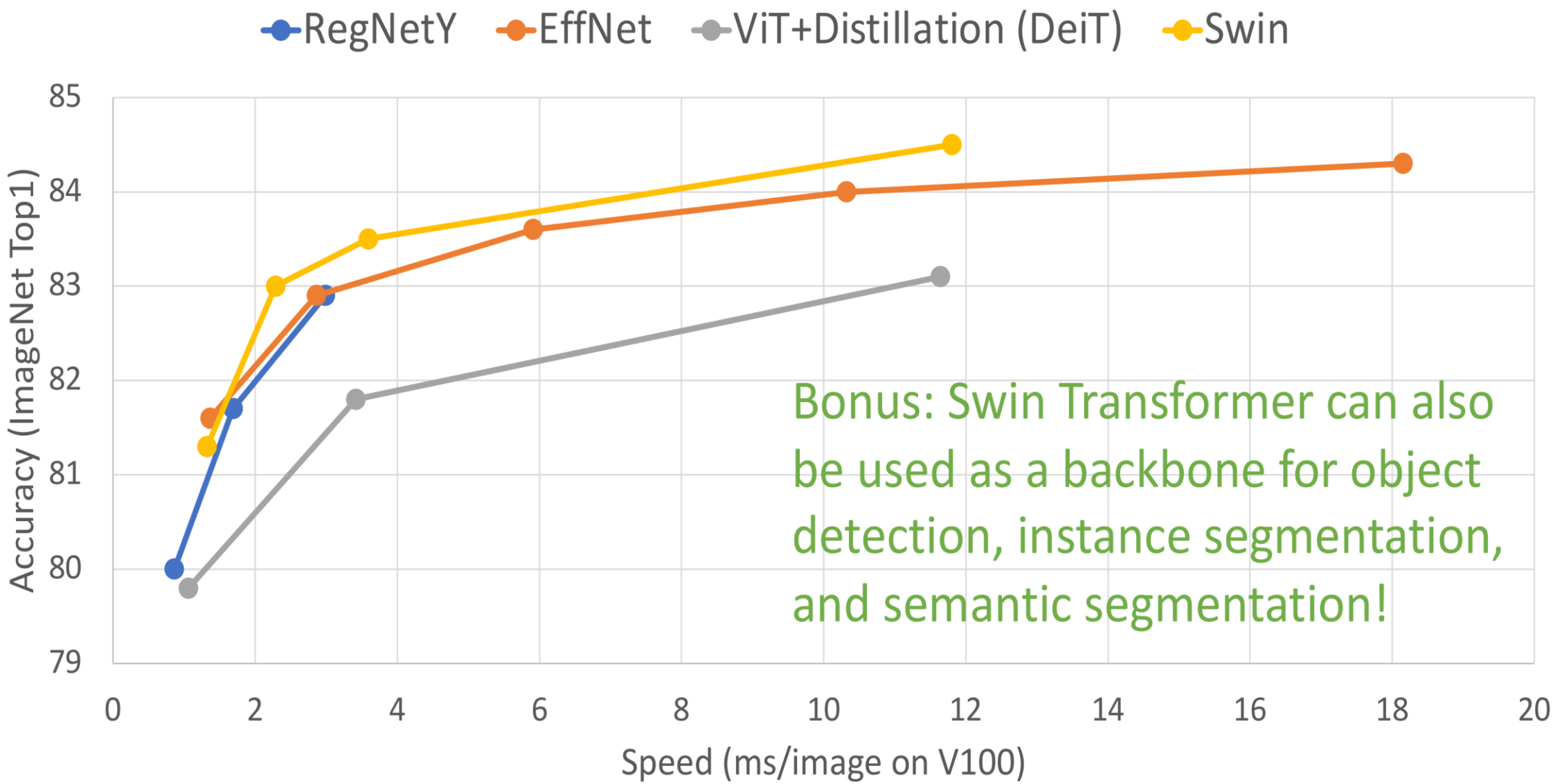
Detail: Relative Positional Bias

ViT adds positional embedding to input tokens, encodes *absolute position* of each token in the image

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

Slide credit: Justin Johnson

Swin Transformer: Speed vs Accuracy



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

Slide credit: Justin Johnson

Part 3

Vision Encoders as Foundation Models

Patch tokens · scaling · self-supervision · dense visual features

Seminal anchor: An Image is Worth 16×16 Words — ViT (Dosovitskiy et al., ICLR 2021)
Frontier anchor: DINOv3 (Siméoni et al., 2025)

Where we are, and where we are going

- Part 1 — Attention, self-attention, the transformer block ✓
- Part 2 — ViT: an image is a sequence of patch tokens; Swin: hierarchy + windowed attention ✓
- **Part 3 — From supervised ViTs to vision foundation models**
  - Where does supervision come from when you have no labels?
  - MAE, DINO, DINOv2, registers → DINOv3 (today's frontier anchor)
  - Global features vs. dense features — and why they come apart
- **Part 4 — What a VLM actually needs from its vision encoder**
  - Encoder → connector → LLM; token budgets, resolution, frozen vs. tuned
- **Next lecture (L4) — Vision–language alignment: CLIP → SigLIP 2**

ViT Supervision

- **ViT scaling result: ViT only beats ResNets when pretrained on JFT-300M**
  - 300M labeled images, internal to Google; ImageNet-1k alone is not enough
  - "Transformers have less inductive bias" → they must get structure from data instead
- **Labels are the bottleneck**
  - Expensive, and a closed vocabulary: you can only learn what the label set names
  - Annotation does not scale to billions of images
- **Two ways to get supervision for free**
  - From the pixels themselves → self-supervised learning (DINO family) (today)
  - From text that already accompanies web images → CLIP / SigLIP (next lecture)
- **Both produce a frozen encoder that turns an image into tokens**

Dosovitskiy et al., "An Image is Worth 16x16 Words", ICLR 2021.

What do we want from a vision foundation model?

- **1. General purpose: one frozen backbone, many tasks, no per-task finetuning**
  - "Foundation model" in vision = features good enough that a linear head is competitive
- **2. Global features: one vector per image**
  - Classification, retrieval, image–text matching. This is what a [CLS] token gives you.
- **3. Dense features: one vector per patch**
  - Segmentation, depth, correspondence, tracking, grounding
- **4. Scales: gets better with more data and compute, without more annotation**
- **5. A token interface a language model can consume**
  - A grid of h×w patch embeddings is already a sequence
- **Theme for the rest of the lecture: global quality and dense quality are surprisingly decoupled**
  - **A model can get better at ImageNet while getting worse at segmentation.**

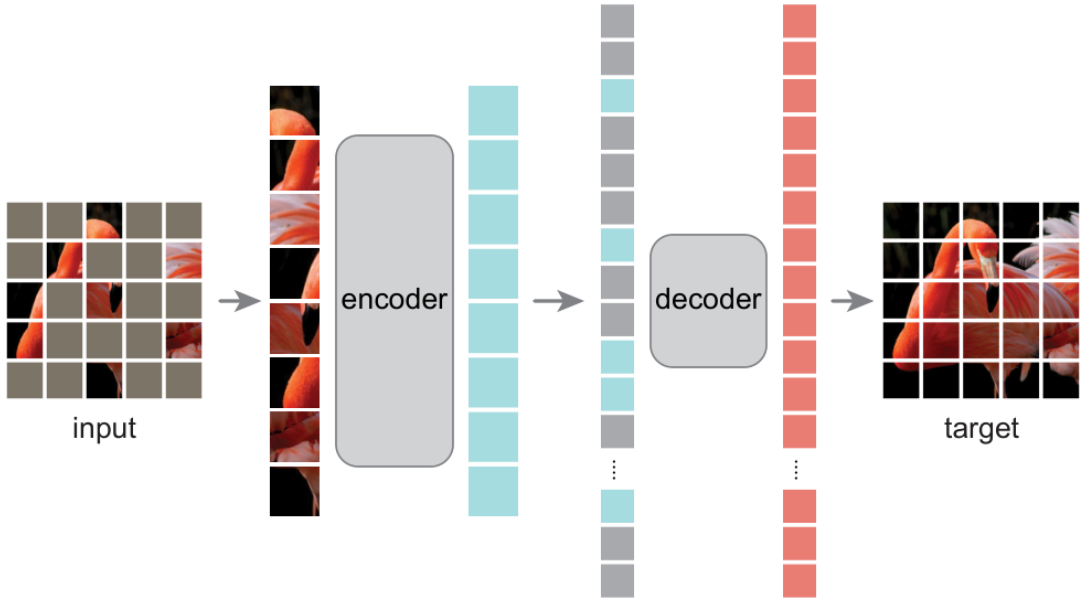
Three families of self-supervised objectives

Family	Idea	Examples	Strong at	Weak at
Contrastive / joint embedding	Two augmented views of the same image attract; different images repel	SimCLR, MoCo	Global, linear-probe semantics	Needs negatives & large batches; dense features mediocre
Masked image modeling (MIM)	Hide most patches, reconstruct the missing pixels / tokens	BEiT, MAE, iBOT	Scalable pretraining; strong after finetuning	Frozen / linear-probe features are weak
Self-distillation (no negatives)	Student matches the output distribution of an EMA teacher across views	BYOL, DINO, DINOv2/v3	Frozen features; emergent object structure	Collapse must be prevented explicitly

DINOv2 / DINOv3 deliberately combine the last two: an image-level self-distillation loss for global semantics + a patch-level masked loss for dense features.

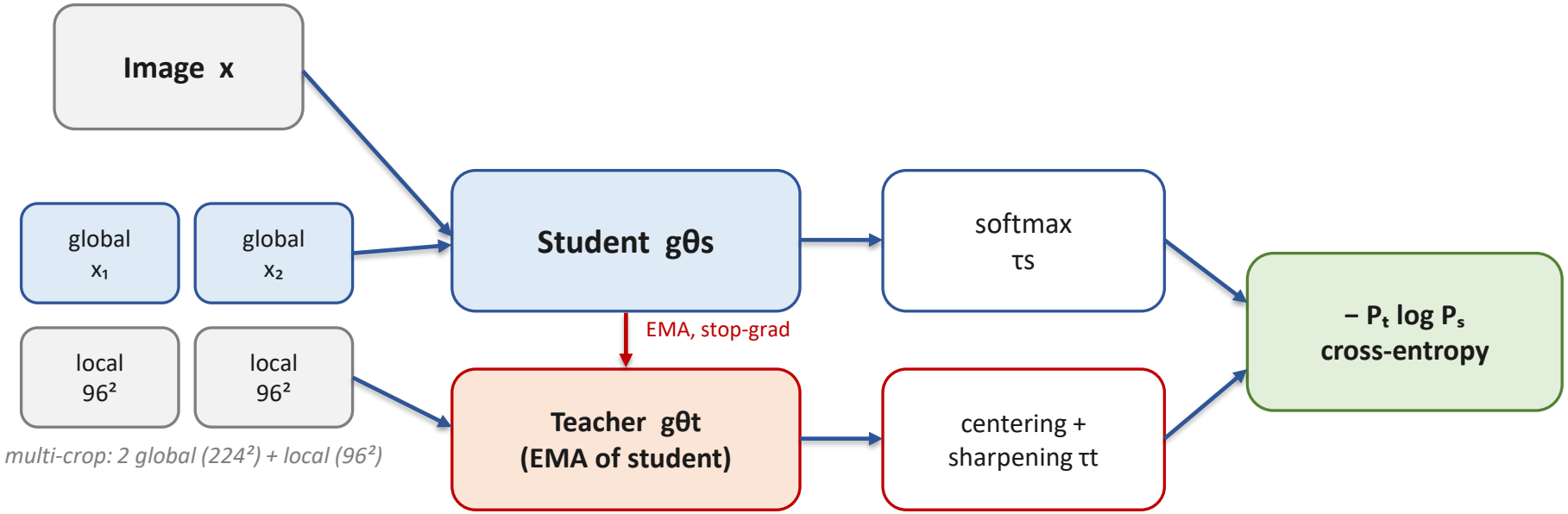
Masked image modeling: MAE

- **Mask a very high fraction of patches (75%)**
  - Images are redundant: Unlike text, a low mask rate is nearly trivial
- **Asymmetric encoder–decoder**
  - Encoder sees only the ~25% visible patches → large models train cheaply
  - Small decoder reconstructs raw pixels, then is thrown away
- **Result: excellent pretraining for finetuning**
- **But: frozen / linear-probe features are noticeably weaker**
  - Reconstruction rewards low-level detail, not semantic abstraction
  - For a frozen VLM backbone we need discriminative features → DINO



He et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022 (figure from the paper).

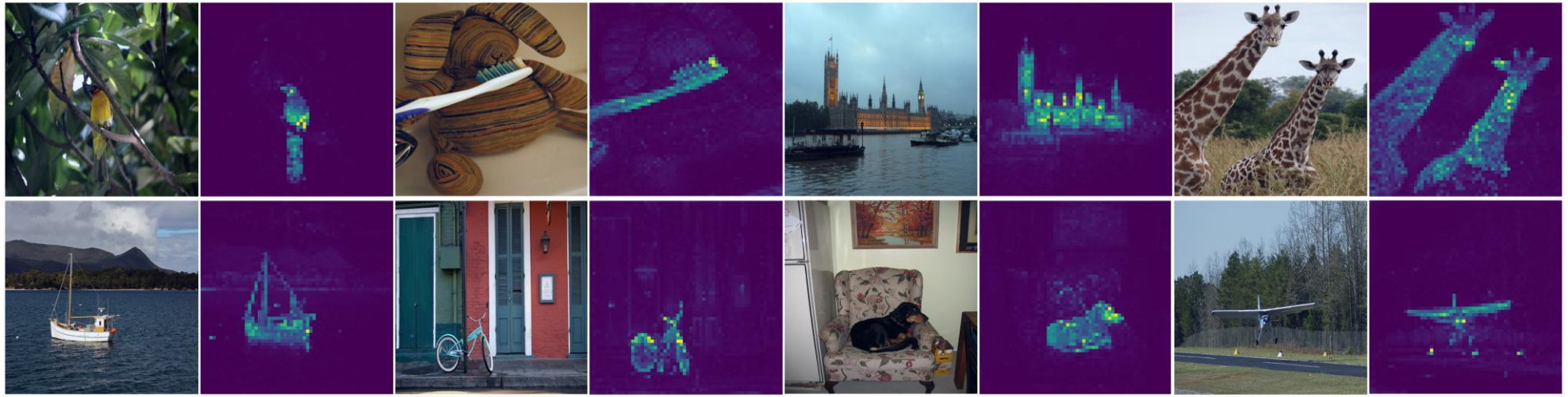
DINO: self-distillation with no labels



- **No labels, no negative pairs, no contrastive queue**
- Teacher = exponential moving average of the student; gradients never flow into it
- **All crops → student; only global crops → teacher**
  - "Local-to-global": a 96^2 crop must predict what the teacher sees in the whole image
- **Collapse is prevented by two cheap tricks**
  - Centering: subtract a running mean from teacher logits (stops one dimension dominating)
  - Sharpening: low teacher temperature τt (stops the uniform-output solution)

Caron et al., "Emerging Properties in Self-Supervised Vision Transformers" (DINO), ICCV 2021.

DINO: Segments for free



- **Self-attention of the [CLS] token in a ViT-S/8 trained with DINO**
  - **Different heads segment different objects. No masks, no labels, no segmentation loss.**
- k-NN classifier directly on frozen features: 78.3% top-1 on ImageNet (no finetuning, not even a linear layer)
- Frozen features transfer to video object tracking (DAVIS), copy detection and retrieval
- The same emergent structure does not appear in supervised ViTs of the same size
  - A property of the objective, not the architecture

Caron et al., DINO, ICCV 2021 (figure from the paper).

DINOv2: turning DINO into a foundation model

- **Data is the first contribution: LVD-142M**
  - Automatic curation pipeline: embed ~1.2B uncurated web images, retrieve nearest neighbours of a curated seed set, deduplicate, rebalance
  - No labels and no text
- **Objective: combine image-level and patch-level self-distillation**
  - L_DINO (global, [CLS]-level) + L_iBOT (masked patch-level) + KoLeo (spreads features in the batch) + a short high-resolution phase at the end
  - Untied heads for the two losses; Sinkhorn-Knopp centering
- **Scale then distill**
  - Train ViT-g/14 (1.1B) once, distill into ViT-S / B / L

- **Payoff: frozen features + a linear head beat weakly-supervised (OpenCLIP) features on dense tasks without any finetuning**

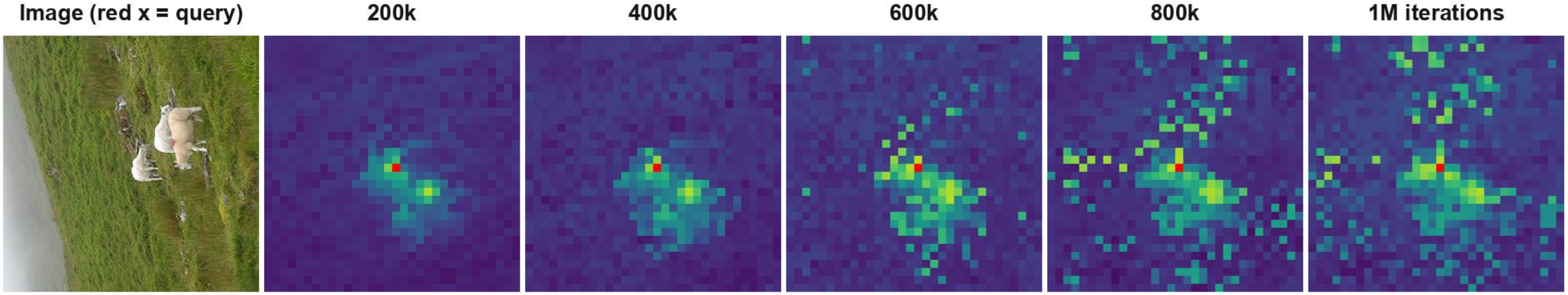
Oquab et al., "DINOv2: Learning Robust Visual Features without Supervision", TMLR 2024.

DINOv3: scaling self-supervision to a frontier model

- **Scale, with no labels and no text at all**
  - 7B-parameter ViT trained on LVD-1689M (1.689B curated web images)
  - 40 blocks, embedding dim 4096, patch size 16 (DINOv2 g/14: 1.1B, dim 1536, patch 14)
  - Axial RoPE with position jittering (instead of learned position embeddings) + 4 registers
  - Constant hyper-parameter schedules, 1M iterations — no cosine schedule to "land"
- **Three contributions**
  - 1. Data + model scaling of the DINOv2 recipe
  - 2. Gram anchoring
    - Fixes dense-feature degradation over long training ← the key idea
  - 3. Post-hoc flexibility: high-resolution adaptation, distillation into a model family, and text alignment added after the fact
- **Result: a single frozen backbone that is state of the art on dense tasks and competitive on global**

Siméoni et al., "DINOv3", arXiv:2508.10104, 2025.

The problem: dense features degrade during long training



- **Each map: cosine similarity between the patch marked with a red cross and every other patch, at increasing training iterations**
- **Global metrics (ImageNet linear probe) keep improving the whole time**
  - **Patch-level maps get noisier and less localized**
- Global and dense performance are decoupled: the global objective progressively dominates, and the patch-level iBOT loss cannot hold the balance
- This is why “just train longer / bigger” had stopped paying off for dense tasks

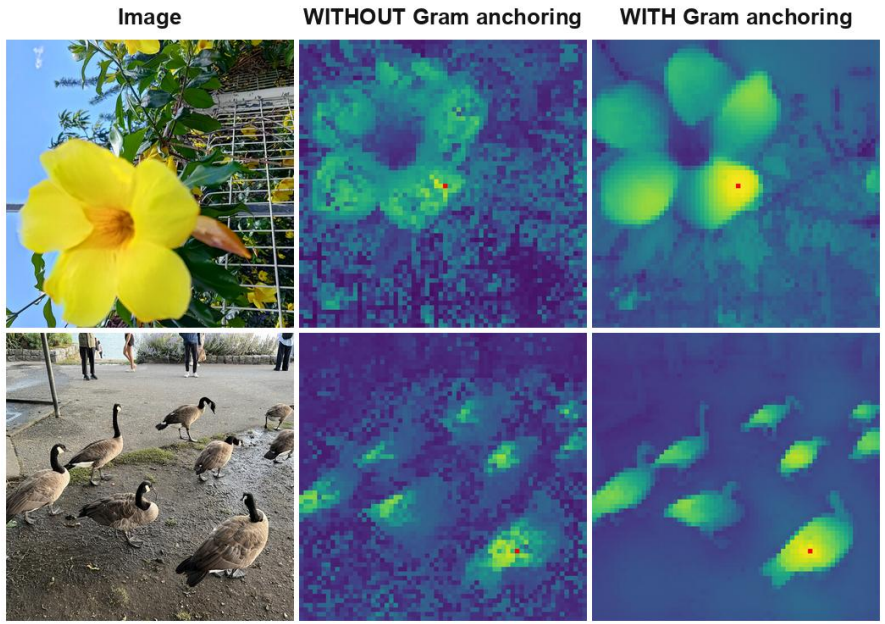
Siméoni et al., DINOv3, 2025, Fig. 6 (panels from the paper).

The fix: Gram anchoring

- **Idea: regularize the structure of patch-to-patch similarities**
- Let X_s, X_G be the P×d matrices of L2-normalized patch features of the student and of a Gram teacher — an earlier EMA checkpoint whose dense maps were still clean

$$L\_Gram = \| X_s X_s^T - X\_G X\_G^T \|^2\_F$$

- **Because the loss only constrains the Gram matrix, individual features are free to keep moving**
- Applied as a refinement stage after 1M iterations; the Gram teacher is refreshed every 10k iterations
  - Remarkably, it repairs already-degraded maps rather than merely preventing the damage
- **L_Ref = w_D·L_DINO + L_iBOT + w_DK·L_DKoleo + w_Gram·L_Gram**
- **ADE20k dense linear probe: 50.3 → 55.7 mIoU; NYU depth RMSE 0.307 → 0.281, with ImageNet linear unchanged**



Siméoni et al., DINOv3, 2025, §4.2 and Fig. 10 (figure panels from the paper).

Post-training: one model, many deployment shapes

- **High-resolution adaptation**
  - Trained at 256×256; a short adaptation phase makes features stable from 512 up to 4096×4096 inputs
- **Distillation into a family**
  - One 7B teacher → parallel multi-student distillation → ViT-S (21M), S+ (29M), B (86M), L (300M), H+ (840M), plus ConvNeXt T/S/B/L for constrained hardware
  - Same teacher for all students: the family is consistent, not a set of unrelated models
- **Text alignment, added afterwards (dino.txt)**
  - Freeze the vision backbone, train a text encoder against it → zero-shot classification and retrieval without retraining the visual features
  - Keeps the dense advantage: on zero-shot segmentation it roughly doubles CLIP/SigLIP-style models (ADE20k 24.7 vs. 10.8 mIoU for SigLIP 2 at ViT-L), while remaining a bit behind on retrieval
- **Language alignment can be later added onto a strong visual representation**

Siméoni et al., DINOv3, 2025, §5 and Tab. 16.

Frozen backbone, no finetuning: where each supervision wins

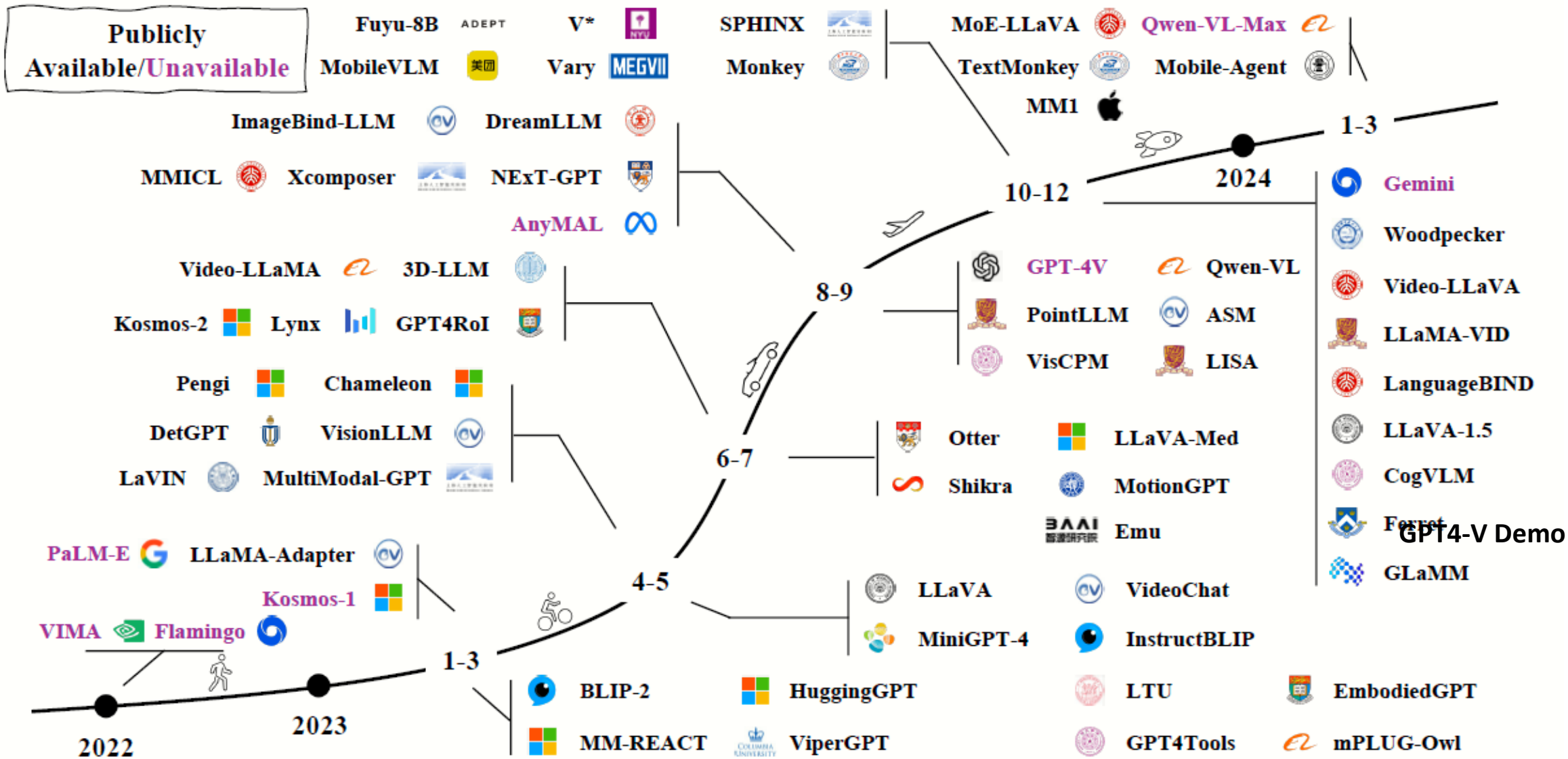
Backbone	Supervision	ADE20k mIoU ↑	NYUv2 depth RMSE ↓	DAVIS track J&F ↑	IN1k linear top-1 ↑
SigLIP 2 g/16	image–text (weak)	42.7	0.494	62.3	89.1
PE core G/14	image–text (weak)	38.9	0.590	53.1	89.3
DINOv2 g/14	self-supervised	49.5	0.372	73.6	87.3
PE spatial G/14	image–text + distill	49.3	0.362	74.5	—
DINOv3 7B/16	self-supervised	55.9	0.309	79.7	88.4

- **Language-supervised encoders lead on global, nameable semantics (ImageNet linear)**
- **Self-supervised encoders lead on everything dense and spatial**
  - **Segmentation, depth, tracking — often by a very wide margin**
- **DINOv3 extends the dense lead while nearly closing the global gap**

Siméoni et al., DINOv3, 2025, Tabs. 3, 5, 7. Linear probes on frozen features; DAVIS at medium resolution.

Introduction to Vision-Language Models

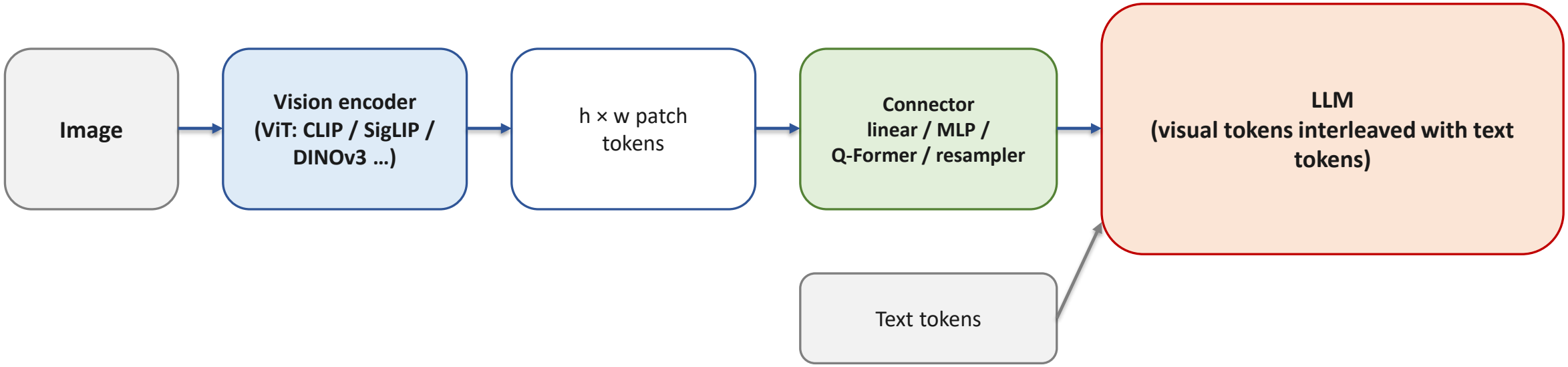
Introduction



Yin et al., A Survey on Multimodal Large Language Models


Pre-Trained
Vision Model
?
Pre-Trained
Language
Model

Anatomy of a VLM: the encoder is the interface



- **The encoder decides what is even representable downstream**
- Design questions we will keep returning to (L5 Flamingo/InternVL, L6 LLaVA/Qwen3-VL):
  - Which encoder, and trained with which supervision? ← today vs. next lecture
  - Which layer's features: Last layer is the most language-aligned, penultimate often more spatial
  - How many visual tokens can the LLM afford, and at what input resolution?
  - Freeze the encoder, or unfreeze it late in training?

The visual token budget

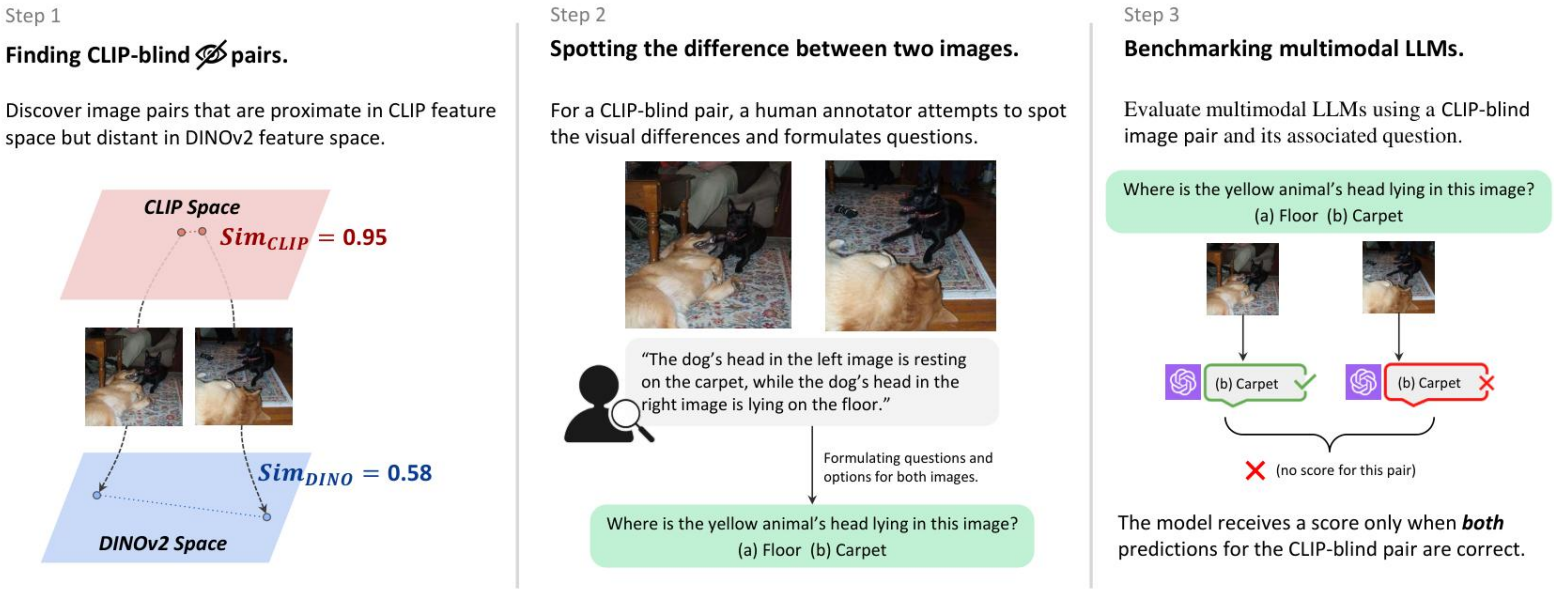
- **Patch size and resolution set the sequence length**
  - ViT-L/14 @ 224^2 → 256 tokens · @ 336^2 → 576 tokens · patch 16 @ 512^2 → 1024 tokens
  - Tiling a high-res image into 4 crops + a thumbnail → ~2.9k tokens for one image
  - A 4096^2 input at patch 16 would be 65k tokens
- **Solution: Compression**
  - Perceiver resampler (Flamingo) / Q-Former (BLIP-2): fixed number of learned queries
  - Pooling and pixel-shuffle: trade spatial detail for tokens
  - Native dynamic resolution (Qwen-VL) and tiling (InternVL, LLaVA-NeXT): spend tokens where the image needs them
- **Resolution is the single biggest lever for OCR, documents, charts, and fine-grained grounding**

Which encoder do real VLMs actually use?

System	Vision encoder	Supervision	Interface to the LLM
Flamingo (2022)	NFNet-F6, frozen	image–text	Perceiver resampler + gated cross-attention
BLIP-2 (2023)	CLIP / EVA ViT-g, frozen	image–text	Q-Former (32 learned queries)
LLaVA-1.5 (2023)	CLIP ViT-L/14 @336	image–text	MLP projector, tokens in-line
InternVL 3.5 (2025)	InternViT	image–text (+ distill)	Tiling + pixel-shuffle, encoder tuned
Qwen3-VL (2025)	Native-resolution ViT	image–text	Dynamic resolution, window attention
Dense / VLA pipelines	DINOv2, DINOv3	self-supervised	Frozen features for depth, seg, 3D, control

- **The default for VLM understanding is a language-supervised encoder: CLIP, then SigLIP**
- **Self-supervised encoders dominate wherever geometry and precise localization matter, and are increasingly mixed in alongside CLIP-style features**

Choice of Encoder



- **"Eyes Wide Shut?" finds CLIP-blind pairs: images that CLIP embeds almost identically (cos > 0.95) but DINOv2 embeds far apart (cos < 0.6)**
- **Ask a simple visual question about those pairs (MMVP) and strong VLMs**
  - **GPT-4V included — fall to near chance: the detail was never in the tokens the LLM received**
- **Mixture-of-Features: interleave DINOv2 tokens with CLIP tokens → visual grounding improves; too large a DINOv2 share degrades instruction following**
- **Language alignment buys nameability, self-supervision results in visual detail**

Tong et al., "Eyes Wide Shut? Exploring the Visual Shortcomings of Multimodal LLMs", CVPR 2024 (figure from the paper).

Next lecture (L4): Vision–Language Alignment

- **Seminal anchor: CLIP — Radford et al., 2021**
  - Contrastive image–text pretraining on 400M web pairs; zero-shot transfer by writing the classifier as text; the encoder almost every early VLM used
- **Frontier anchor: SigLIP 2 — Tschannen et al., 2025**
  - Sigmoid pairwise loss instead of softmax-over-batch; captioning and self-distillation objectives added; multilingual data; NaFlex variable aspect ratio and resolution