

CENG501 – Deep Learning

Week 11

Spring 2026

Sinan Kalkan

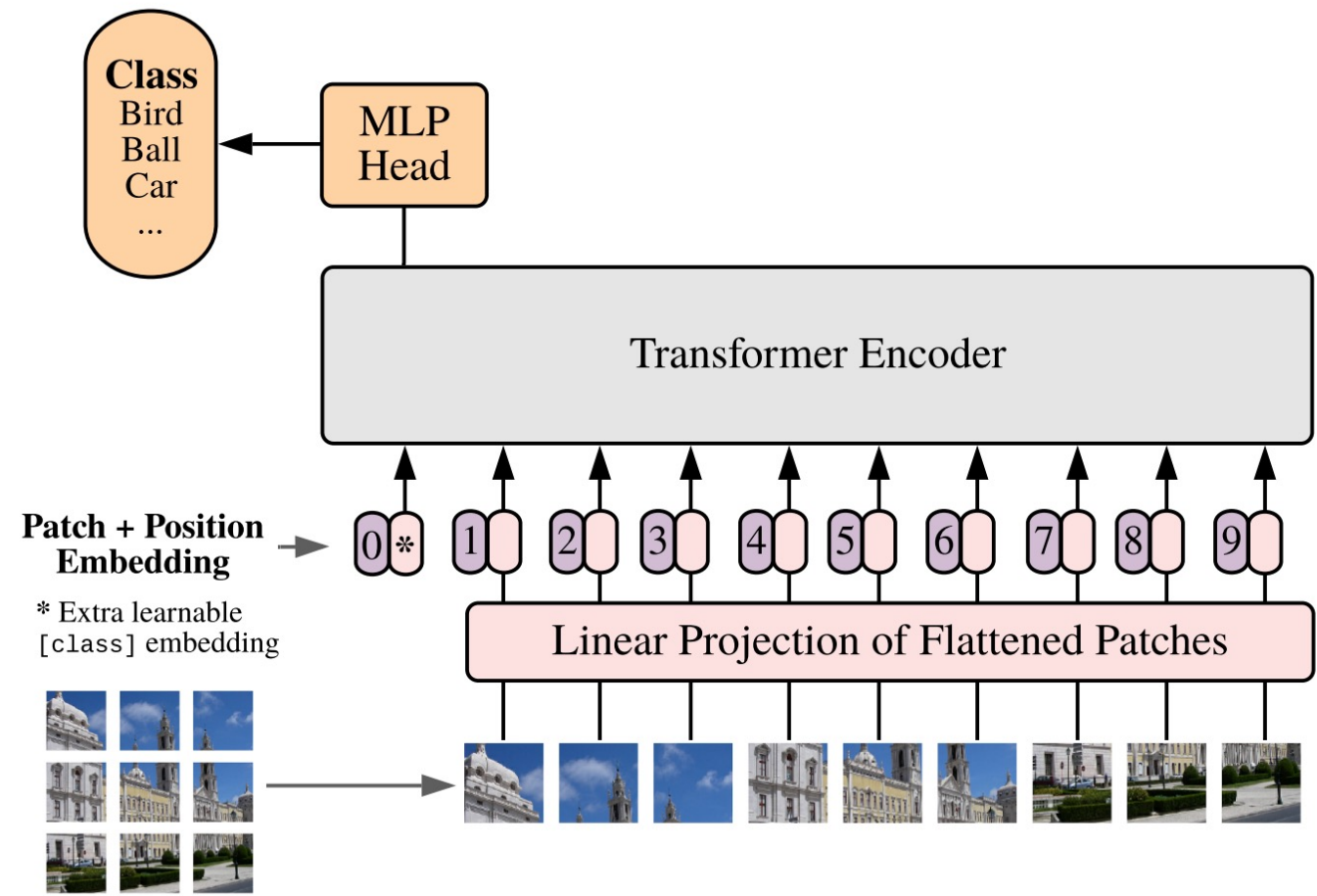
Dept. of Computer Engineering, METU

Previously on CENG501

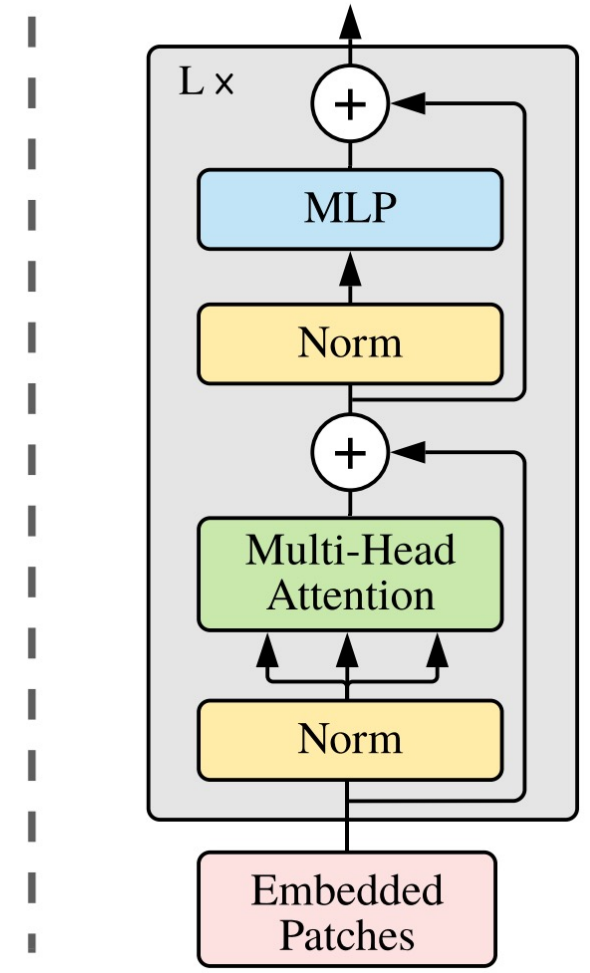
ViT: Architecture

Applied to classification tasks only!

Vision Transformer (ViT)



Transformer Encoder



ViT vs. CNNs

Previously on CENG501

- “When trained on mid-sized datasets such as ImageNet without strong regularization, these models yield modest accuracies of a few percentage points below ResNets of comparable size.

This seemingly discouraging outcome may be expected: Transformers lack some of the inductive biases inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data.”

- “However, the picture changes if the models are trained on larger datasets (14M-300M images). We find that large scale training trumps inductive bias.”

Swin Transformer v1

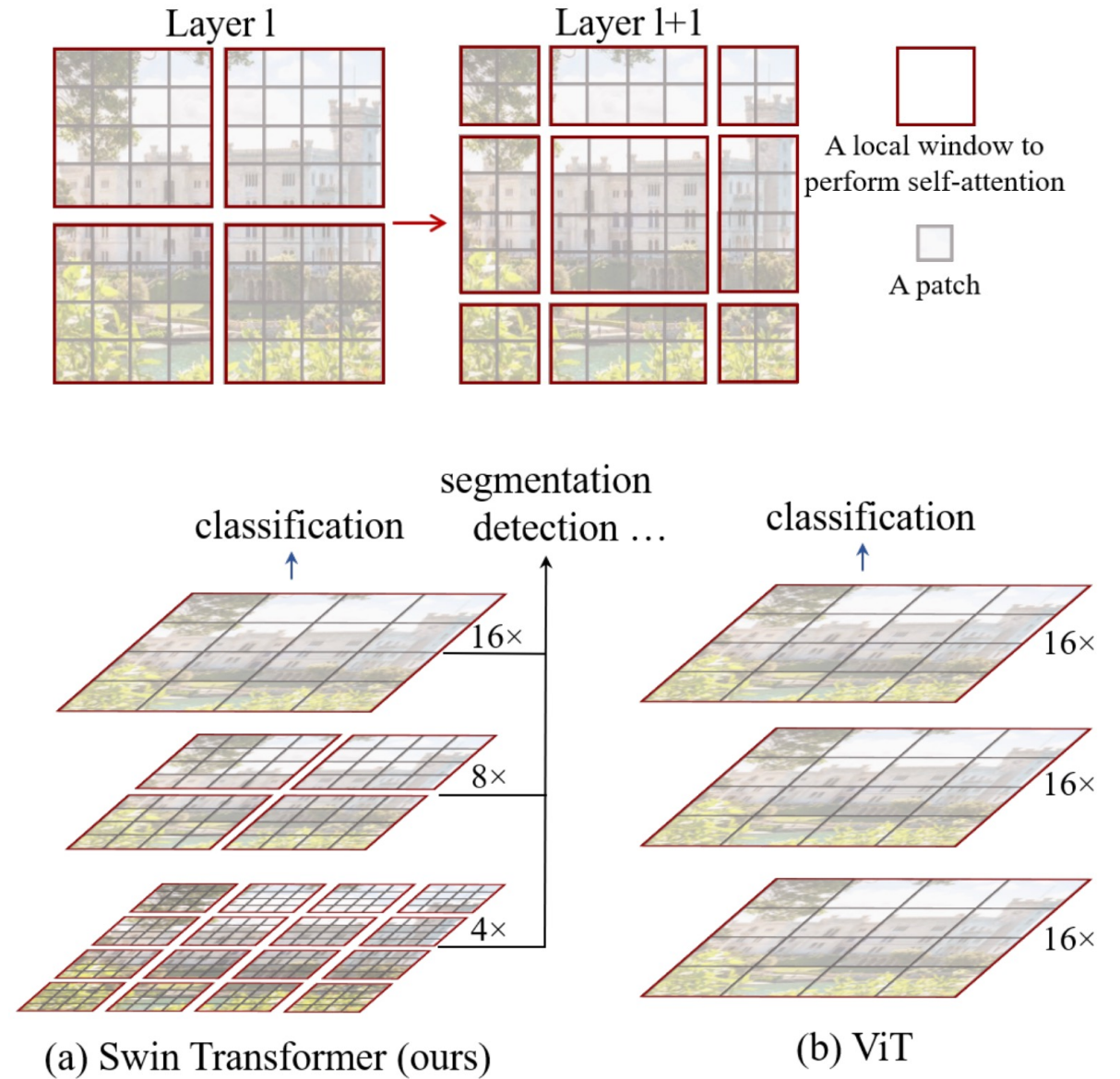
Previously on ENG501

- Motivation:

- ViT is promising but limited to classification
- Challenges in using Transformers:
 - large variations in scales of visual entities,
 - more pixels compared to words in text
- Existing Transformers use fixed token size across layers

- Contributions:

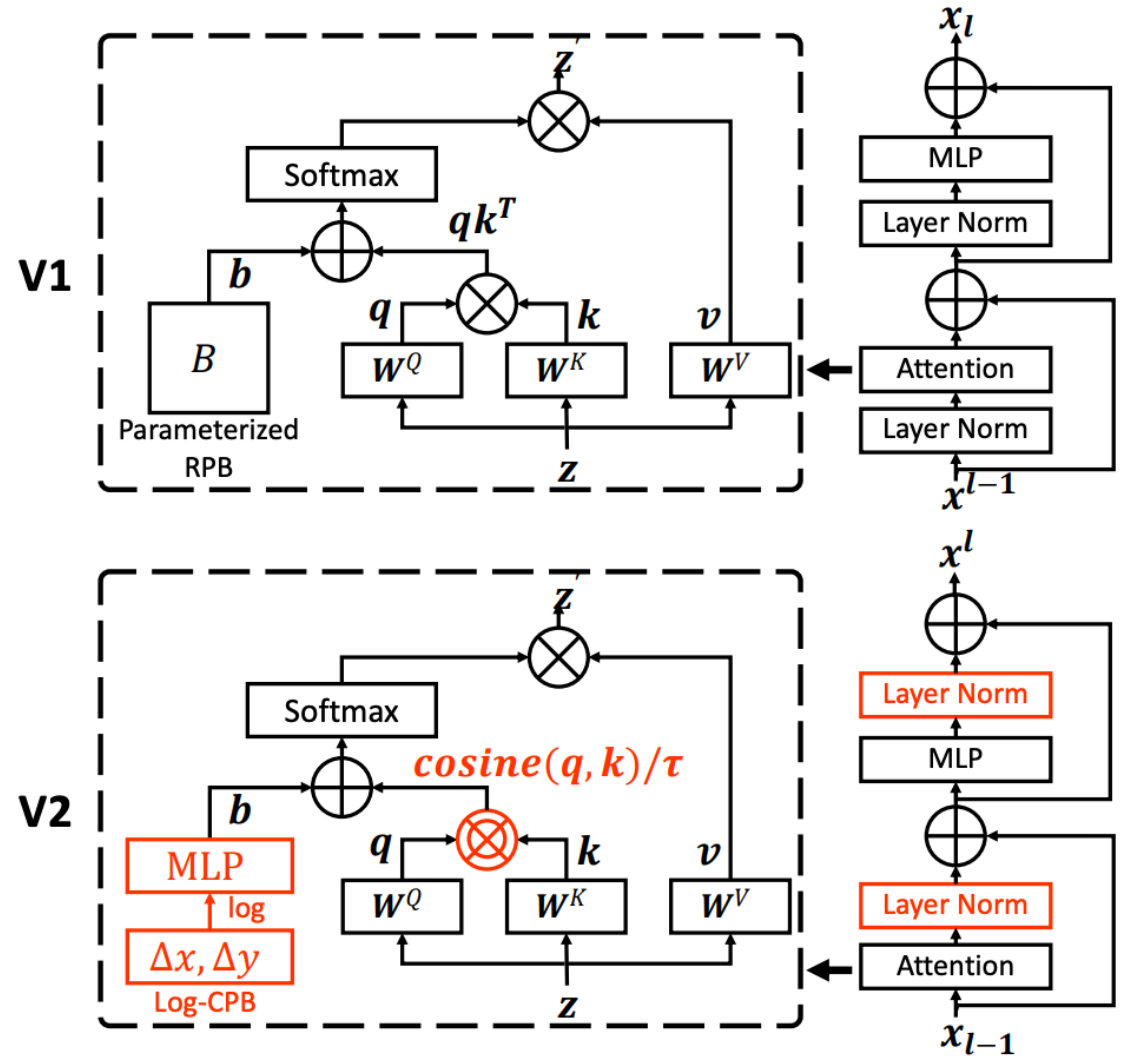
- Limit self-attention to non-overlapping windows while allowing cross-window attention
- Change token size across layers



Scaling up Transformer v2

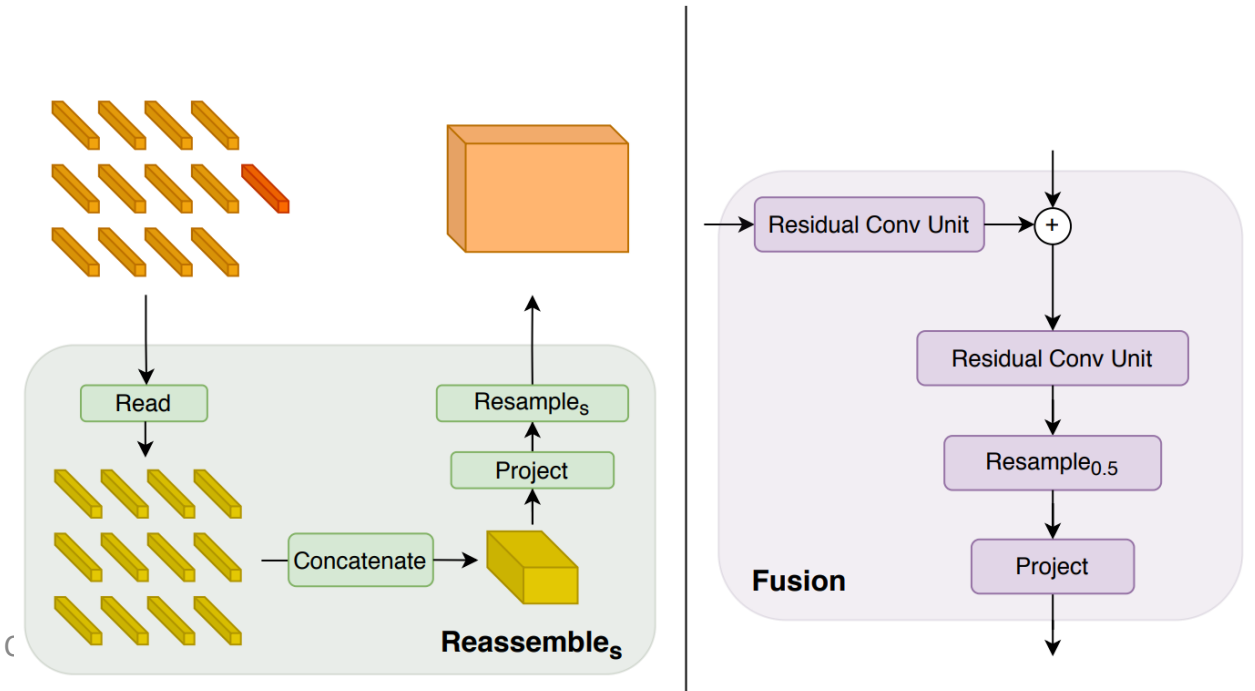
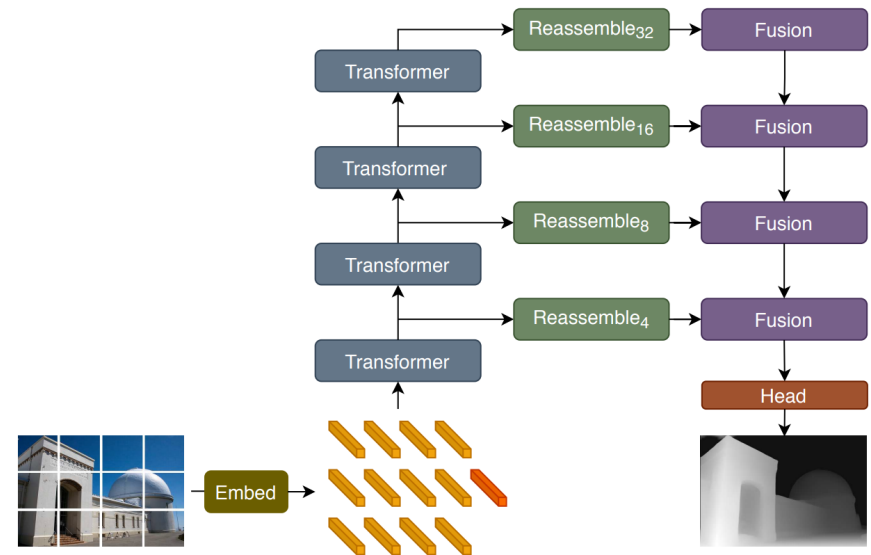
Scaling up model capacity

- Residual post normalization:
Prevents output to diverge
- Scaled cosine attention:
 - learnt attention maps of some blocks and heads are frequently dominated by a few pixel pairs, especially in the res-post-norm configuration => cosine yields values in a smaller range



Previously on **DP-ENG501**

- Extend a ViT for dense-prediction tasks
- Each Reassemble block is a light-weight module that combines tokens and resamples them (1x1 conv)
- Fusion combines features from different scales



Previously on **ViT Adapter**

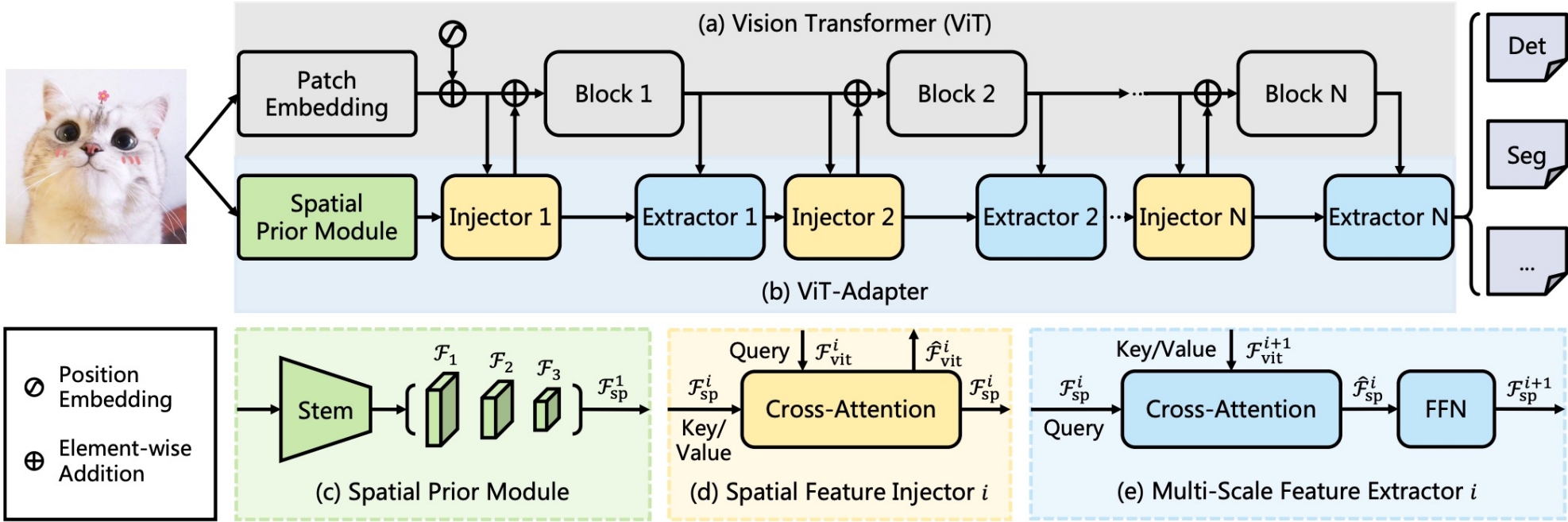
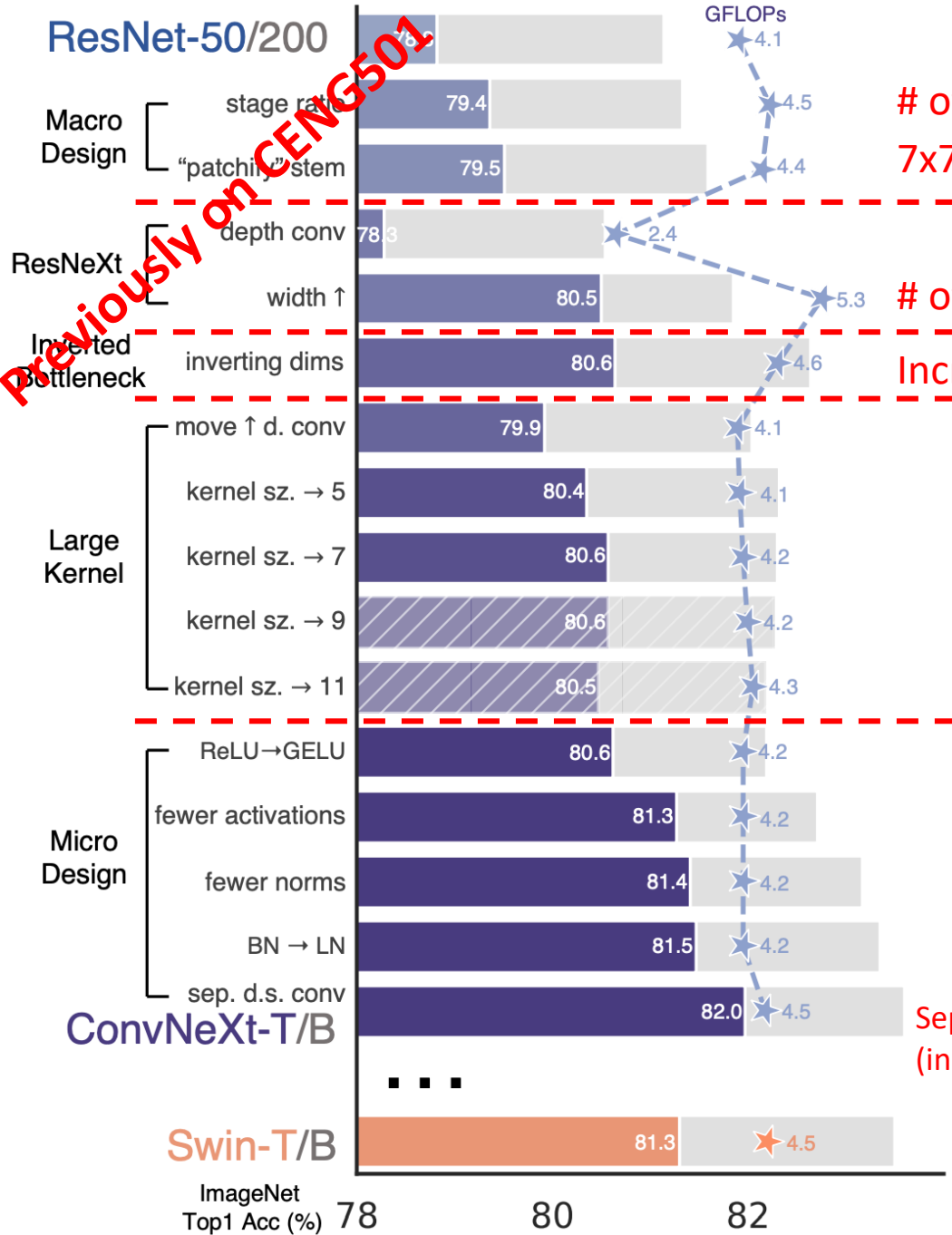


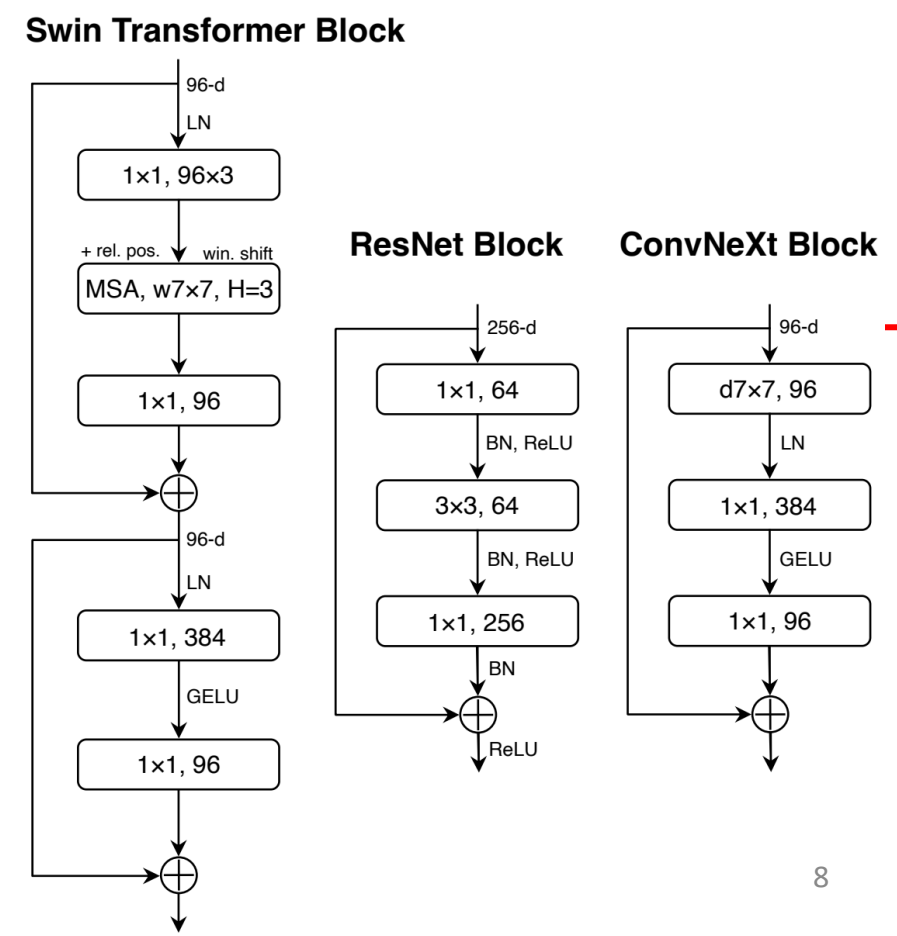
Figure 4: **Overall architecture of ViT-Adapter.** (a) The ViT, whose encoder layers are divided into N (usually $N = 4$) equal blocks for feature interaction. (b) Our ViT-Adapter, which contains three key designs, including (c) a spatial prior module for modeling local spatial contexts from the input image, (d) a spatial feature injector for introducing spatial priors into ViT, and (e) a multi-scale feature extractor for reorganizing multi-scale features from the single-scale features of ViT.



of blocks in each stage from [3,4,6,3] to [3,3,9,3]
 7x7 conv with stride 2 + max-pool (4x downsamp) => 4x4 conv with stride 4

of channels increased to match Swin-T
 Increase embedding dim

Separate downsampling layers (instead of residual blocks)



Previously on CENG501

Faster ViT

Figure 2: Visualization of the proposed Hierarchical Attention in the feature space. By performing local window attention and hierarchical attention we can achieve global information propagation at reduced costs.

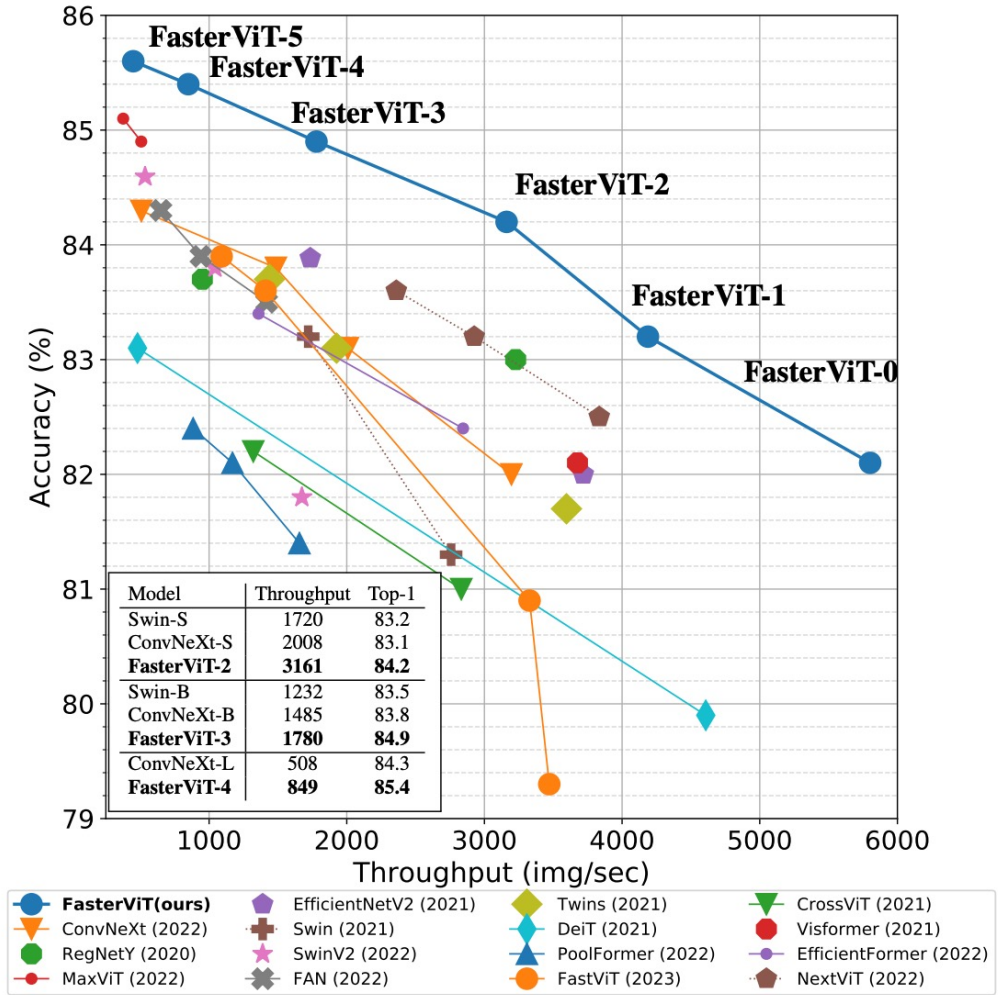
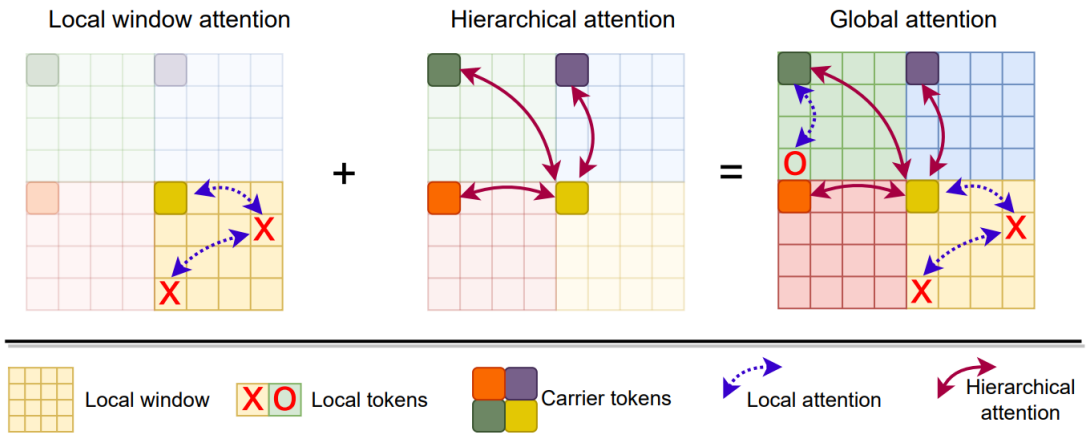


Figure 1: Comparison of image throughput and ImageNet-1K Top-1 accuracy. Throughput is measured on A100 GPU with batch size of 128.

Previously on ENG501

Masked Autoencoders

Masked Autoencoders Are Scalable Vision Learners

Kaiming He^{*,†} Xinlei Chen^{*} Saining Xie Yanghao Li Piotr Dollár Ross Girshick

^{*}equal technical contribution [†]project lead

Facebook AI Research (FAIR)

CVPR 2022

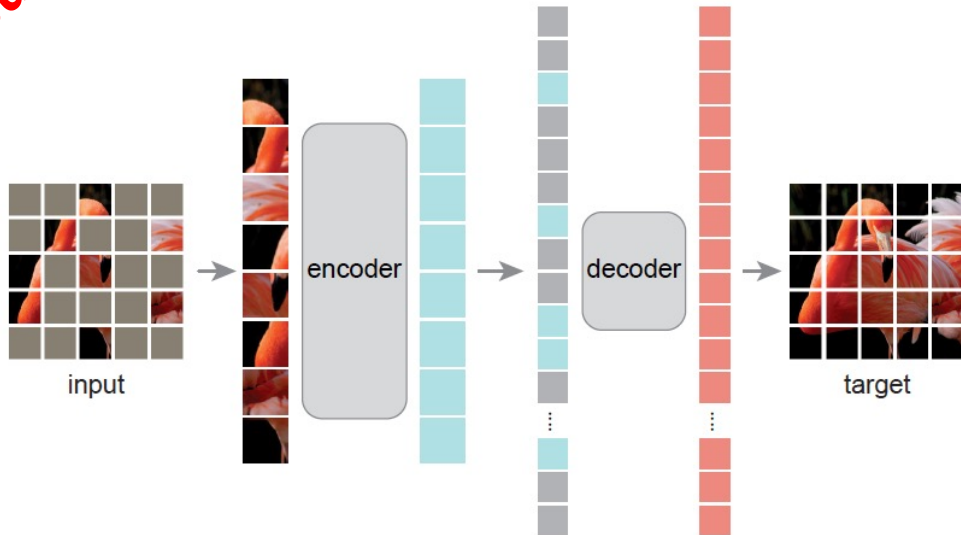


Figure 1. **Our MAE architecture.** During pre-training, a large random subset of image patches (e.g., 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images (full sets of patches) for recognition tasks.

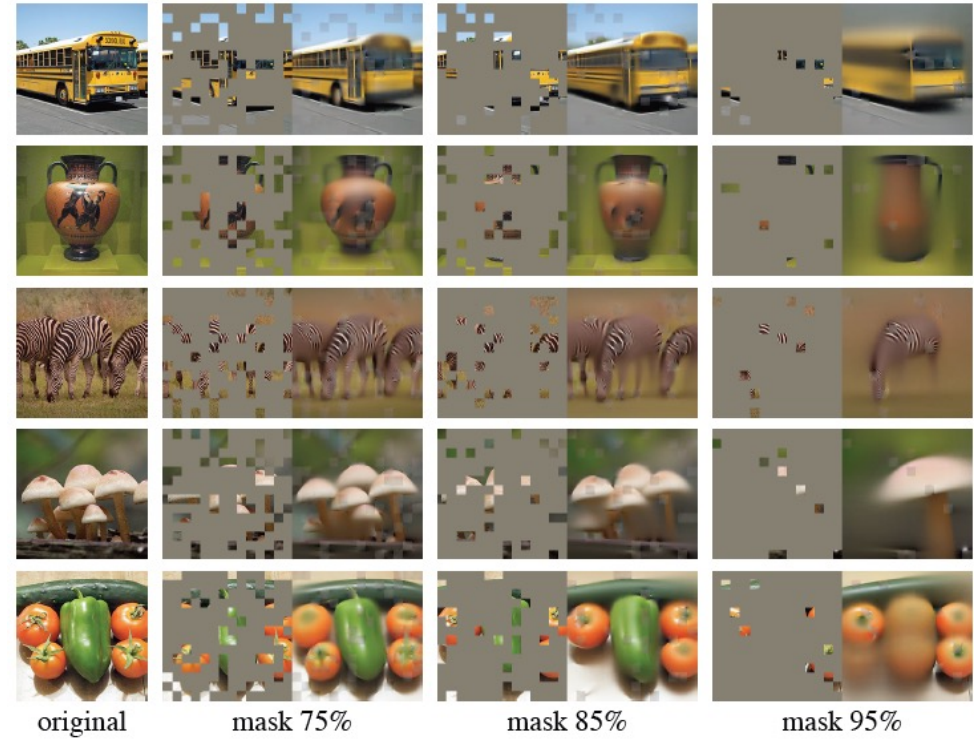


Figure 4. Reconstructions of ImageNet *validation* images using an MAE pre-trained with a masking ratio of 75% but applied on inputs with higher masking ratios. The predictions differ plausibly from the original images, showing that the method can generalize.

MSE loss between pixels for masked tokens only!

Previously on **SimMIM**
Tsinghua University

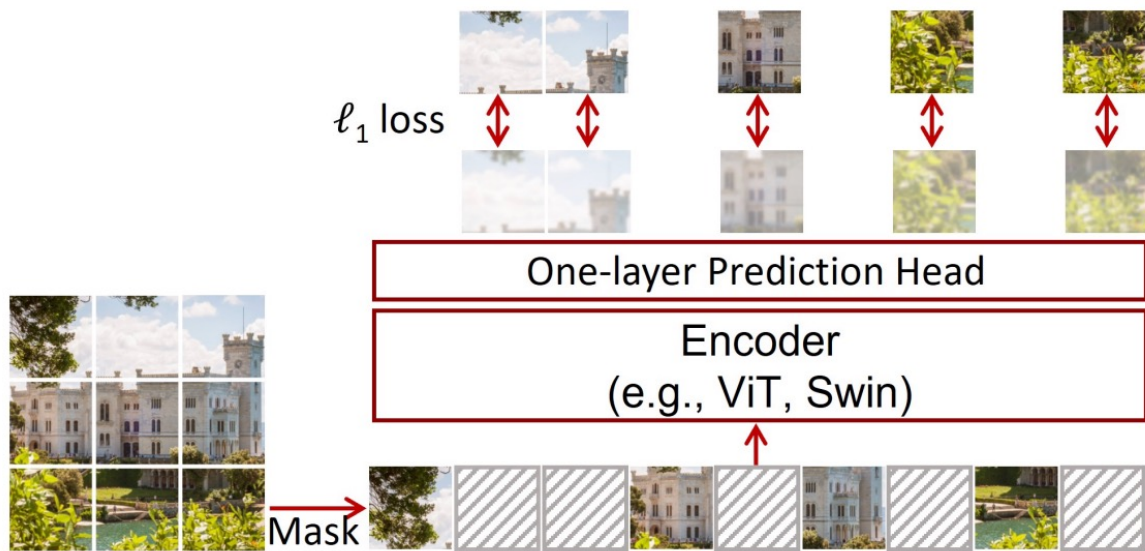


Figure 1. An illustration of our simple framework for masked language modeling, named *SimMIM*. It predicts raw pixel values of the randomly masked patches by a lightweight one-layer head, and performs learning using a simple ℓ_1 loss.

SimMIM: a Simple Framework for Masked Image Modeling

CVPR 2022

Zhenda Xie^{1*} Zheng Zhang^{2*} Yue Cao^{2*}

Yutong Lin³ Jianmin Bao² Zhuliang Yao¹ Qi Dai² Han Hu^{2*}

¹Tsinghua University ²Microsoft Research Asia ³Xi'an Jiaotong University

{t-zhxie, zhez, yuecao, t-yutonglin, jianmin.bao, t-zhuyao, qid, hanhu}@microsoft.com

Methods	Input Size	Fine-tuning	Linear eval	Pre-training
		Top-1 acc (%)	Top-1 acc (%)	costs
Sup. baseline [44]	224 ²	81.8	-	-
DINO [5]	224 ²	82.8	78.2	2.0×
MoCo v3 [9]	224 ²	83.2	76.7	1.8×
ViT [15]	384 ²	79.9	-	~4.0×
BEiT [1]	224 ²	83.2	56.7	1.5× [†]
Ours	224 ²	83.8	56.7	1.0×

Table 6. System-level comparison using ViT-B as the encoder. Training costs are counted in relative to our approach. [†] BEiT requires an additional stage to pre-train dVAE, which is not counted.

Previously on CVENG501

DINO v1 (ICCV'21)

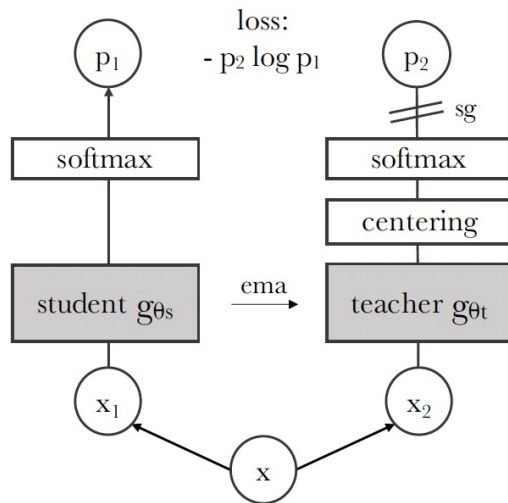


Figure 2: **Self-distillation with no labels.** We illustrate DINO in the case of one single pair of views (x_1, x_2) for simplicity. The model passes two different random transformations of an input image to the student and teacher networks. Both networks have the same architecture but different parameters. The output of the teacher network is centered with a mean computed over the batch. Each networks outputs a K dimensional feature that is normalized with a temperature softmax over the feature dimension. Their similarity is then measured with a cross-entropy loss. We apply a stop-gradient (sg) operator on the teacher to propagate gradients only through the student. The teacher parameters are updated with an exponential moving average (ema) of the student parameters.

Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

DIWO v2 (TMLR'24)

Previously on CENG501

- **Patch-Level Loss:** Unlike v1, which looked at the whole image, v2 forced the model to match individual patches. This made it much better at tasks like depth estimation and semantic segmentation.

Patch-level objective (Zhou et al., 2022a). We randomly mask some of the input patches given to the student, but not to the teacher. We then apply the student iBOT head to the student mask tokens. Similarly, we apply the teacher iBOT head to the (visible) teacher patch tokens corresponding to the ones masked in the student. We then apply the softmax and centering steps as above, and obtain the iBOT loss term:

$$\mathcal{L}_{iBOT} = - \sum_i p_{ti} \log p_{si}$$

, where i are patch indices for masked tokens. Similarly to above, we learn the parameters of the student, and build the teacher head through exponential moving average.

Previously on CVENG501

DINO v2 (TMLR'24)

- **Data Curation:** Meta developed an automated pipeline to curate a massive, balanced dataset (LVD-142M) from the web, proving that data quality matters as much as quantity.

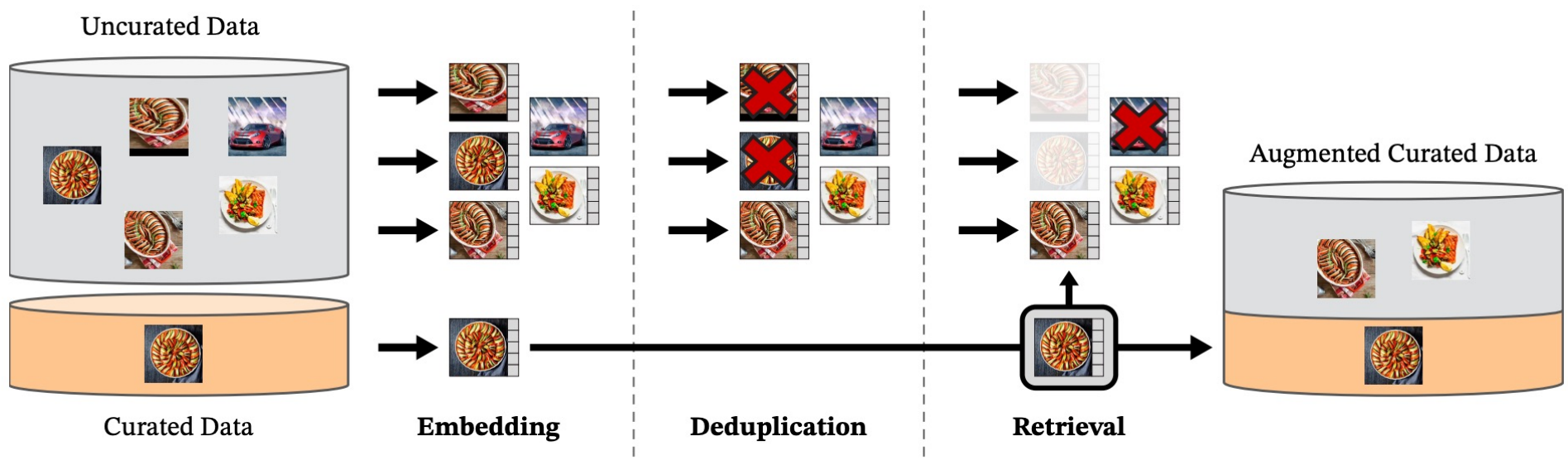


Figure 3: **Overview of our data processing pipeline.** Images from curated and uncurated data sources are first mapped to embeddings. Uncurated images are then deduplicated before being matched to curated images. The resulting combination augments the initial dataset through a self-supervised retrieval system.

Previously on CENG501

DINO v3 (2025/6)

- Larger dataset
- Larger model

Table 2: Comparison of the teacher architectures used in DINOv2 and DINOv3 models. We keep the model 40 blocks deep, and increase the embedding dimension to 4096. Importantly, we use a patch size of 16 pixels, changing the effective sequence length for a given resolution.

Teacher model	DINOv2	DINOv3
Backbone	ViT-giant	ViT-7B
#Params	1.1B	6.7B
#Blocks	40	40
Patch Size	14	16
Pos. Embeddings	Learnable	RoPE
Registers	4	4
Embed. Dim.	1536	4096
FFN Type	SwiGLU	SwiGLU
FFN Hidden Dim.	4096	8192
Attn. Heads	24	32
Attn. Heads Dim.	64	128
DINO Head MLP	4096-4096-256	8192-8192-512
DINO Prototypes	128k	256k
iBOT Head MLP	4096-4096-256	8192-8192-384
iBOT Prototypes	128k	96k

DINO v3 (2025/6)

- Gram Anchoring:
 - Why: Feature collapse emerges where local details become noisy over long training runs.
 - Gram Anchoring: a regularization technique that keeps the relationship between image patches stable.

superior dense properties. By operating on the Gram matrix rather than the feature themselves, the local features are free to move, provided the structure of similarities remains the same. Suppose we have an image composed of P patches, and a network that operates in dimension d . Let us denote by \mathbf{X}_S (respectively \mathbf{X}_G) the $P \times d$ matrix of \mathbf{L}_2 -normalized local features of the student (respectively the Gram teacher). We define the loss $\mathcal{L}_{\text{Gram}}$ as follows:

$$\mathcal{L}_{\text{Gram}} = \|\mathbf{X}_S \cdot \mathbf{X}_S^\top - \mathbf{X}_G \cdot \mathbf{X}_G^\top\|_{\text{F}}^2. \quad (2)$$

Previously on CENG501

Autoregressive Image Modeling (AIM)

<https://machinelearning.apple.com/research/autoregressive-image-models>

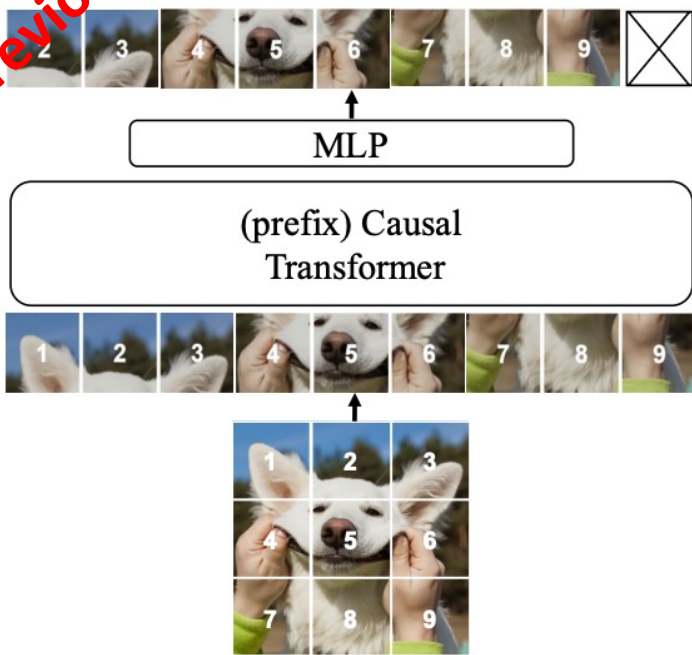


Figure 2. AIM pre-training overview. Input images are split into non-overlapping patches and embedded linearly following Dosovitskiy et al. [29]. The patch features are fed to a transformer in which the self-attention operation is causally masked to prevent attending to preceding positions. Afterward, a heavily parameterized MLP processes each of the patch features independently and finally projects it to pixel space. The targets correspond to the input sequence shifted one position to the left, requiring the model to predict the next patch in raster order.

Model	Arch.	Data	IN-1k	iNAT-18	Cifar10	Cifar100	Food101	DTD	Pets	Cars	iWildCam	Camelyon17	PCAM	RxRX1	EuroSAT	fMoW	Infographic	Avg
DINO [17]	ViT-B/8	IN-1k	80.1	66.0	97.8	87.3	89.5	78.4	92.3	89.2	58.5	93.7	90.2	6.1	98.2	57.0	41.1	75.0
iBOT [88]	ViT-L/16	IN-21k	83.5	70.5	99.2	93.3	93.5	81.6	92.8	90.8	61.8	94.5	90.0	5.9	98.0	60.3	47.7	77.6
DINOv2 [58]	ViT-g/14 ₅₁₆	LVD	86.4	84.5	99.6	95.2	96.3	86.3	96.4	95.6	68.2	96.5	90.7	8.0	98.6	66.7	58.8	81.9
BEiT [5]	ViT-L/14	IN-21k	62.2	44.4	94.4	78.7	79.0	64.0	80.9	69.5	52.0	92.8	88.2	4.2	97.5	47.7	25.9	65.4
MAE [41, 70]	ViT-H/14	IN-1k	80.9	64.6	97.1	85.8	90.2	78.1	95.0	93.7	58.1	94.2	89.8	5.4	98.1	56.9	42.2	75.3
	ViT-2B/14	IG-3B	82.2	70.8	97.5	87.3	93.4	81.2	95.1	94.9	57.8	94.4	90.3	7.3	98.2	60.1	50.2	77.4
AIM-0.6B	ViT-H/14	DFN-2B+	78.5	64.0	97.2	86.8	90.1	80.1	93.0	93.0	57.9	94.3	90.0	7.8	98.4	58.3	45.2	75.6
AIM-1B	ViT-1B/14		80.6	67.2	98.2	88.3	91.6	81.8	93.4	93.9	58.6	94.5	90.0	9.0	98.6	59.8	47.5	76.9
AIM-3B	ViT-3B/14		82.2	69.7	98.4	89.9	92.7	81.9	94.1	93.8	58.8	94.3	90.4	9.7	98.5	60.9	48.9	77.6
AIM-7B	ViT-7B/14		82.4	70.9	98.6	90.0	93.1	82.3	93.8	92.1	59.5	93.6	90.7	10.1	98.6	61.7	49.6	77.8
AIM-7B†	ViT-7B/14		DFN-2B+	84.0	75.5	98.9	91.8	94.1	85.6	95.4	95.0	61.4	94.2	90.5	8.4	98.5	63.5	57.7

Table 6. Downstream evaluation with a frozen trunk. We assess the quality of AIM features by evaluating against a diverse set of 15 image recognition benchmarks. AIM and the baseline methods are evaluated using attentive probing with a frozen trunk. AIM models exhibit a strong performance across all benchmarks, especially the AIM-7B. AIM outperforms all other methods, using joint-embedding or generative approaches, except for DINOv2 which utilizes higher-resolution images, that typically results in a 1-1.5% improvement on ImageNet for instance. †: Extracting features from the 20th layer instead of the last (32nd), see Table 7 for more details.

Previously on CAVG501

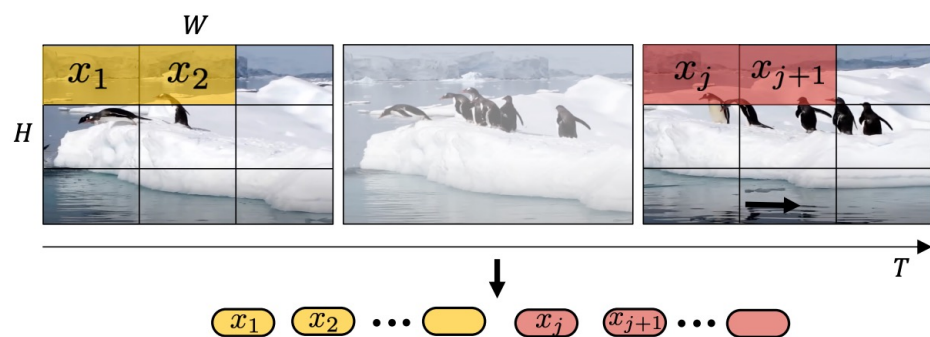
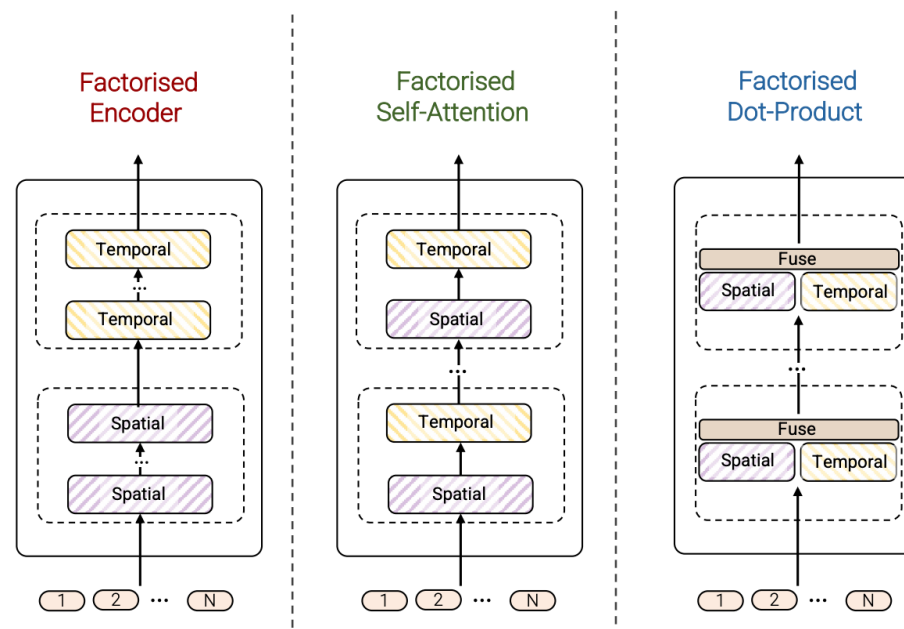
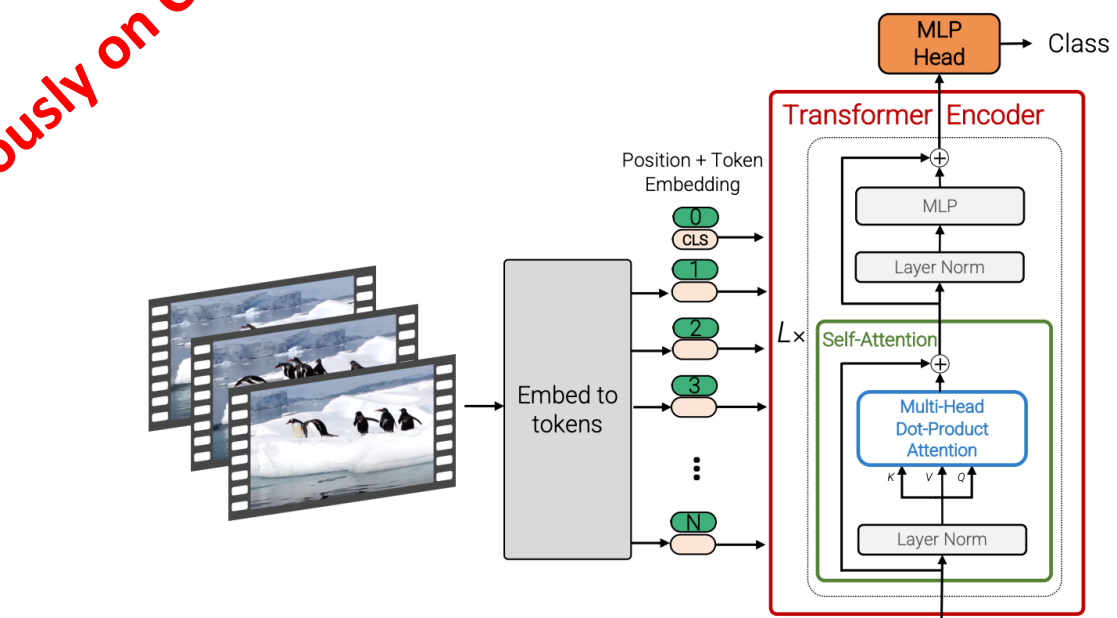


Figure 2: Uniform frame sampling: We simply sample n_t frames, and embed each 2D frame independently following ViT [18].

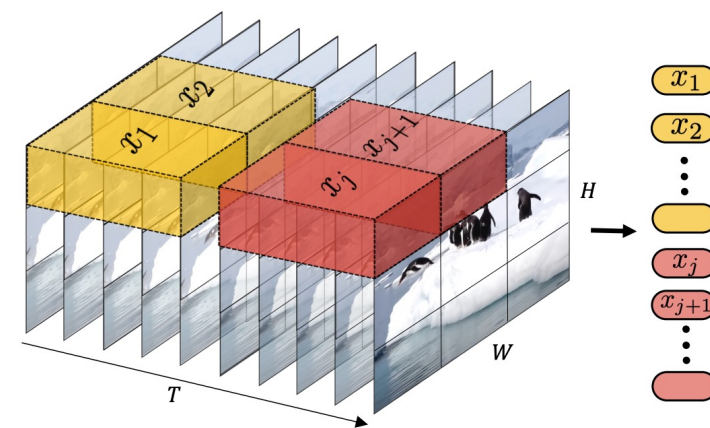


Figure 3: Tubelet embedding. We extract and linearly embed non-overlapping tubelets that span the spatio-temporal input volume.

Previously on **CVPR501**
VideoMAE

Zhan Tong^{1,2*} Yibing Song² Jue Wang² Limin Wang^{1,3†}

¹State Key Laboratory for Novel Software Technology, Nanjing University

²Tencent AI Lab ³Shanghai AI Lab

tongzhan@smail.nju.edu.cn {yibingsong.cv, arphid}@gmail.com lmwang@nju.edu.cn

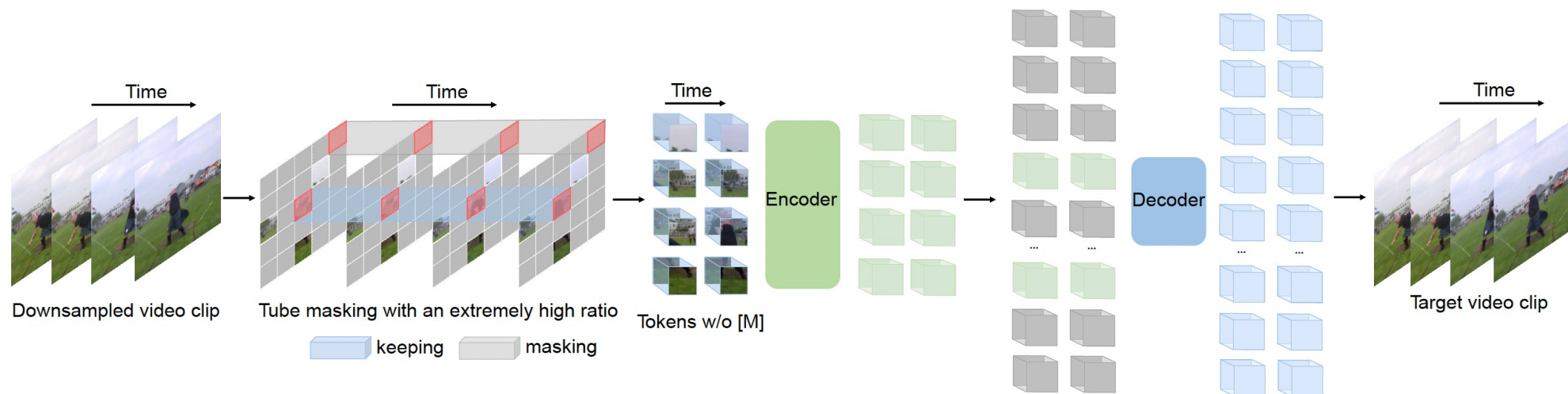


Figure 1: **VideoMAE** performs the task of masking random cubes and reconstructing the missing ones with an asymmetric encoder-decoder architecture. Due to high redundancy and temporal correlation in videos, we present the customized design of tube masking with an extremely high ratio (90% to 95%). This simple design enables us to create a more challenging and meaningful self-supervised task to make the learned representations capture more useful spatiotemporal structures.

Today

- Vision-Language Models

Administrative Notes

- Project next steps:
 - Milestones:
 1. Milestone (April 10, midnight):
 - Read & understand the paper
 - Download the datasets
 - Prepare the Readme file excluding the results & conclusion
 2. Milestone (May 4, midnight)
 - The results of the first experiment
 3. Milestone (June 1, midnight)
 - Final report (Readme file)
 - Repo with all code & trained models

Firindan Sıcak Sıcak

Philippe Laban Tobias Schnabel Jennifer Neville
Microsoft Research
{plaban, tobias.schnabel, jenneville}@microsoft.com April 2026

Delegating editing work to LLMs corrupts documents: across 52 professional domains, **current frontier models degrade 25% of document content** after just 20 interactions.

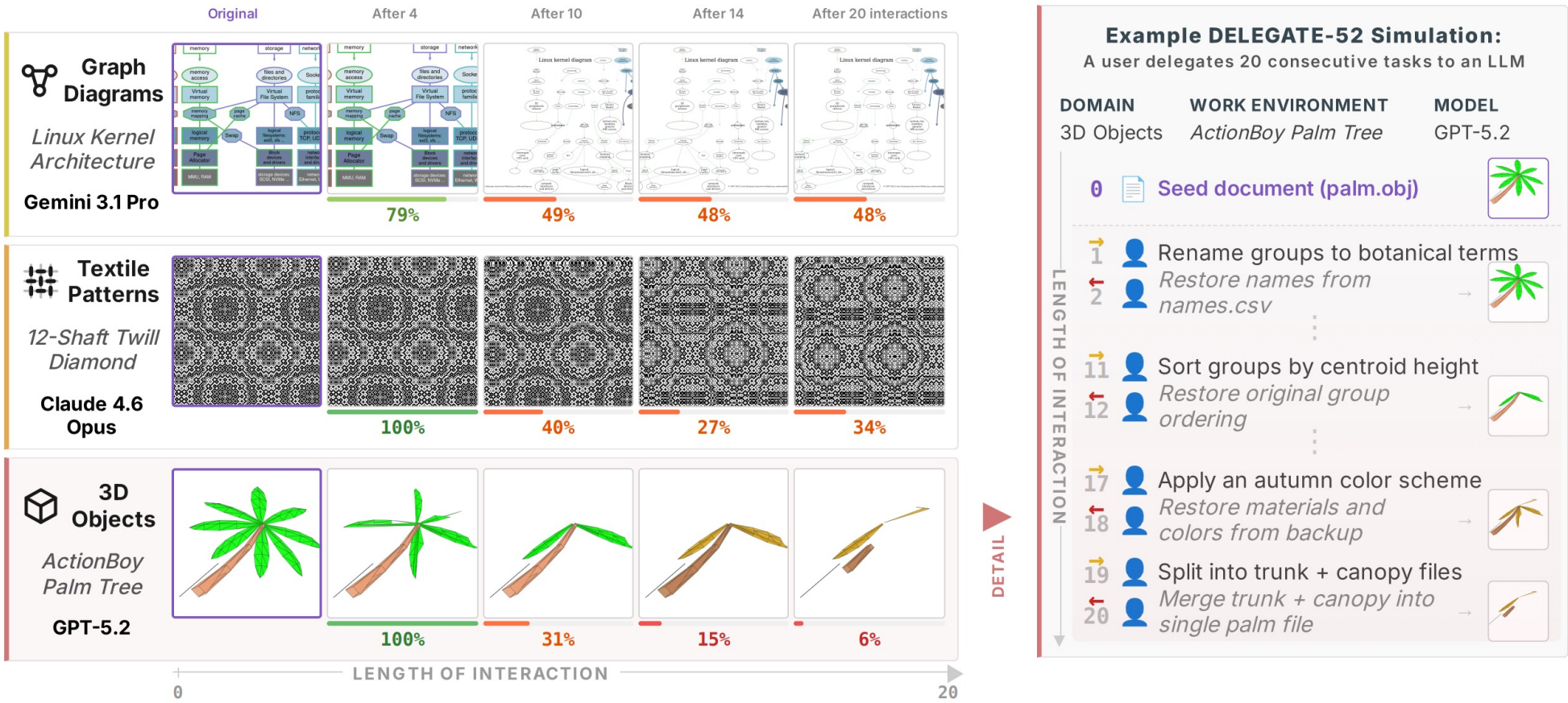
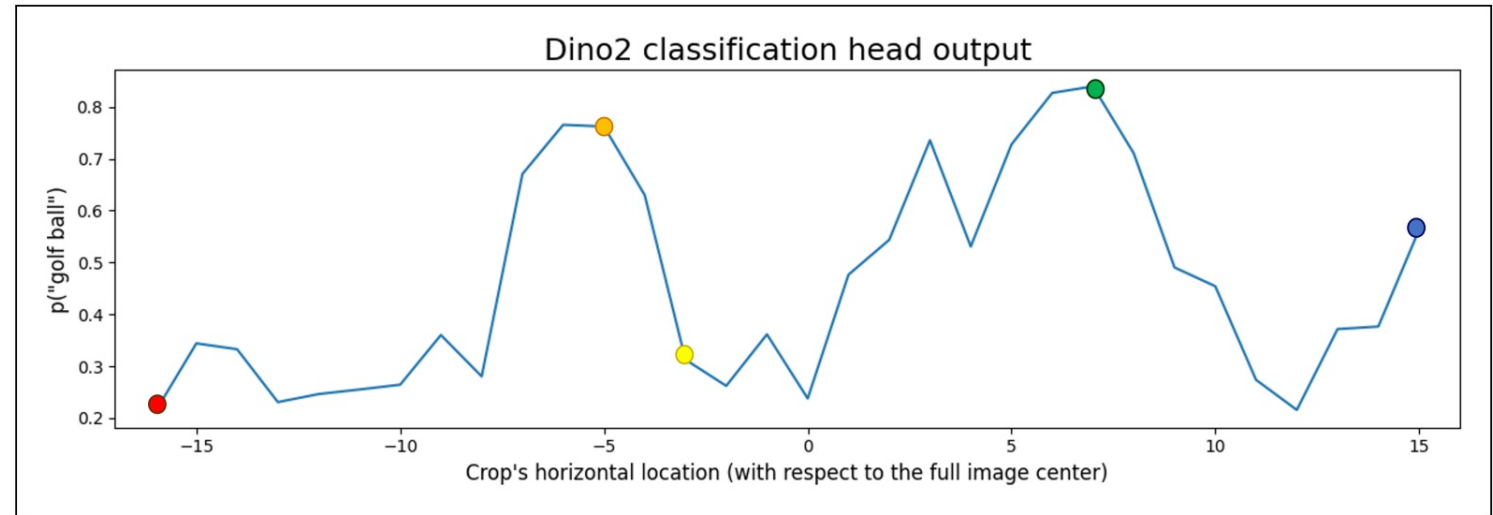
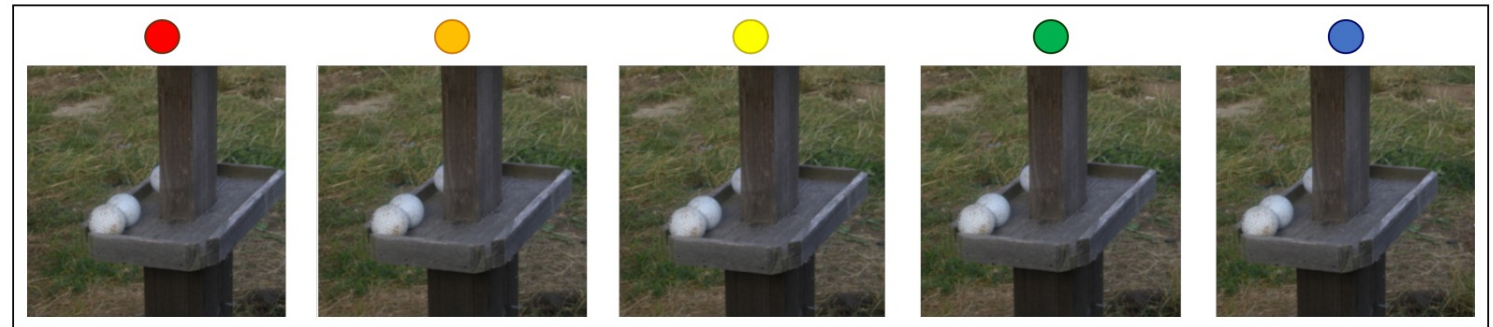


Figure 1: Illustrative examples of how LLMs corrupt documents over long workflows in the DELEGATE-52 benchmark. As LLMs edit files that represent graph diagrams, textile patterns or 3D objects, they introduce sparse but severe errors that silently corrupt documents, compounding over long interaction.¹

Firindan Sıcak Sıcak



(a) DINOv2's output score for the true label ("golf ball") as a function of the crop location.



(b) Five Examples of different crops, the colored circles connect the image to its predicted probability

Lost in Translation: Modern Neural Networks Still Struggle With Small Realistic Image Transformations

April 2026

Ofir Shifman and Yair Weiss

The Hebrew University of Jerusalem
{Ofir.Shifman, Yair.Weiss}@mail.huji.ac.il

Fig. 1: The modern DNN DINOv2 [23] still suffers from significant changes in the output probability for the true label ("golf ball") as a function of minor realistic translations. (a) shows DINOv2's probability for "golf ball" for 32 crops, each measuring 224x224 pixels, extracted from the same 256x256 pixels image with lateral translation only. (b) Demonstrates that this translation by a few pixels translation is nearly imperceptible. We limit our experiments to evaluation set in which the true label object remains fully observable within all the assessed crops, as can be seen here.

Firindan Sıcak Sıcak

Damiano Fornasiero*, Mirko Bronzi*, Spencer Kitts*,
Alessandro Palmas, Yoshua Bengio†, Oliver Richardson†

{d.fornasiero, m.bronzi, s.kitts, a.palmas, y.bengio, o.richardson}@lawzero.org

LawZero

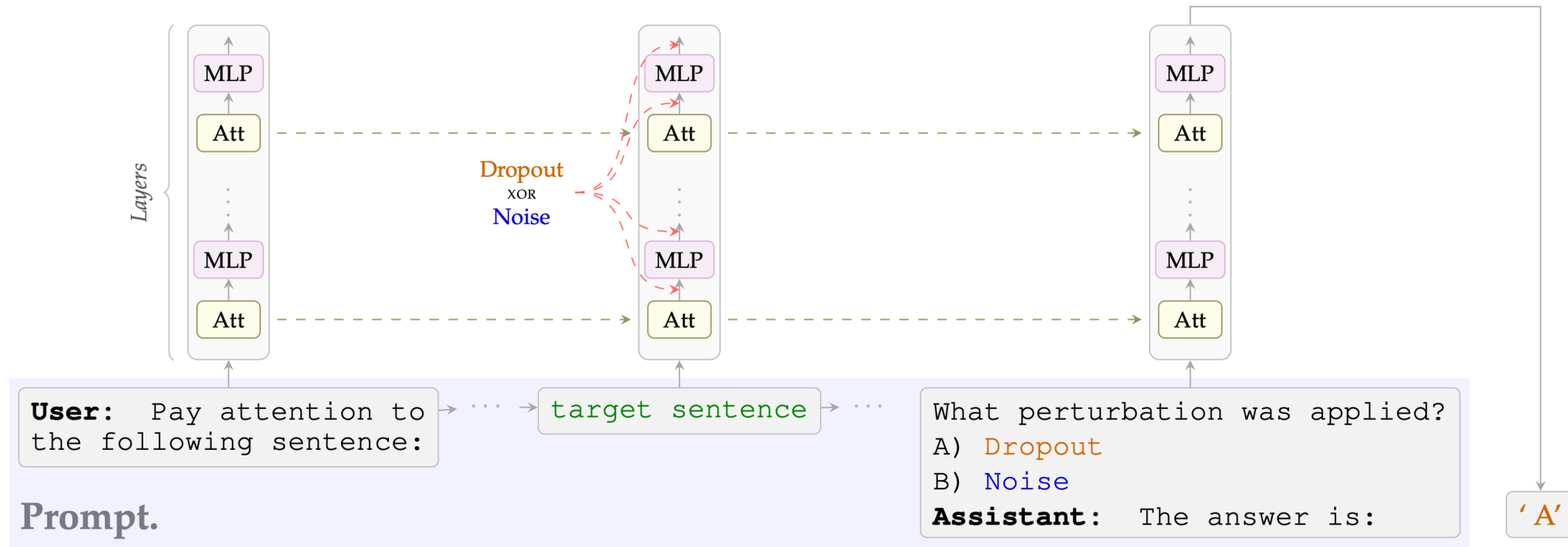


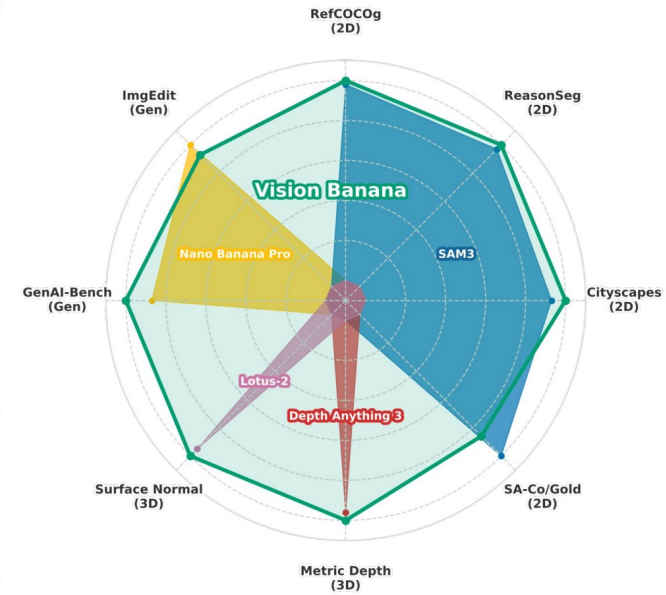
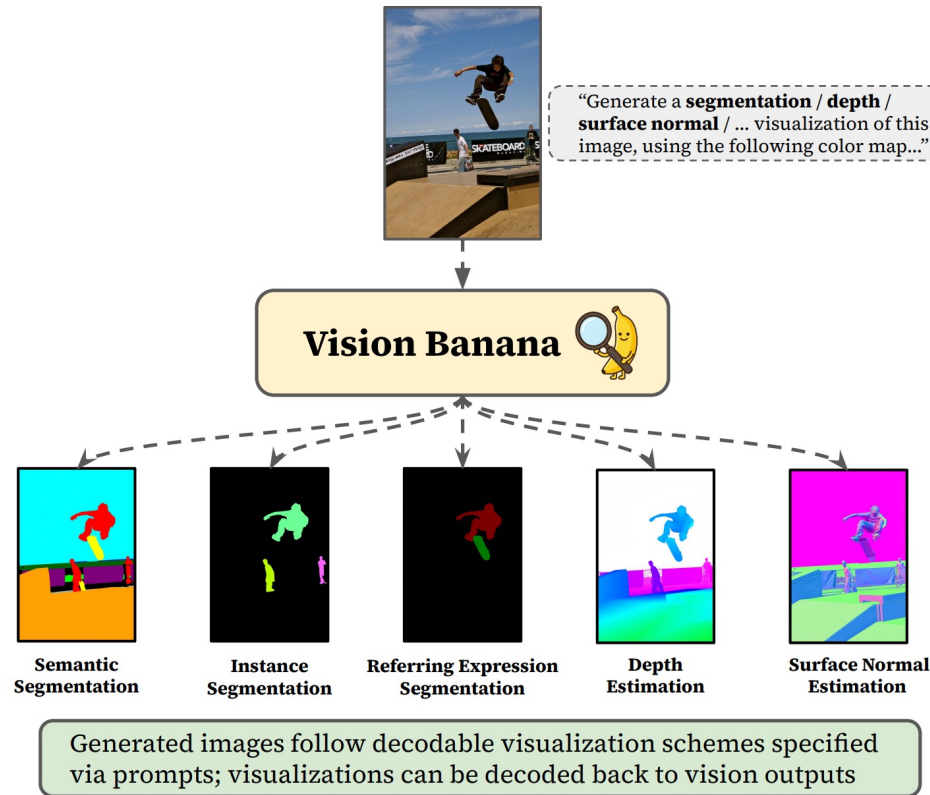
Figure 1: We perturb the activations of a **target sentence** by either **masking activations** or **adding Gaussian noise**. In the same prompt we then ask the model to identify which perturbation was applied. Success is measured as **accuracy** of the most-likely next-token.

Firindan Sıcak Sıcak

Image Generators are Generalist Vision Learners

Valentin Gabeur*, Shangbang Long*, Songyou Peng*, Paul Voigtlaender[∇], Shuyang Sun[∇], Yanan Bao[∇], Karen Truong[∇], Zhicheng Wang[∇], Wenlei Zhou[∇], Jonathan T. Barron[∇], Kyle Genova[∇], Nithish Kannan[∇], Sherry Ben[∇], Yandong Li[∇], Mandy Guo[∇], Suhas Yogin[∇], Yiming Gu[†], Huizhong Chen[†], Oliver Wang[‡], Saining Xie[‡], Howard Zhou[‡], Kaiming He[‡], Thomas Funkhouser[‡], Jean-Baptiste Alayrac[‡] and Radu Soricut[‡]

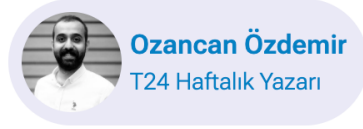
*project leads and equal contributions, [∇]core contributors, [†]project advisors, [‡]leadership sponsors



Vision Banana is a generalist vision model in both visual generation and understanding, surpassing or rivaling specialist models.

Figure 1 | We demonstrate the hidden visual understanding capabilities of image generators by instruction-tuning Nano Banana Pro. The instruction-tuned model, **Vision Banana** 🍌, can produce visualizations in a precise format that can enable evaluation on established benchmarks.

Fırından Sıcak Sıcak



https://t24.com.tr/yazarlar/ozancan-ozdemir/yapay-zek-yazilim-muhendisligini-ve-diger-meslekleri-oldurecek-mi,54925?rnd=1777182637880181&_t=1777395326441

Yapay zekâ yazılım mühendisliğini (ve diğer meslekleri) öldürecek mi?

Sadece yazılım mühendisliği değil, diğer alanlardaki veriler de en azından şu an için bizlerin yapay zekâ ve meslekler arasındaki etkileşim hakkında bir sonuca varmamızı sağlıyor. Büyük dil modelleri özelinde yapay zekâ teknolojisi işlerimizi elimizden almıyor, zaten mevcut LLM teknolojisi itibari ile de alması aslında bir hayli zor. Mevcut işten çıkarmaların en temel sebebi ise iş gücünü yapay zeka ile değiştirmekten ziyade, yapay zeka yatırımları için kaynak sıkıntısı yaşayan şirketlerin çalışan giderlerini bir kaynak olarak görmesi. Kamuoyunun görmesi gereken ancak maalesef görmediği ya da görmek istemediği ayrım burada yatıyor. Çünkü korku, etkileşimi artırır

Vision-Language Models

Reference: Bordes et al., “An Introduction to Vision-Language Modeling”, 2024.
<https://arxiv.org/pdf/2405.17247>

What is a Vision-Language Model?

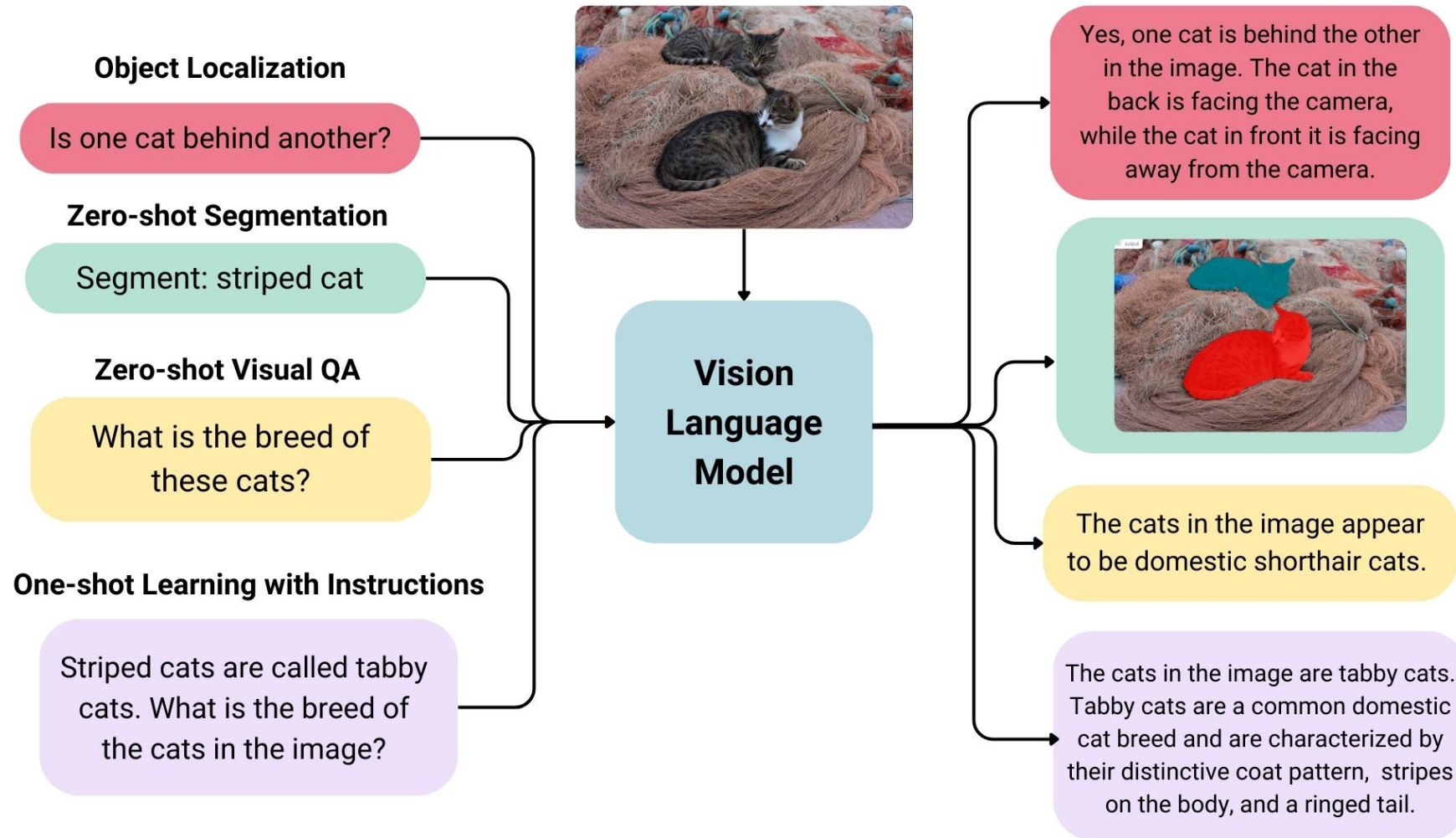


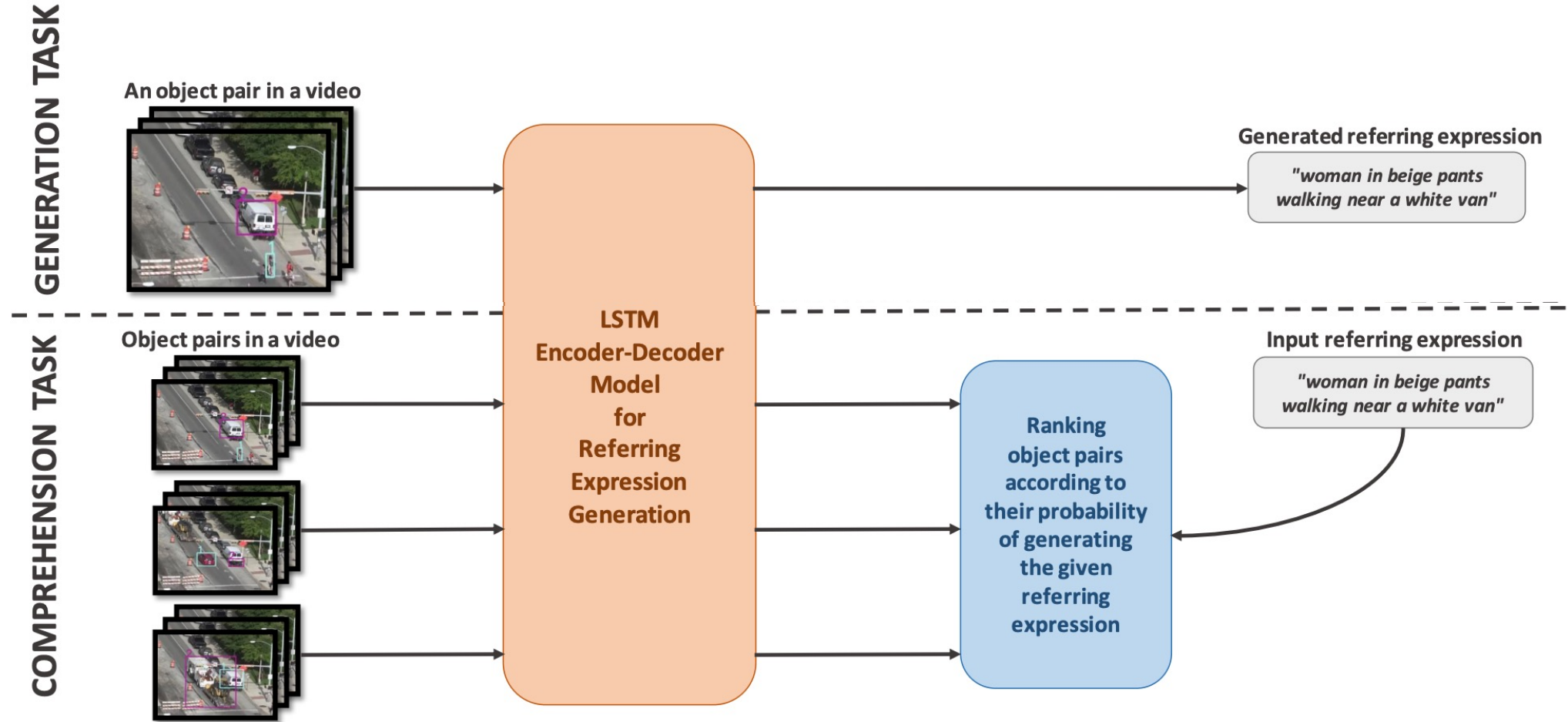
Fig: <https://huggingface.co/blog/vlms>

Earlier Attempts:

Hazan Anayurt*
hazan.anayurt@metu.edu.tr
Sezai Artun Ozyegin*

Department of Computer Engineering,
Middle East Technical University,
Ankara, Turkey

2019



Earlier Attempts:

ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks



2019

Remember BERT?

(Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, 2018)

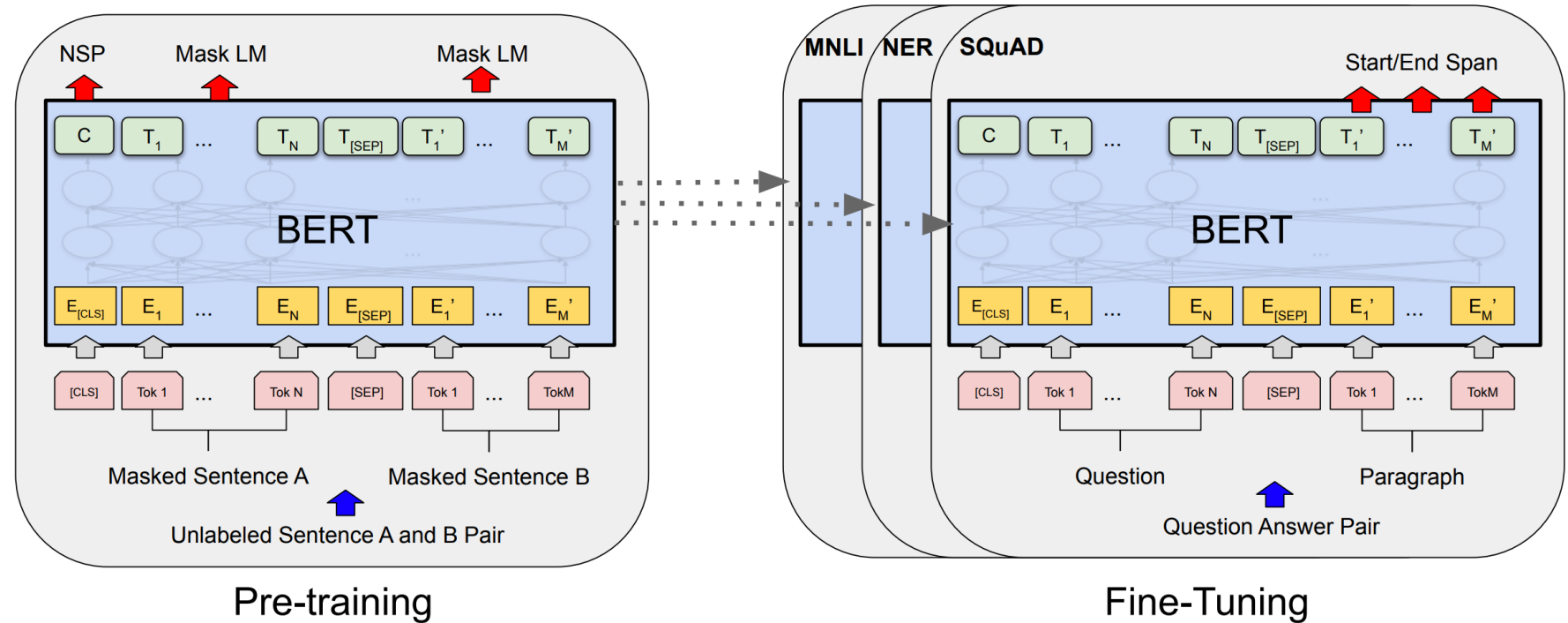


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

Earlier Attempts:

ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks



2019

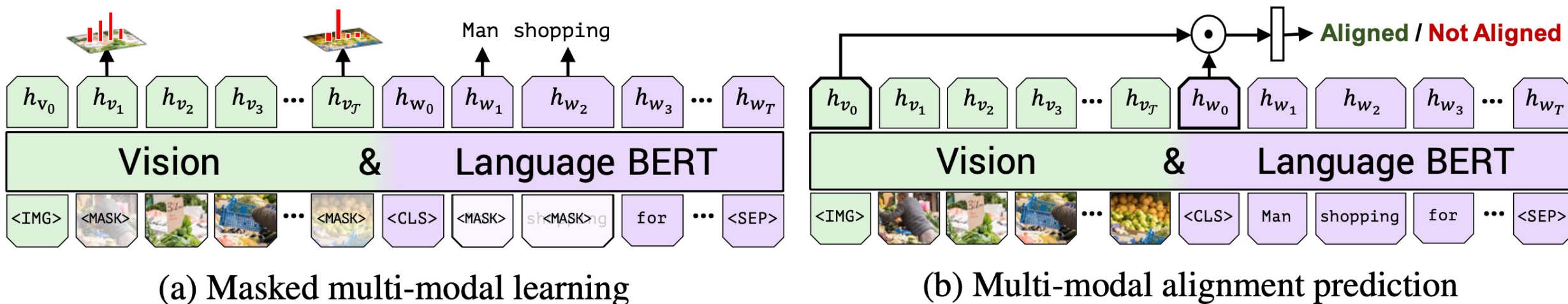


Figure 3: We train ViLBERT on the Conceptual Captions [24] dataset under two training tasks to learn visual grounding. In masked multi-modal learning, the model must reconstruct image region categories or words for masked inputs given the observed inputs. In multi-modal alignment prediction, the model must predict whether or not the caption describes the image content.

Earlier Attempts:

ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks



2019

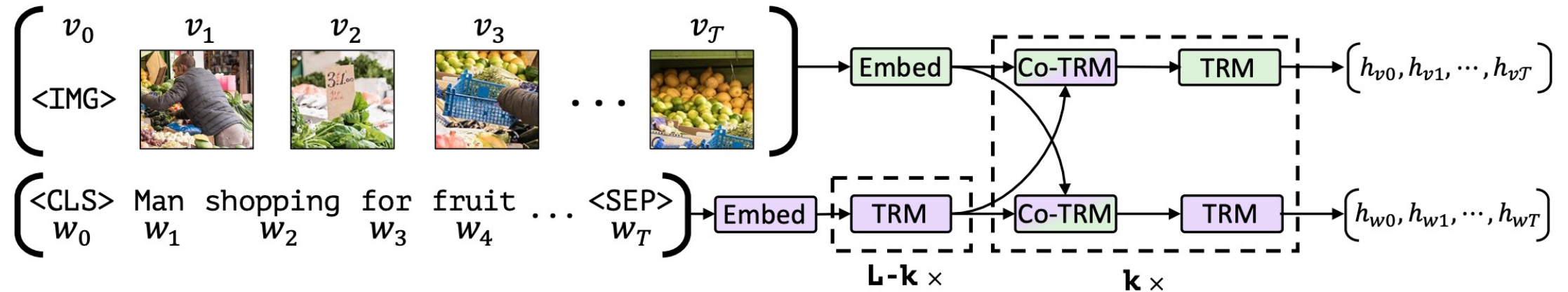


Figure 1: Our ViLBERT model consists of two parallel streams for visual (green) and linguistic (purple) processing that interact through novel co-attentional transformer layers. This structure allows for variable depths for each modality and enables sparse interaction through co-attention. Dashed boxes with multiplier subscripts denote repeated blocks of layers.

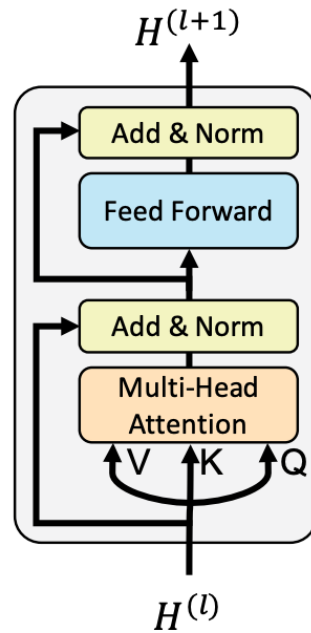
Co-TRM: Co-attentional Transformer

Earlier Attempts:

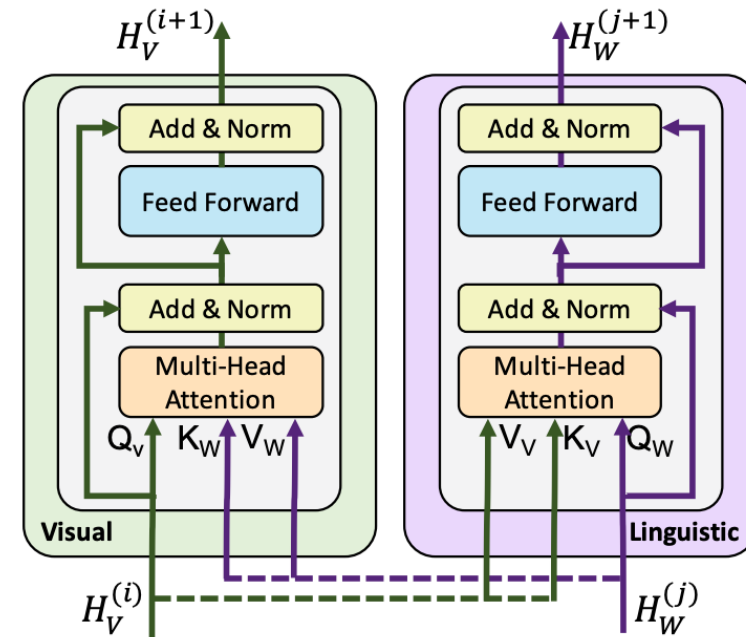
ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks



2019



(a) Standard encoder transformer block



(b) Our co-attention transformer layer

Figure 2: We introduce a novel co-attention mechanism based on the transformer architecture. By exchanging key-value pairs in multi-headed attention, this structure enables vision-attended language features to be incorporated into visual representations (and vice versa).

VISUALBERT: A SIMPLE AND PERFORMANT BASELINE FOR VISION AND LANGUAGE

Earlier Attempts:

Liunian Harold Li[†], Mark Yatskar^{*}, Da Yin[°], Cho-Jui Hsieh[†] & Kai-Wei Chang[†]

[†]University of California, Los Angeles

^{*}Allen Institute for Artificial Intelligence

[°]Peking University

2019



A person hits a ball with a tennis racket

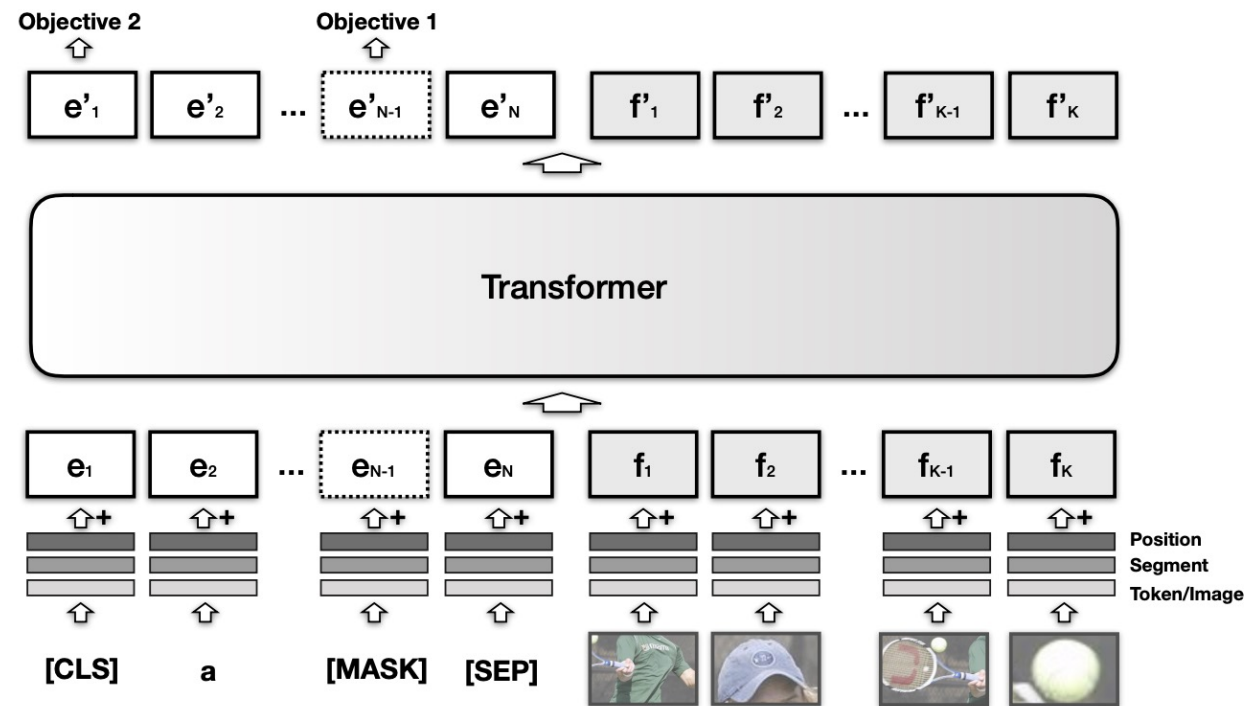
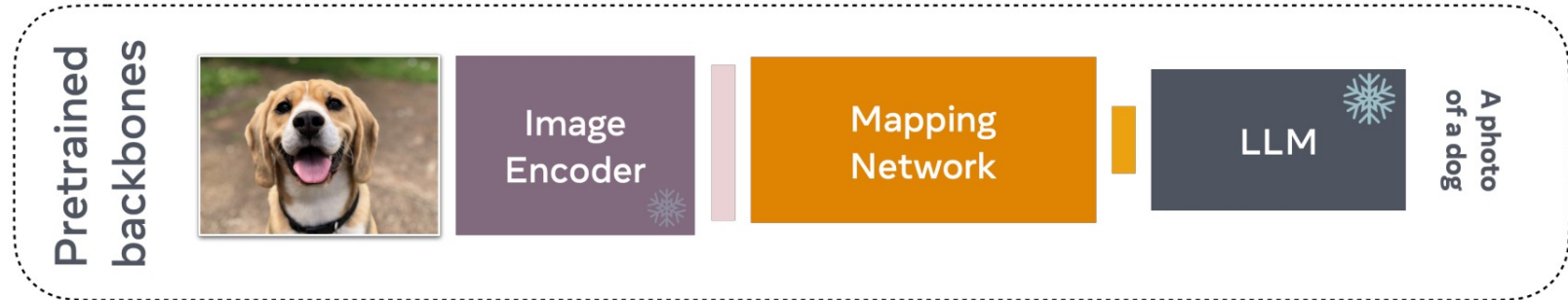
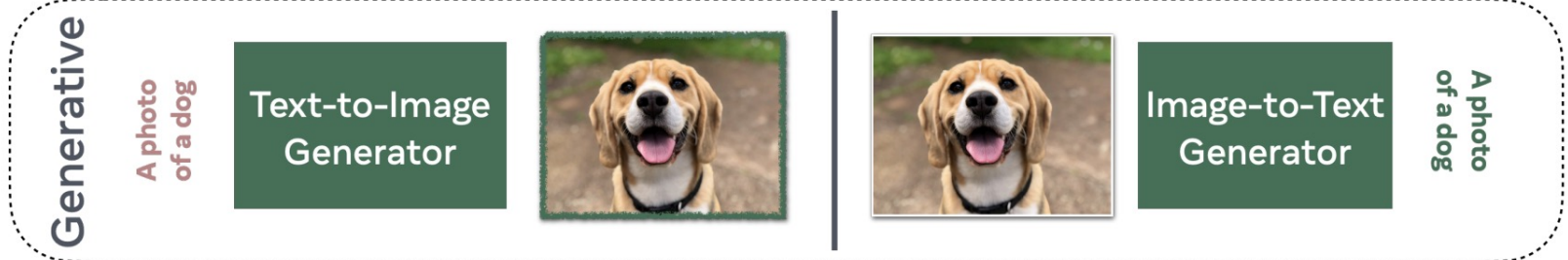
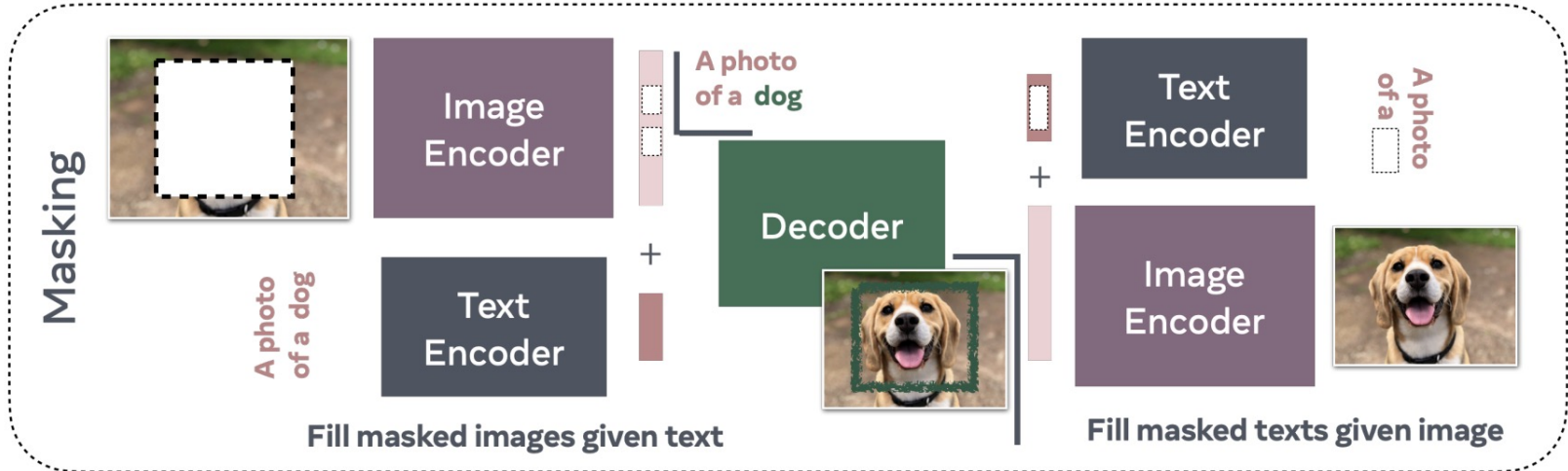
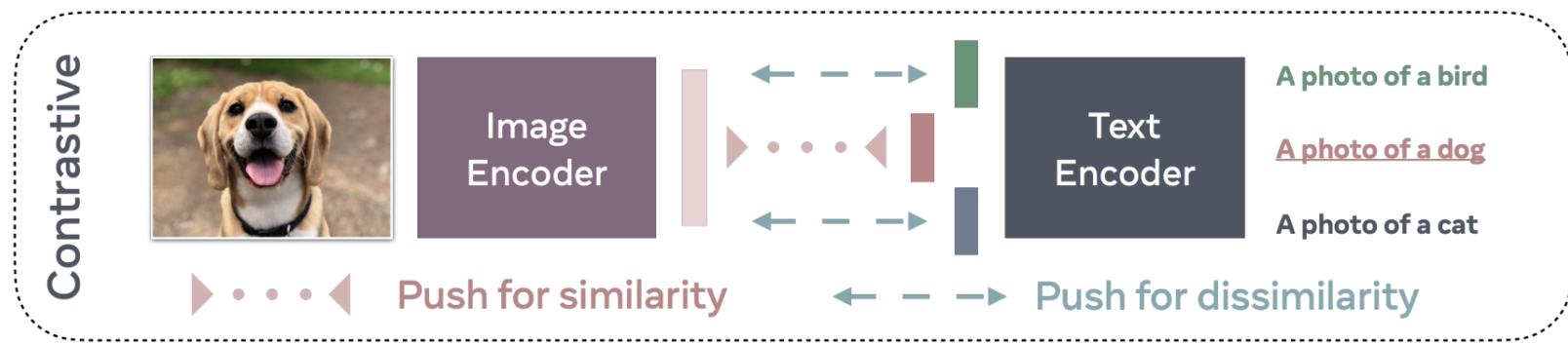


Figure 2: The architecture of VisualBERT. Image regions and language are combined with a Transformer to allow the self-attention to discover implicit alignments between language and vision. It is pre-trained with a masked language modeling (Objective 1), and sentence-image prediction task (Objective 2), on caption data and then fine-tuned for different tasks. See §3.3 for more details.

Overview



Bordes et al., "An Introduction to Vision-Language Modeling", 2024.
<https://arxiv.org/pdf/2405.17247>

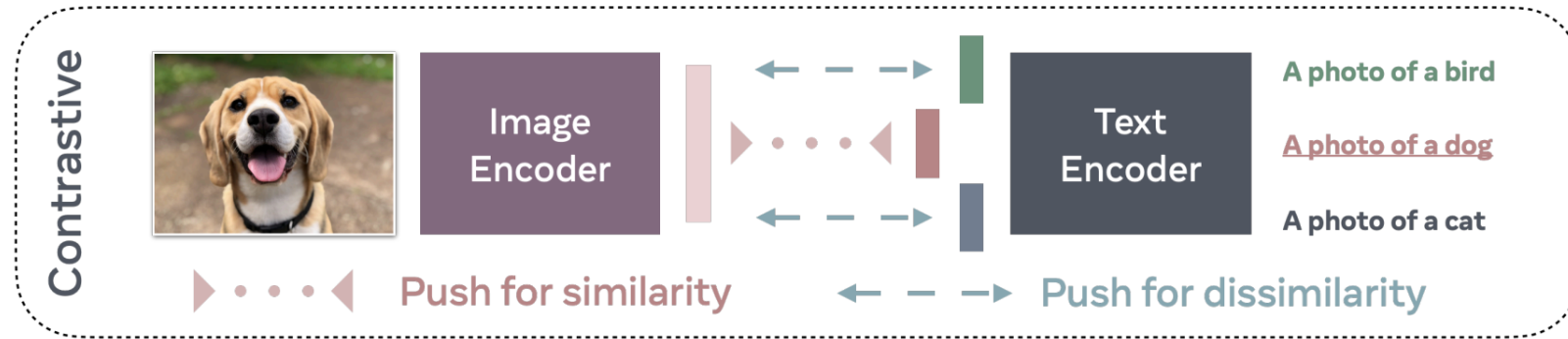


Fig: Bordes et al., "An Introduction to Vision-Language Modeling", 2024.

Contrastive Approaches

CLIP

Alec Radford^{*1} Jong Wook Kim^{*1} Chris Hallacy¹ Aditya Ramesh¹ Gabriel Goh¹ Sandhini Agarwal¹
Girish Sastry¹ Amanda Askell¹ Pamela Mishkin¹ Jack Clark¹ Gretchen Krueger¹ Ilya Sutskever¹

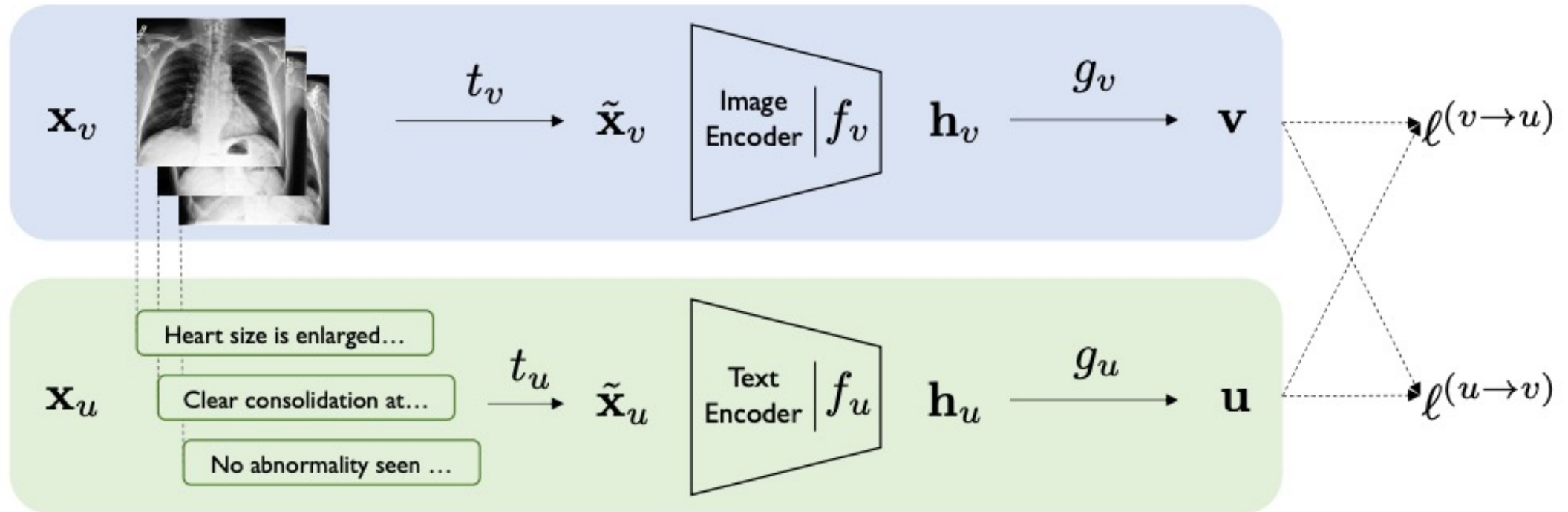
From OpenAI. ICML, 2021

- Motivation:
 - NLP models have benefited significantly from large amounts of web data
 - “Could scalable pre-training methods which learn directly from web text result in a similar breakthrough in computer vision?”
- Contribution:
 - Using natural language supervision for image representation learning at large scale
 - 400M pairs of (image, text)
 - Adapted from ConVIRT (train it from scratch with 400M pairs)

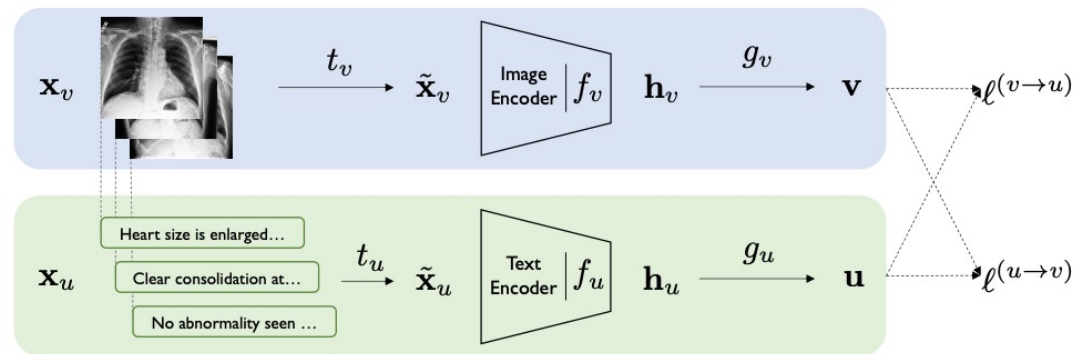
ConVIRT

Contrastive Learning of Medical Visual Representations from Paired Images and Text

Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, Curtis P. Langlotz



ConVIRT



The training objective of ConVIRT involves two loss functions. The first loss function is an image-to-text contrastive loss for the i -th pair:

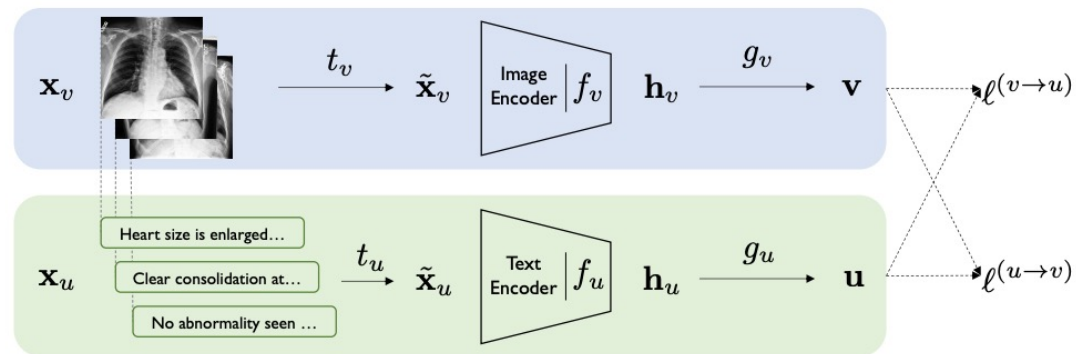
$$\ell_i^{(v \rightarrow u)} = -\log \frac{\exp(\langle \mathbf{v}_i, \mathbf{u}_i \rangle / \tau)}{\sum_{k=1}^N \exp(\langle \mathbf{v}_i, \mathbf{u}_k \rangle / \tau)}, \quad (2)$$

where $\langle \mathbf{v}_i, \mathbf{u}_i \rangle$ represents the cosine similarity, i.e., $\langle \mathbf{v}, \mathbf{u} \rangle = \mathbf{v}^\top \mathbf{u} / \|\mathbf{v}\| \|\mathbf{u}\|$; and $\tau \in \mathbb{R}^+$ represents a temperature parameter. This loss takes the same form as the InfoNCE loss

unlike previous work which use a contrastive loss between inputs of the same modality (Chen et al., 2020a; He et al., 2020), our image-to-text contrastive loss is asymmetric for each input modality. We therefore define a similar text-to-image contrastive loss as:

$$\ell_i^{(u \rightarrow v)} = -\log \frac{\exp(\langle \mathbf{u}_i, \mathbf{v}_i \rangle / \tau)}{\sum_{k=1}^N \exp(\langle \mathbf{u}_i, \mathbf{v}_k \rangle / \tau)}. \quad (3)$$

ConVIRT



$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \left(\lambda \ell_i^{(v \rightarrow u)} + (1 - \lambda) \ell_i^{(u \rightarrow v)} \right), \quad (4)$$

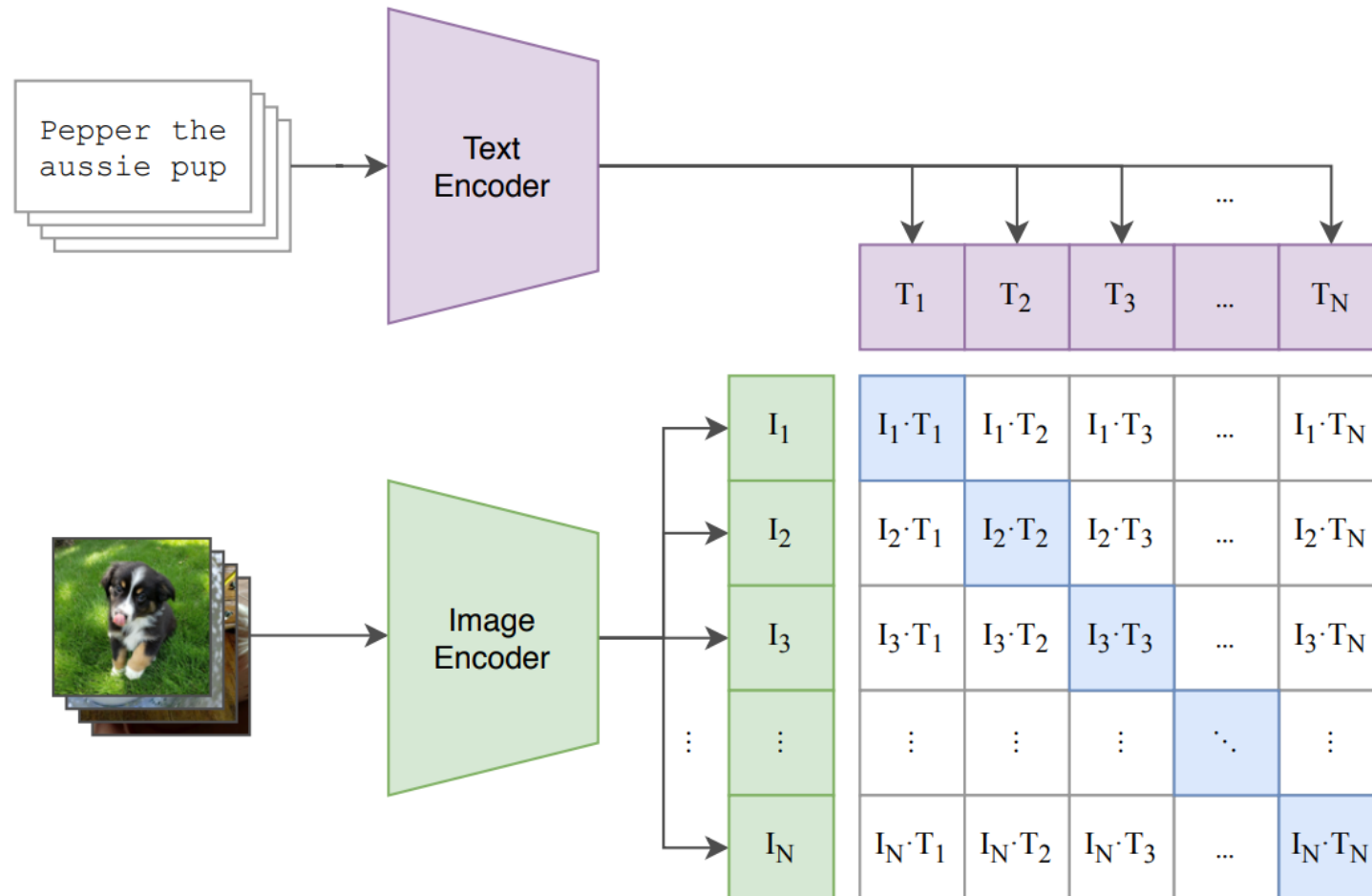
where $\lambda \in [0, 1]$ is a scalar weight.

CLIP

Alec Radford^{*1} Jong Wook Kim^{*1} Chris Hallacy¹ Aditya Ramesh¹ Gabriel Goh¹ Sandhini Agarwal¹
 Girish Sastry¹ Amanda Askell¹ Pamela Mishkin¹ Jack Clark¹ Gretchen Krueger¹ Ilya Sutskever¹

2021

(1) Contrastive pre-training



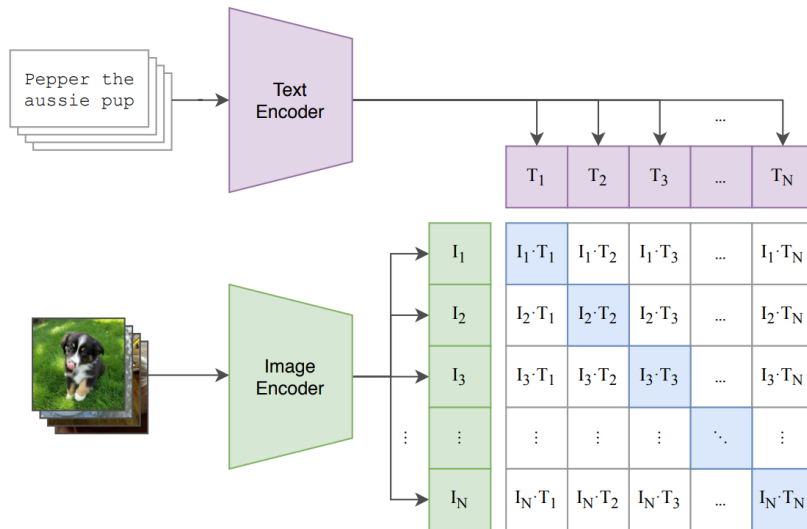
Learning Transferable Visual Models From Natural Language Supervision

CLIP

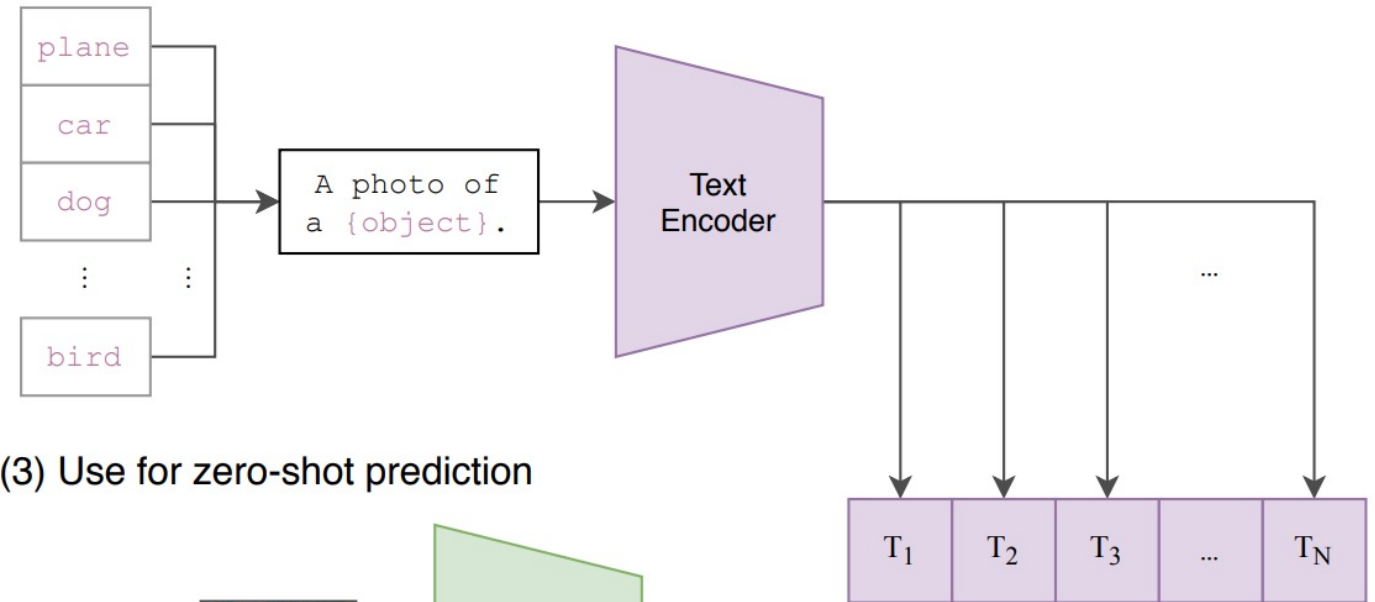
Alec Radford^{*1} Jong Wook Kim^{*1} Chris Hallacy¹ Aditya Ramesh¹ Gabriel Goh¹ Sandhini Agarwal¹
Girish Sastry¹ Amanda Askell¹ Pamela Mishkin¹ Jack Clark¹ Gretchen Krueger¹ Ilya Sutskever¹

2021

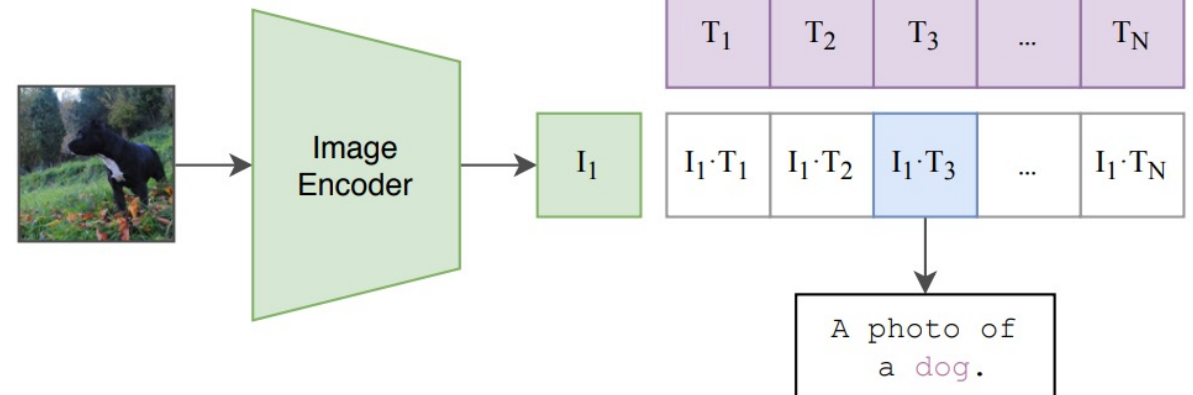
(1) Contrastive pre-training



(2) Create dataset classifier from label text



(3) Use for zero-shot prediction



Dataset Construction

CLIP: Dataset

- NL supervision requires a large-scale dataset.
 - MS-COCO, Visual Genome, .. are limited in size
 - YFCC100M: Includes 100M images but metadata for each image is sparse and low quality. When filtered, the # of images is around the size of ImageNet
 - The CLIP dataset includes 400M (image, text) pairs.
 - Similar to the size of WebText used to train GPT-2
- **Step 1: Raw Data.** Collect a raw, noisy set of billions of images with associated text.
 - **Step 2: Base queries (500K keywords).**
 - “The base query list is all words occurring at least 100 times in the English version of Wikipedia.”
 - “This is augmented with bi-grams with high pointwise mutual information as well as the names of all Wikipedia articles above a certain search volume.
 - “Finally all WordNet synsets not already in the query list are added.”
 - **Step 3: Search (filtering).** They kept noisy (image, text) pairs if text includes a keyword from the base query list.
 - **Step 4: Class balancing.** 20K (image, text) pairs are included for each query.

amounts of compute. Mahajan et al. (2018) required 19 GPU years to train their ResNeXt101-32x48d and Xie et al. (2020) required 33 TPUv3 core-years to train their Noisy Student EfficientNet-L2. When considering that both these systems were trained to predict only 1000 ImageNet classes, the task of learning an open set of visual concepts from natural language seems daunting. In the course of our ef-

Our initial approach, similar to VirTex, jointly trained an image CNN and text transformer from scratch to predict the caption of an image. However, we encountered difficulties efficiently scaling this method. In Figure 2 we show that a 63 million parameter transformer language model, which already uses twice the compute of its ResNet-50 image encoder, learns to recognize ImageNet classes three times slower than a much simpler baseline that predicts a bag-of-words encoding of the same text.

Both these approaches share a key similarity. They try to predict the *exact* words of the text accompanying each image. This is a difficult task due to the wide variety of descriptions, comments, and related text that co-occur with images. Recent work in contrastive representation learning for images has found that contrastive objectives can learn better representations than their equivalent predictive objective (Tian

CLIP: Efficiency

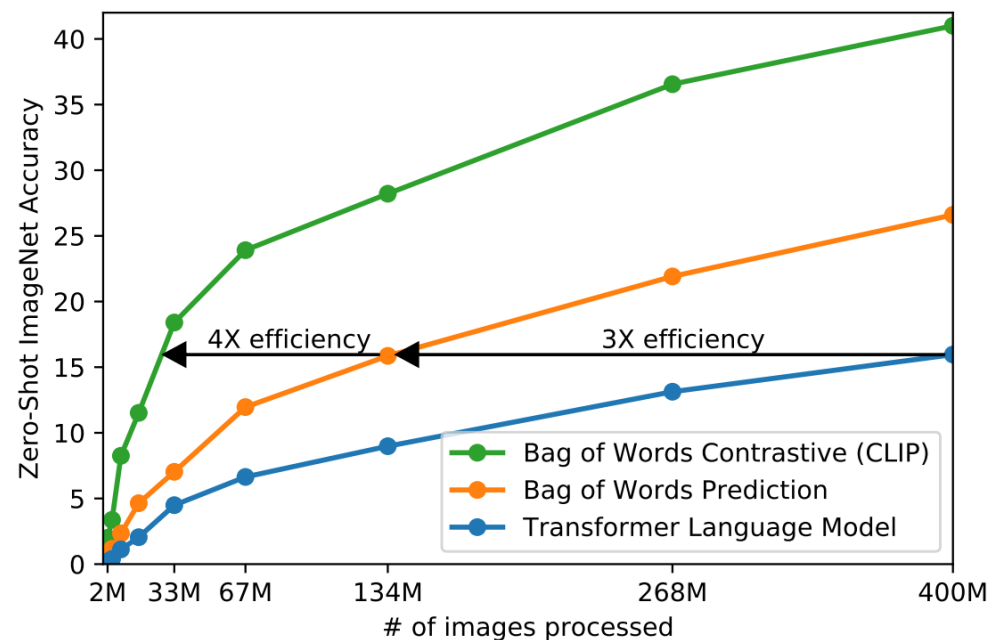
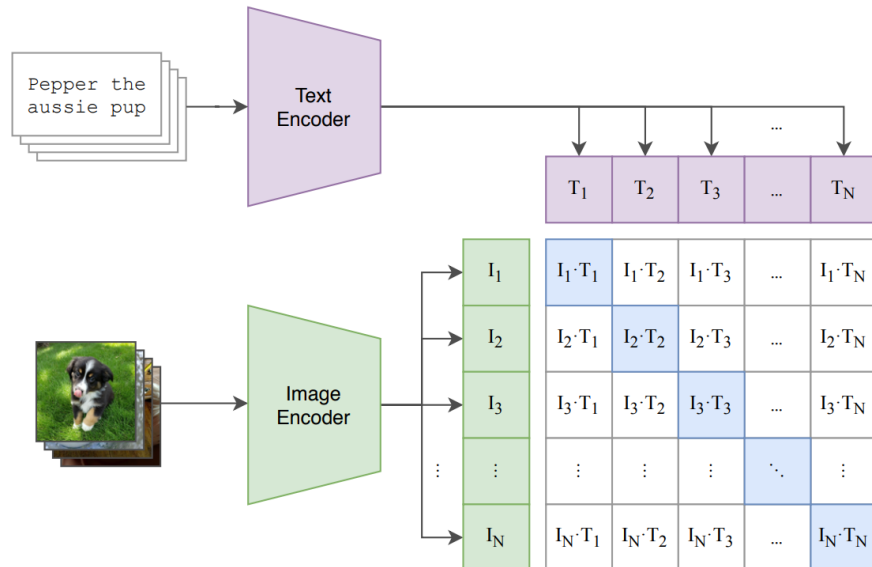


Figure 2. CLIP is much more efficient at zero-shot transfer than our image caption baseline. Although highly expressive, we found that transformer-based language models are relatively weak at zero-shot ImageNet classification. Here, we see that it learns 3x slower than a baseline which predicts a bag-of-words (BoW) encoding of the text (Joulin et al., 2016). Swapping the prediction objective for the contrastive objective of CLIP further improves efficiency another 4x.

CLIP: Method

$$\mathcal{L}_{\text{infoNCE}} = - \sum_{(i,j) \in \mathbb{P}} \log \left(\frac{e^{\text{CoSim}(\mathbf{z}_i, \mathbf{z}_j) / \tau}}{\sum_{k=1}^N e^{\text{CoSim}(\mathbf{z}_i, \mathbf{z}_k) / \tau}} \right),$$

(1) Contrastive pre-training



```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter
```

```
# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]
```

```
# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)
```

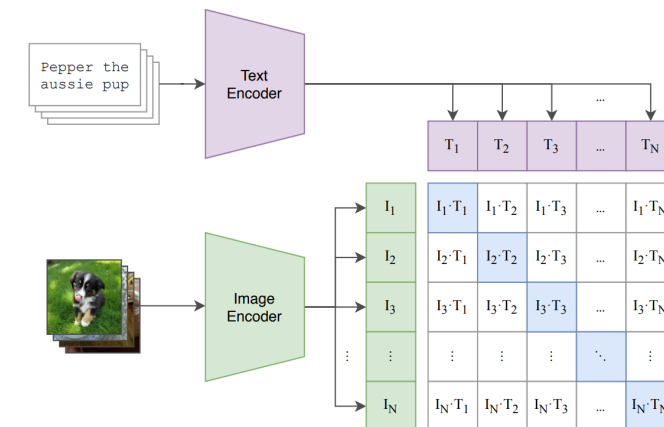
```
# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)
```

```
# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss    = (loss_i + loss_t) / 2
```

Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

CLIP: Architecture

(1) Contrastive pre-training



- Image Encoder

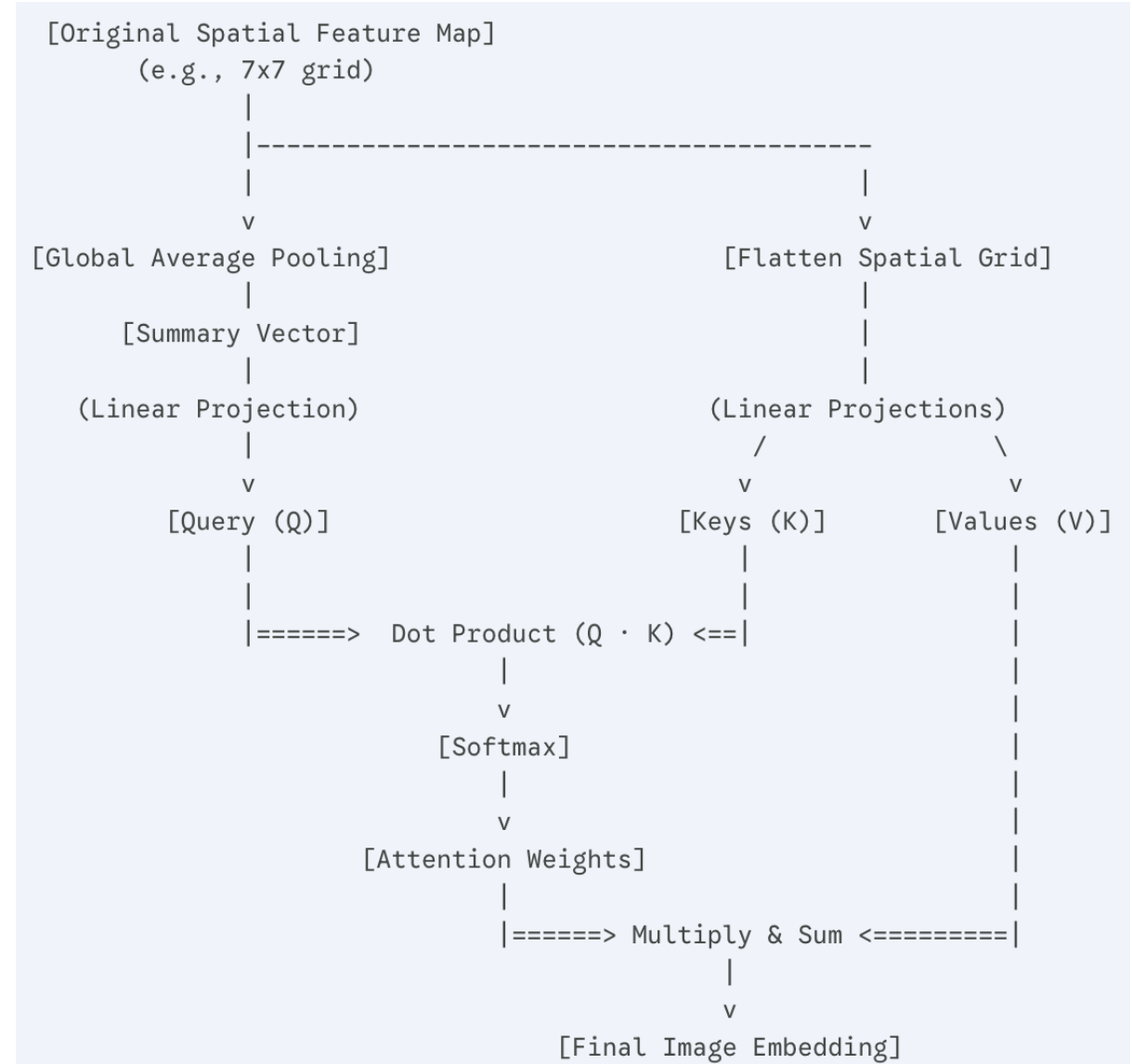
- Option 1: ResNet-D, some modifications over ResNet-50 + antialiased rect-2 blur pooling + **attention pooling** (instead of global average pooling)
- Option 2: ViT + additional layer norm after combining patch and position embeddings

- Text Encoder

- Transformer (same architecture as GPT-2)
- Different sizes (base version: 63M params, 12 layers, 512-wide model with 8 attention heads)
- Lowercased byte-pair encoding
- Max sequence length: Trimmed at 76
- [SOS] and [EOS] tokens

CLIP: ResNet-D's Attention Pooling

- “We also replace the global average pooling layer with an attention pooling mechanism.
- The attention pooling is implemented as a single layer of “transformer-style” multi-head QKV attention where the query is conditioned on the global average-pooled representation of the image.”



Generated with Gemini

CLIP: Training

- Train 5 ResNets, 3 ViTs:
 - ResNet-50, ResNet-101, 3 More following EfficientNet-style modifications on ResNet-50
 - ViT-B/32, ViT-B/16, ViT-L/14
- 32 epochs, Adam, weight decay (on all weights except for biases), LR with cosine scheduling
- Hyperparams:
 - Grid search, random search, manual tuning for 1 epoch for ResNet-50-baseline
 - Then based on heuristics adapted for larger models
- Temperature: Learnable, initialized at 0.07 and clipped to prevent explosion
- **Batchsize: 32,768.**
- “The largest ResNet model, RN50x64, **took 18 days to train on 592 V100 GPUs** while the largest Vision Transformer took **12 days on 256 V100 GPUs.**”

CLIP: Results

	aYahoo	ImageNet	SUN
Visual N-Grams	72.4	11.5	23.0
CLIP	98.4	76.2	58.5

Table 1. Comparing CLIP to prior zero-shot transfer image classification results. CLIP improves performance on all three datasets by a large amount. This improvement reflects many differences in the 4 years since the development of Visual N-Grams (Li et al., 2017).

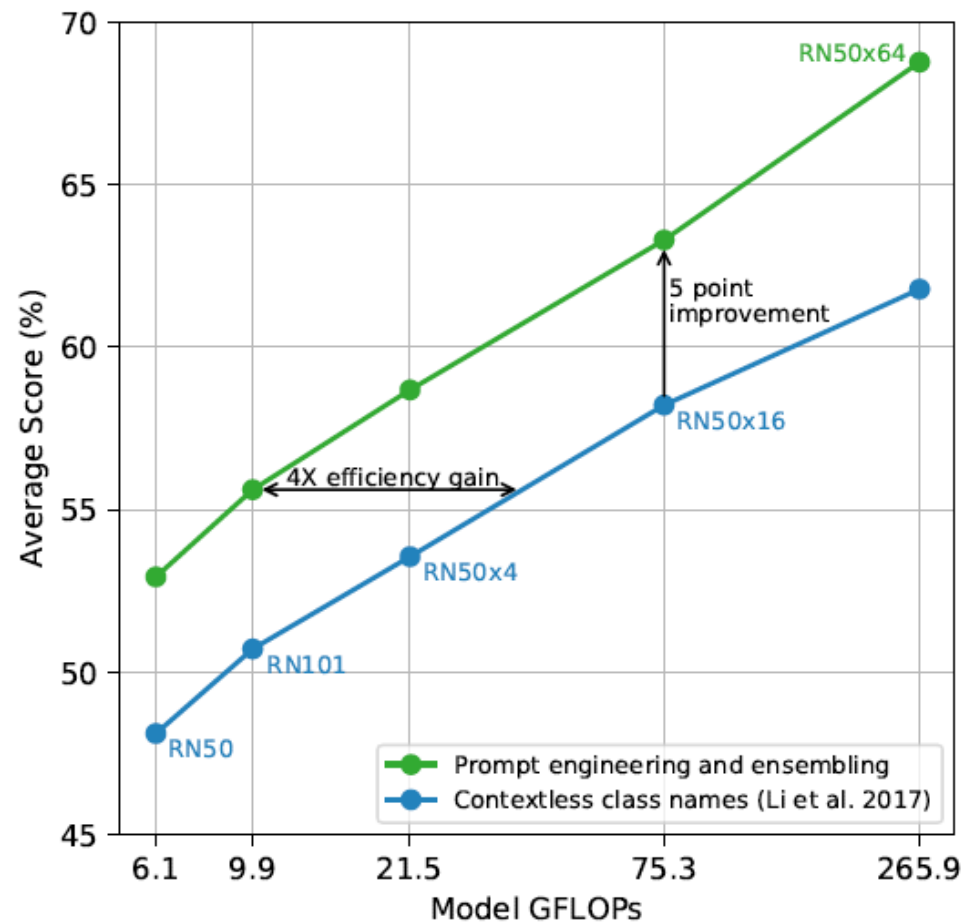


Figure 4. Prompt engineering and ensembling improve zero-shot performance. Compared to the baseline of using contextless class names, prompt engineering and ensembling boost zero-shot classification performance by almost 5 points on average across 36 datasets. This improvement is similar to the gain from using 4 times more compute with the baseline zero-shot method but is “free” when amortized over many predictions.

CLIP: Results

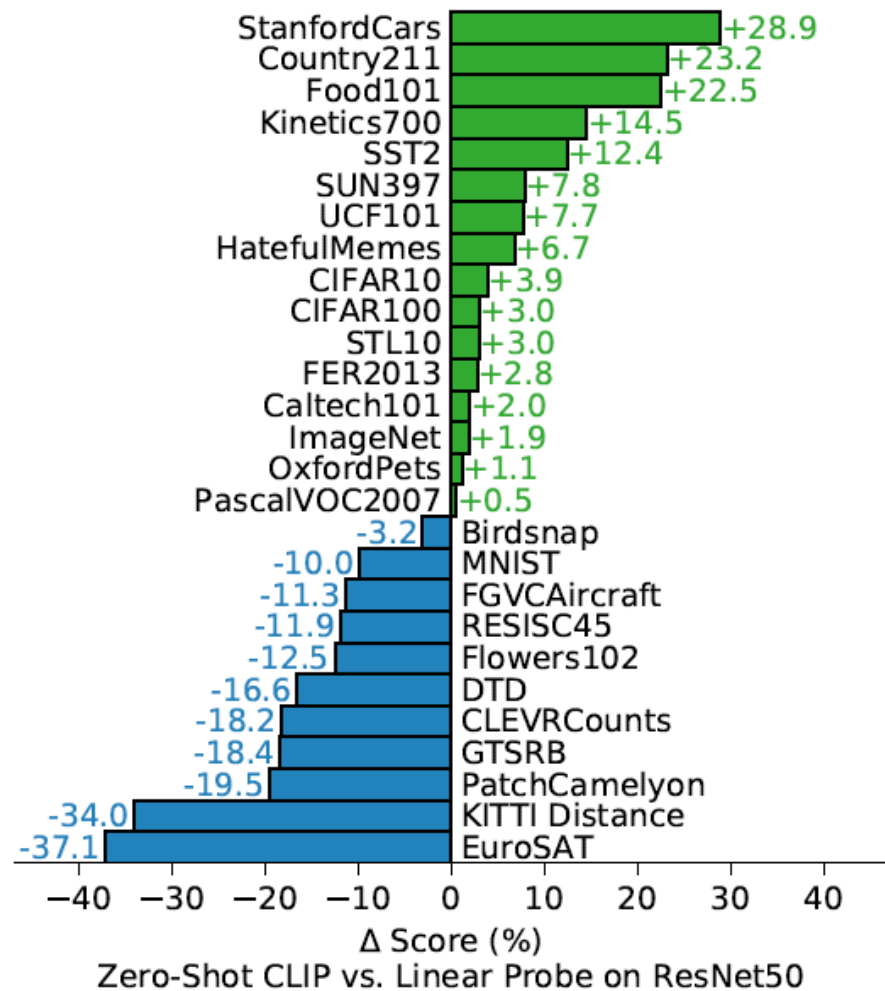


Figure 5. Zero-shot CLIP is competitive with a fully supervised baseline. Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

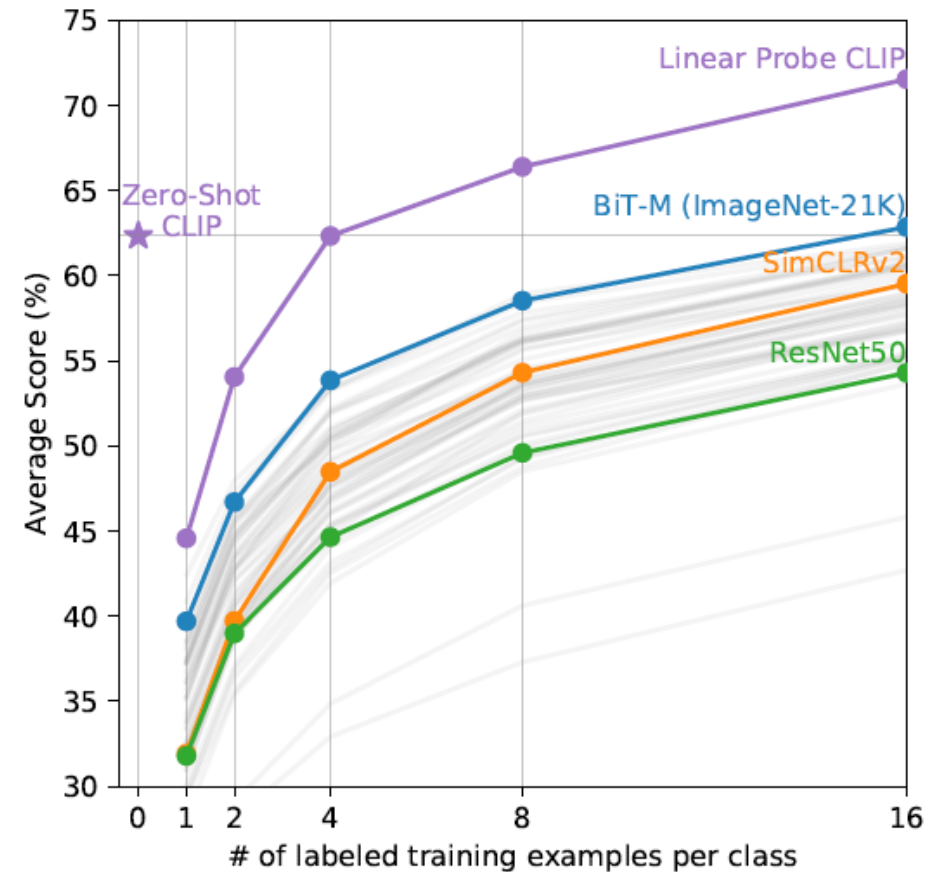


Figure 6. Zero-shot CLIP outperforms few-shot linear probes. Zero-shot CLIP matches the average performance of a 4-shot linear classifier trained on the same feature space and nearly matches the best results of a 16-shot linear classifier across publicly available models. For both BiT-M and SimCLRv2, the best performing model is highlighted. Light gray lines are other models in the eval suite. The 20 datasets with at least 16 examples per class were used in this analysis.

CLIP: Extensions

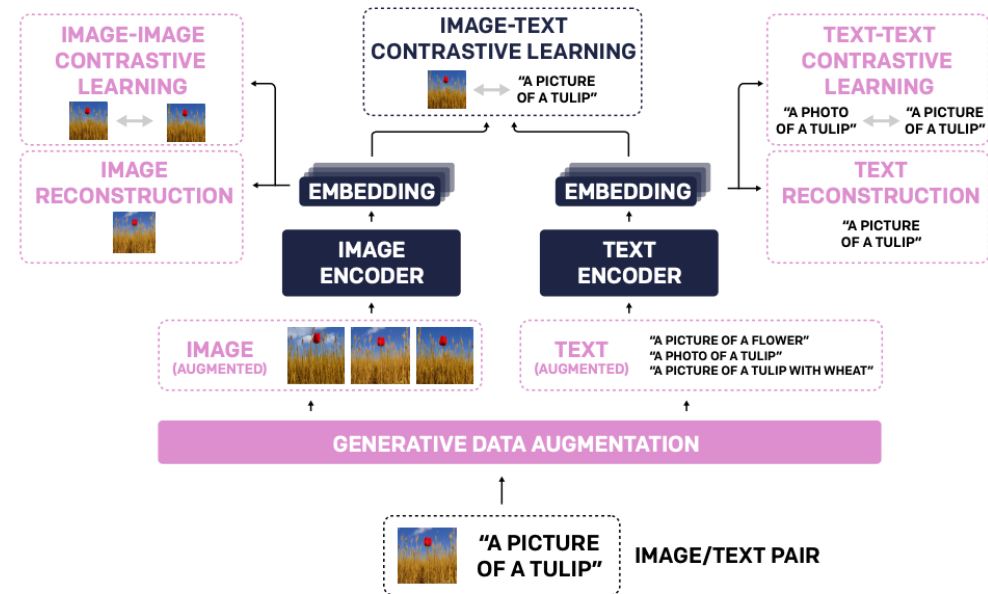
- SigLIP (Zhai et al., 2023)
 - CLIP with NCE Loss with binary CE instead of CLIP's multi-class InfoNCE
 - Better zero-shot performance on smaller batch sizes
- SLIP
 - CLIP combined with Self-Supervised Learning

TULIP: Towards Unified Language-Image Pretraining

Zineng Tang, Long Lian, Seun Eisape, XuDong Wang,
Roei Herzig, Adam Yala, Alane Suhr, Trevor Darrell, David M. Chan
University of California, Berkeley

Abstract

Despite the recent success of image-text contrastive models like CLIP and SigLIP, these models often struggle with vision-centric tasks that demand high-fidelity image understanding, such as counting, depth estimation, and fine-grained object recognition. These models, by performing language alignment, tend to prioritize high-level semantics over visual understanding, weakening their image understanding. On the other hand, vision-focused models are great at processing visual information but struggle to understand language, limiting their flexibility for language-



Perception Encoder

Bolya et al., “Perception Encoder: The best visual embeddings are not at the output of the network”, 2025.

- “Traditionally, vision encoders have relied on a variety of pretraining objectives, each tailored to specific downstream tasks such as classification, captioning, or localization. Surprisingly, after scaling our carefully tuned image pretraining recipe and refining with our robust video data engine, **we find that contrastive vision-language training alone can produce strong, general embeddings for all of these downstream tasks.**
- There is only one caveat: **these embeddings are hidden within the intermediate layers of the network. To draw them out, we introduce two alignment methods:**
 - language alignment for multimodal language modeling, and
 - spatial alignment for dense prediction.”

Perception Encoder

Bolya et al., “Perception Encoder: The best visual embeddings are not at the output of the network”, 2025.

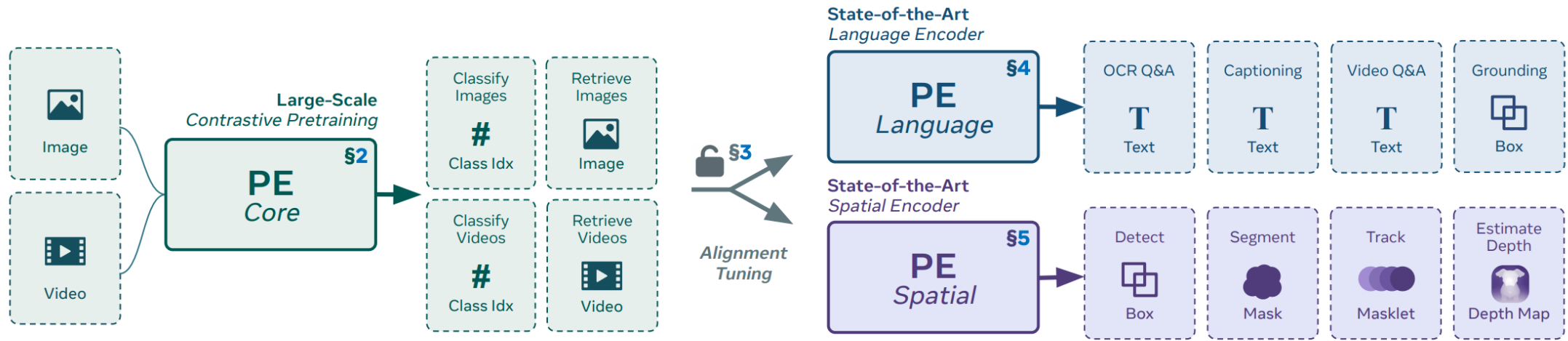
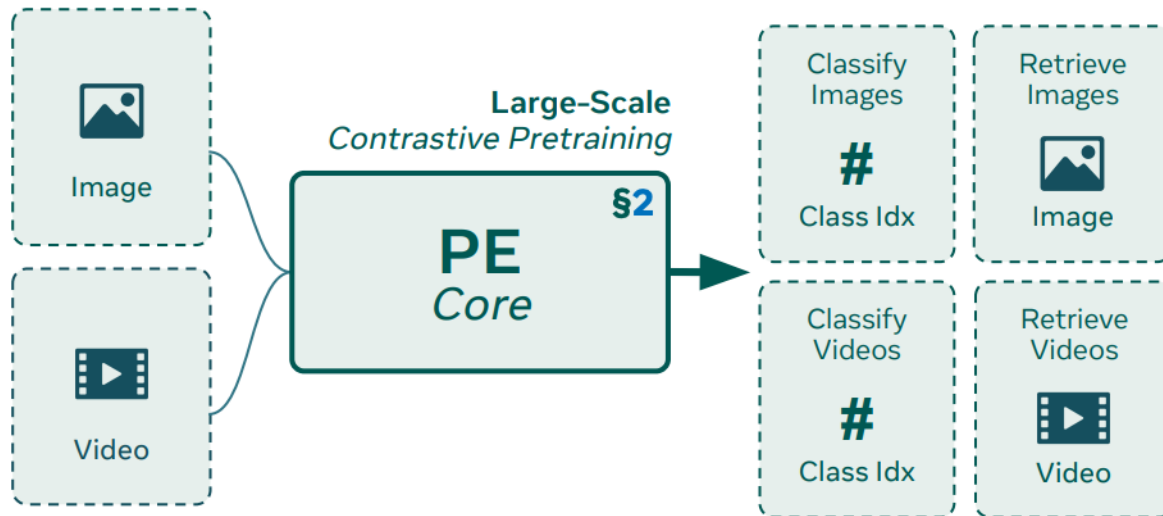


Figure 1 Perception Encoder (PE) is a family of large-scale vision encoder models with state-of-the-art performance on a large variety of vision tasks. By using a robust contrastive pretraining recipe and finetuning on synthetically aligned videos, PE not only outperforms all existing models on classification and retrieval (§2), but it also internally produces strong, general features that *scale* for downstream tasks (§3). PE unlocks the ability for large-scale contrastive pretraining to transfer to downstream tasks with alignment tuning to capitalize on those general features (§4, §5).

Perception Encoder

Bolya et al., “Perception Encoder: The best visual embeddings are not at the output of the network”, 2025.



- “we first focus on developing a strong image-only contrastive pretraining recipe to extract general knowledge from billion-scale image-text data.
- We then use the resulting model as a frame-based encoder to develop a video data engine for generating well-aligned video captions. Finetuning on this synthetic video-text data substantially improves performance on both image and video classification and retrieval tasks.
- Finally, we scale our robust image pretraining and well-aligned video finetuning strategy to 2B parameters to produce PEcoreG (§2.4), a single unified encoder that outperforms SigLIP2 [138] on zero-shot image tasks and InternVideo2 [146] on most zero-shot video tasks.”

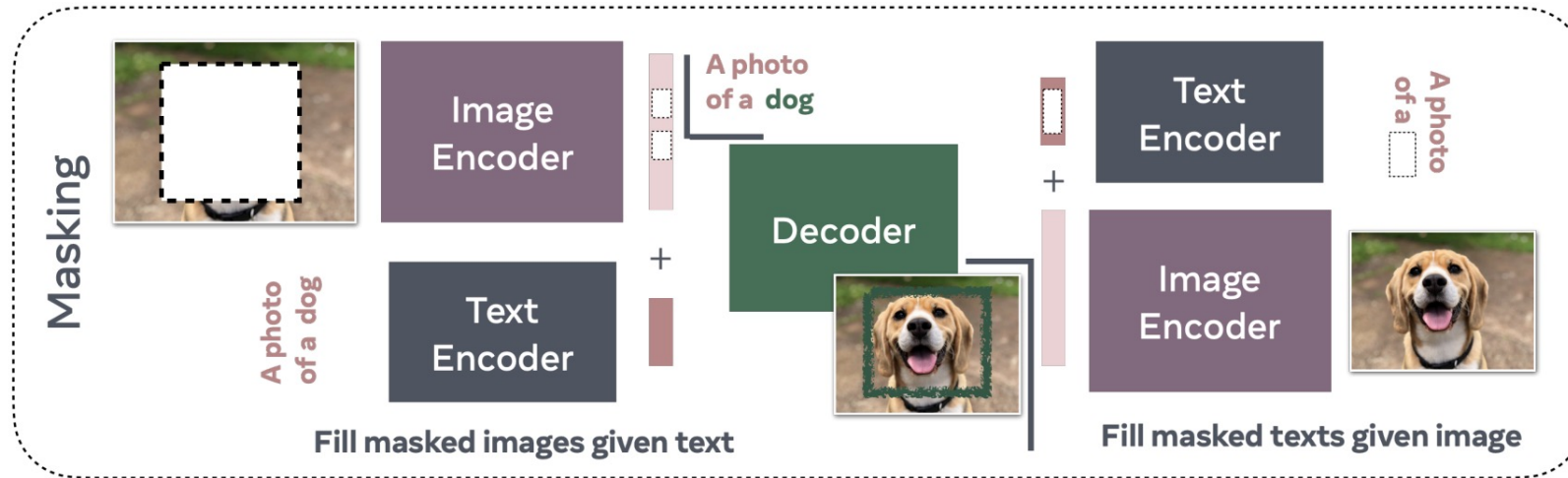


Fig: Bordes et al., "An Introduction to Vision-Language Modeling", 2024.

Masking Approaches

FLAVA

Foundational Language And Vision Alignment (FLAVA)
[Singh et al., 2022].

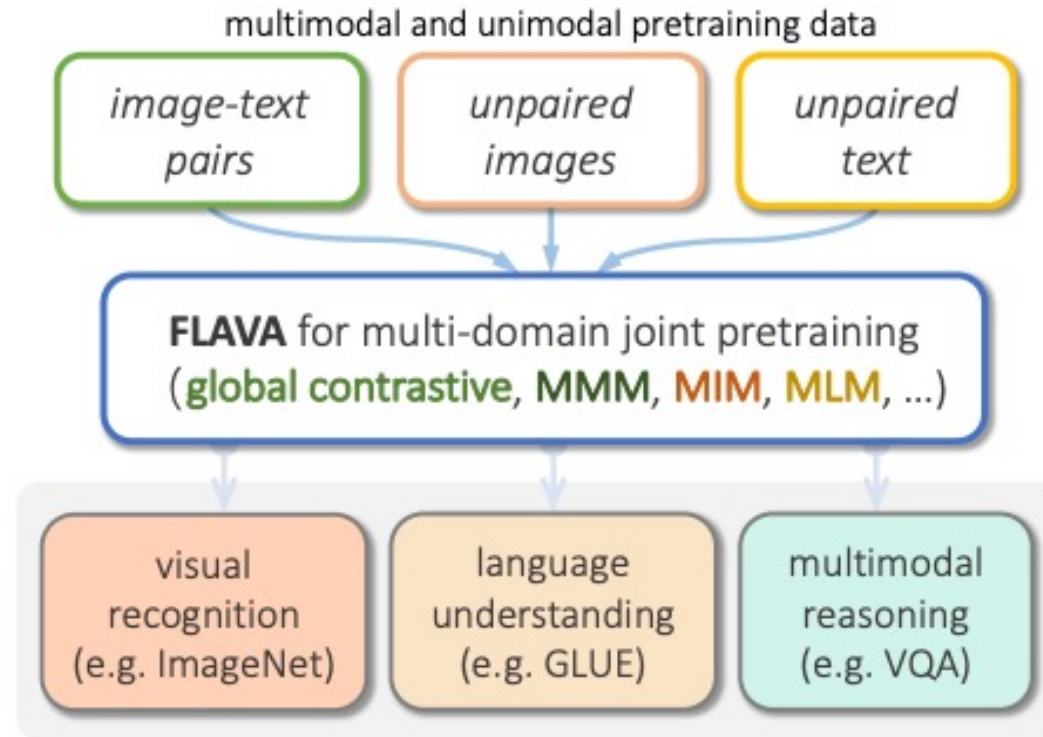


Figure 1. We present FLAVA, a language and vision alignment model that learns strong representations from multimodal (image-text pairs) and unimodal data (unpaired images and text) and can be applied to target a broad scope of tasks from three domains (visual recognition, language understanding, and multimodal reasoning) under a common transformer model architecture.

FLAVA

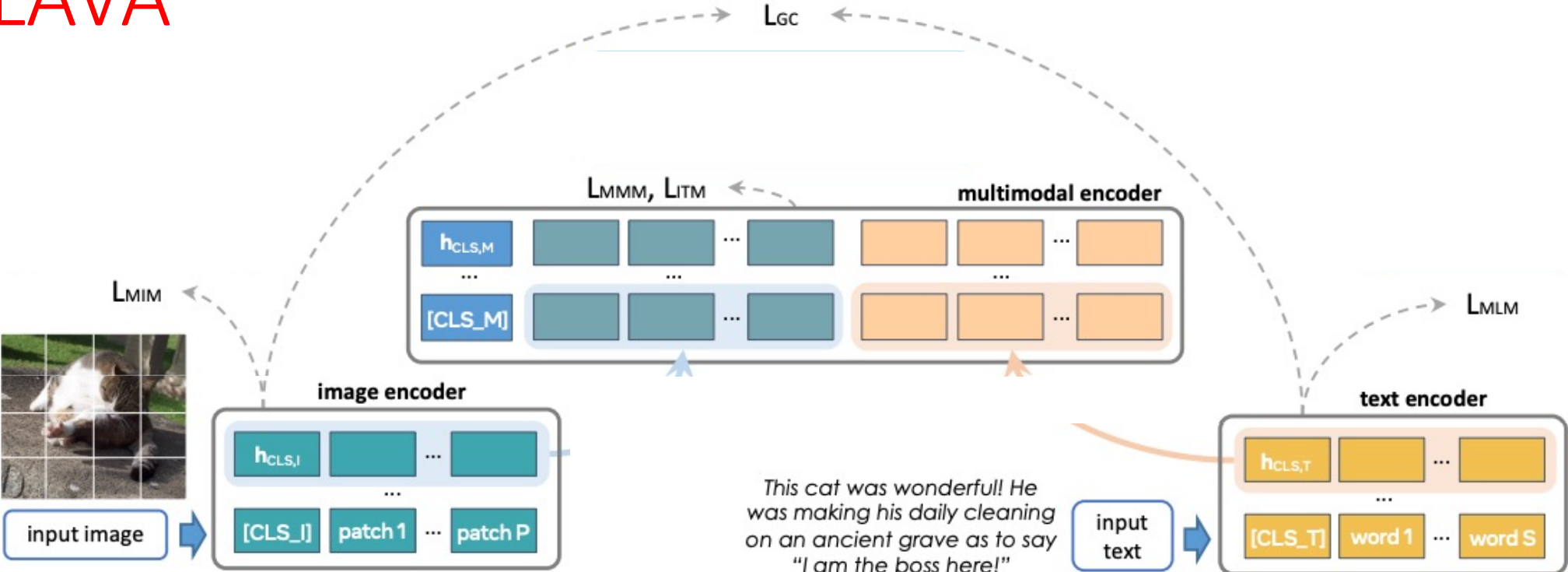


Figure 2. **An overview of our FLAVA model**, with an image encoder transformer to capture unimodal image representations, a text encoder transformer to process unimodal text information, and a multimodal encoder transformer that takes as input the encoded unimodal image and text and integrates their representations for multimodal reasoning. **During pretraining**, masked image modeling (MIM) and mask language modeling (MLM) losses are applied onto the image and text encoders over a single image or a text piece, respectively, while contrastive, masked multimodal modeling (MMM), and image-text matching (ITM) loss are used over paired image-text data. **For downstream tasks**, classification heads are applied on the outputs from the image, text, and multimodal encoders respectively for visual recognition, language understanding, and multimodal reasoning tasks.

FLAVA

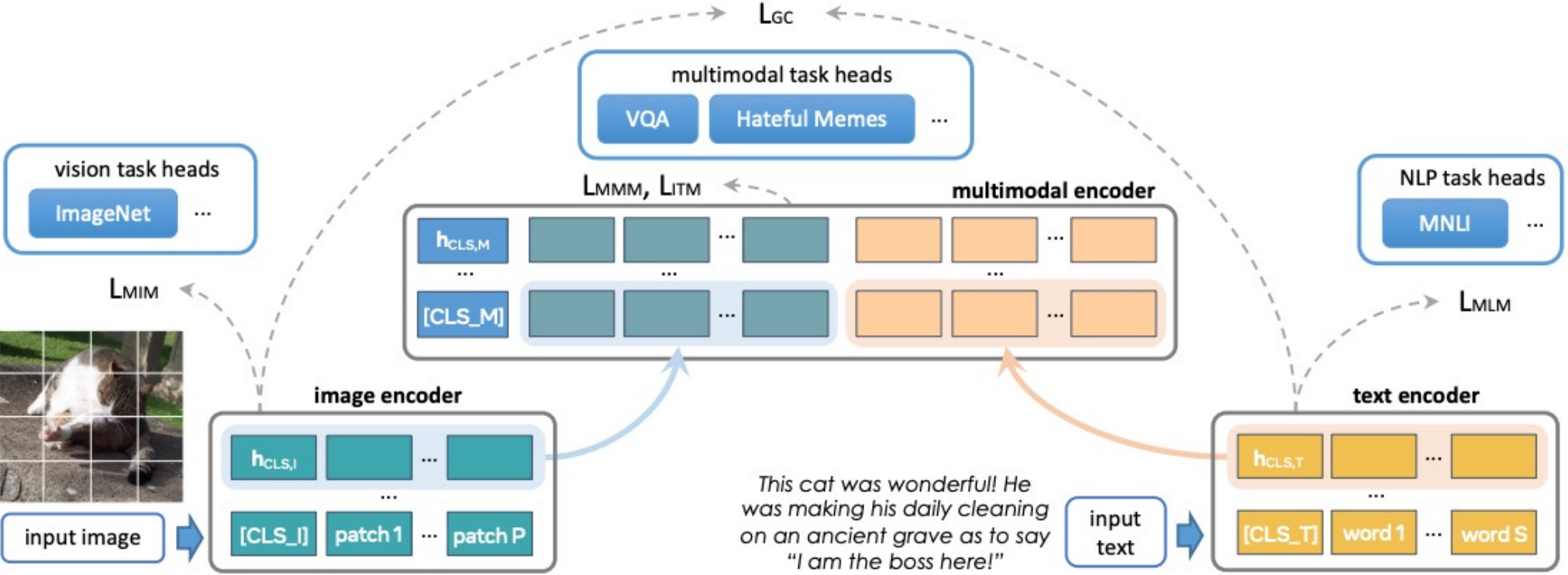


Figure 2. **An overview of our FLAVA model**, with an image encoder transformer to capture unimodal image representations, a text encoder transformer to process unimodal text information, and a multimodal encoder transformer that takes as input the encoded unimodal image and text and integrates their representations for multimodal reasoning. **During pretraining**, masked image modeling (MIM) and mask language modeling (MLM) losses are applied onto the image and text encoders over a single image or a text piece, respectively, while contrastive, masked multimodal modeling (MMM), and image-text matching (ITM) loss are used over paired image-text data. **For downstream tasks**, classification heads are applied on the outputs from the image, text, and multimodal encoders respectively for visual recognition, language understanding, and multimodal reasoning tasks.

FLAVA: Results

Method	Vision Avg.	NLP Avg.	Multi-modal Avg.	Macro Avg.
1 MIM	57.46	–	–	19.15
2 MLM	–	71.55	–	23.85
3 FLAVA _C	64.80	79.14	66.25	70.06
4 FLAVA _{MM}	74.22	79.35	69.11	74.23
5 FLAVA w/o unimodal init	75.55	78.29	67.32	73.72
6 FLAVA	78.19	79.44	69.92	75.85

Table 3. Our full FLAVA pretraining (row 6) achieves the best average scores on vision, language, and multimodal tasks compared to ablations. Row 1 to 4 are pretrained on PMD while row 5 and 6 also involve unimodal IN-1k, CCNews, and BookCorpus datasets.

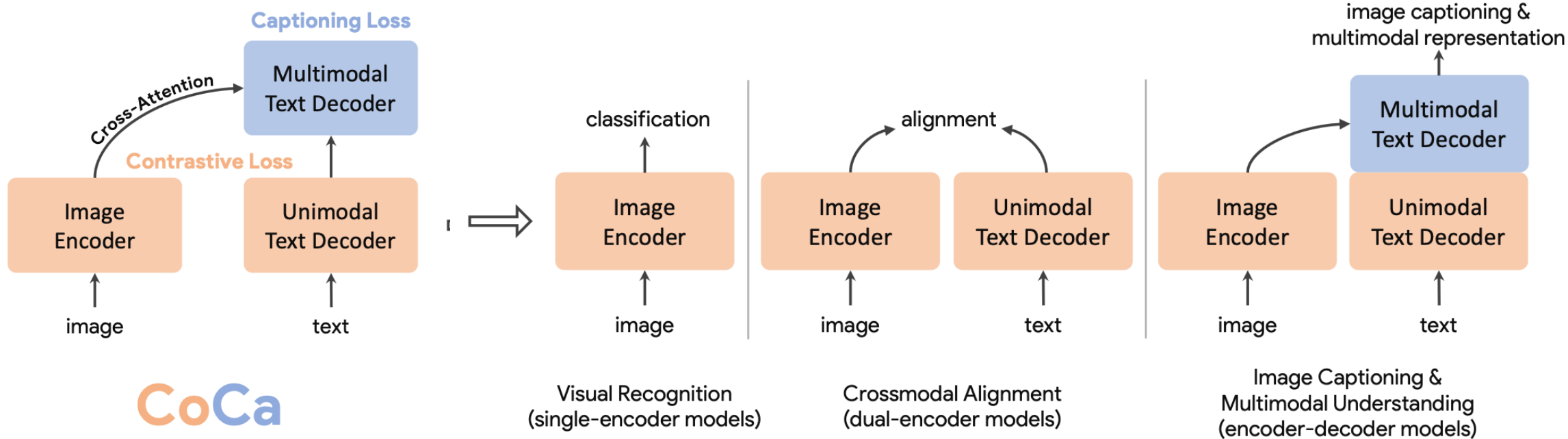


Fig: Bordes et al., "An Introduction to Vision-Language Modeling", 2024.

Generative Approaches

CoCa

Contrastive Captioner (CoCa),
Yu et al., 2022.

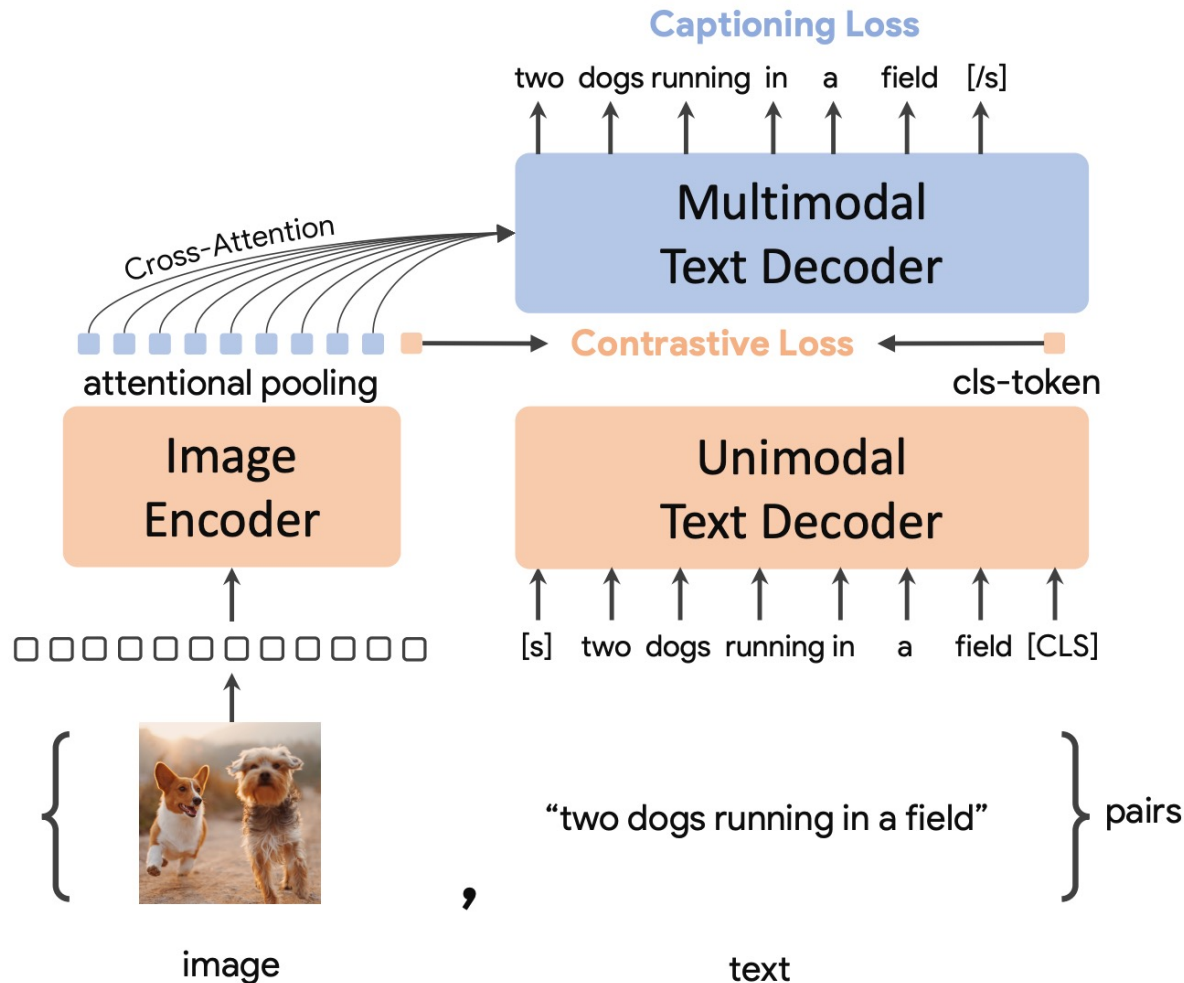


CoCa
Pretraining

Zero-shot, frozen-feature or finetuning

CoCa

Contrastive Captioner (CoCa),
Yu et al., 2022.



Single-Encoder Classification. The classic single-encoder approach pretrains a visual encoder through image classification on a large crowd-sourced image annotation dataset (*e.g.*, ImageNet [9], Instagram [20] or JFT [21]), where the vocabulary of annotation texts is usually fixed. These image annotations are usually mapped into discrete class vectors to learn with a cross-entropy loss as

$$\mathcal{L}_{\text{Cls}} = -p(y) \log q_{\theta}(x), \quad (1)$$

Dual-Encoder Contrastive Learning. Compared to pretraining with single-encoder classification, which requires human-annotated labels and data cleaning, the dual-encoder approach exploits noisy web-scale text descriptions and introduces a learnable text tower to encode free-form texts. The two encoders are jointly optimized by contrasting the paired text against others in the sampled batch:

$$\mathcal{L}_{\text{Con}} = -\frac{1}{N} \left(\underbrace{\sum_i \log \frac{\exp(x_i^{\top} y_i / \sigma)}{\sum_{j=1}^N \exp(x_i^{\top} y_j / \sigma)}}_{\text{image-to-text}} + \underbrace{\sum_i \log \frac{\exp(y_i^{\top} x_i / \sigma)}{\sum_{j=1}^N \exp(y_i^{\top} x_j / \sigma)}}_{\text{text-to-image}} \right), \quad (2)$$

where x_i and y_j are normalized embeddings of the image in the i -th pair and that of the text in the j -th pair. N is the batch size, and σ is the temperature to scale the logits. In addition to the image

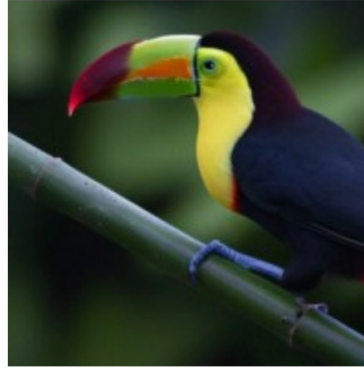
Encoder-Decoder Captioning. While the dual-encoder approach encodes the text as a whole, the generative approach (*a.k.a.* captioner) aims for detailed granularity and requires the model to predict the exact tokenized texts of y autoregressively. Following a standard encoder-decoder architecture, the image encoder provides latent encoded features (*e.g.*, using a Vision Transformer [39] or ConvNets [40]) and the text decoder learns to maximize the conditional likelihood of the paired text y under the forward autoregressive factorization:

$$\mathcal{L}_{\text{Cap}} = -\sum_{t=1}^T \log P_{\theta}(y_t | y_{<t}, x). \quad (3)$$

Chameleon

Chameleon: Mixed-Modal Early-Fusion Foundation Models, Meta, 2024.

Prompt: I'm bored. Could you show me some cool, quirky-looking birds? Also tell me about them in 2-3 sentences.



Response: Absolutely! Here are three birds with some unique features that you might find interesting:

Keel-Billed Toucan The Keel-Billed Toucan is a brightly colored bird native to South American rainforests and woodlands. Its most distinctive feature is its colorful beak, which can grow up to seven inches long and has a distinctive, curved shape. The beak is used for plucking fruit from trees and for social displays. Despite its large size, the Keel-Billed Toucan is a skilled flyer and can often be seen flitting through the treetops. Here is an image of this vibrant bird: ****

Puffin Puffins are small, aquatic birds that are found in the northern hemisphere. They have distinctive black and white feathers, as well as a brightly colored beak that is used for attracting mates and defending their territory. Puffins are known for their distinctive "puffin" call, which sounds like a loud, guttural chuckle. They feed on fish, which they catch by diving into the water from the air. Puffins are colonial birds, and they breed in large colonies on rocky cliffs and islands. Here is an image of this adorable bird: ****



Golden Pheasant The Golden Pheasant is a brightly colored bird native to the forests of China. It is a member of the pheasant family and is known for its striking plumage, which features a golden-yellow body, red face, and green tail feathers. The Golden Pheasant is a ground-dwelling bird, and it feeds on a diet of insects, seeds, and fruits. It is a popular game bird, and is often hunted for its meat and feathers. In captivity, the Golden Pheasant is sometimes kept as an ornamental bird, due to its striking appearance. Here is an image of this beautiful bird: ****

I hope these images and descriptions have helped to brighten your day! If you have any other questions or requests, please don't hesitate to ask.

Chameleon

Chameleon: Mixed-Modal Early-Fusion Foundation Models, Meta, 2024.

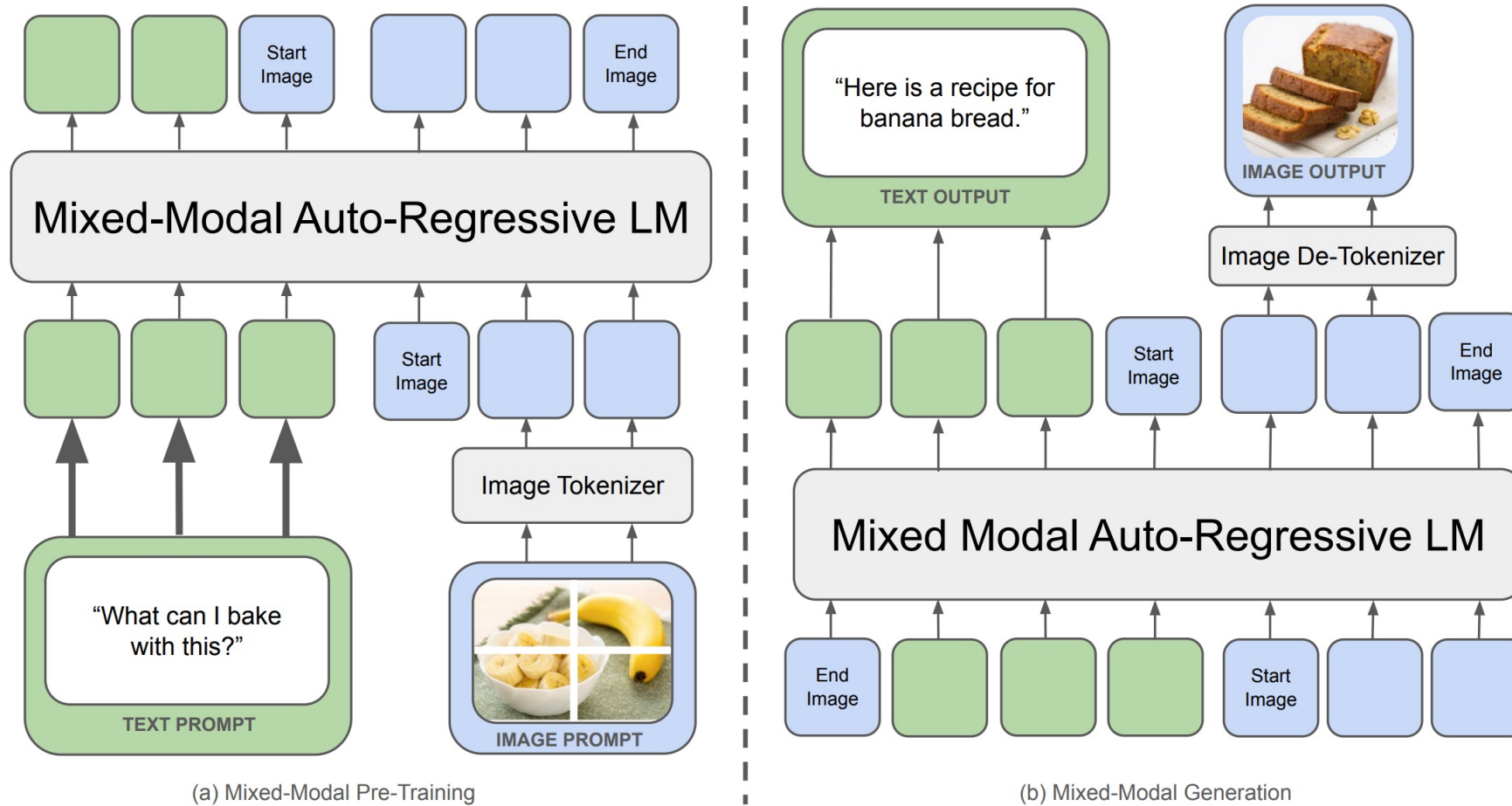


Figure 1 Chameleon represents all modalities — images, text, and code, as discrete tokens and uses a uniform transformer-based architecture that is trained from scratch in an end-to-end fashion on $\sim 10T$ tokens of interleaved mixed-modal data. As a result, Chameleon can both reason over, as well as generate, arbitrary mixed-modal documents. Text tokens are represented in green and image tokens are represented in blue.

Chameleon

2 Pre-Training

Chameleon represents images, in addition to text, as a series of discrete tokens and takes advantage of the scaling properties of auto-regressive Transformers (Ramesh et al., 2021; Aghajanyan et al., 2022, 2023; Yu et al., 2023). We present any ordering of images and text during training ranging from text-only, to single text/image pairs to full interleaved text-image documents.



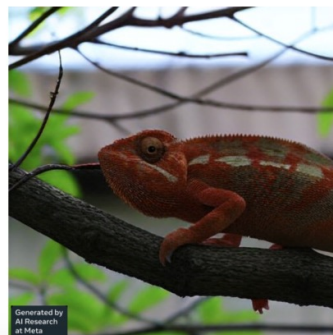
Prompt: `` While chameleons are masters of camouflage in the natural world, let's see how they might fare in a man-made setting.

1. Analyze the colors, patterns, and textures in this image and identify the most challenging spot for a chameleon to blend in.
2. Then generate an image of a chameleon with the optimal pattern to camouflage itself.
3. Then, rate and explain how successful each attempt is on a scale from 1-10.

Response: The image is of a stop sign in a forested area, with some sunlight filtering through the trees. The sign has a red octagonal shape with a white border and the word "STOP" written in white letters. In the background, there are green leaves and trees.

Challenging Spot for a Chameleon: The stop sign would be the most challenging spot for a chameleon to blend in due to its bright red color and bold white letters, which would stand out in the forested environment.

Optimal Pattern for Camouflage: To camouflage itself on the stop sign, a chameleon would need to have a pattern that would break up its outline and make it less conspicuous. A pattern with shades



of brown and red would help the chameleon blend in more effectively. Additionally, a mottled or dappled pattern would help to scatter light and further reduce the chameleon's visibility.

``

Rating and Explanation: A rating of 2/10 would be given for the chameleon's ability to blend in on the stop sign. While the chameleon's green and brown pattern helps to break up its outline, the bright red color of the stop sign will still make it stand out in the forested environment.



Fig: Bordes et al., "An Introduction to Vision-Language Modeling", 2024.

Approaches Using Pre-trained Backbones

Motivation

- Large-scale pretrained networks already contain rich, useful and similar representations.

The Platonic Representation Hypothesis

2024

Minyoung Huh^{*1} Brian Cheung^{*1} Tongzhou Wang^{*1} Phillip Isola^{*1}

CENG501

The Platonic Representation Hypothesis

Neural networks, trained with different objectives on different data and modalities, are converging to a shared statistical model of reality in their representation spaces.

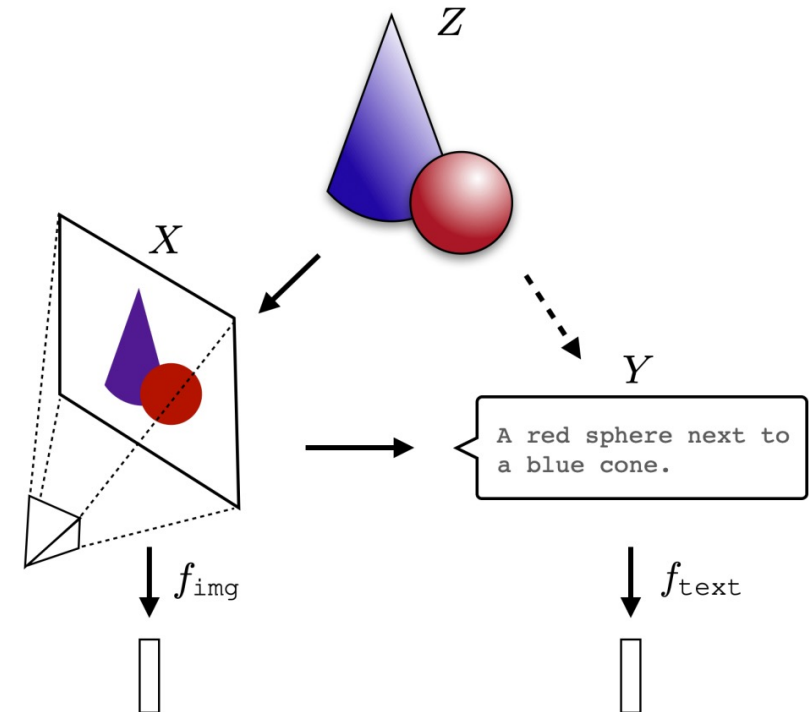


Figure 1. The Platonic Representation Hypothesis: Images (X) and text (Y) are projections of a common underlying reality (Z). We conjecture that representation learning algorithms will converge on a shared representation of Z , and scaling model size, as well as data and task diversity, drives this convergence.

Frozen

Multimodal Few-Shot Learning with Frozen Language Models, 2021



Figure 1: Curated samples with about five seeds required to get past well-known language model failure modes of either repeating text for the prompt or emitting text that does not pertain to the image. These samples demonstrate the ability to generate open-ended outputs that adapt to both images and text, and to make use of facts that it has learned during language-only pre-training.

Frozen

Multimodal Few-Shot Learning with
Frozen Language Models, 2021

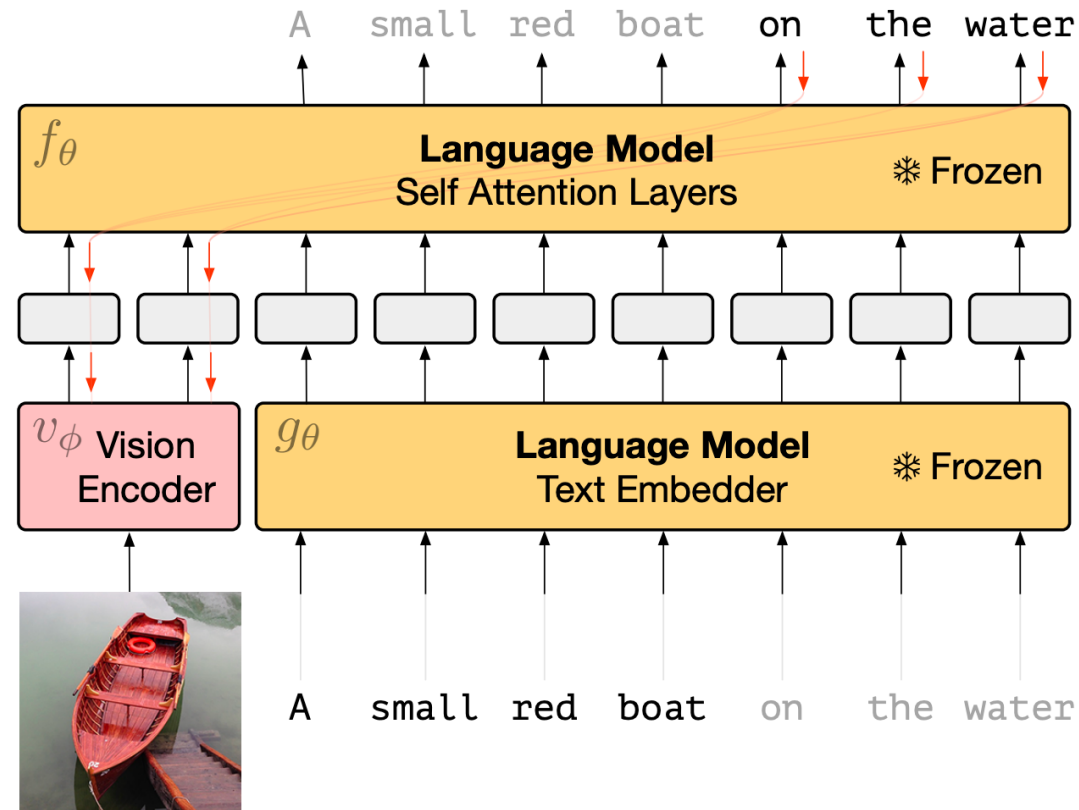


Figure 2: Gradients through a frozen language model's self attention layers are used to train the vision encoder.

Frozen

Multimodal Few-Shot Learning with Frozen Language Models, 2021

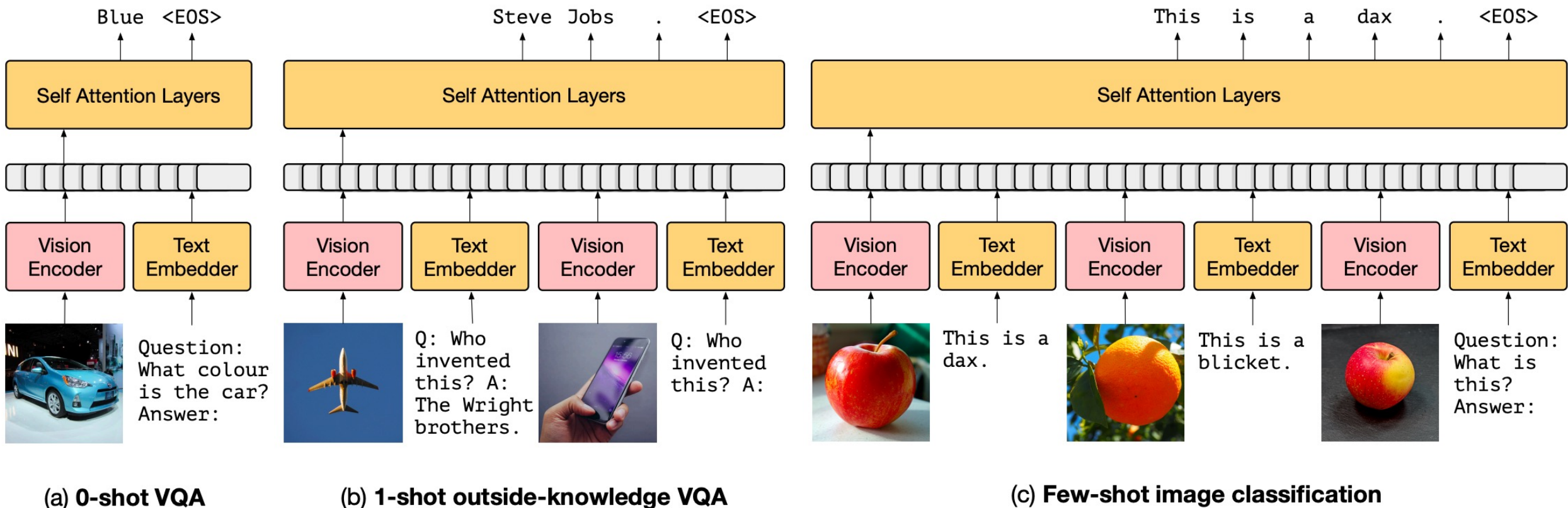


Figure 3: Inference-Time interface for *Frozen*. The figure demonstrates how we can support (a) visual question answering, (b) outside-knowledge question answering and (c) few-shot image classification via in-context learning.

Frozen

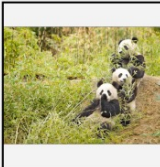
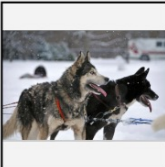
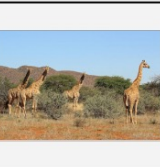








Multimodal Few-Shot Learning with
Frozen Language Models, 2021

Vision Encoder Our vision encoder is based on NF-ResNet-50 [3]. We define v_ϕ as a function that takes a raw image and emits a continuous sequence to be consumed by the transformer. We use the final output vector of the NF-Resnet *after* the global pooling layer.

Visual Prefix One important requirement is to represent images in a form that the transformer *already* understands: a sequence of continuous embeddings, each having the same dimensionality D as a token embedding t_l . We therefore form the visual prefix by linearly mapping the vision encoder's output to $D * n$ channels, and then reshaping the result as a sequence of n embeddings, each with dimensionality D . We call this sequence a *visual prefix* since it plays the same functional role in the transformer architecture as (part of) an embedding sequence of prefix tokens. We experimented using different number of tokens, specifically 1, 2 and 4 and found that 2 performs best, though certainly this would be sensitive to other architectural details. See Appendix for more details on the architecture.

Flamingo

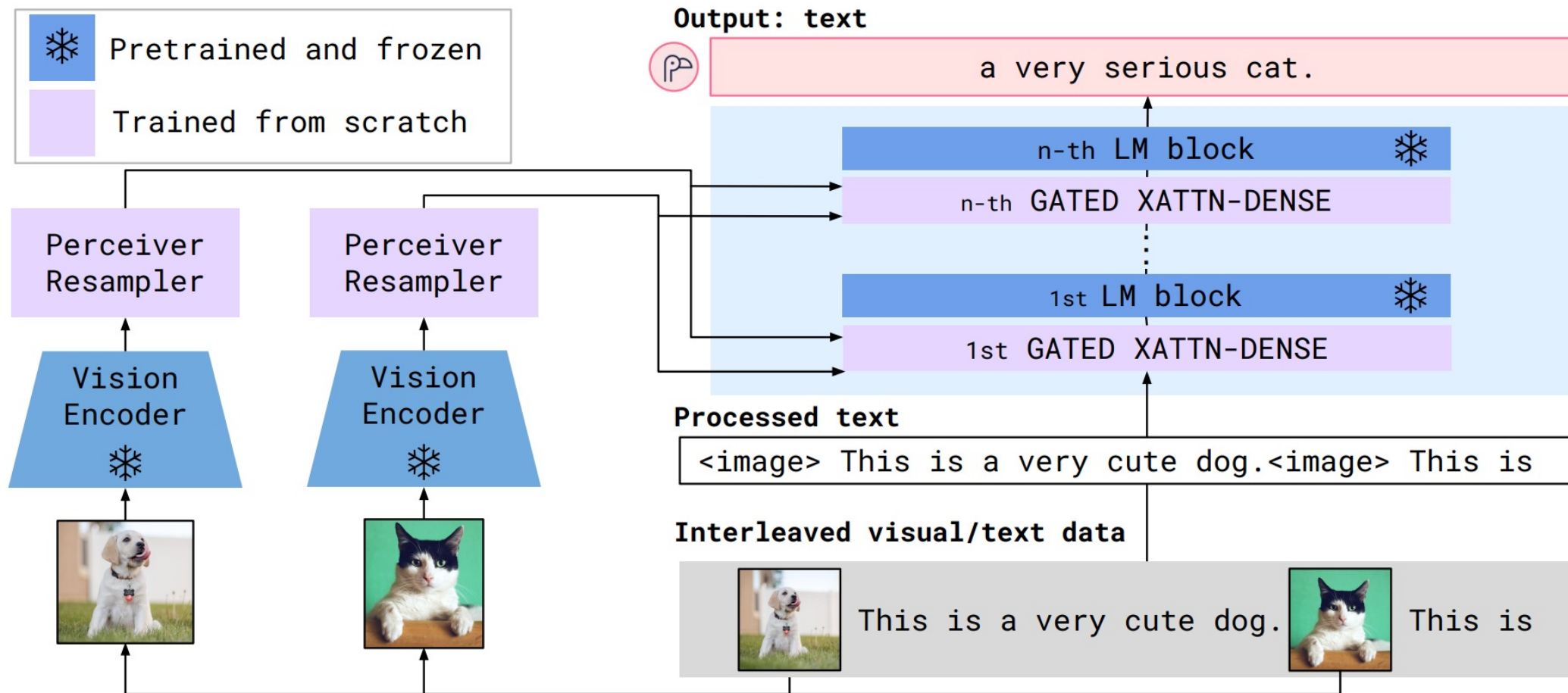
Flamingo: a Visual Language Model for Few-Shot Learning, Deepmind, 2022

	<p>pandas: 3</p>		<p>dogs: 2</p>			<p>giraffes: 4</p>
<p>I like reading</p>		<p>, my favourite play is Hamlet. I also like</p>		<p>, my favorite book is</p>	<p>Dreams from my Father.</p>	
					<p>What happens to the man after hitting the ball? Answer:</p>	<p>he falls down.</p>
 <p>This is a picture of two teddy bears on the moon.</p> <p>What are they doing?</p> <p>They are having a conversation.</p> <p>What object are they using?</p> <p>It looks like a computer.</p> <p>Is this surprising?</p> <p>Yes, it is surprising.</p> <p>Why is this picture surprising to you?</p> <p>I think it is surprising because teddy bears are not usually found on the moon.</p>	 <p>What is the common thing about these three images?</p> <p>They are all flamingos.</p> <p>What is the difference between these three images?</p> <p>The first one is a cartoon, the second one is a real flamingo, and the third one is a 3D model of a flamingo.</p>	 <p>This is an apple with a sticker on it.</p> <p>What does the sticker say?</p> <p>The sticker says "iPod".</p> <p>Where is the photo taken?</p> <p>It looks like it's taken in a backyard.</p> <p>Do you think it is printed or handwritten?</p> <p>It looks like it's handwritten.</p> <p>What color is the sticker?</p> <p>It's white.</p>	 <p>This is a cityscape. It looks like Chicago.</p> <p>What makes you think this is Chicago?</p> <p>I think it's Chicago because of the Shedd Aquarium in the background.</p>  <p>What about this one? Which city is this and what famous landmark helped you recognise the city?</p> <p>This is Tokyo. I think it's Tokyo because of the Tokyo Tower.</p>			

Flamingo

Flamingo: a Visual Language Model for Few-Shot Learning, Deepmind, 2022

Vision Encoder: Normalizer-Free ResNet (NFNet)
Perceiver Sampler: Fixed # of queries attend to variable length of visual tokens. (input images can have different resolutions)
LLM: Chinchilla



Flamingo

Flamingo: a Visual Language Model
for Few-Shot Learning, Deepmind, 2022

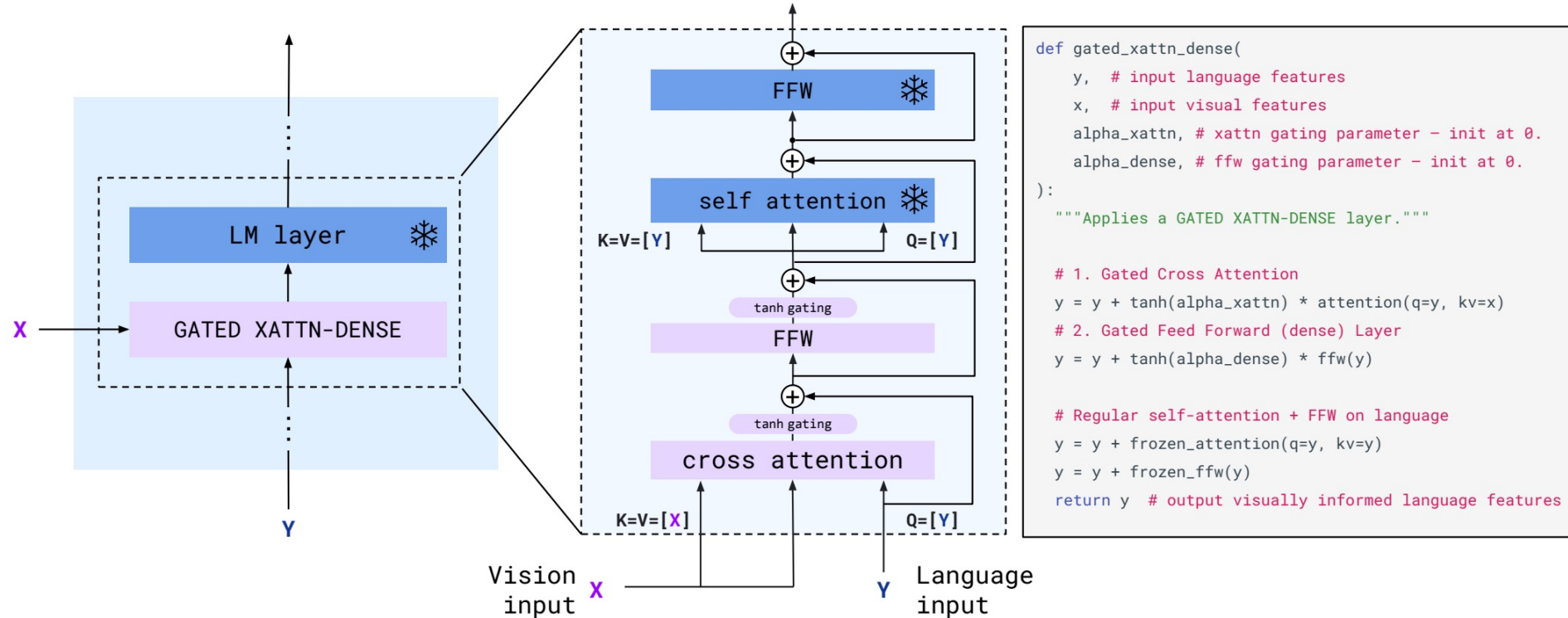


Figure 4: **GATED XATTN-DENSE layers.** To condition the LM on visual inputs, we insert new cross-attention layers between existing pretrained and frozen LM layers. The keys and values in these layers are obtained from the vision features while the queries are derived from the language inputs. They are followed by dense feed-forward layers. These layers are *gated* so that the LM is kept intact at initialization for improved stability and performance.

BLIP

BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation, Salesforce, 2022.

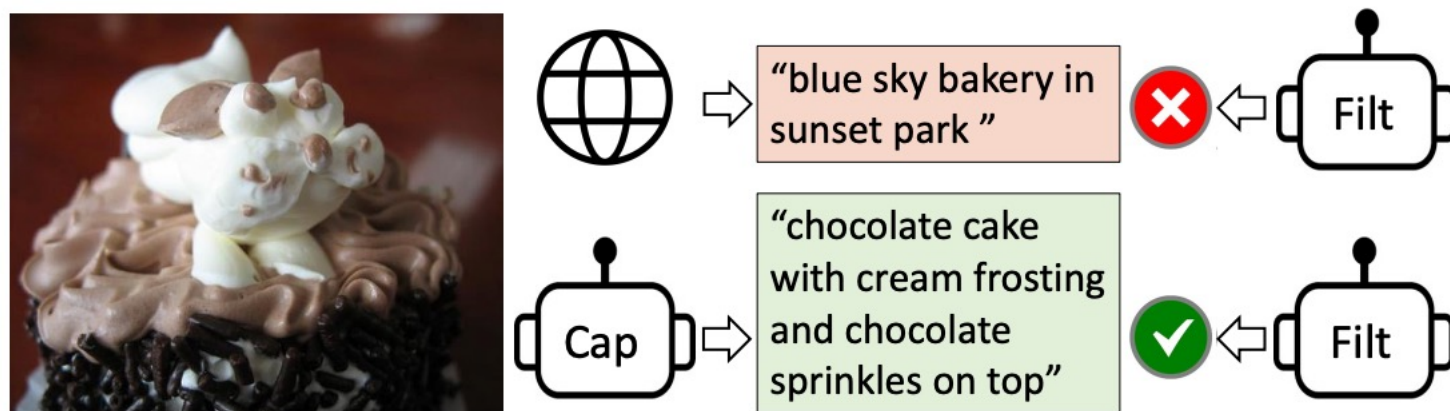


Figure 1. We use a Captioner (Cap) to generate synthetic captions for web images, and a Filter (Filt) to remove noisy captions.

BLIP

BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation, Salesforce, 2022.

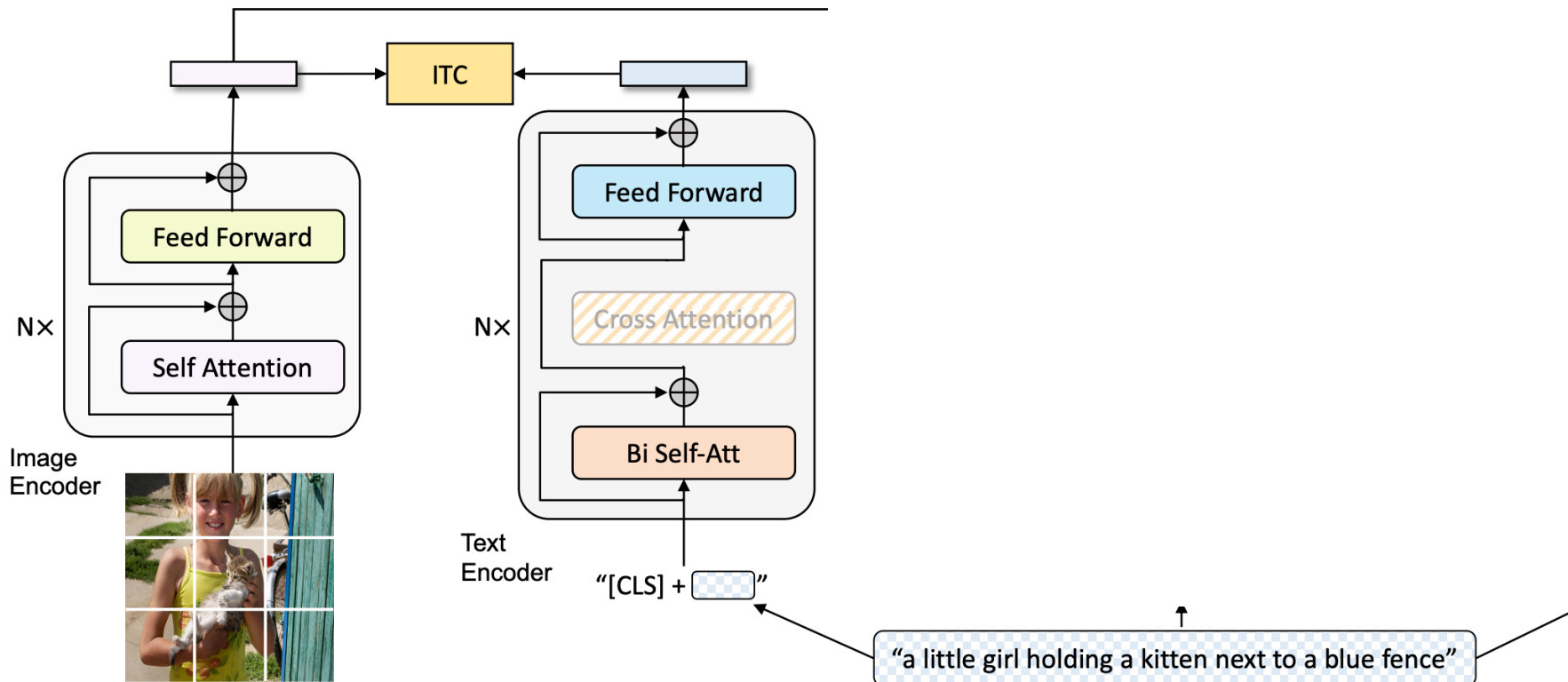


Figure 2. Pre-training model architecture and objectives of BLIP (same parameters have the same color). We propose multimodal mixture of encoder-decoder, a unified vision-language model which can operate in one of the three functionalities: (1) Unimodal encoder is trained with an image-text contrastive (ITC) loss to align the vision and language representations. (2) Image-grounded text encoder uses additional cross-attention layers to model vision-language interactions, and is trained with a image-text matching (ITM) loss to distinguish between positive and negative image-text pairs. (3) Image-grounded text decoder replaces the bi-directional self-attention layers with causal self-attention layers, and shares the same cross-attention layers and feed forward networks as the encoder. The decoder is trained with a language modeling (LM) loss to generate captions given images.

BLIP

BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation, Salesforce, 2022.

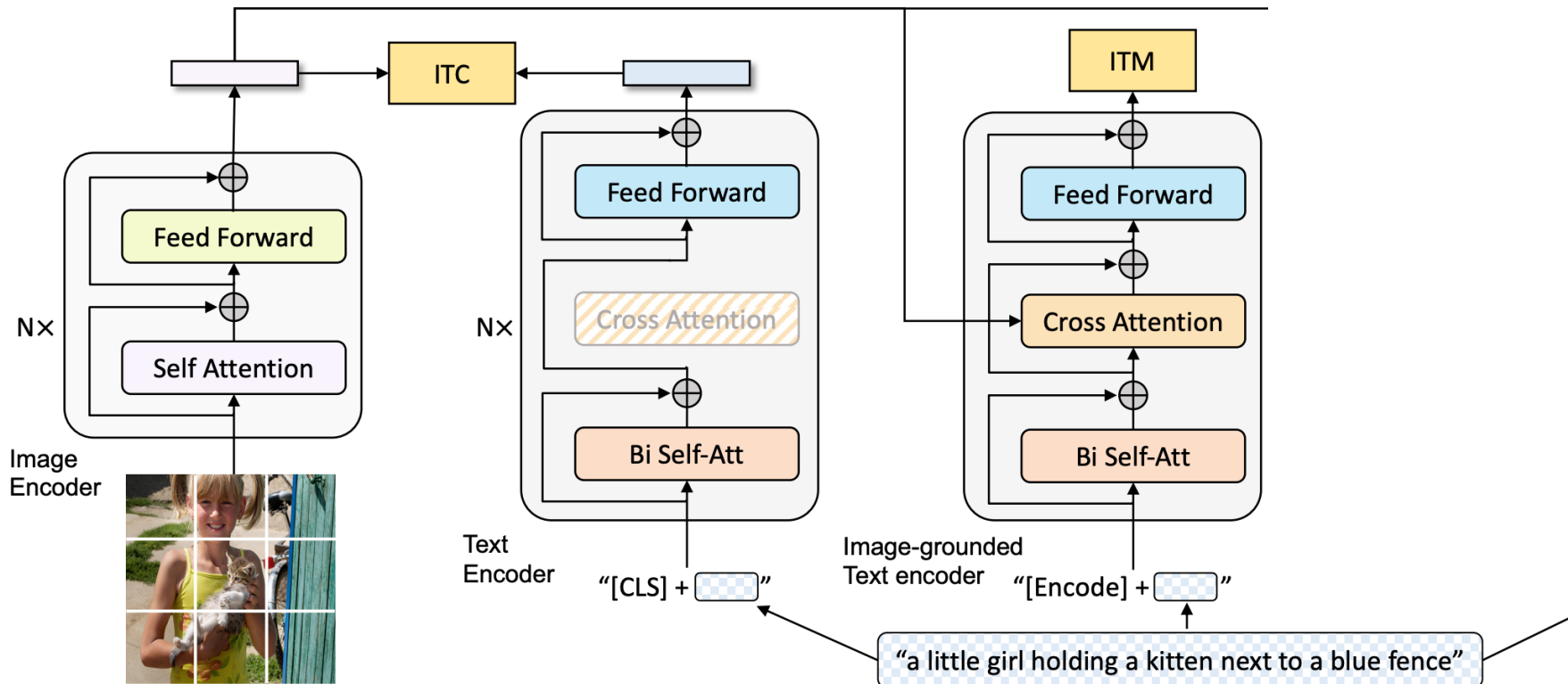


Figure 2. Pre-training model architecture and objectives of BLIP (same parameters have the same color). We propose multimodal mixture of encoder-decoder, a unified vision-language model which can operate in one of the three functionalities: (1) Unimodal encoder is trained with an image-text contrastive (ITC) loss to align the vision and language representations. (2) Image-grounded text encoder uses additional cross-attention layers to model vision-language interactions, and is trained with a image-text matching (ITM) loss to distinguish between positive and negative image-text pairs. (3) Image-grounded text decoder replaces the bi-directional self-attention layers with causal self-attention layers, and shares the same cross-attention layers and feed forward networks as the encoder. The decoder is trained with a language modeling (LM) loss to generate captions given images.

BLIP

BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation, Salesforce, 2022.

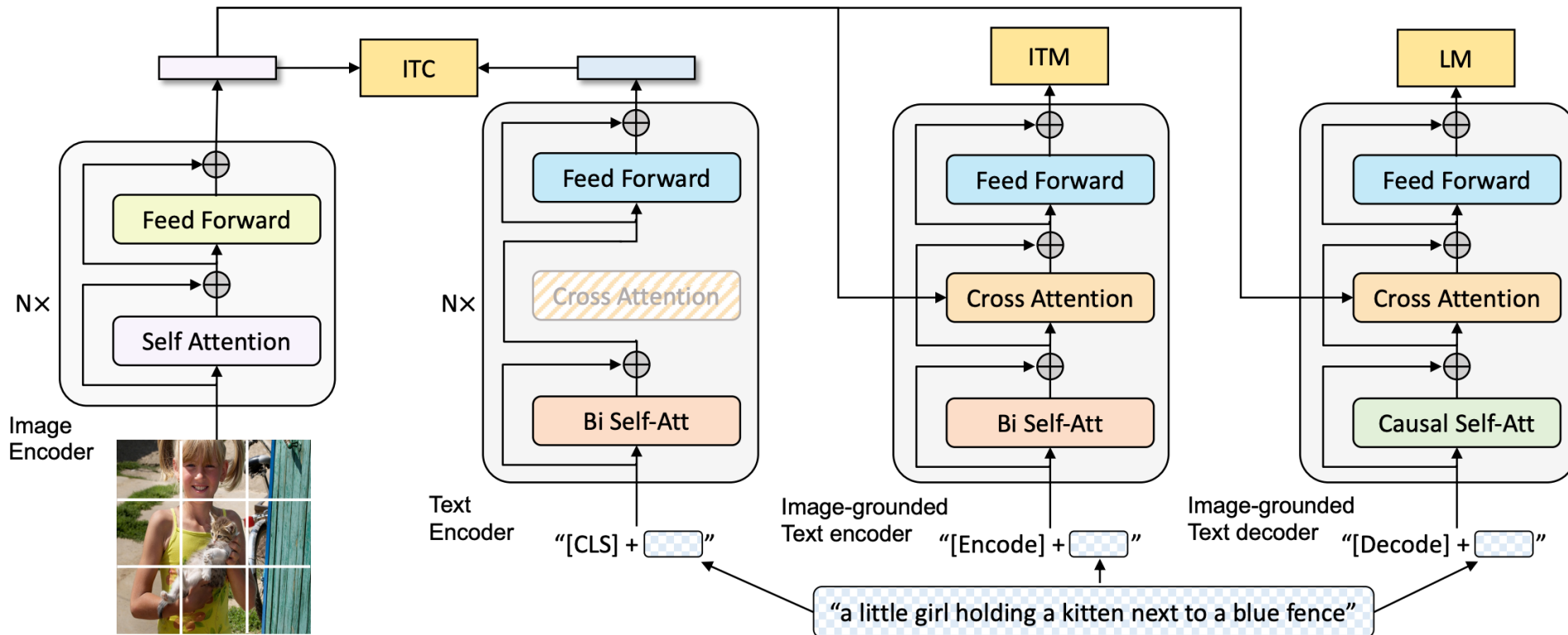


Figure 2. Pre-training model architecture and objectives of BLIP (same parameters have the same color). We propose multimodal mixture of encoder-decoder, a unified vision-language model which can operate in one of the three functionalities: (1) Unimodal encoder is trained with an image-text contrastive (ITC) loss to align the vision and language representations. (2) Image-grounded text encoder uses additional cross-attention layers to model vision-language interactions, and is trained with a image-text matching (ITM) loss to distinguish between positive and negative image-text pairs. (3) Image-grounded text decoder replaces the bi-directional self-attention layers with causal self-attention layers, and shares the same cross-attention layers and feed forward networks as the encoder. The decoder is trained with a language modeling (LM) loss to generate captions given images.

BLIP-2

BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models, Salesforce, 2023.

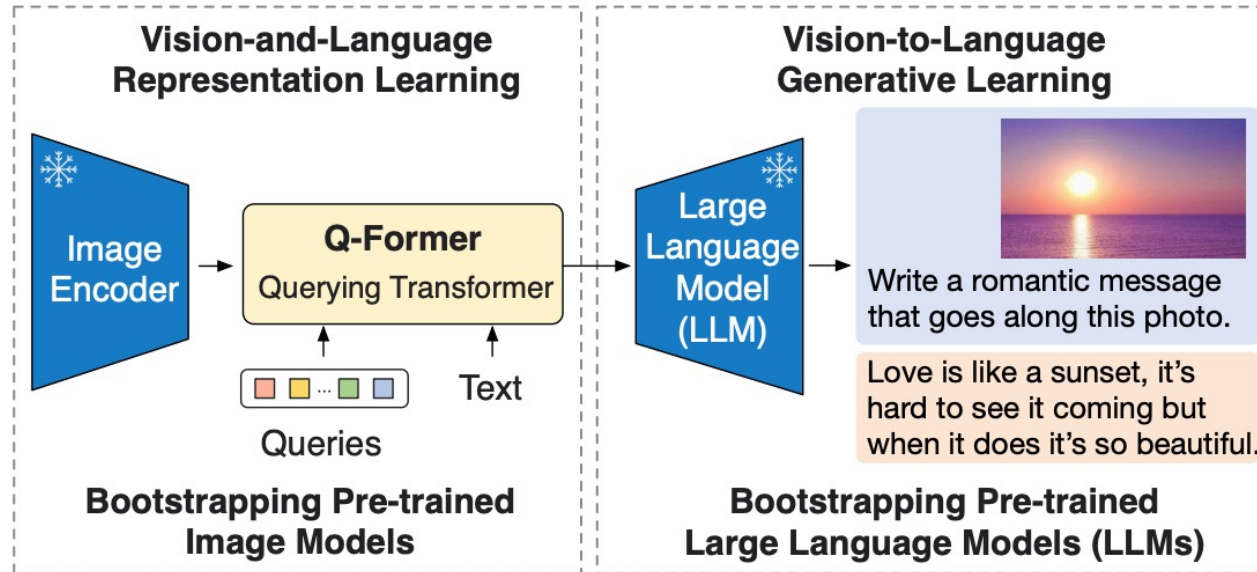


Figure 1. Overview of BLIP-2’s framework. We pre-train a lightweight Querying Transformer following a two-stage strategy to bridge the modality gap. The first stage bootstraps vision-language representation learning from a frozen image encoder. The second stage bootstraps vision-to-language generative learning from a frozen LLM, which enables zero-shot instructed image-to-text generation (see Figure 4 for more examples).

BLIP-2

BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models, Salesforce, 2023.

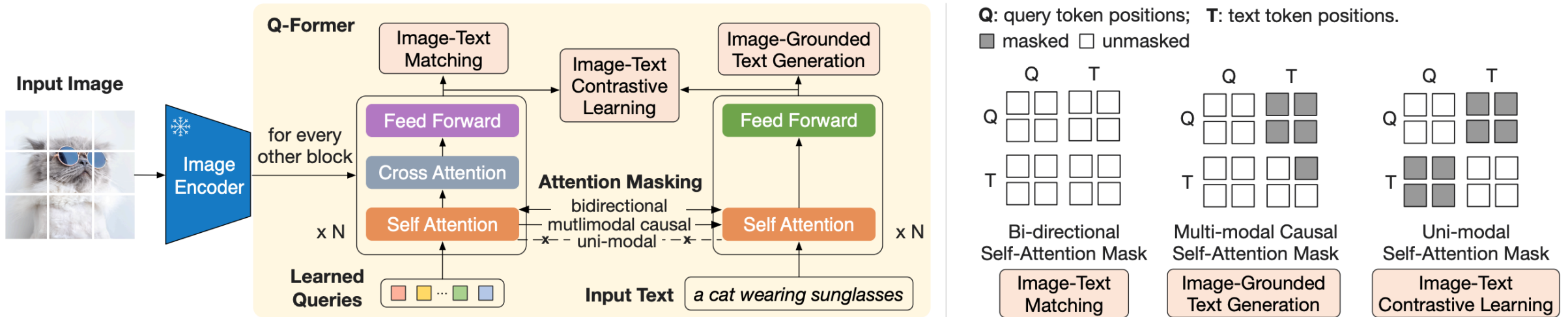


Figure 2. (Left) Model architecture of Q-Former and BLIP-2’s first-stage vision-language representation learning objectives. We jointly optimize three objectives which enforce the queries (a set of learnable embeddings) to extract visual representation most relevant to the text. (Right) The self-attention masking strategy for each objective to control query-text interaction.

BLIP-2

BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models, Salesforce, 2023.

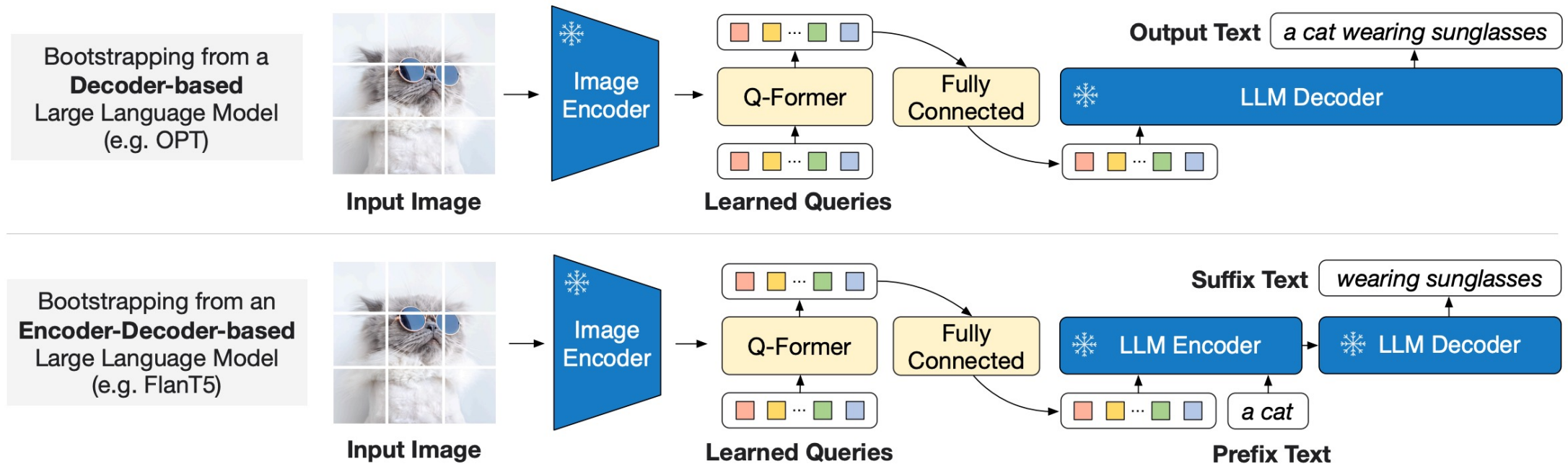


Figure 3. BLIP-2’s second-stage vision-to-language generative pre-training, which bootstraps from frozen large language models (LLMs). **(Top)** Bootstrapping a decoder-based LLM (e.g. OPT). **(Bottom)** Bootstrapping an encoder-decoder-based LLM (e.g. FlanT5). The fully-connected layer adapts from the output dimension of the Q-Former to the input dimension of the chosen LLM.

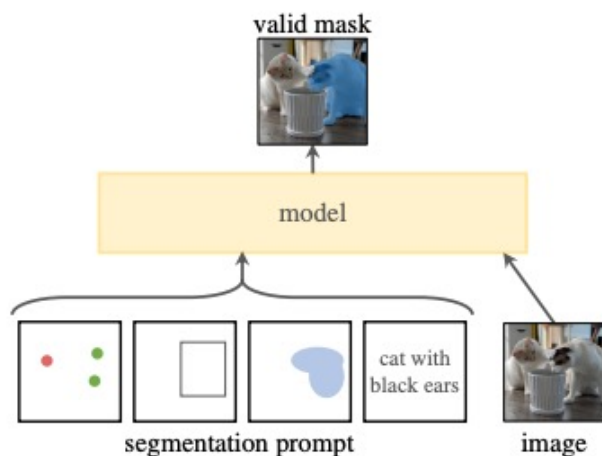
Misc

Segment Anything Model (SAM) 2023

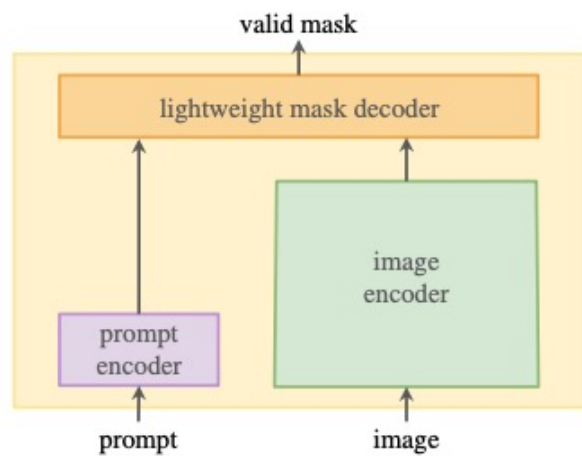
Segment Anything

Alexander Kirillov^{1,2,4} Eric Mintun² Nikhila Ravi^{1,2} Hanzi Mao² Chloe Rolland³ Laura Gustafson³
Tete Xiao³ Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár⁴ Ross Girshick⁴
¹project lead ²joint first author ³equal contribution ⁴directional lead

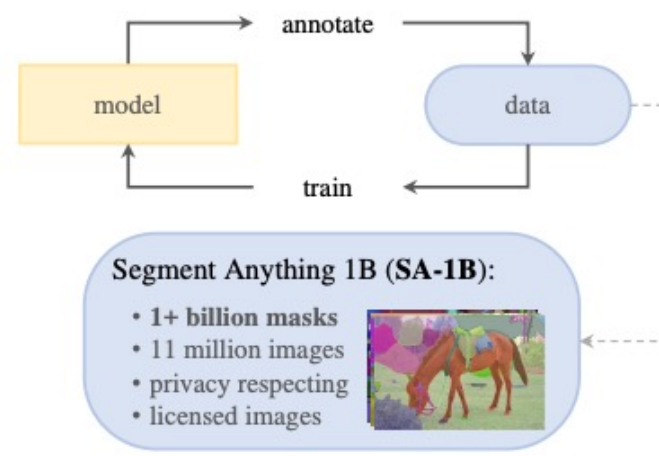
Meta AI Research, FAIR



(a) **Task:** promptable segmentation



(b) **Model:** Segment Anything Model (SAM)



(c) **Data:** data engine (top) & dataset (bottom)

Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation *task*, a segmentation *model* (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a *data* engine for collecting SA-1B, our dataset of over 1 billion masks.

Segment Anything Model (SAM) v2 2024

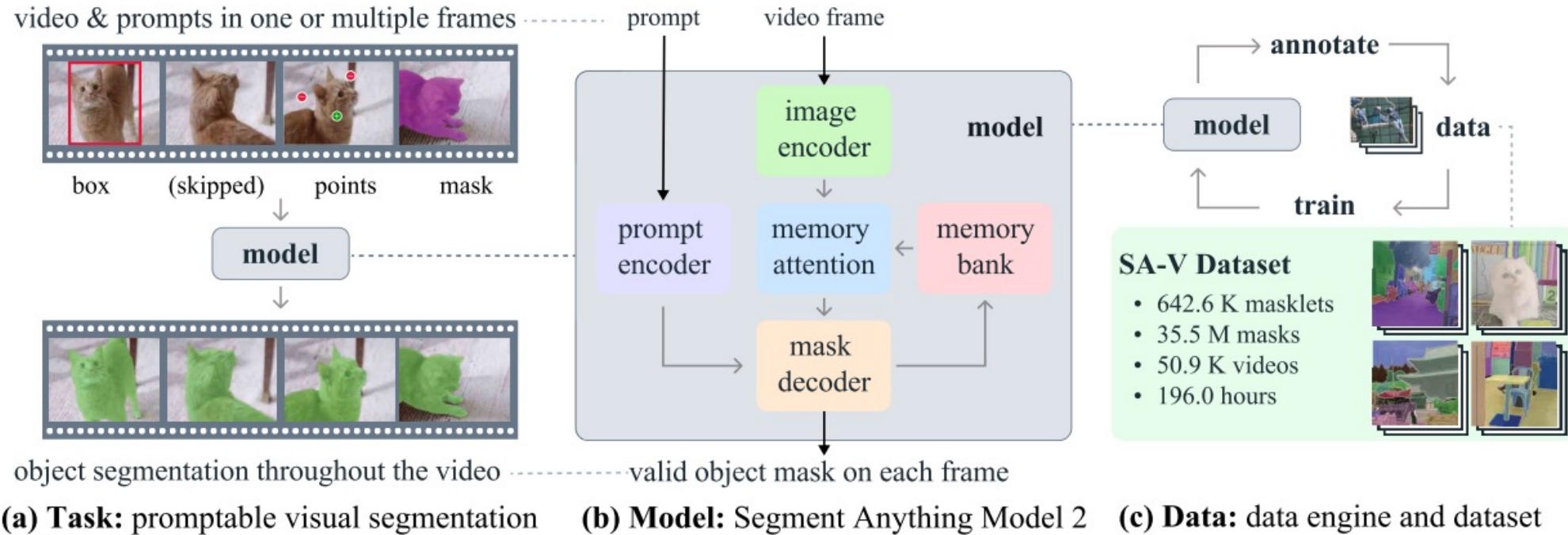


Figure 1 We introduce the Segment Anything Model 2 (SAM 2), towards solving the promptable visual segmentation task (a) with our foundation model (b), trained on our large-scale SA-V dataset collected through our data engine (c). SAM 2 is capable of interactively segmenting regions through prompts (clicks, boxes, or masks) on one or multiple video frames by utilizing a streaming memory that stores previous prompts and predictions.

Segment Everything Everywhere All At Once 2023

Segment Everything Everywhere All at Once

Xueyan Zou^{*§2}, Jianwei Yang^{*†1}, Hao Zhang^{*‡}, Feng Li^{*‡}, Linjie Li[†], Jianfeng Wang[†]
Lijuan Wang[†], Jianfeng Gao^{†‡}, Yong Jae Lee^{¶§}

[§] University of Wisconsin-Madison [†] Microsoft Research, Redmond [‡] HKUST [†] Microsoft Cloud & AI

^{*}Equal Contribution [¶]Equal Advisory Contribution 1. Project Lead 2. Main Technical Contribution

{xueyan, yongjaelee}@cs.wisc.edu {jianwyan, jfgao, linjli}@microsoft.com {hzhangcx, fliay}@connect.ust.hk

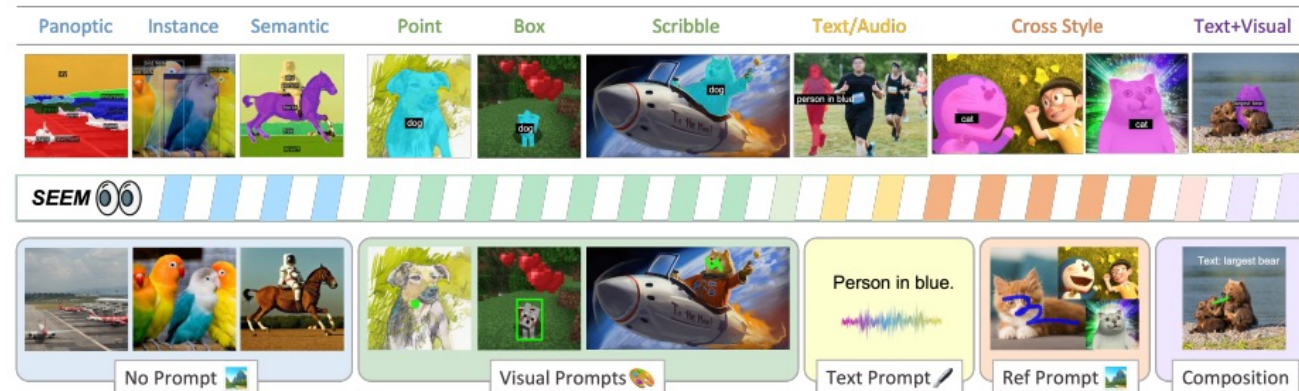


Figure 1: *SEEM* supports generic segmentation tasks—including semantic, instance, and panoptic segmentation—in an open-set fashion when no prompt is provided. *SEEM* also enables the use of visual, textual, and referring region prompts in flexible combinations, making it a promptable and interactive segmentation interface.

INTERNVIDEO2: SCALING FOUNDATION MODELS FOR MULTIMODAL VIDEO UNDERSTANDING

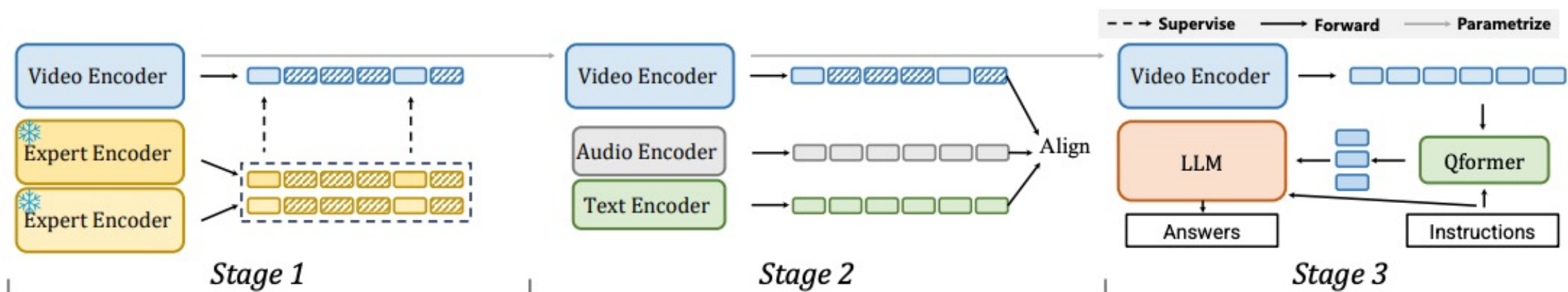


Figure 2: Framework of **InternVideo2**. It consists of three consecutive training phases: unmasked video token reconstruction, multimodal contrastive learning, and next token prediction. In stage 1, the video encoder is trained from scratch, while in stages 2 and 3, it is initialized from the version used in the previous stage.

Stage 1: Reconstructing Unmasked Video Tokens

Stage 2: Aligning Video to Audio-Speech-Text

Stage 3: Predicting Next Token with Video-Centric Inputs

INTERNVIDEO2: SCALING FOUNDATION MODELS FOR MULTIMODAL VIDEO UNDERSTANDING

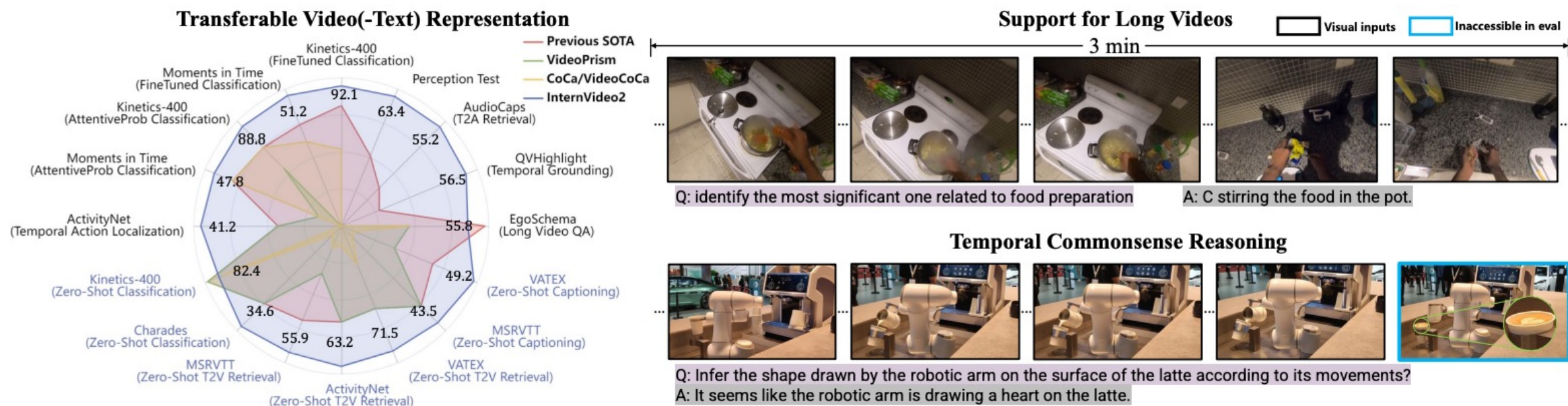


Figure 1: **InternVideo2** yields strong transferable visual and visual-linguistic representations across a total of 70 video understanding tasks, ranging from action recognition, video-text understanding, to video-centric dialogue. It also exhibits capability of long-form video understanding and procedure-aware reasoning.