SoTA Multimodal Multitask Architectures.
The Survey

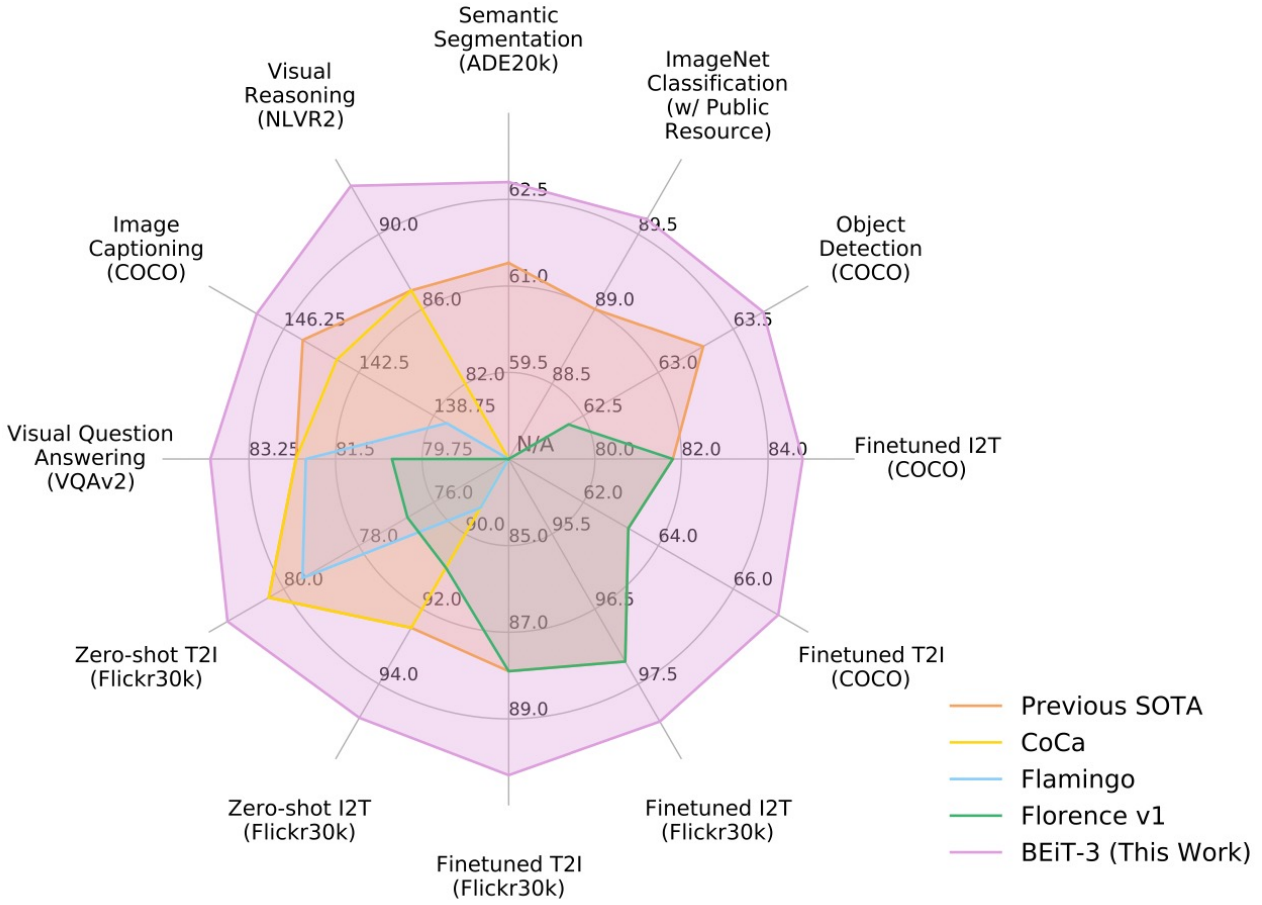
Image as a Foreign Language: BEIT Pretraining for All Vision and Vision-Language Tasks

Current state in terms of BEiT-3

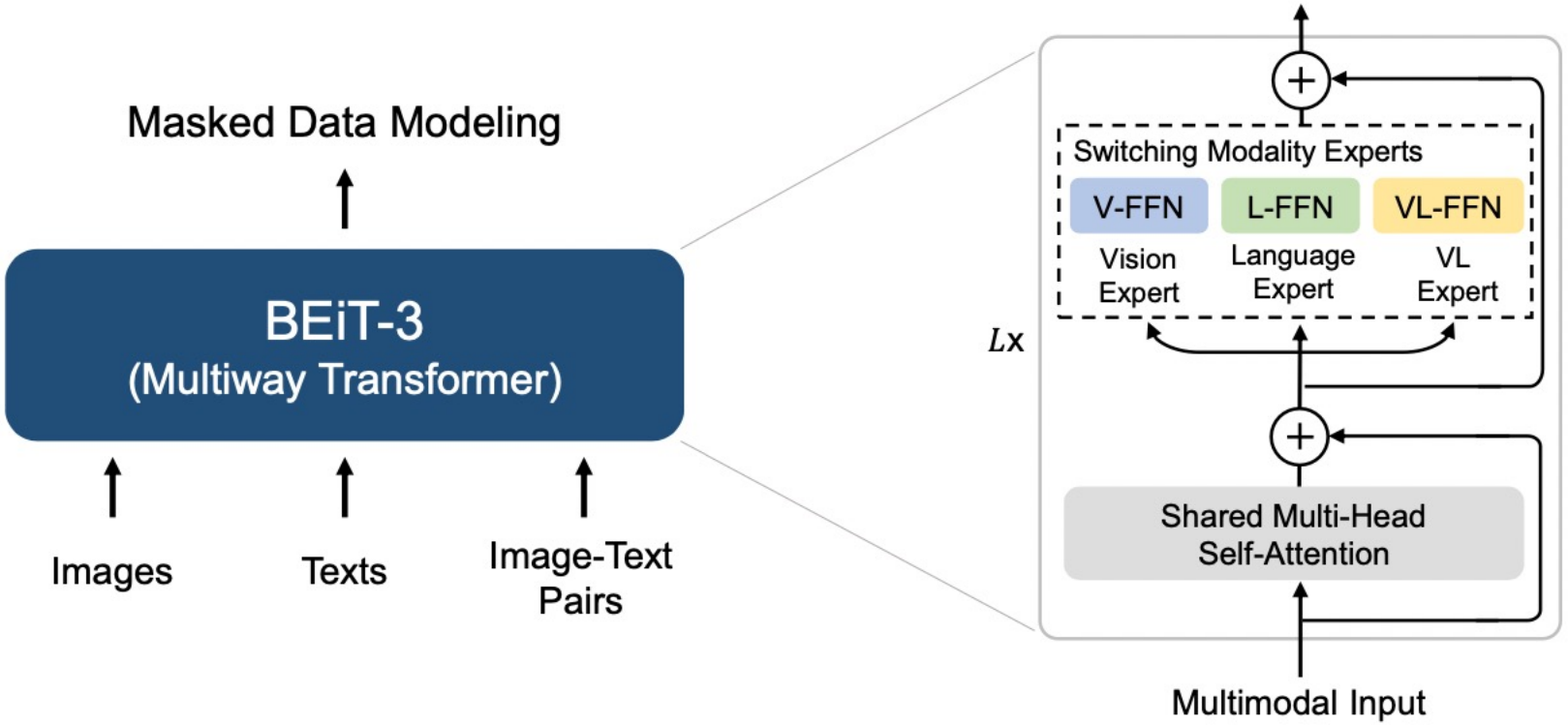
BERT image Transformers (BEiT)

3 aspects to use transformers:

1. Successful translation from text to visual and multimodal problems
2. The pretraining task based on masked data modeling is successful to various modalities: text, images, text-image pairs
3. Scaling up the model size and data size universally improves the generalization quality of foundation models -> easy to transfer to various downstream tasks.



BEiT pretrain



Masked data modeling on **monomodal** (texts and images) and **multimodal** (image-text pairs) data with a **shared Multiway Transformer** as the backbone network.

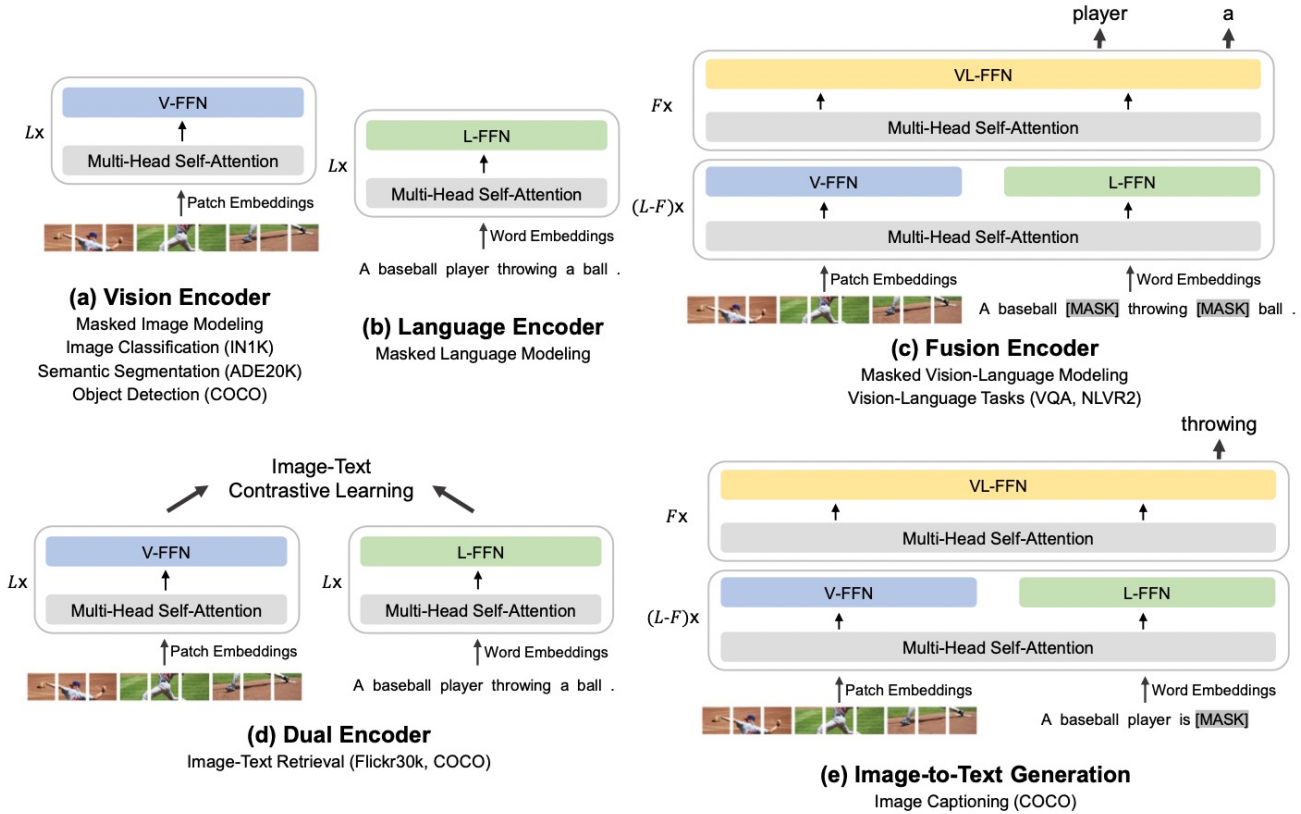
Images = Imglish😀

Multiway Transformer

Multiway Transformer block:

- a shared self-attention module
- a pool of feed-forward networks (i.e., modality experts)

Capture more modality-specific information!



Pretrain setup

- 1M steps
- BS = 6144 samples in total
  - 2048 images
  - 2048 texts
  - 2048 image-text pairs
- 14 × 14 patch size, 224 × 224 resolution

Finetuning on downstream tasks

Model	VQAv2		NLVR2		COCO Captioning			
	test-dev	test-std	dev	test-P	B@4	M	C	S
Oscar [LYL+20]	73.61	73.82	79.12	80.37	37.4	30.7	127.8	23.5
VinVL [ZLH+21]	76.52	76.60	82.67	83.98	38.5	30.4	130.8	23.4
ALBEF [LSG+21]	75.84	76.04	82.55	83.14	-	-	-	-
BLIP [LLXH22]	78.25	78.32	82.15	82.24	40.4	-	136.7	-
SimVLM [WYY+21]	80.03	80.34	84.53	85.15	40.6	33.7	143.3	**25.4**
Florence [YCC+21]	80.16	80.36	-	-	-	-	-	-
OFA [WYM+22]	82.00	82.00	-	-	43.9	31.8	145.3	24.8
Flamingo [ADL+22]	82.00	82.10	-	-	-	-	138.1	-
CoCa [YWV+22]	82.30	82.30	86.10	87.00	40.9	**33.9**	143.6	24.7
BEIT-3	**84.19**	**84.03**	**91.51**	**92.58**	**44.1**	32.4	**147.6**	**25.4**

Table 4: Results of visual question answering, visual reasoning, and image captioning tasks. We report *vqa-score* on VQAv2 test-dev and test-standard splits, accuracy for NLVR2 development set and public test set (test-P). For COCO image captioning, we report BLEU@4 (B@4), METEOR (M), CIDEr (C), and SPICE (S) on the Karpathy test split. For simplicity, we report captioning results without using CIDEr optimization.

Model	Flickr30K (1K test set)					
	Image → Text			Text → Image		
	R@1	R@5	R@10	R@1	R@5	R@10
FLAVA [SHG+21]	67.7	94.0	-	65.2	89.4	-
CLIP [RKH+21]	88.0	98.7	99.4	68.7	90.6	95.2
ALIGN [JYX+21]	88.6	98.7	99.7	75.7	93.8	96.8
FILIP [YHH+21]	89.8	99.2	99.8	75.0	93.4	96.3
Florence [YCC+21]	90.9	99.1	-	76.7	93.6	-
Flamingo [ADL+22]	89.3	98.8	99.7	79.5	95.3	**97.9**
CoCa [YWV+22]	92.5	99.5	99.9	80.4	**95.7**	97.7
BEIT-3	**94.9**	**99.9**	**100.0**	**81.5**	95.6	97.8

Table 6: Zero-shot image-to-text retrieval and text-to-image retrieval on Flickr30K.

Model	Extra OD Data	Maximum Image Size	COCO test-dev	
			AP^{box}	AP^{mask}
ViT-Adapter [CDW+22]	-	1600	60.1	52.1
DyHead [DCX+21]	ImageNet-Pseudo Labels	2000	60.6	-
Soft Teacher [XZH+21]	Object365	-	61.3	53.0
GLIP [LZZ+21]	FourODs	-	61.5	-
GLIPv2 [ZZH+22]	FourODs	-	62.4	-
Florence [YCC+21]	FLOD-9M	2500	62.4	-
SwinV2-G [LHL+21]	Object365	1536	63.1	54.4
Mask DINO [LZX+22]	Object365	1280	-	54.7
DINO [ZLL+22]	Object365	2000	63.3	-
BEIT-3	Object365	1280	**63.7**	**54.8**

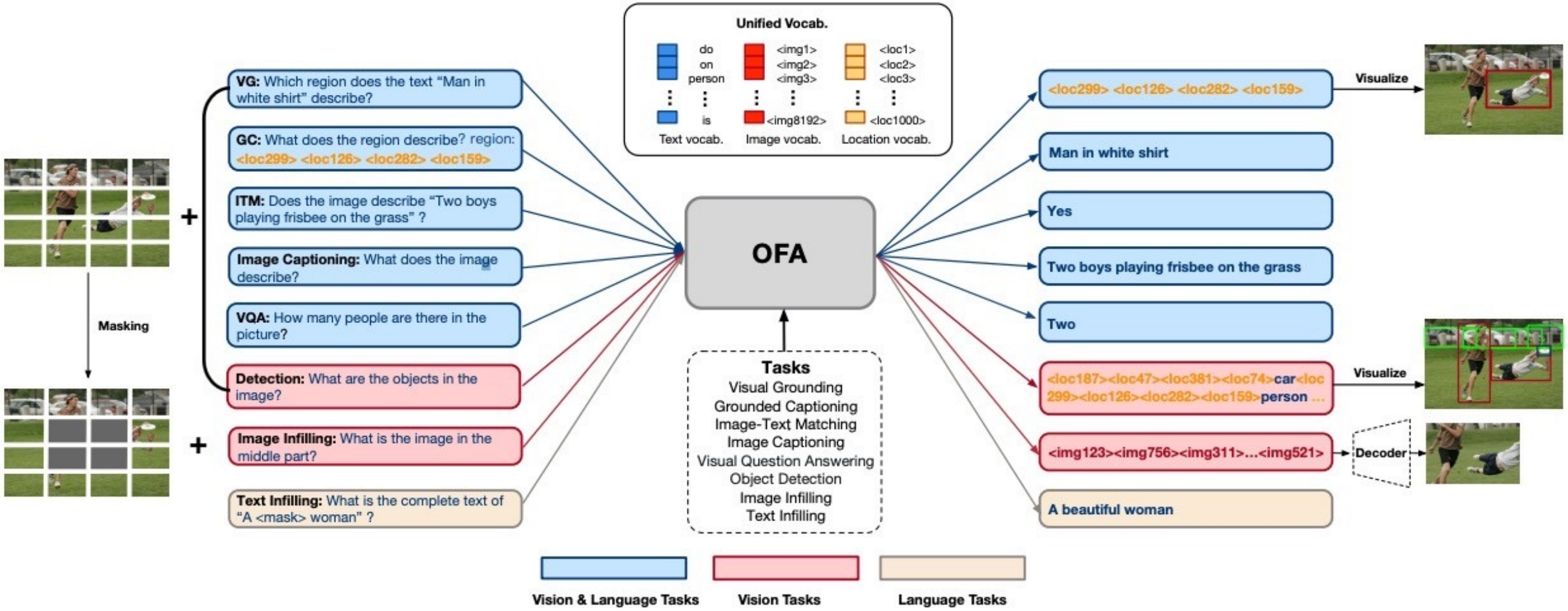
Table 7: Results of object detection and instance segmentation on COCO benchmark. BEIT-3 uses Cascade Mask R-CNN [CV21] as the detection head. Our results are reported with multi-scale evaluation. We report the maximum image size used for training. FLOD-9M and FourODs also contain Object365. The results of the comparison systems are from the paperswithcode.com leaderboard (timestamp: 08/22/2022).

Model	Crop Size	ADE20K	
		mIoU	+MS
HorNet [RZT+22]	640^2	57.5	57.9
SeMask [JSO+21]	640^2	57.0	58.3
SwinV2-G [LHL+21]	896^2	59.3	59.9
ViT-Adapter [CDW+22]	896^2	59.4	60.5
Mask DINO [LZX+22]	-	59.5	60.8
FD-SwinV2-G [WHX+22]	896^2	-	61.4
BEIT-3	896^2	**62.0**	**62.8**

Table 8: Results of semantic segmentation on ADE20K. "MS" is short for multi-scale. The results of the comparison systems are from the paperswithcode.com leaderboard (timestamp: 08/22/2022).

OFA: Unifying Architectures, Tasks and Modalities Through a Simple Sequence-to-Sequence Learning Framework

Pretraining tasks



Input batches construction

1. All tasks are presented in a seq-to-seq fashion
2. Each task is specified via handcrafted instructions
3. Each batch consists of
    - 2048 vision & language samples
    - 256 object detection samples
    - 256 image-only samples
    - 512 text-only samples

Architecture

1. Encoder-decoder Transformer
2. OFA-base
    - 6 encoder and 6 decoder layers
    - 768 hidden size
    - 12 attention heads in each layer
    - 3072 FFN size
3. OFA-large
    - 12 encoder and 12 decoder layers
    - 1024 hidden size
    - 16 attention heads in each layer
    - 4096 FFN size

Dataset

Pretraining datasets are constructed using

- image-text pairs
- raw image data
- object-labeled data
- plain texts

Type	Pretraining Task	Source	#Image	#Label
Vision&Language	Image Captioning Image-Text Matching	CC12M, CC3M, SBU, COCO, VG-Cap	14.78M	15.25M
	Visual Question Answering	VQAv2, VG-QA, GQA	178K	2.92M
	Visual Grounding Grounded Captioning	RefCOCO, RefCOCO+, RefCOCOg, VG-Cap	131K	3.20M
Vision	Detection	OpenImages, Object365, VG, COCO	2.98M	3.00M
	Image Infilling	OpenImages, YFCC100M, ImageNet-21K	36.27M	-
Language	Masked Language Modeling	Pile (Filter)	-	140G*

Downstream data

Task	Dataset	Instruction	Target
Image Captioning	COCO	[**Image**] What does the image describe?	{**Caption**}
Visual Question Answering	VQA	[**Image**] {**Question**}	{**Answer**}
Visual Entailment	SNLI-VE	[**Image**] Can image and text1 "{**Text1**}" imply text2 "{**Text2**}"?	Yes/No/Maybe
Referring Expression Comprehension	RefCOCO, RefCOCO+, RefCOCOg	[**Image**] Which region does the text "{**Text**}" describe?	{**Location**}
Image Generation	COCO	What is the complete image? caption: {**Caption**}	{**Image**}
Image Classification	ImageNet-1K	[**Image**] What does the image describe?	{**Label**}
Single-Sentence Classification	COLA	Is the text "{**Text**}" grammatically correct?	Yes/No
	SST-2	Is the sentiment of text "{**Text**}" positive or negative?	Positive/Negative
Sentence-Pair Classification	RTE	Can text1 "{**Text1**}" imply text2 "{**Text2**}"?	Yes/No
	MRPC	Does text1 "{**Text1**}" and text2 "{**Text2**}" have the same semantics?	Yes/No
	QQP	Is question "{**Question1**}" and question "{**Question2**}" equivalent?	Yes/No
	MNLI	Can text1 "{**Text1**}" imply text2 "{**Text2**}"?	Yes/No/Maybe
	QNLI	Does "{**Text**}" contain the answer to question "{**Question**}"?	Yes/No
	WNLI	Can text1 "{**Text1**}" imply text2 "{**Text2**}"?	Yes/No
Text Summarization	Gigaword	What is the summary of article "{**Article**}"?	{**Summary**}

Cross-modal tasks

Model	COCO Captions B@4 / M / C / S	VQA test-dev / test-std	SNLI-VE dev / test	RefCOCO val / testA / testB	RefCOCO+ val / testA / testB	RefCOCOg val-u / test-u
VL-BERT [8]	-	71.79 / 72.22	-	-	72.59 / 78.57 / 62.30	-
UNITER [14]	-	73.82 / 74.02	79.39 / 79.38	81.41 / 87.04 / 74.17	75.90 / 81.45 / 66.70	74.86 / 75.77
OSCAR [15]	41.7 / 30.6 / 140.0 / 24.5	73.61 / 73.82	-	-	-	-
VILLA [16]	-	74.69 / 74.87	80.18 / 80.02	82.39 / 87.48 / 74.84	76.17 / 81.54 / 66.84	76.18 / 76.71
MDETR [65]	-	70.64 / 70.63	-	86.75 / 89.58 / 81.41	79.52 / 84.09 / 70.62	81.64 / 80.89
UNICORN [55]	35.8 / 28.4 / 119.1 / 21.5	69.2 / 69.4	-	88.29 / 90.42 / 83.06	80.30 / 85.05 / 71.88	83.44 / 83.93
VinVL [17]	41.0 / 31.1 / 140.9 / 25.2	76.52 / 76.60	-	-	-	-
UNIMO [43]	39.6 / - / 127.7 / -	75.06 / 75.27	81.11 / 80.63	-	-	-
METER [66]	-	77.68 / 77.64	80.86 / 81.19	-	-	-
VLMO [46]	-	**79.94** / 79.98	-	-	-	-
SimVLM [45]	40.3 / **33.4** / 142.6 / 24.7	79.32 / 79.56	85.68 / 85.62	-	-	-
OFA	**43.5** / 31.9 / **149.6** / **26.1**	79.87 / **80.02**	**90.30 / 90.20**	**90.05 / 92.93 / 85.26**	**84.49 / 90.10 / 77.77**	**84.54 / 85.20**

Qualitative results on unseen task

Q: what color is the car in the region? region: <loc301> <loc495> <loc501> <loc596>

A: tan

Q: what color is the car in the region? region: <loc512> <loc483> <loc675> <loc576>

A: gray

Qualitative results on unseen domain VQA

Q: what is grown on the plant?

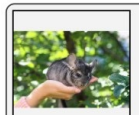
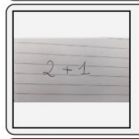
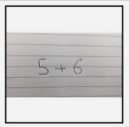
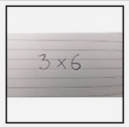
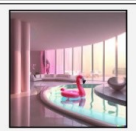
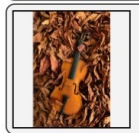
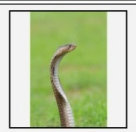
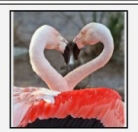
A: money

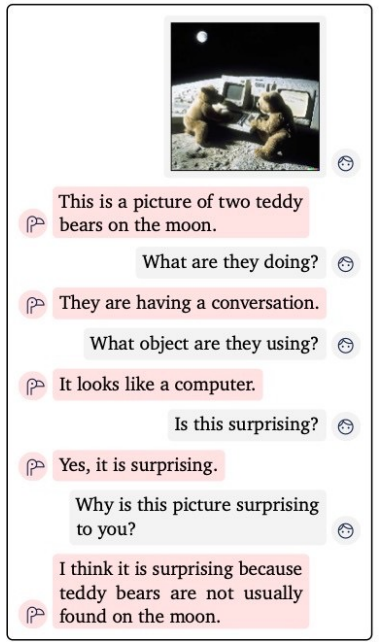
Q: what does the red-roofed building right to the big airship look like?

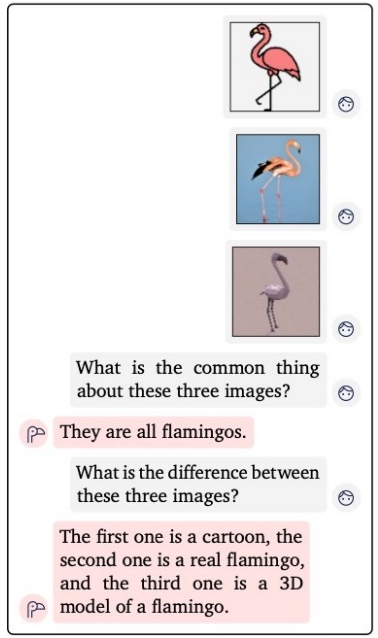
A: a mushroom

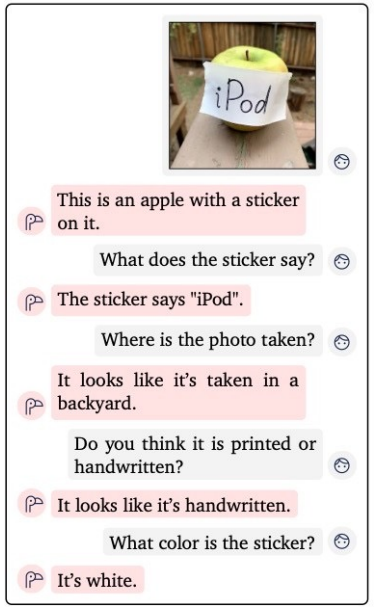
Flamingo: a Visual Language Model for Few-Shot Learning

Flamingo examples of inputs and outputs

Input Prompt	Completion
 This is a chinchilla. They are mainly found in Chile.  This is a shiba. They are very popular in Japan.  This is	a flamingo. They are found in the Caribbean and South America.
 What is the title of this painting? Answer: The Hallucinogenic Toreador.  Where is this painting displayed? Answer: Louvres Museum, Paris.  What is the name of the city where this was painted? Answer:	Arles.
 Output: "Underground"  Output: "Congress"  Output:	"Soulomes"
 2+1=3  5+6=11 	3x6=18
 Output: A propaganda poster depicting a cat dressed as French emperor Napoleon holding a piece of cheese.  Output: A pink room with a flamingo pool float.  Output:	A portrait of Salvador Dali with a robot head.
 Les sanglots longs des violons de l'automne blessent mon coeur d'une langueur monotone.  Pour qui sont ces serpents qui sifflent sur vos têtes? 	Je suis un cœur qui bat pour vous.
 pandas: 3  dogs: 2 	giraffes: 4
I like reading  , my favourite play is Hamlet. I also like  , my favorite book is	Dreams from my Father.
 What happens to the man after hitting the ball? Answer:	he falls down.







Flamingo models architecture

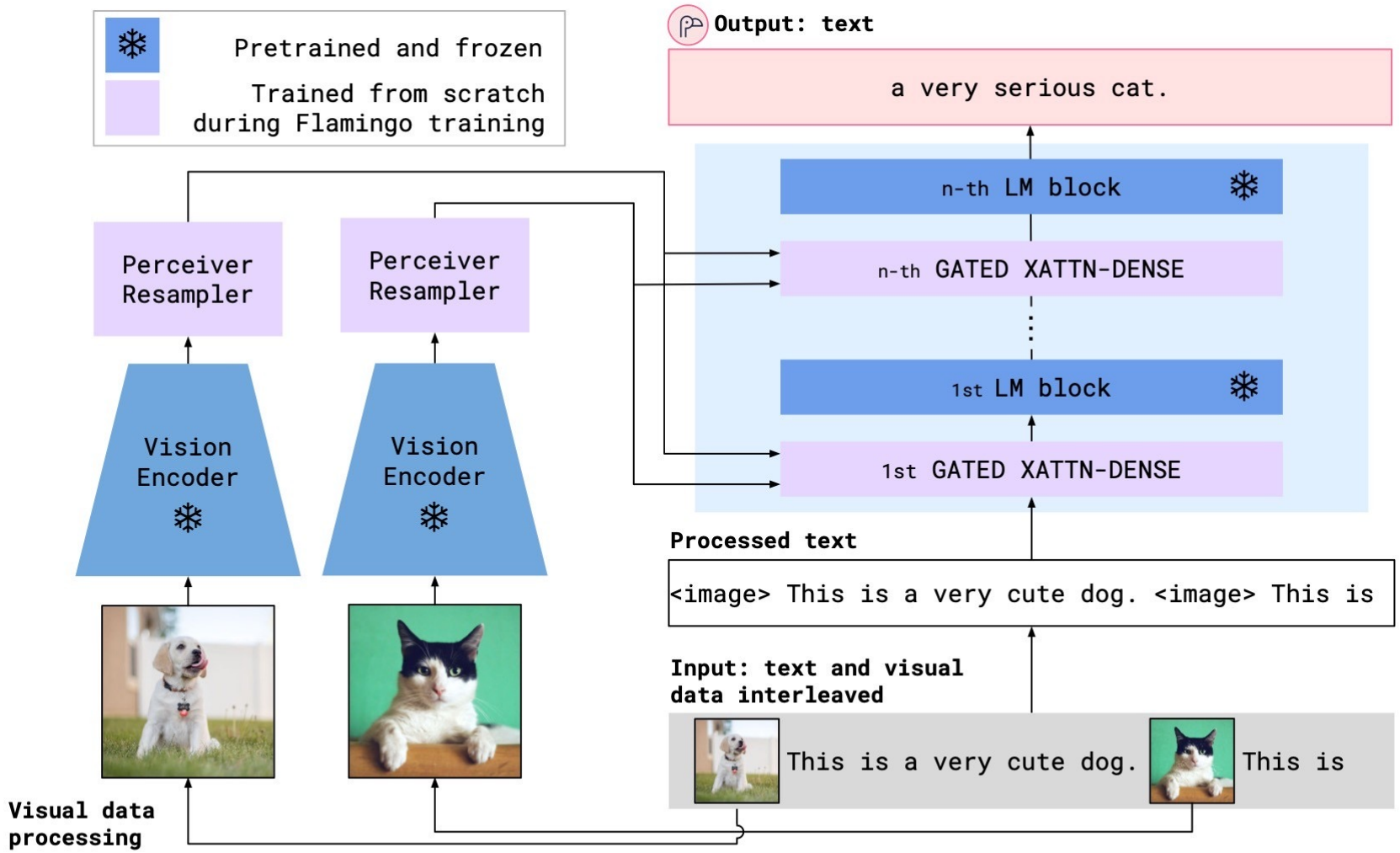
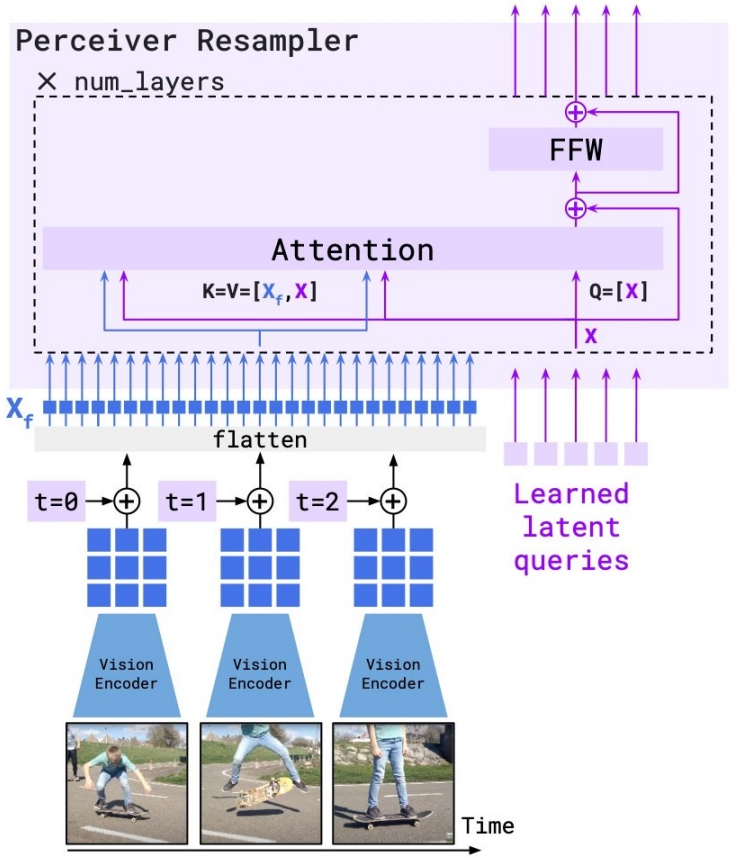


Figure 3 | **Overview of the Flamingo model.** The Flamingo models are a family of visual language model (VLM) that can take as input visual data interleaved with text and can produce free-form text as output. Key to its performance are novel architectural components and pretraining strategies described in Section 3.

Flamingo models architecture

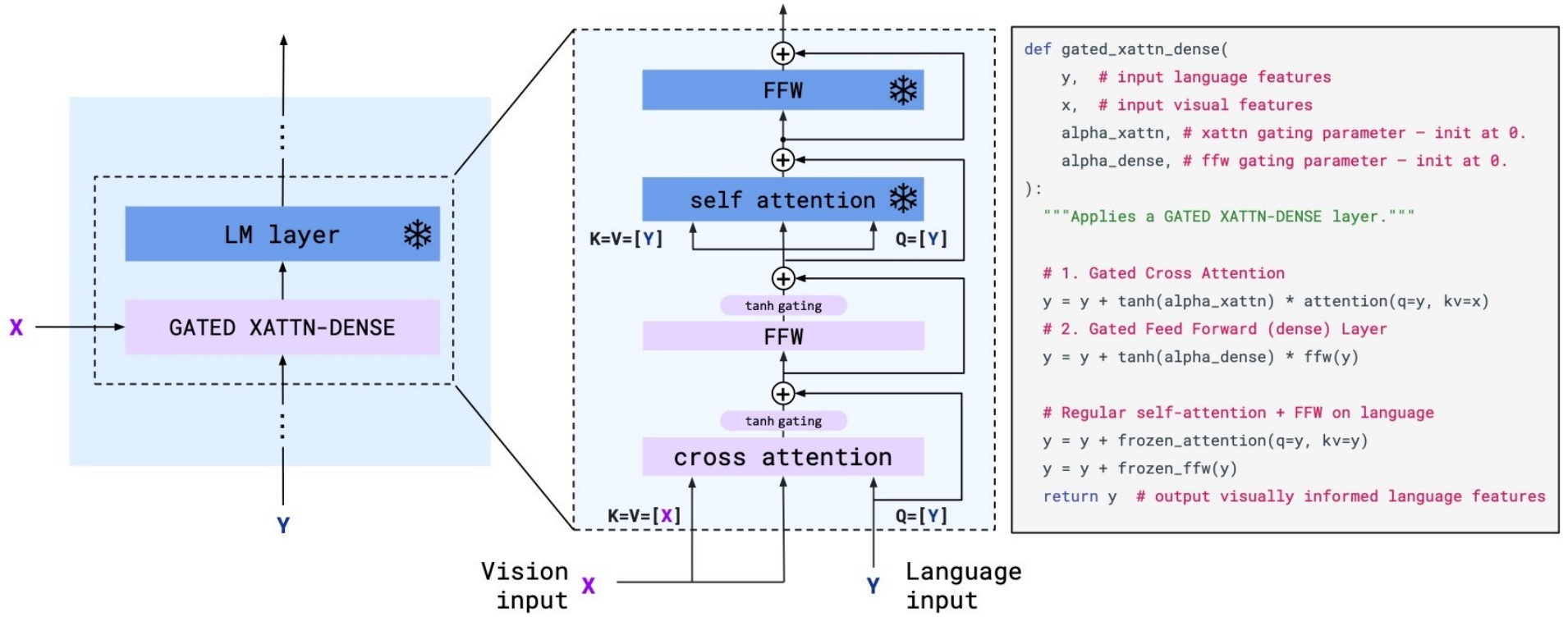



```python
def perceiver_resampler(
    x_f,  # The [T, S, d] visual features (T=time, S=space)
    time_embeddings,  # The [T, 1, d] time pos embeddings.
    x,  # R learned latents of shape [R, d]
    num_layers,  # Number of layers
):
  """The Perceiver Resampler model."""

  # Add the time position embeddings and flatten.
  x_f = x_f + time_embeddings
  x_f = flatten(x_f)  # [T, S, d] -> [T * S, d]
  # Apply the Perceiver Resampler layers.
  for i in range(num_layers):
    # Attention.
    x = x + attention_i(q=x, kv=concat([x_f, x]))
    # Feed forward.
    x = x + ffw_i(x)
  return x
```

Figure 4 | **The Perceiver Resampler** module maps a *variable size* grid of spatio-temporal visual features coming out of the Vision Encoder to a *fixed* number of output tokens (five in the figure), independently of the input image resolution or the number of input video frames. This transformer has a set of learned latent vectors as queries, and the keys and values are a concatenation of the spatio-temporal visual features with the learned latent vectors. More details can be found in Section 3.1.1.

Flamingo models architecture




```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter – init at 0.
    alpha_dense, # ffw gating parameter – init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 1. Gated Cross Attention
  y = y + tanh(alpha_xattn) * attention(q=y, kv=x)
  # 2. Gated Feed Forward (dense) Layer
  y = y + tanh(alpha_dense) * ffw(y)

  # Regular self-attention + FFW on language
  y = y + frozen_attention(q=y, kv=y)
  y = y + frozen_ffw(y)
  return y  # output visually informed language features
```

Figure 5 | **GATED XATTN-DENSE layers.** We insert new cross-attention layers, whose keys and values are obtained from the vision features while using language queries, followed by dense feed forward layers in between existing pretrained and frozen LM layers in order to condition the LM on visual inputs. These layers are *gated* so that the LM is kept intact at initialization for improved stability and performance.

Flamingo models architecture

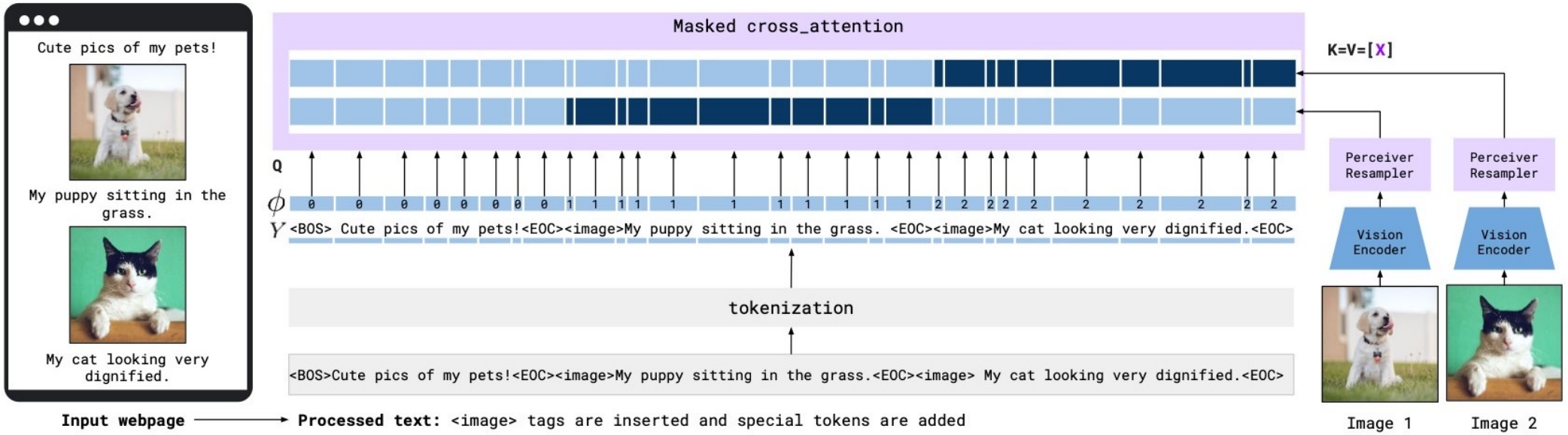


Figure 6 | **Interleaved visual data and text support.** Given text interleaved with images/videos, e.g. coming from a webpage, we first process the text by inserting `<image>` tags at the location of the visual data in the text as well as special tokens (`<BOS>` for "begining of sentence" or `<EOC>` for "end of chunk"). The images are processed independently by the Vision Encoder and Perceiver Resampler to extract visual tokens. Following our modeling choice motivated in Section 3.1.3, each text token only cross-attends to the visual tokens corresponding to the last preceding image. The function ϕ illustrated above indicates for each token what is the index of the last preceding image (and 0 if there are no preceding images). In practice, this selective cross-attention is achieved via a masked cross attention mechanism – illustrated here with the dark blue entries (non masked) and light blue entries (masked).

Datasets

Figure 7 | **Training datasets.** Mixture of training datasets of different nature. N corresponds to the number of visual inputs for a single example. For paired image (or video) and text datasets, $N = 1$. T is the number of video frames with $T = 1$ being the special case of images. H, W, C are height, width and color channels.

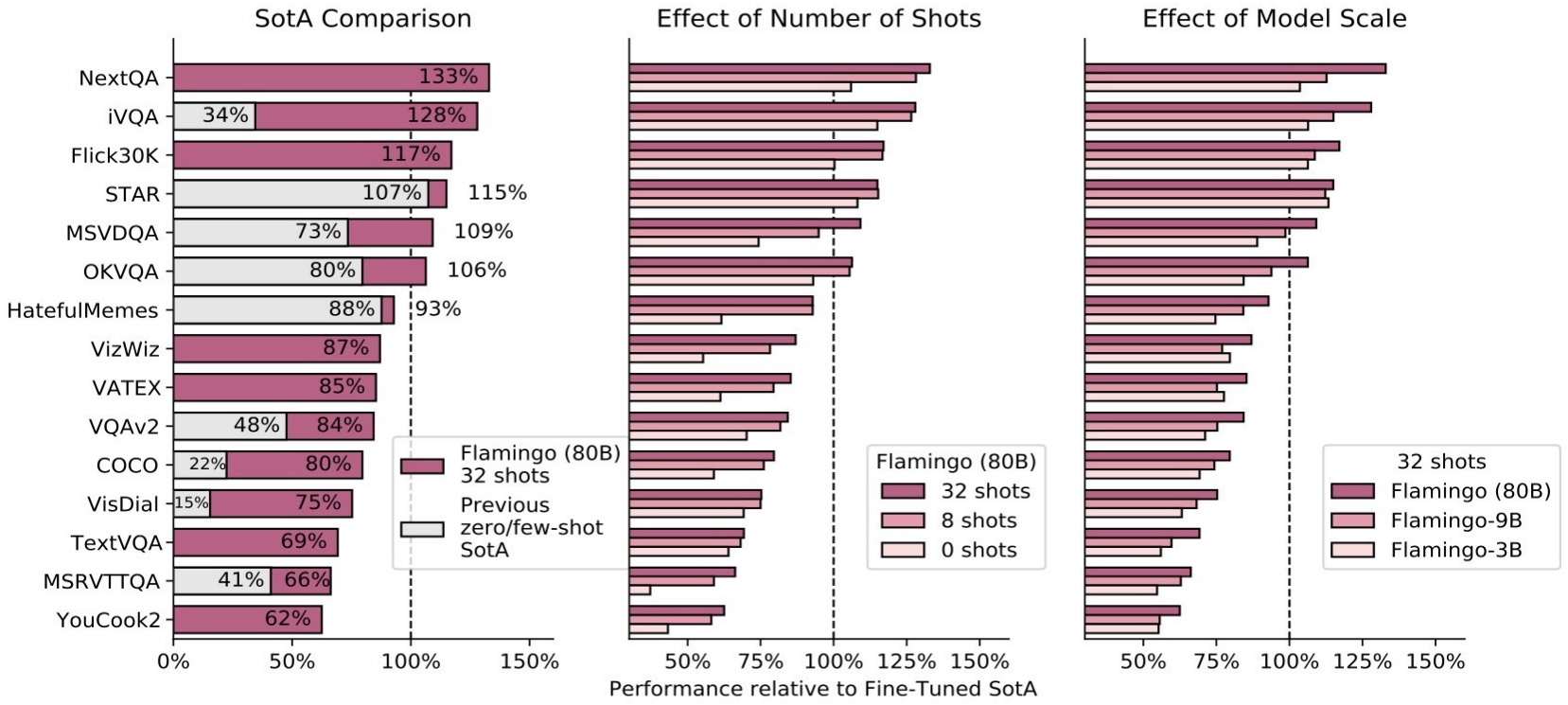
Training objective and optimisation strategy

$$\sum_{m=1}^{M} \lambda_m \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}_m} \left[-\sum_{\ell=1}^{L} \log p(y_\ell | y_{<\ell}, x_{\leq\ell}) \right],$$

where $\mathcal{D}_m$ and λ_m is the m-th dataset and the positive scalar weighting its influence in the loss

Experiments

	Requires model sharding	Frozen Language	Frozen Vision	Trainable GATED XATTN-DENSE	Trainable Resampler	Total count
Flamingo-3B	✗	1.4B	435M	1.2B (every)	194M	**3.2B**
Flamingo-9B	✗	7.1B	435M	1.6B (every 4th)	194M	**9.3B**
Flamingo	✓	70B	435M	10B (every 7th)	194M	**80B**



Experiments

Method	VQAV2		COCO	VATEX	VizWiz		MSRVTTQA	VisDial		YouCook2	TextVQA		HatefulMemes
	test-dev	test-std	test	test	test-dev	test-std	test	valid	test-std	valid	valid	test-std	test seen
Flamingo - 32 shots	67.6	-	113.8	65.1	49.8	-	31.0	56.8	-	86.8	36.0	-	70.0
SimVLM [134]	80.0	80.3	**143.3**	-	-	-	-	-	-	-	-	-	-
OFA [129]	79.9	80.0	149.6	-	-	-	-	-	-	-	-	-	-
Florence [150]	80.2	80.4	-	-	-	-	-	-	-	-	-	-	-
Flamingo Fine-tuned	**82.0**	**82.1**	138.1	**84.2**	**65.7**	**65.4**	**47.4**	61.8	59.7	118.6	**57.1**	54.1	**86.6**
Restricted SotA†	80.2 [150]	80.4 [150]	**143.3** [134]	76.3 [165]	- -	- -	46.8 [57]	**75.2** [87]	**74.5** [87]	**138.7** [142]	54.7 [147]	**73.7** [92]	75.4 [60]
Unrestricted SotA	81.3 [143]	81.3 [143]	149.6 [129]	81.4 [165]	57.2 [70]	60.6 [70]	- -	- -	75.4 [133]	- -	- -	- -	84.6 [164]

Table 6 | **Comparison to SotA when fine-tuning *Flamingo*.** We fine-tune *Flamingo* on all nine tasks where *Flamingo* was SotA overall with few-shot learning. *Flamingo* sets a new SotA on five of these tasks sometimes even beating methods that resorts to known performance optimization tricks such as model ensembling (on VQAv2, VATEX, VizWiz and HatefulMemes). Best numbers among the restricted SotA are in **bold**. Best numbers overall are underlined. Restricted SotA†: only includes methods that use a single model (not ensembles) and do not directly optimise the test metric (no CIDEr optimisation).

Examples

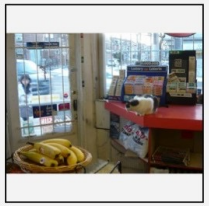
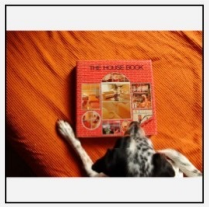
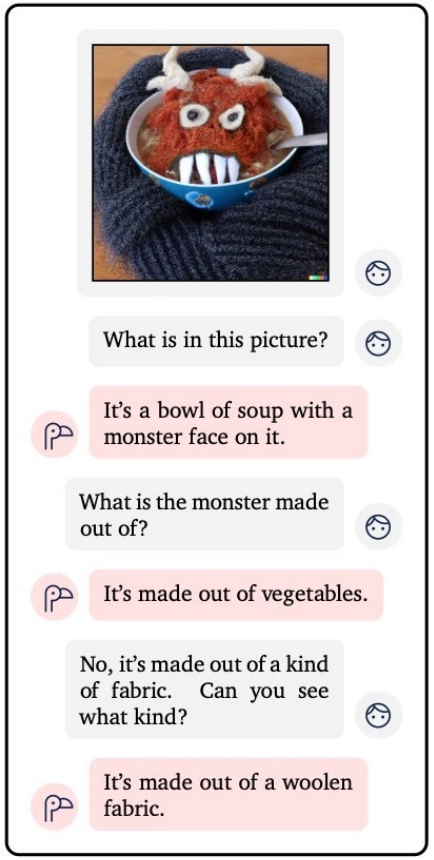
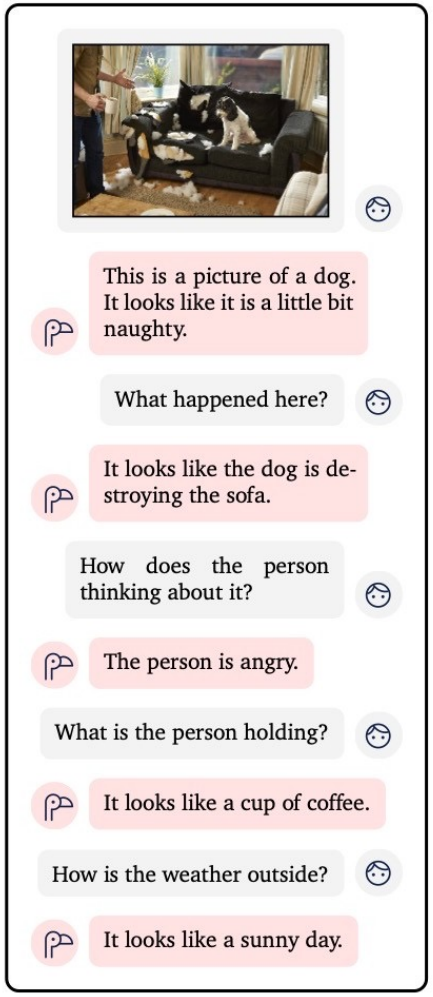
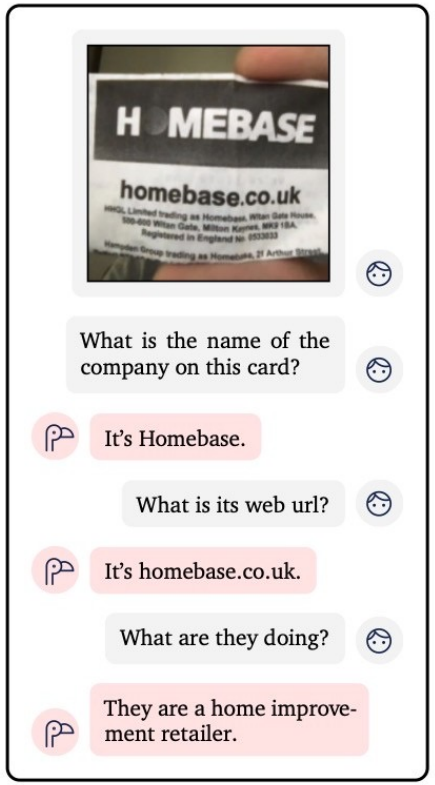
Input Prompt	 Question: What nutrients is the fruit in the image rich in? Answer:	 Question: What is the purpose of this? Answer:	 Question: What season is this? Explain why. Answer:
Completion	**Vitamin C, potassium, and fiber.**	**To keep the mosquitoes out of the bed.**	**This is winter season. The Christmas tree and the presents are the main reasons why it is winter.**
Input Prompt	 Question: What is the title of the book? Answer:	 The ingredients of this dish are	 The soundtrack includes
Completion	**The House Book.**	**: beef, potatoes, carrots, broccoli, and lemon.**	**the songs "Let It Go" and "For the First Time in Forever" by Tony Award® winner Idina Menzel, who plays Elsa [...]**

Figure 10 | **Selected single image samples.** Gray boxes are user input and the pink boxes are *Flamingo* output.

Examples







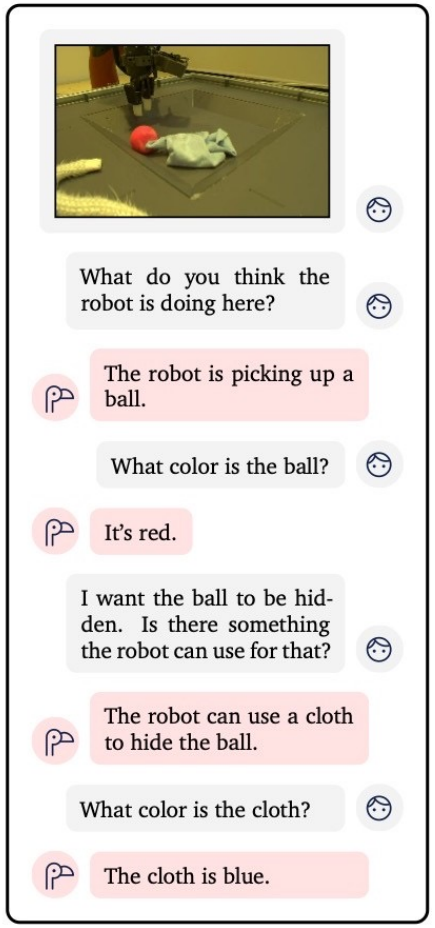


Figure 11 | **Selected dialogue samples.** Gray boxes are user input and the pink boxes are *Flamingo* output. For dialogue, *Flamingo* is provided with a custom prompt (hidden from the visualization but shown in Appendix D) containing a dialogue with 3 random images, but it is not fine-tuned for dialogue in any other way.

Examples

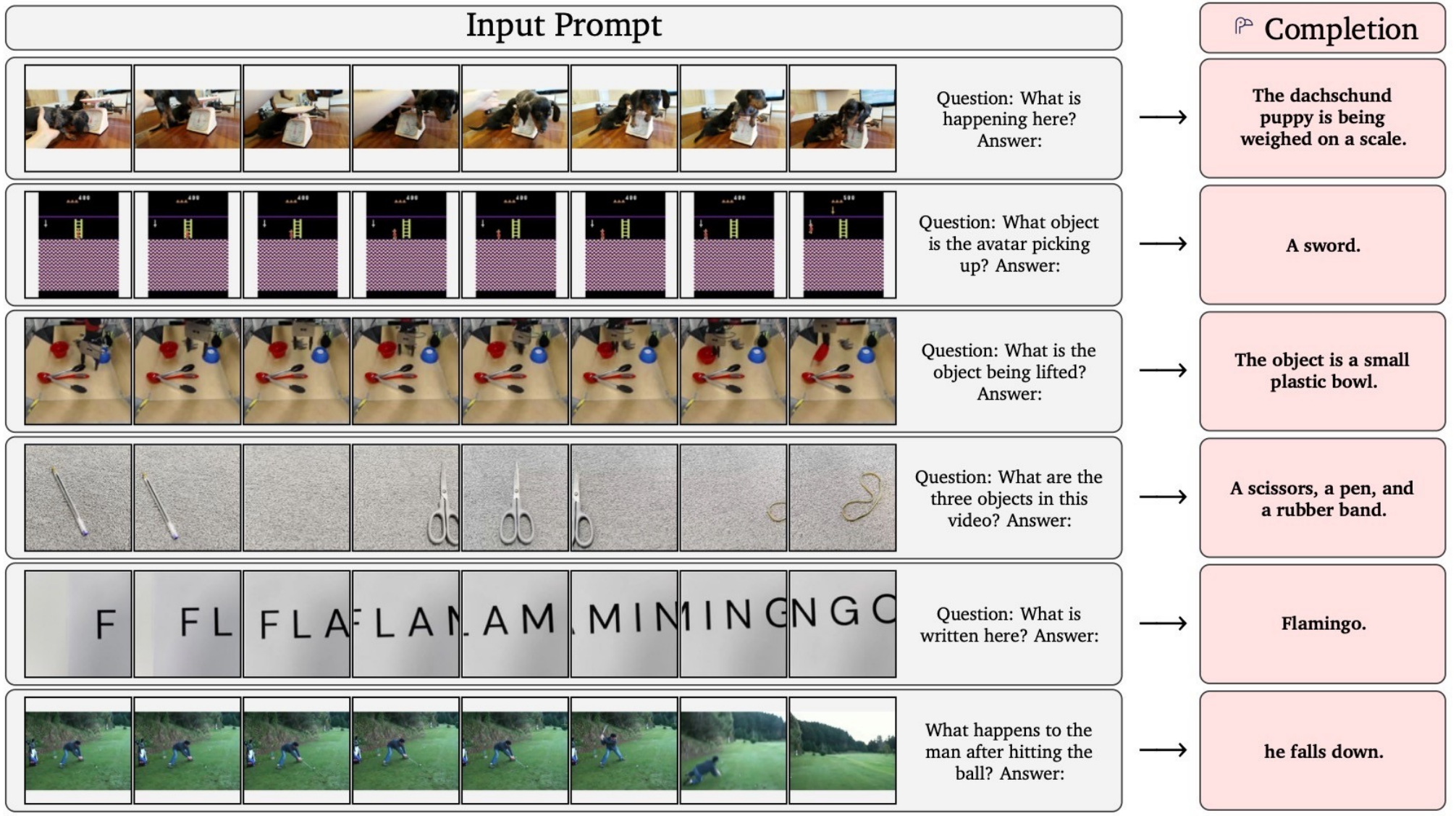
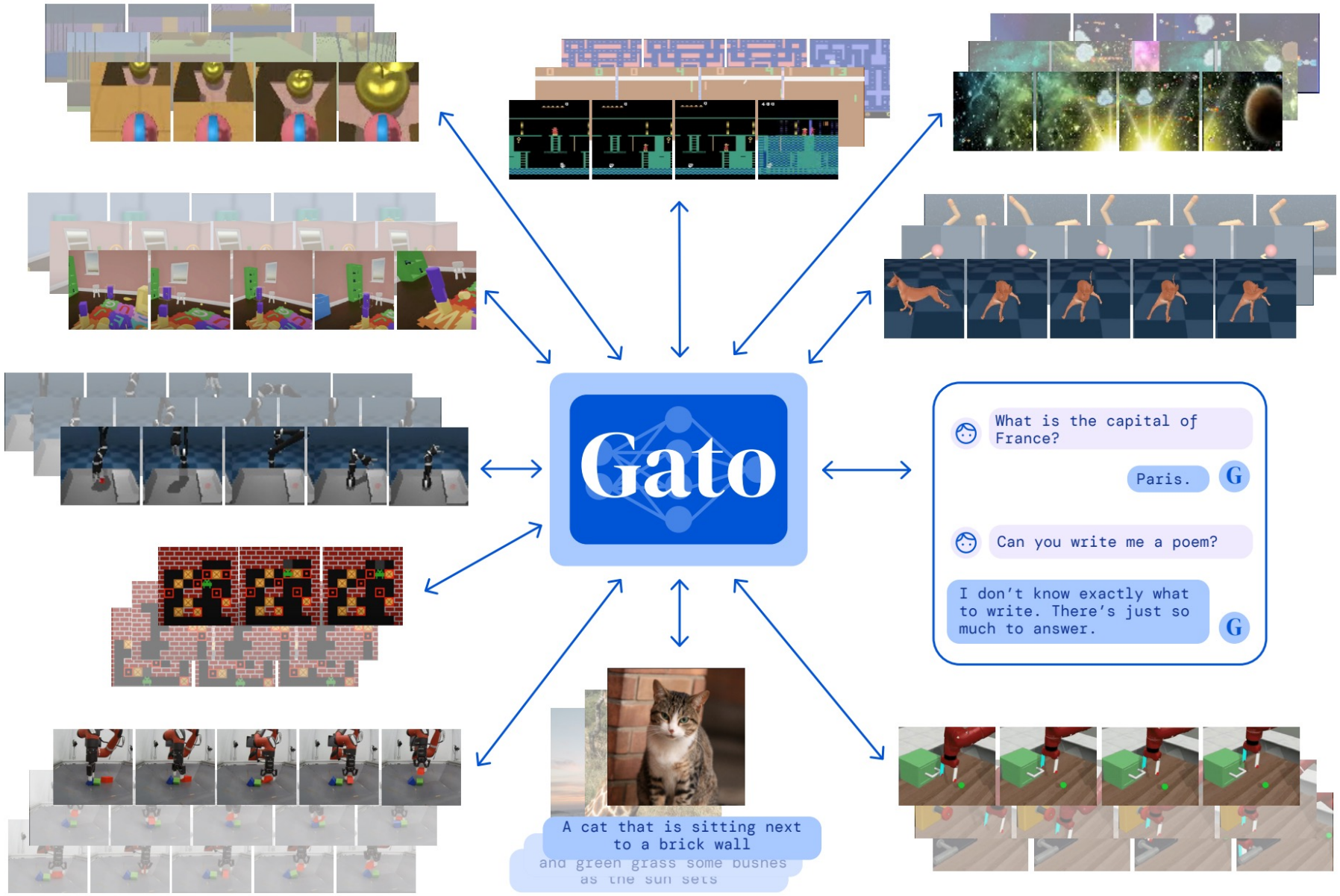
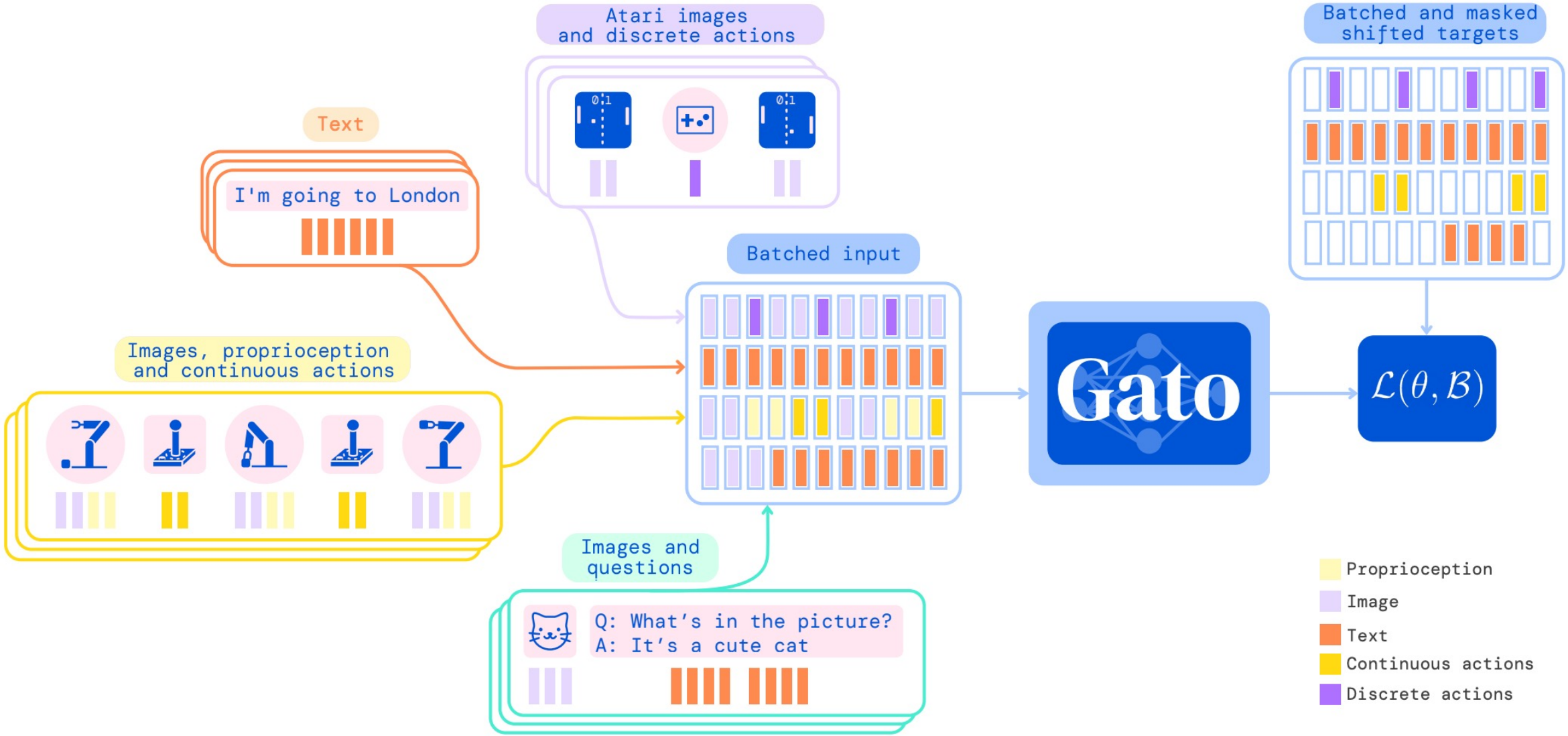


Figure 12 | **Selected video samples.** These are all of the frames the model sees. (Best viewed with zoom.)

GATO


Gato
What is the capital of France?
Paris.
Can you write me a poem?
I don't know exactly what to write. There's just so much to answer.
A cat that is sitting next to a brick wall
and green grass some busnes as the sun sets


Atari images
and discrete actions
Batched and masked
shifted targets
Text
I'm going to London
Batched input
Images, proprioception
and continuous actions
Gato
$\mathcal{L}(\theta, \mathcal{B})$
Images and
questions
Q: What's in the picture?
A: It's a cute cat
Proprioception
Image
Text
Continuous actions
Discrete actions

Tokenization

- Text: SentencePiece ([0, 32000) tokens)
- Image: patch image en 16x16 patches
- Discrete values: sequence of integers in range of [0, 1024)
- Continuous values: discretized to range of [32000, 33024)

$$s_{1:L} = [[y^1_{1:k}, x^1_{1:m}, z^1_{1:n}, '|', a^1_{1:A}], \ldots, [y^T_{1:k}, x^T_{1:m}, z^T_{1:n}, '|', a^T_{1:A}]]$$

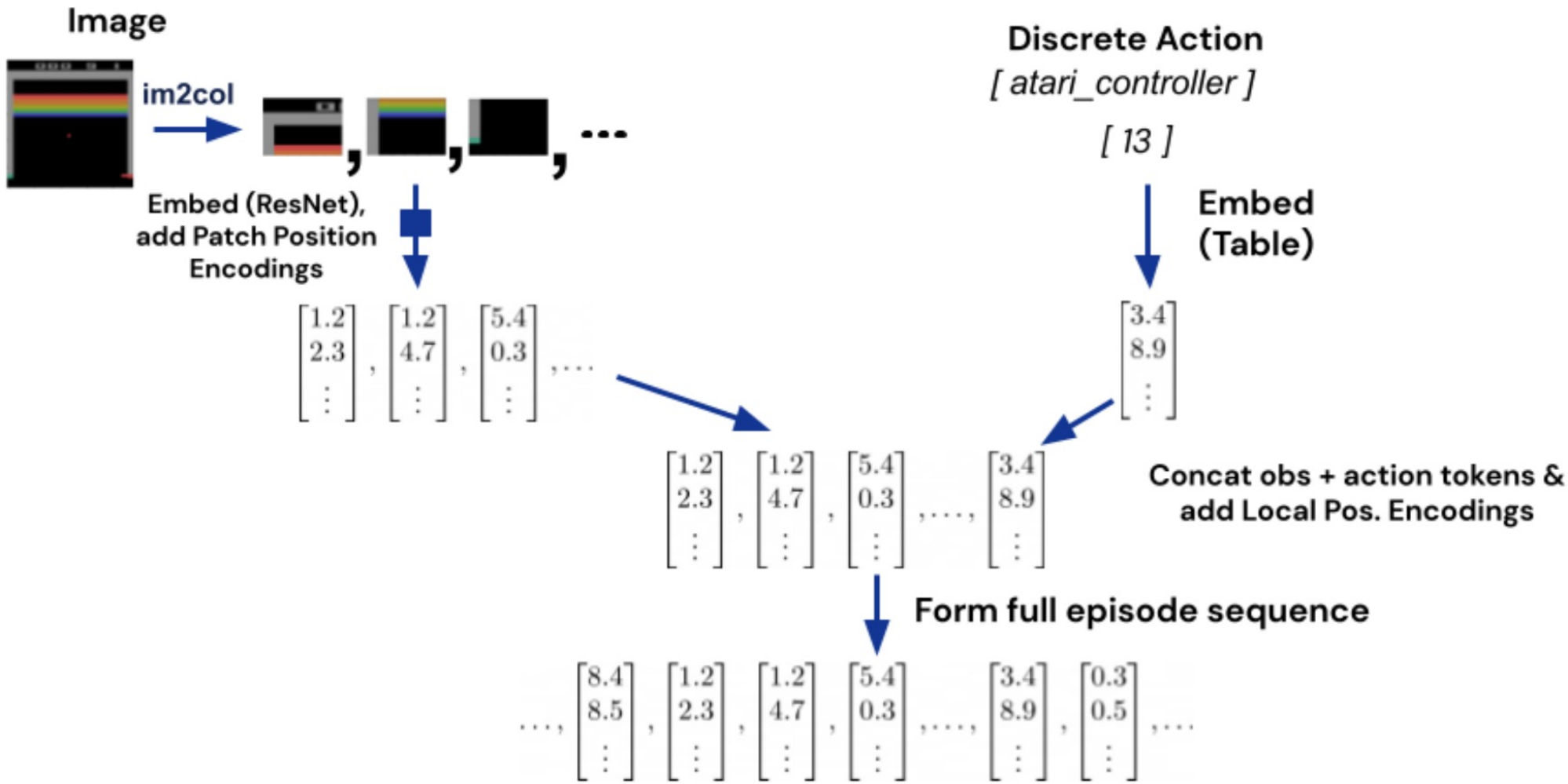


Image
im2col
Embed (ResNet), add Patch Position Encodings
Discrete Action
[atari_controller]
[13]
Embed (Table)
Concat obs + action tokens & add Local Pos. Encodings
Form full episode sequence

Proprioception

[cart_x, velocity_x, pole_x, pole_y]

[0.3, 1.5, 0.25, -0.4]

Continuous Action

[cart_move_x]

[-0.7]

$$F(x) = \text{sgn}(x)\frac{\log(|x|\mu + 1.0)}{\log(M\mu + 1.0)}$$

Mu-law Encode, Discretize & Embed (Table)

Discretize & Embed (Table)

$$\begin{bmatrix}5.9\\8.4\\\vdots\end{bmatrix}, \begin{bmatrix}8.7\\2.6\\\vdots\end{bmatrix}, \begin{bmatrix}7.6\\5.3\\\vdots\end{bmatrix}, \begin{bmatrix}4.6\\3.4\\\vdots\end{bmatrix} \qquad \begin{bmatrix}9.0\\3.3\\\vdots\end{bmatrix}$$

Token Embedding

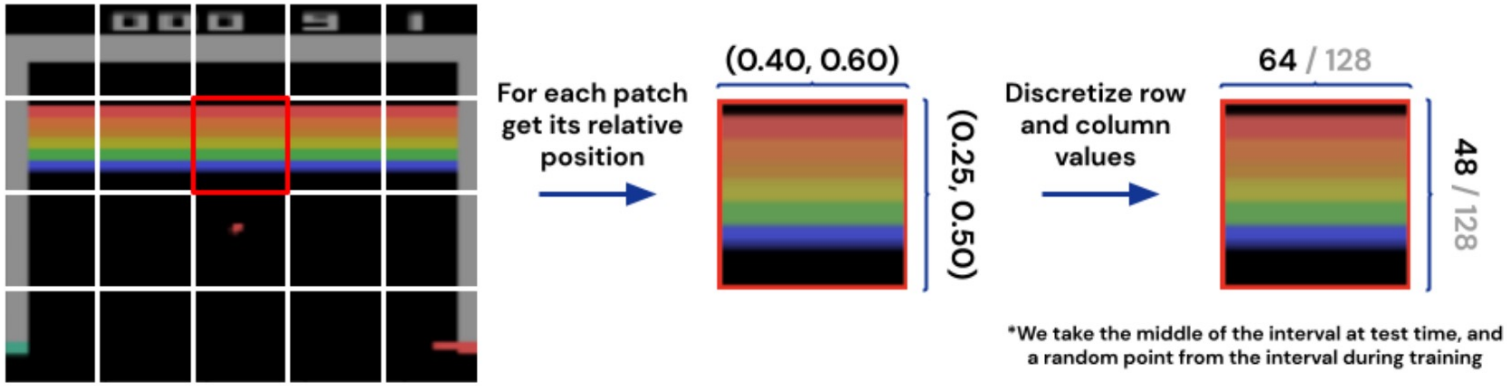
$$\begin{bmatrix}5.9\\8.4\\\vdots\end{bmatrix}, \begin{bmatrix}8.7\\2.6\\\vdots\end{bmatrix}, \begin{bmatrix}7.6\\5.3\\\vdots\end{bmatrix}, \begin{bmatrix}4.6\\3.4\\\vdots\end{bmatrix}, \begin{bmatrix}9.0\\3.3\\\vdots\end{bmatrix}$$

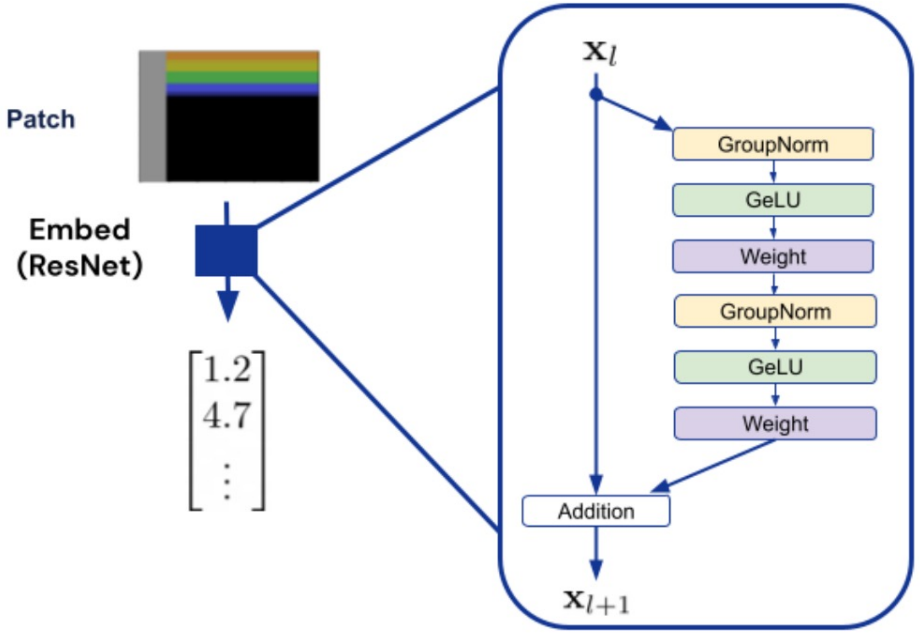
Concat obs + action tokens & add Local Pos. Encodings

Form full episode sequence

$$\ldots, \begin{bmatrix}2.3\\8.3\\\vdots\end{bmatrix}, \begin{bmatrix}5.9\\8.4\\\vdots\end{bmatrix}, \begin{bmatrix}8.7\\2.6\\\vdots\end{bmatrix}, \begin{bmatrix}7.6\\5.3\\\vdots\end{bmatrix}, \begin{bmatrix}4.6\\3.4\\\vdots\end{bmatrix}, \begin{bmatrix}9.0\\3.3\\\vdots\end{bmatrix}, \begin{bmatrix}2.3\\0.9\\\vdots\end{bmatrix}, \ldots$$

Image Embeddings

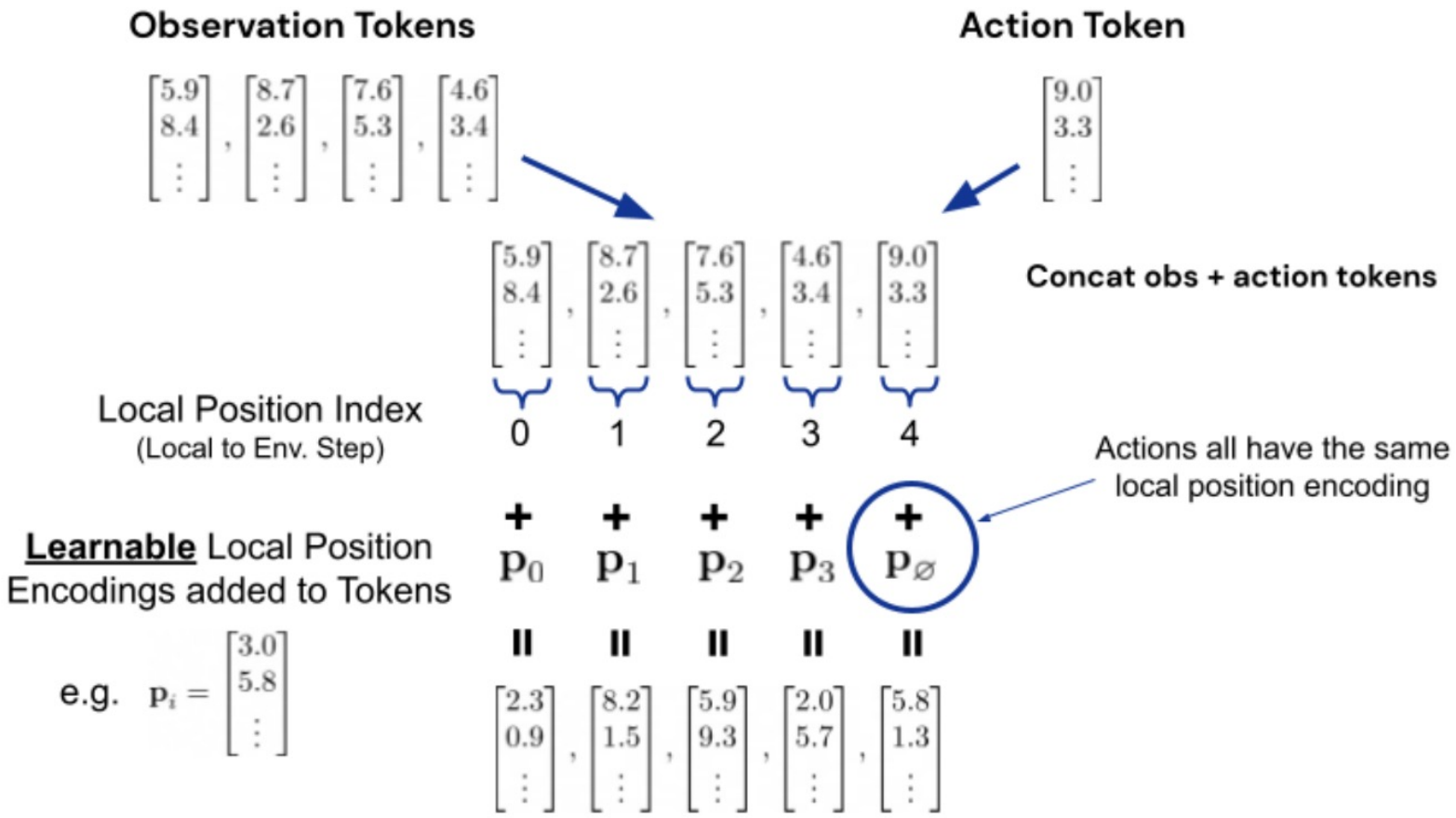




Final patch embedding | ResNet-based embedding | Column position embedding | Row position embedding

$$\begin{bmatrix} 4.6 \\ 3.4 \\ \vdots \end{bmatrix} = \begin{bmatrix} 7.6 \\ 5.3 \\ \vdots \end{bmatrix} + col_{64} + row_{48}$$

Position tokens



Model

Hyperparameter	Gato 1.18B	364M	79M
Transformer blocks	24	12	8
Attention heads	16	12	24
Layer width	2048	1536	768
Feedforward hidden size	8192	6144	3072
Key/value size	128	128	32
Shared embedding	True		
Layer normalization	Pre-norm		
Activation Function	GEGLU		

Training 1.18B

- 16x16 TPU v3
- Steps: 1M
- Batch size: 512
- Token sequence length: 1024
- Days: 4

- Loss: $\mathcal{L}(\theta, \mathcal{B}) = -\sum_{b=1}^{|\mathcal{B}|}\sum_{l=1}^{L} m\,(b,l)\log p_\theta\left(s_l^{(b)}|s_1^{(b)},\ldots,s_{l-1}^{(b)}\right)$

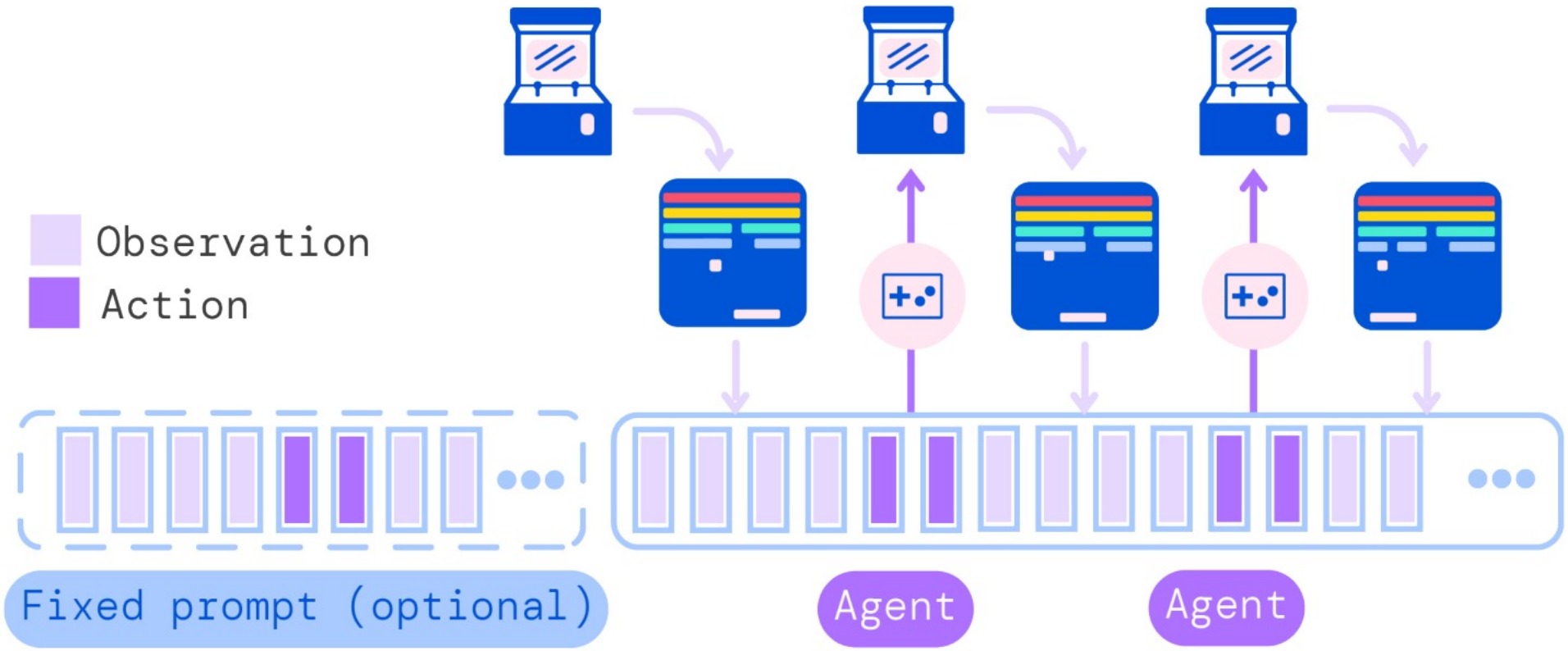
Datasets

Control environment	Tasks	Episodes	Approx. Tokens	Sample Weight
DM Lab	254	16.4M	194B	9.35%
ALE Atari	51	63.4K	1.26B	9.5%
ALE Atari Extended	28	28.4K	565M	10.0%
Sokoban	1	27.2K	298M	1.33%
BabyAI	46	4.61M	22.8B	9.06%
DM Control Suite	30	395K	22.5B	4.62%
DM Control Suite Pixels	28	485K	35.5B	7.07%
DM Control Suite Random Small	26	10.6M	313B	3.04%
DM Control Suite Random Large	26	26.1M	791B	3.04%
Meta-World	45	94.6K	3.39B	8.96%
Procgen Benchmark	16	1.6M	4.46B	5.34%
RGB Stacking simulator	1	387K	24.4B	1.33%
RGB Stacking real robot	1	15.7K	980M	1.33%
Modular RL	38	843K	69.6B	8.23%
DM Manipulation Playground	4	286K	6.58B	1.68%
Playroom	1	829K	118B	1.33%
Total	596	63M	1.5T	85.3%

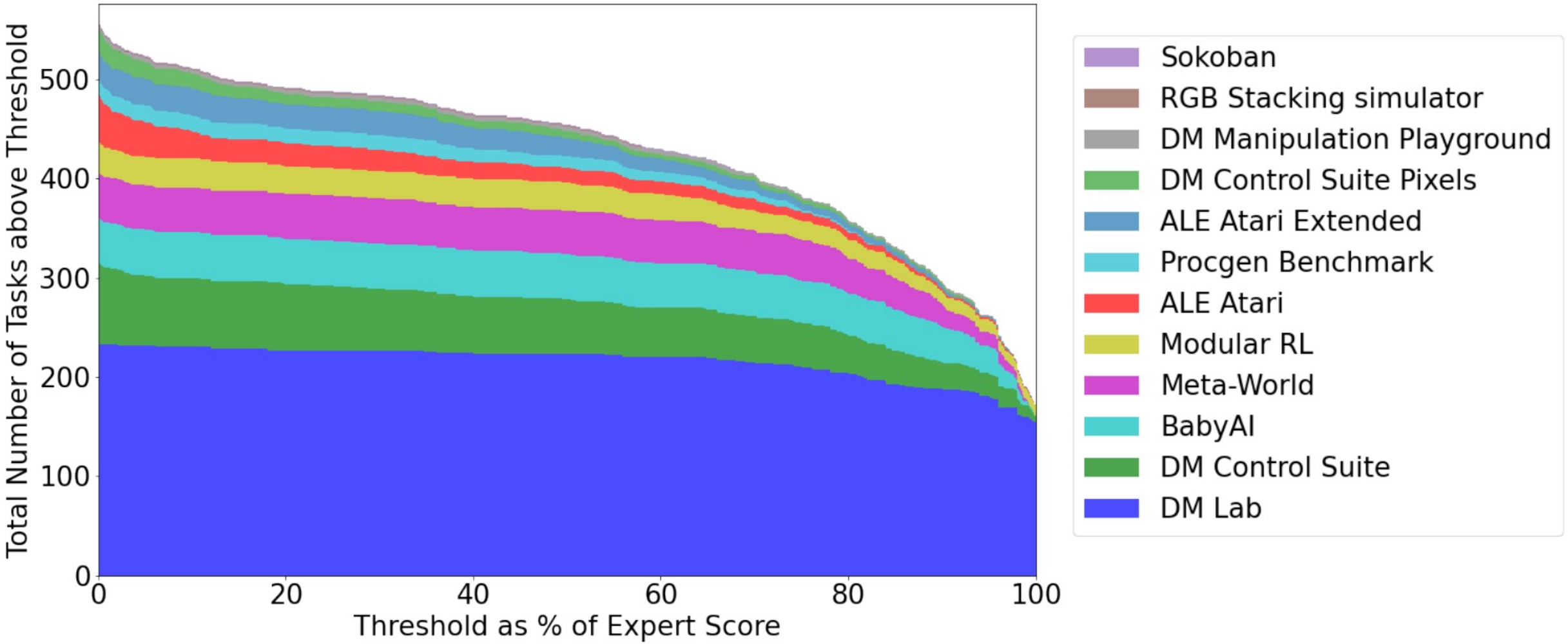
Vision / language dataset	Sample Weight
MassiveText	6.7%
M3W	4%
ALIGN	0.67%
MS-COCO Captions	0.67%
Conceptual Captions	0.67%
LTIP	0.67%
OKVQA	0.67%
VQAV2	0.67%
Total	14.7%

- Games: generated by SoTA RL Agents
- Texts: MassiveText (web pages, books, news articles, code)
- Images: ALIGN (1.8B), LTIP (312M)

Gato as control policy



Results


Total Number of Tasks above Threshold
0
100
200
300
400
500
0
20
40
60
80
100
Threshold as % of Expert Score
Sokoban
RGB Stacking simulator
DM Manipulation Playground
DM Control Suite Pixels
ALE Atari Extended
Procgen Benchmark
ALE Atari
Modular RL
Meta-World
BabyAI
DM Control Suite
DM Lab

The colorful ceramic toys are on the living room floor.

a living room with three different color deposits on the floor

a room with a long red rug a tv and some pictures

Man standing in the street wearing a suit and tie.

A man in a blue suit with a white bow tie and black shoes.

A man with a hat in his hand looking at the camera

A bearded man is holding a plate of food.

Man holding up a banana to take a picture of it.

a man smiles while holding up a slice of cake

a group of people that is next to a big horse

A tan horse holding a piece of cloth lying on the ground.

Two horses are laying on their side on the dirt.

Man biting a kite while standing on a construction site

a big truck in the middle of a road

A truck with a kite painted on the back is parked by rocks.

a white horse with a blue and silver bridle

A white horse with blue and gold chains.

A horse is being shown behind a wall.

a couple of people are out in the ocean

A surfer riding a wave in the ocean.

A surfer with a wet suit riding a wave.

A baseball player pitching a ball on top of a baseball field.

A man throwing a baseball at a pitcher on a baseball field.

A baseball player at bat and a catcher in the dirt during a baseball game

Pistachios on top of a bowl with coffee on the side.

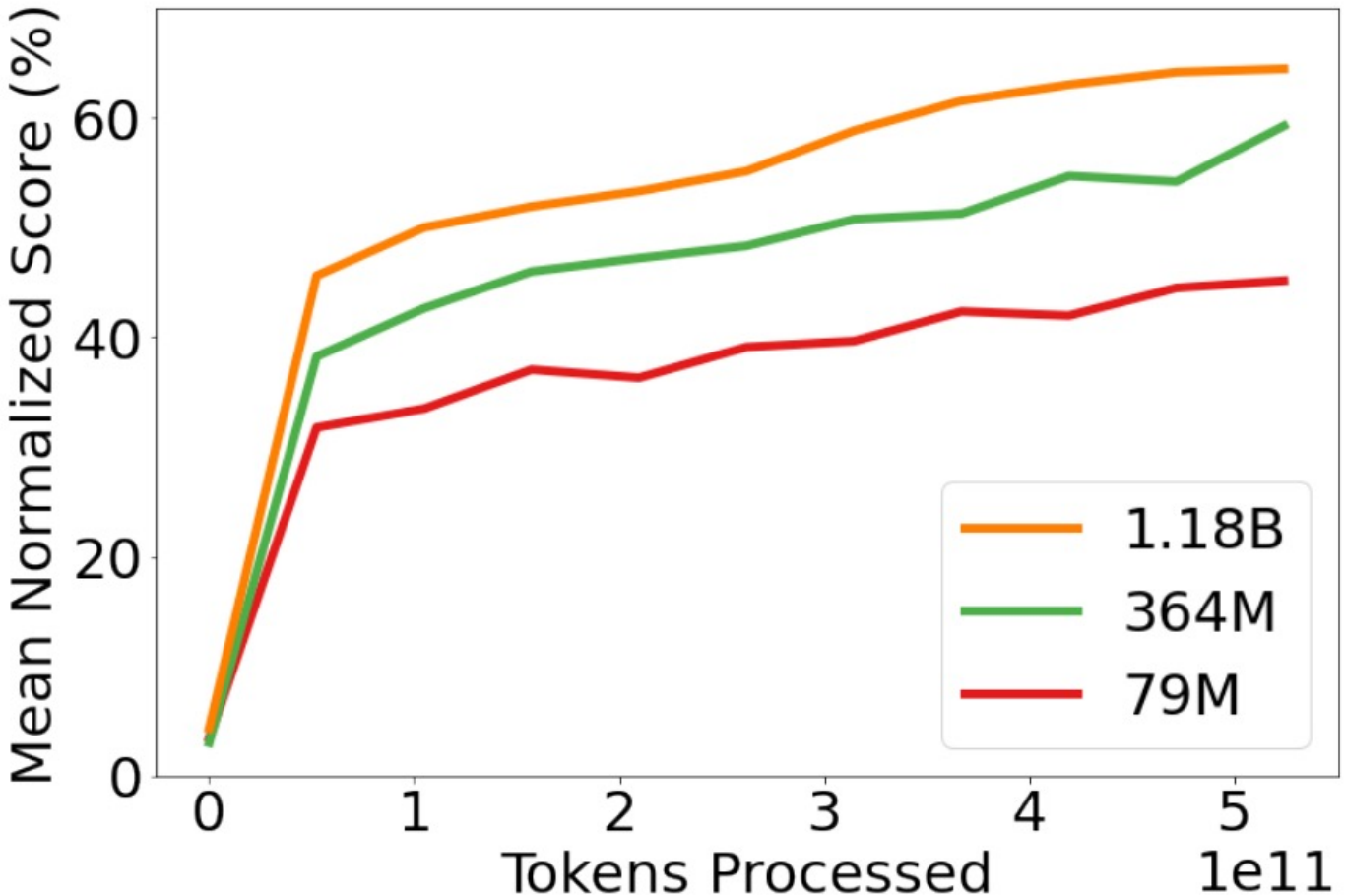
A bowl and a glass of liquid sits on a table.

A white plate filled with a banana bread next to a cup of coffee.

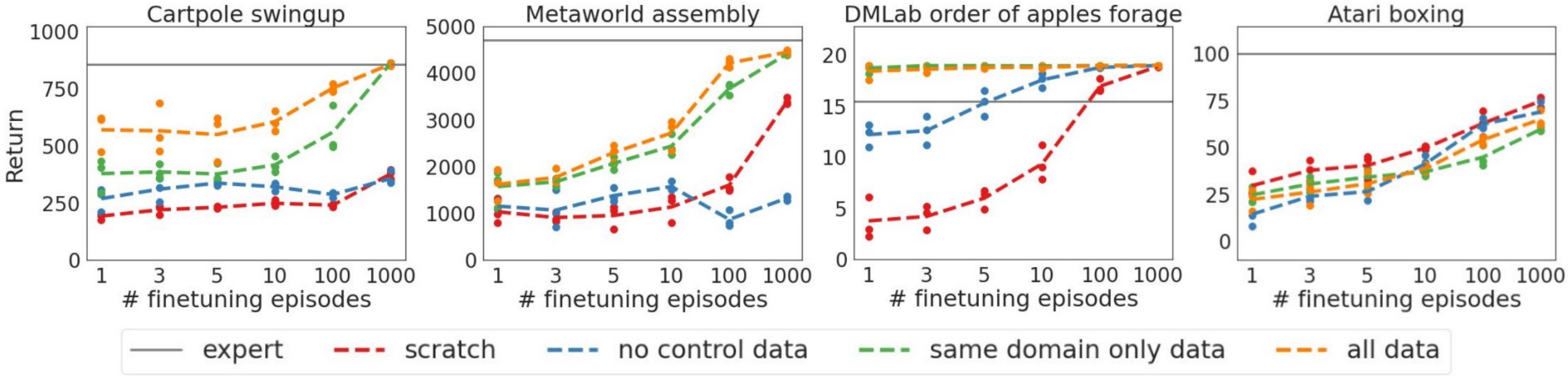
A group of children eating pizza at a table.

Two boys having pizza for lunch with their friends.

The boys are eating pizza together at the table.

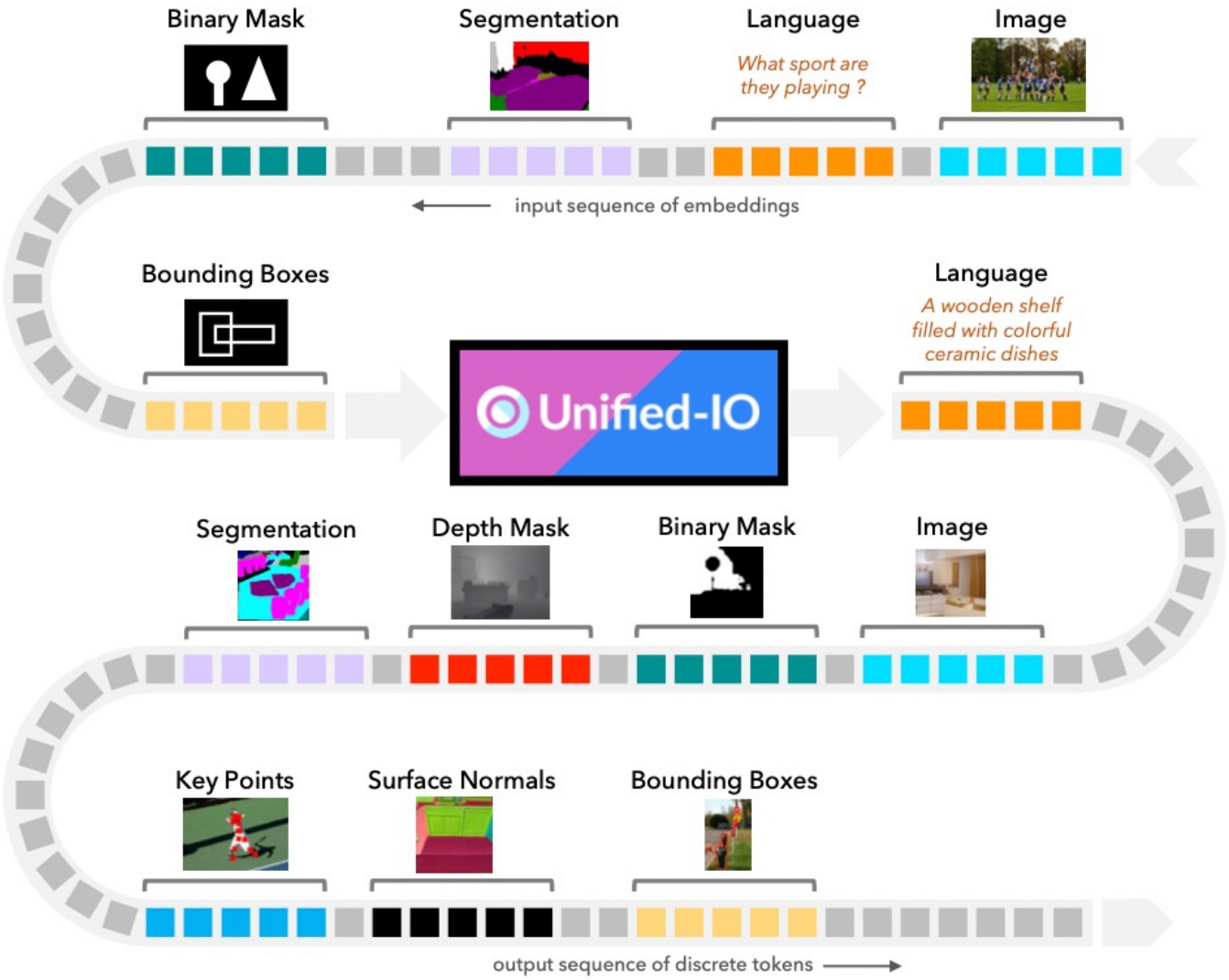

Mean Normalized Score (%)
0
20
40
60
0
1
2
3
4
5
Tokens Processed
1e11
1.18B
364M
79M

Generalization



Unified IO

Architecture



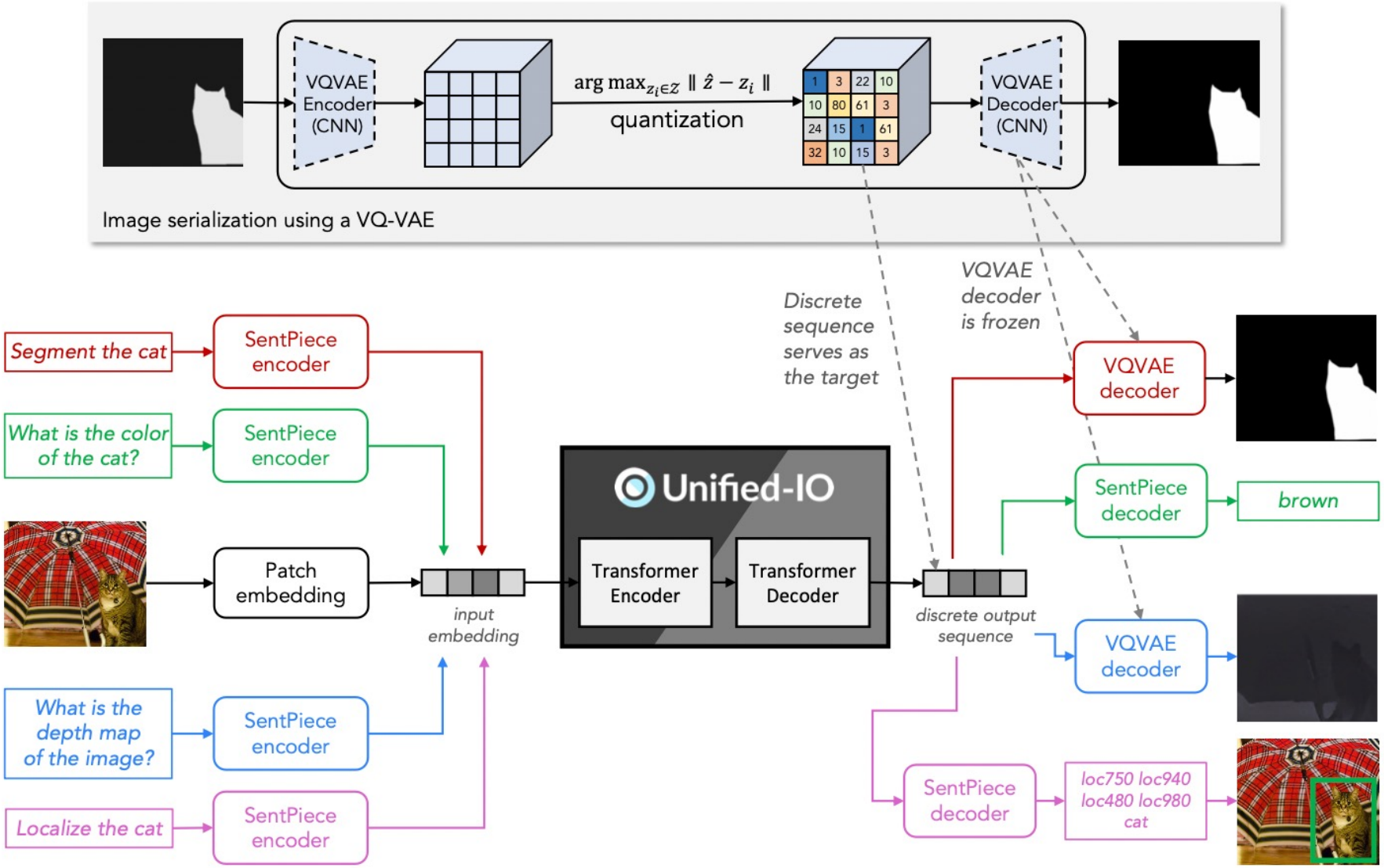
Tasks

Image Classification
Object Detection
Semantic Segmentation
Depth Estimation
Surface Normal Estimation
Segment-based Image Generation
Image Inpainting
Pose Estimation
Relationship Detection
Image Captioning
Visual QA
Referring Expressions
Situation Recognition
Text-based Image Generation
Visual Commonsense
Classification in context
Region Captioning
GLUE Benchmark tasks
Reading comprehension
Natural Language Inference
Grounded Commonsense Inference

Tasks and Data

	Example Source	Size				Input Modalities				Output Modalities			
		Datasets	Size	Percent	Rate	Text	Image	Sparse	Dense	Text	Image	Sparse	Dense
Image Synthesis		**14**	**56m**	**43.0**	**18.7**	✓	✓	✓	✓	-	✓	-	-
Image Synthesis from Text	*RedCaps*	9	55m	41.9	16.7	✓	-	-	-	-	✓	-	-
Image Inpainting	*VG*	3	1.2m	0.9	1.5	✓	✓	✓	-	-	✓	-	-
Image Synthesis from Seg.	*LVIS*	2	220k	0.2	0.6	✓	-	-	✓	-	✓	-	-
Sparse Labelling		**10**	**8.2m**	**6.3**	**12.5**	✓	✓	✓	-	-	-	✓	-
Object Detection	*Open Images*	3	1.9m	1.5	3.6	-	✓	-	-	-	-	✓	-
Object Localization	*VG*	3	6m	4.6	7.1	✓	✓	-	-	-	-	✓	-
Keypoint Estimation	*COCO*	1	140k	0.1	0.7	-	✓	✓	-	-	-	✓	-
Referring Expression	*RefCoco*	3	130k	0.1	1.1	✓	✓	-	-	-	-	✓	-
Dense Labelling		**6**	**2.4m**	**1.8**	**6.2**	✓	✓	-	-	-	-	-	✓
Depth Estimation	*NYU Depth*	1	48k	0.1	0.4	-	✓	-	-	-	-	-	✓
Surface Normal Estimation	*Framenet*	2	210k	0.2	1.1	-	✓	-	-	-	-	-	✓
Object Segmentation	*LVIS*	3	2.1m	1.6	4.7	✓	✓	-	-	-	-	-	✓
Image Classification		**9**	**22m**	**16.8**	**12.5**	-	✓	✓	-	✓	-	-	-
Image Classification	*ImageNet*	6	16m	12.2	8.1	✓	✓	-	-	✓	-	-	-
Object Categorization	*COCO*	3	6m	4.6	4.4	-	✓	✓	-	✓	-	-	-
Image Captioning		**7**	**31m**	**23.7**	**12.5**	-	✓	✓	-	✓	-	-	-
Webly Supervised Captioning	*CC12M*	3	26m	19.7	8.8	-	✓	-	-	✓	-	-	-
Supervised Captioning	*VizWiz*	3	1.4m	1.1	1.7	-	✓	-	-	✓	-	-	-
Region Captioning	*VG*	1	3.8m	2.9	2.0	-	✓	✓	-	✓	-	-	-
Vision & Language		**16**	**4m**	**3.0**	**12.5**	✓	✓	✓	-	✓	-	-	✓
Visual Question Answering	*VQA 2.0*	13	3.3m	2.5	10.4	✓	✓	✓	-	✓	-	-	-
Relationship Detection	*VG*	2	640k	0.5	1.9	-	✓	✓	-	✓	-	-	-
Grounded VQA	*VizWiz*	1	6.5k	0.1	0.1	✓	✓	-	-	✓	-	-	✓
NLP		**31**	**7.1m**	**5.4**	**12.5**	✓	-	-	-	✓	-	-	-
Text Classification	*MNLI*	17	1.6m	1.2	4.8	✓	-	-	-	✓	-	-	-
Question Answering	*SQuAD*	13	1.7m	1.3	5.2	✓	-	-	-	✓	-	-	-
Text Summarization	*Gigaword*	1	3.8m	2.9	2.5	✓	-	-	-	✓	-	-	-
Language Modelling		**2**	-	-	**12.5**	✓	-	-	-	✓	-	-	-
Masked Language Modelling	*C4*	2	-	-	12.5	✓	-	-	-	✓	-	-	-
All Tasks		**95**	**130m**	**100**	**100**	✓	✓	✓	✓	✓	✓	✓	✓

4 Demo Downstream Tasks



Training

1. Pre-train stage

- *text span denoising*
- *masked image denoising*

2. Multi-tasking stage

- 95 datasets
- Mixing in batch
- We equally sample each group (1/8) except for image synthesis (3/16) and dense labeling (1/16) since dense labeling has significantly fewer data and image synthesis has significantly more data than other groups
- Within each group, we sample datasets proportional to the square root of their size to better expose the model to underrepresented tasks.
- Due to the large variance in dataset size, some tasks are still rarely sampled (*e.g*. depth estimation only has a 0.43% chance of being sampled)
- The total vocabulary size is 49536, with 32152 language tokens, 1000 location tokens, and 16384 vision tokens
- *Training:* random sub-sample 128 image patches for pre-training state and 256 image patches (out of 576) for multi-task stage
- Do not use dropout

Experiments

		Categorization		Localization		VQA		Refexp		Segmentation		Keypoint		Normal		All	
		ablation	test	ablation	test	ablation	test	ablation	test	ablation	test	ablation	test	ablation	test	ablation	test
0	NLL-AngMF [4]	-	-	-	-	-	-	-	-	-	-	-	-	**49.6**	**50.5**	7.2	7.1
1	Mask R-CNN [41]	-	-	44.7	45.1	-	-	-	-	26.2	26.2	**70.8**	**70.6**	-	-	20.2	20.3
2	GPV-1 [38]	33.2	33.2	42.8	42.7	50.6	49.8	25.8	26.8	-	-	-	-	-	-	21.8	21.8
3	CLIP [86]	48.1	-	-	-	-	-	-	-	-	-	-	-	-	-	6.9	-
4	OFA$_{\text{LARGE}}$ [107]	22.6	-	-	-	72.4	-	61.7	-	-	-	-	-	-	-	22.4	-
5	GPV-2 [52]	54.7	55.1	53.6	53.6	63.5	63.2	51.5	52.1	-	-	-	-	-	-	31.9	32.0
6	UNIFIED-IO$_{\text{SMALL}}$	42.6	-	50.4	-	52.9	-	51.1	-	40.7	-	46.5	-	33.5	-	45.4	-
7	UNIFIED-IO$_{\text{BASE}}$	53.1	-	59.7	-	63.0	-	68.3	-	49.3	-	60.2	-	37.5	-	55.9	-
8	UNIFIED-IO$_{\text{LARGE}}$	57.0	-	64.2	-	67.4	-	74.1	-	54.0	-	67.6	-	40.2	-	60.7	-
9	UNIFIED-IO$_{\text{XL}}$	**61.7**	**60.8**	**67.0**	**67.1**	**74.5**	**74.5**	**78.6**	**78.9**	**56.3**	**56.5**	68.1	67.7	45.0	44.3	**64.5**	**64.3**

Table 3: Comparison of our UNIFIED-IO models to recent SOTA on GRIT benchmark. UNIFIED-IO is the first model to support all seven tasks in GRIT. Results of CLIP, OFA obtained from GRIT challenge.