Multimodal Foundation Models

Google DeepMind

Image Encoders

Introduction

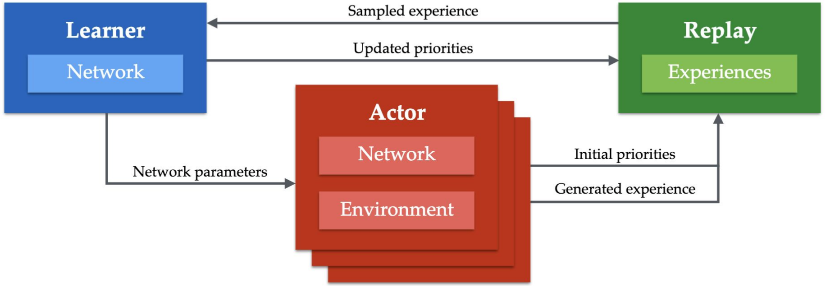
1

About Me

- Background in Reinforcement Learning
- Pioneered Distributed Reinforcement Learning
- Open-sourced Acme, Reverb, and Launchpad
  - https://github.com/google-deepmind/acme
  - https://github.com/google-deepmind/reverb
  - https://github.com/google-deepmind/launchpad

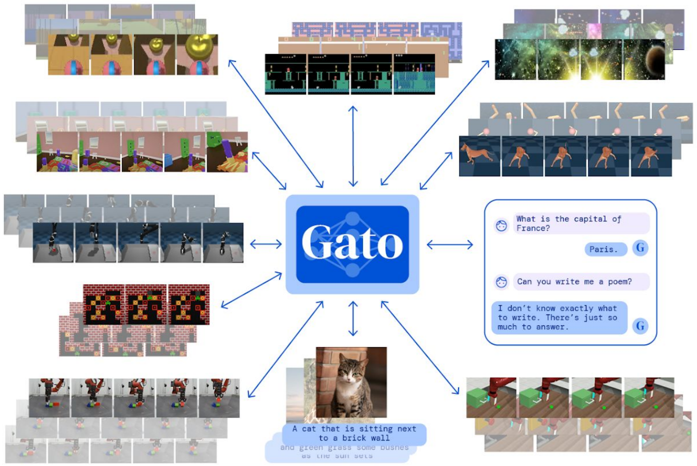
DISTRIBUTED DISTRIBUTIONAL DETERMINISTIC POLICY GRADIENTS

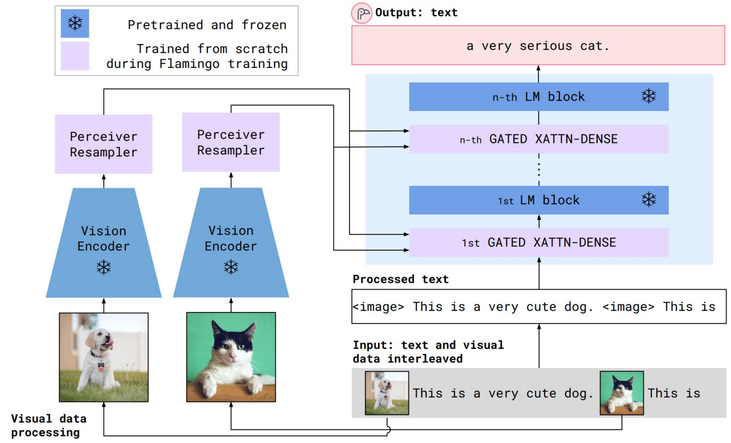
Gabriel Barth-Maron,* Matthew W. Hoffman,* David Budden, Will Dabney, Dan Horgan, Dhruva TB, Alistair Muldal, Nicolas Heess, Timothy Lillicrap
DeepMind
London, UK
{gabrielbm, mwhoffman, budden, wdabney, horgan, dhruvat, alimuldal, heess, countzero}@google.com



About Me

- Worked on A Generalist Agent (Gato).
    - Multimodal, multi-task, multi-embodiment generalist policy.
- Helped build the next-generation of the Flamingo VLM models.





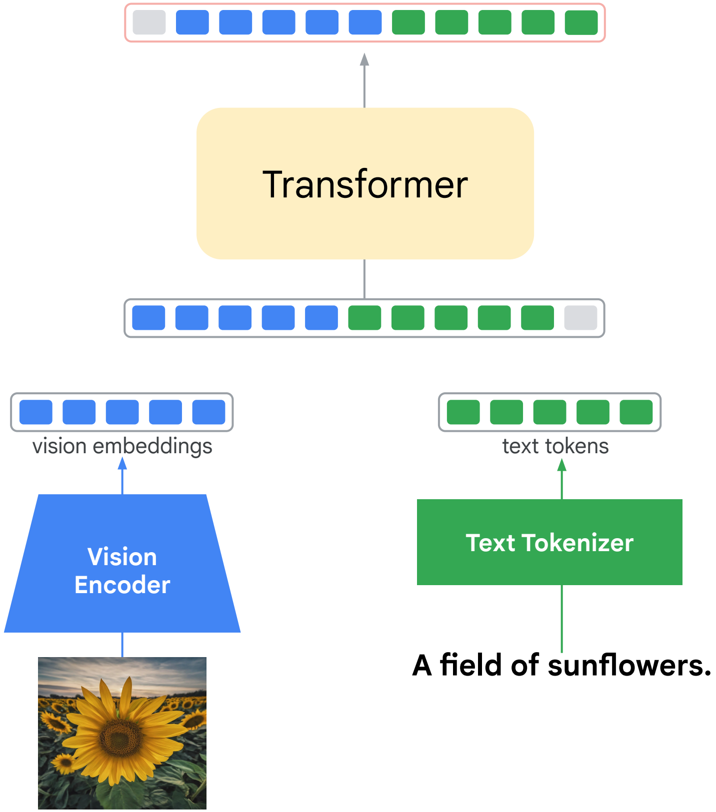
About Me

- Working on Gemini, focused on vision (images and video) generation
- Co-led the development of Google DeepMind’s latest text-to-video model, Veo, capable of generating > 1 min videos at 1080p

Gemini

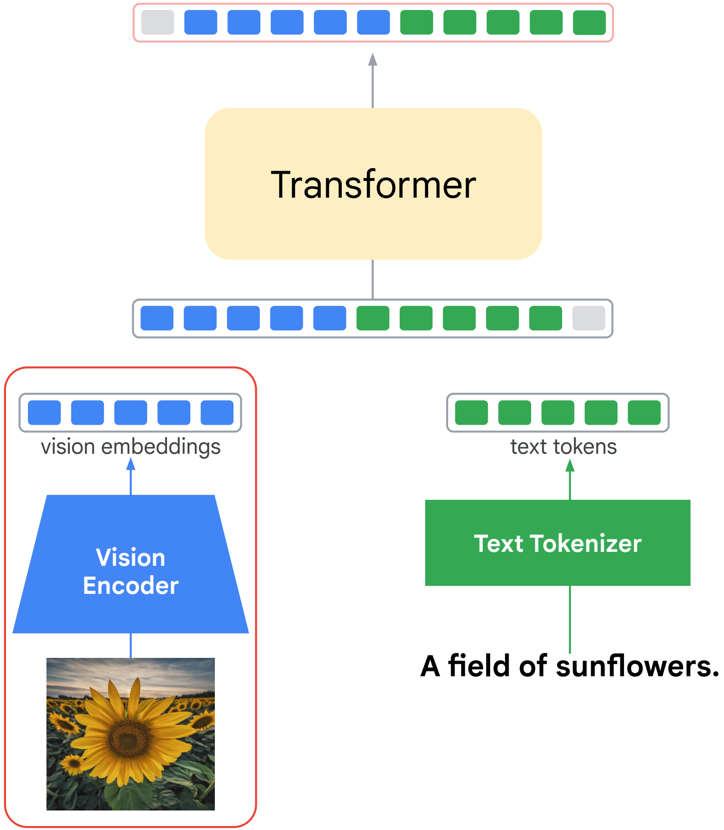
Multimodal Transformers

- Images and videos encoded with a vision encoder.
- Text is tokenized.
- Vision and text tokens are concatenated.
- Decoder-only transformer.
- Next token prediction.



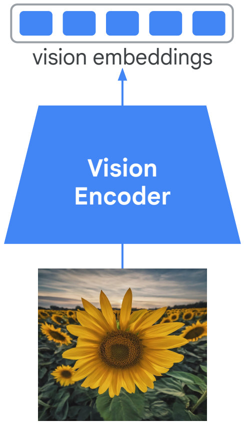
Multimodal Transformers

- Images and videos encoded with a vision encoder.
- Text is tokenized.
- Vision and text tokens are concatenated.
- Decoder-only transformer.
- Next token prediction.



Vision Encoders

- Input are raw pixel values
- Output is an embedding vector
- Embeddings should contain a rich visual description of the image



General Themes

Architecture

What architectures scale best with data and compute?

Training Objective

How can we design training objectives that are both label-efficient and transferable to downstream tasks?

Data and Compute

We have lots of data, but how should we use it?

We have limited compute, how should it be allocated?

Vision Transformer

An Image Is worth 16x16 Words: Transformers For Image Recognition At Scale

Dosovitskiy et al., 2020

2

Vision Transformer (ViT)

Context

- CNNs (mostly ResNets) were SoTA for image tasks
- Attention Is All You Need was published in 2017
- GPT-3 was published in May 2020
- Use the transformer architecture for image classification

Published as a conference paper at ICLR 2021

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

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

[*]equal technical contribution, [†]equal advising

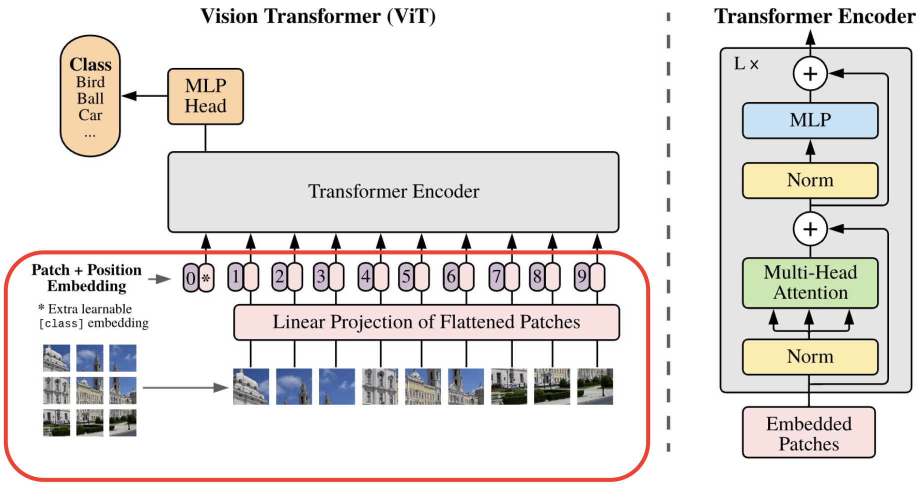
Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

Vision Transformer (ViT)

Preparing Images for the Transformer Encoder

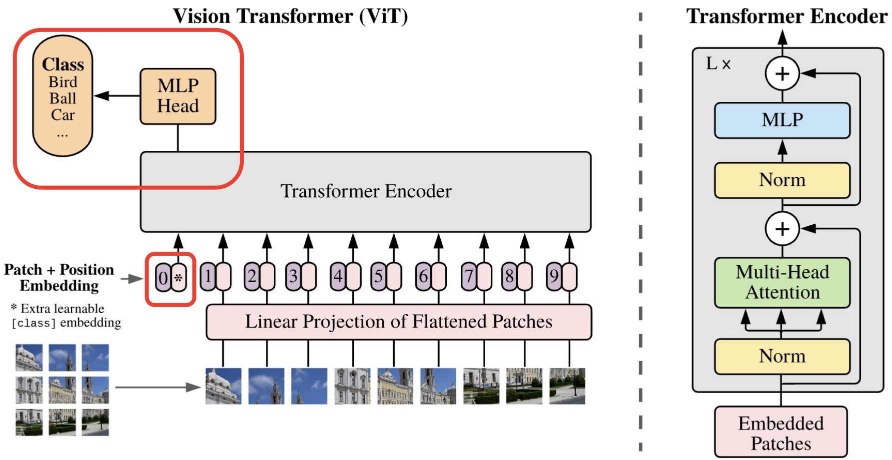
- Images are resized to 224x224 pixels.
- Images are split into non-overlapping patches of size PxP.
- A linear projection is applied to each patch of P^2 pixels (in practice this is 3*P^2 values).
- Add 1-D learnable position embeddings to the projected patches.



Vision Transformer (ViT)

Image Classification

- A `[class]` token is prepended to the flattened patch projections - as done in BERT (Devlin et al., 2019).
- An MLP is applied to the output of the first token, which is used to predict the target class.

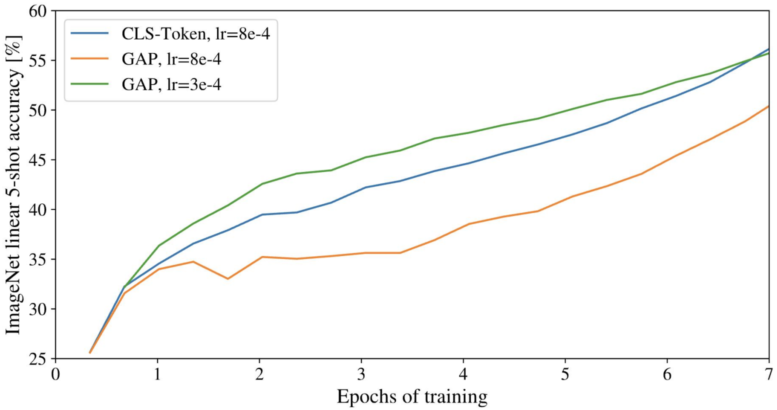


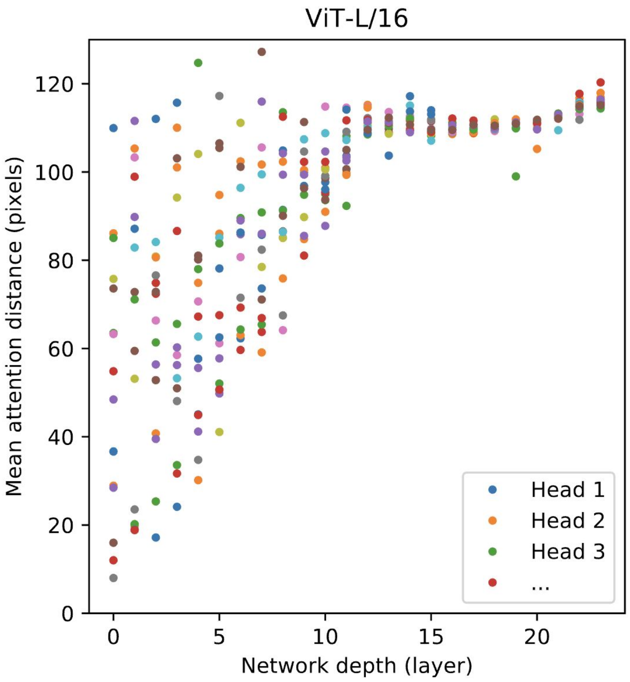
	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Vision Transformer (ViT)

Shedding Inductive Biases

- No receptive field like CNNs. Average attention distance is variable earlier in the network and larger later on.
- Tested 2D positional embeddings but saw no gain over 1D.
- Globally Averaged Pooling (GAP) works better than a separate MLP head for classification.



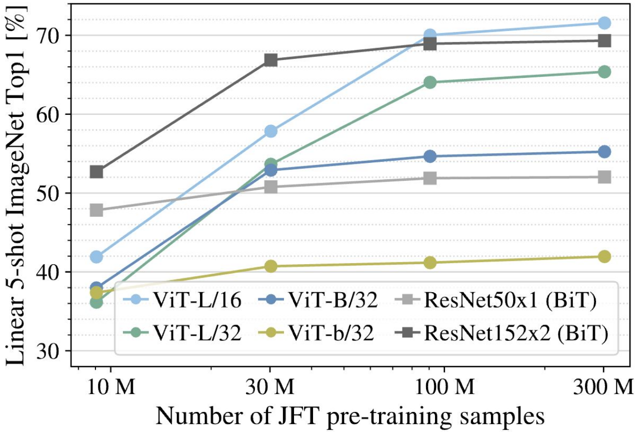


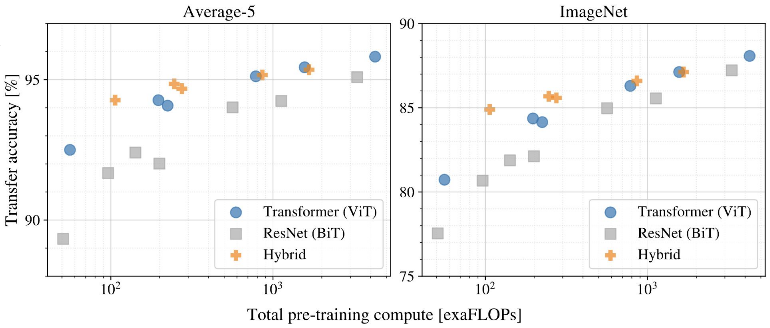
Vision Transformer (ViT)

Scaling ViTs

- Training at sufficient scale is better than the CNN inductive biases.
- ViTs are more compute efficient than ResNets.
- However ViTs require more data.

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k





Vision Transformer (ViT)

In Summary

- ViTs are competitive with CNNs on classification.
- ViTs are more compute efficient than CNNs.

Contrastive Language-Image Pre-training

Learning Transferable Visual Models From Natural Language Supervision

Radford et al., 2021

3

Contrastive Language-Image Pre-training (CLIP)

Contrastive image pre-training for downstream tasks.

Context

- GPT-3 was released in 2020.
- Computer vision was missing sufficient self-supervised objectives and datasets.
- Pre-training on ImageNet was still standard practice (Deng et al. 2009) at the time.
- Some work on contrastive training with language had been explored at smaller scale (Desai & Johnson, 2020), (Bulent Sariyildiz et al., 2020), and (Zhang et al., 2020).
- Built a dataset of 400M (image, text) pairs.

Learning Transferable Visual Models From Natural Language Supervision

Alec Radford [*1] **Jong Wook Kim** [*1] **Chris Hallacy** [1] **Aditya Ramesh** [1] **Gabriel Goh** [1] **Sandhini Agarwal** [1]
Girish Sastry [1] **Amanda Askell** [1] **Pamela Mishkin** [1] **Jack Clark** [1] **Gretchen Krueger** [1] **Ilya Sutskever** [1]

Contrastive Language-Image Pre-training (CLIP)

Dataset

- Previous research used large image datasets or small (image, text) datasets.
- One of the most important innovations is the 400M (image, text) pair dataset.
- Creating this dataset is non-trivial and details are vague.
- However we can compare with ALIGN (Jia, Chao, et al 2021), which used image alt-text.
- Text-image alignment is not great, but what matters is the ability to distinguish/classify the image given the text.


"motorcycle front wheel"


"*thumbnail for version as of 21 57 29 june 2010*"


"file frankfurt airport skyline 2017 05 jpg"


"file london barge race 2 jpg"

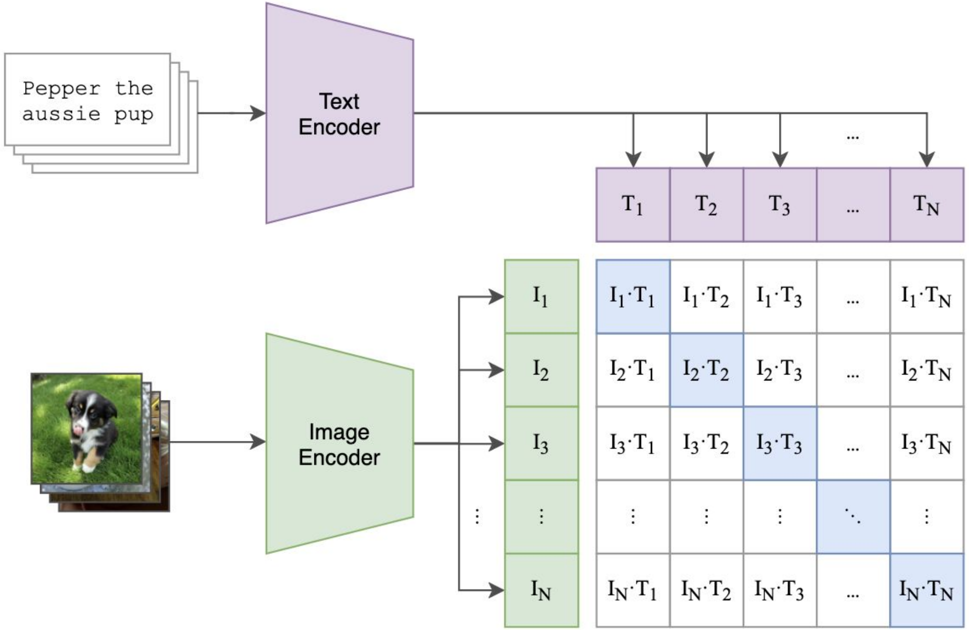

"wild boar head portrait forest creature boar"


"st oswalds way and shops"

Contrastive Language-Image Pre-training (CLIP)

Text Encoder

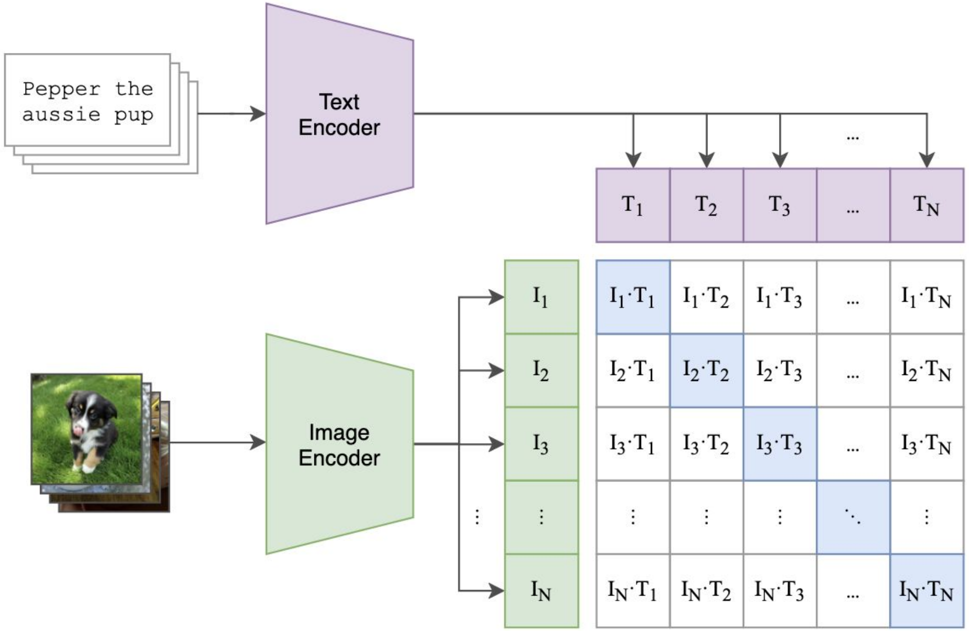
- Uses a Transformer with the architecture modifications described in (Radford et al. 2019).
- Base model had 63M parameters.
- 49,152 vocab size.
- 76 sequence length.
- Text starts with [SOS] and ends with [EOS] tokens.
- The activations from the [EOS] token are treated as the feature representation of the text.



Contrastive Language-Image Pre-training (CLIP)

Vision Encoder

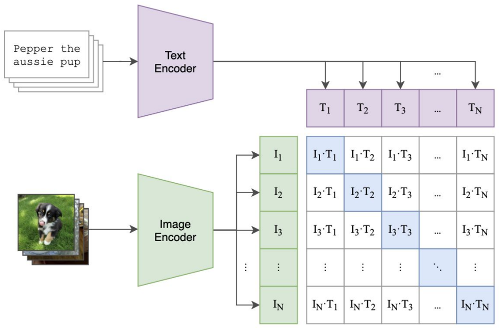
- Experimented with two architectures: one based on ResNet and another on ViT.
- For the ViT implementation, they closely follow the implementation from (Dosovitskiy et al., 2020), with minor modifications.
- Images are resized to 224x224 (except for ViT-L/14-336, trained at 336x336).



Contrastive Language-Image Pre-training (CLIP)

Contrastive loss

- Text and image are encoded separately.
- Calculate the dot-product of the text and image embeddings.
- Cross-entropy loss is used, where each batch element is treated as a class.
- Requires large batch sizes (used 32,768).

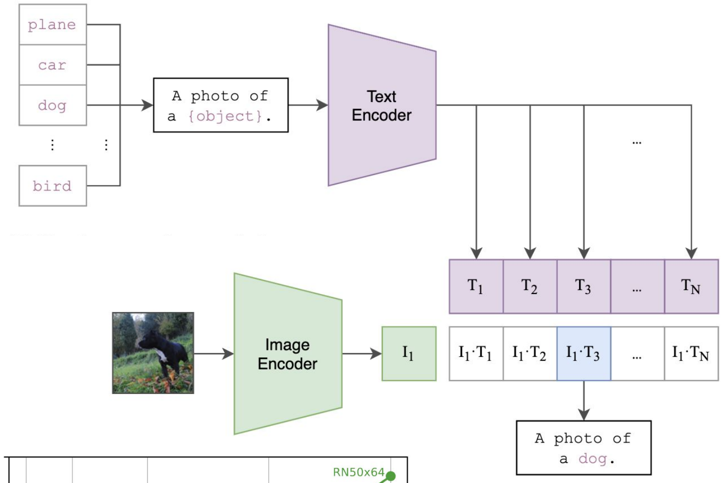


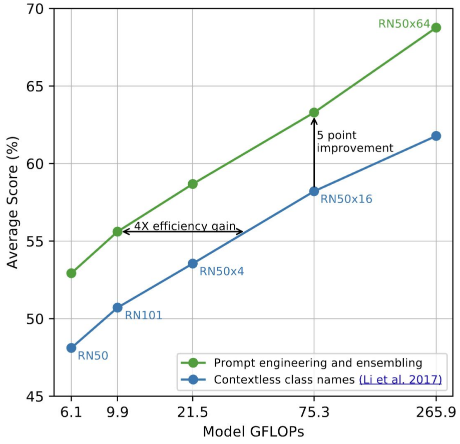

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned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

Contrastive Language-Image Pre-training (CLIP)

Adapting to Downstream Tasks (zero-shot transfer)

- Reformulate the downstream task (e.g. classification) using natural language.
  - *dog ⟶ A photo of a dog.*
- Compute the dot-product between the input image and all possible classes.
- Select the class with the highest cosine similarity.
- Prompt engineering and ensembling improve zero-shot performance.

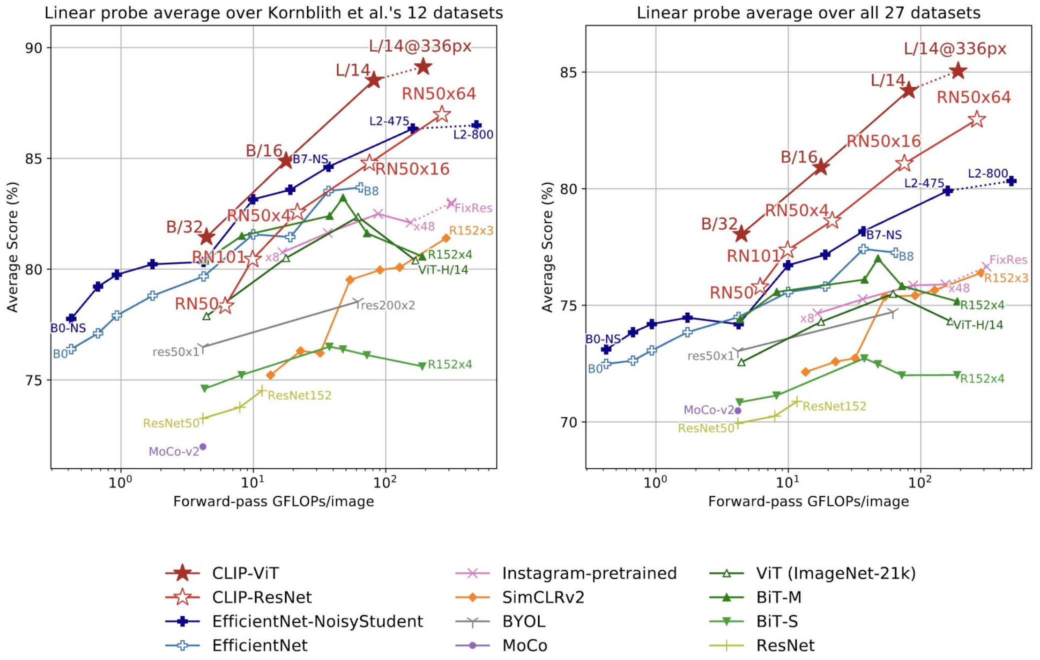




Contrastive Language-Image Pre-training (CLIP)

Scaling Models with CLIP

- Compared Vision Transformers (ViTs) with ResNets.
- Like the original ViT paper, they found ViTs to be more compute efficient.
- All models were trained for 32 epochs.



Contrastive Language-Image Pre-training (CLIP)

In Summary

- CLIP is a training objective that requires little data annotation.
- It can be scaled to large datasets, if you know how to get them!
- Further evidence that ViTs are a good general purpose architecture for encoding images.

Native Resolution Vision Transformer

4

Patch n’ Pack: NaViT, a Vision Transformer for any Aspect Ratio and Resolution

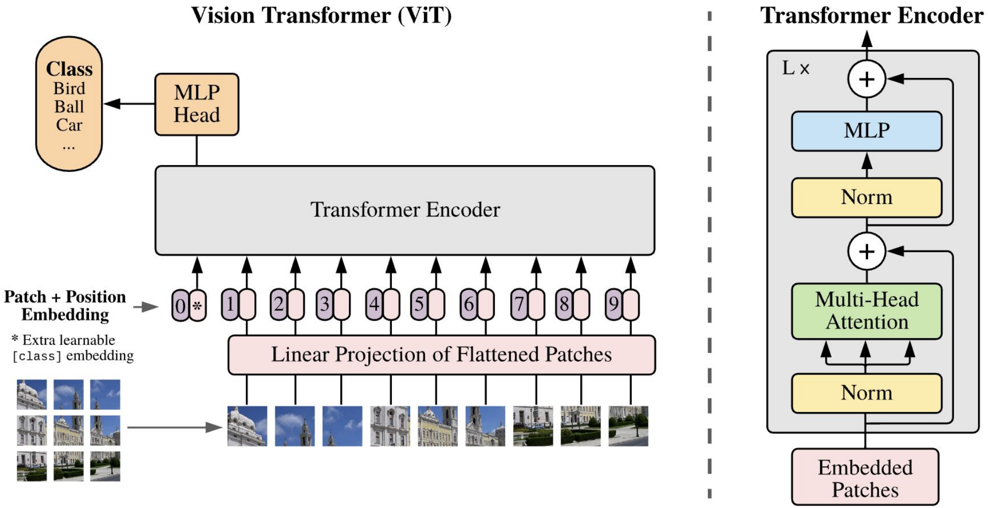
Dehghani et al., July 2023

Native Resolution Vision Transformer (NaViT)

July 2023

ViT Limitations

- The ViT resizes images to a fixed resolution (224x224).
- Problematic because you have to choose between training on lots of images at low resolution or fewer images at high resolution.
- This is mostly due to technical limitations - most hardware requires fixed tensor sizes.
- You can pad images to the largest resolution but will waste lots of FLOPs.

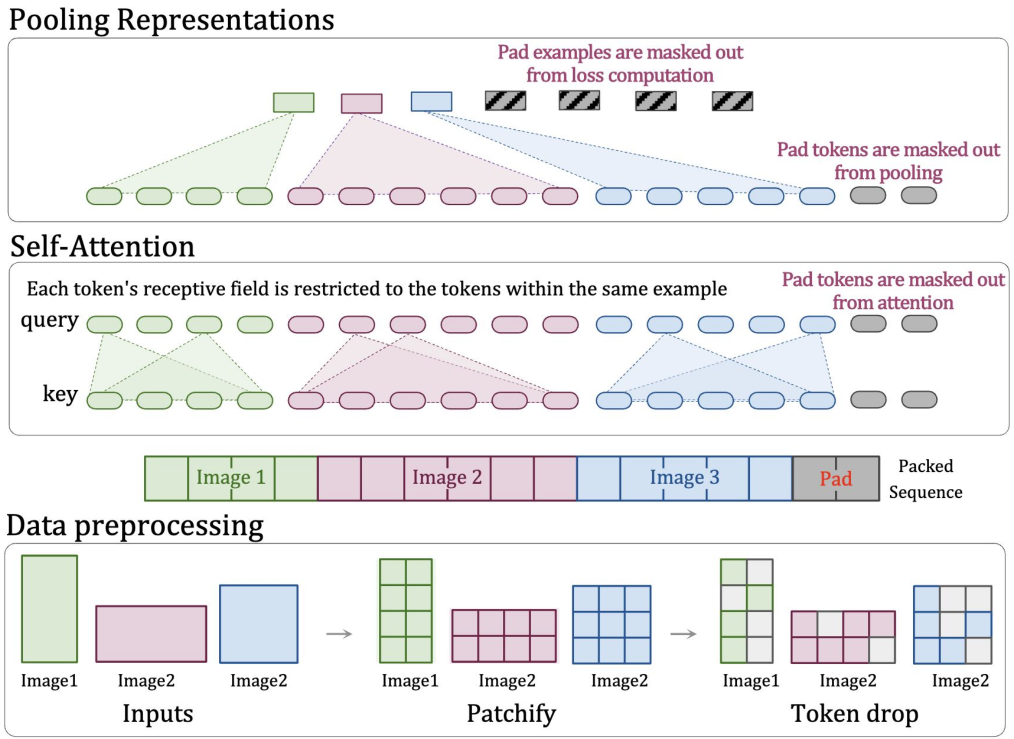


Native Resolution Vision Transformer (NaViT)

July 2023

NaViT Architecture

- NaViT uses native image resolutions.
- Images are broken into patches, as in ViT, and flattened.
- These flattened images can be packed together into a single packed sequence of shape `[L, 3]`.
- Self-attention layers are modified so images do not attend to each other.
- The final pooling layer is also modify so it does not pool across images.

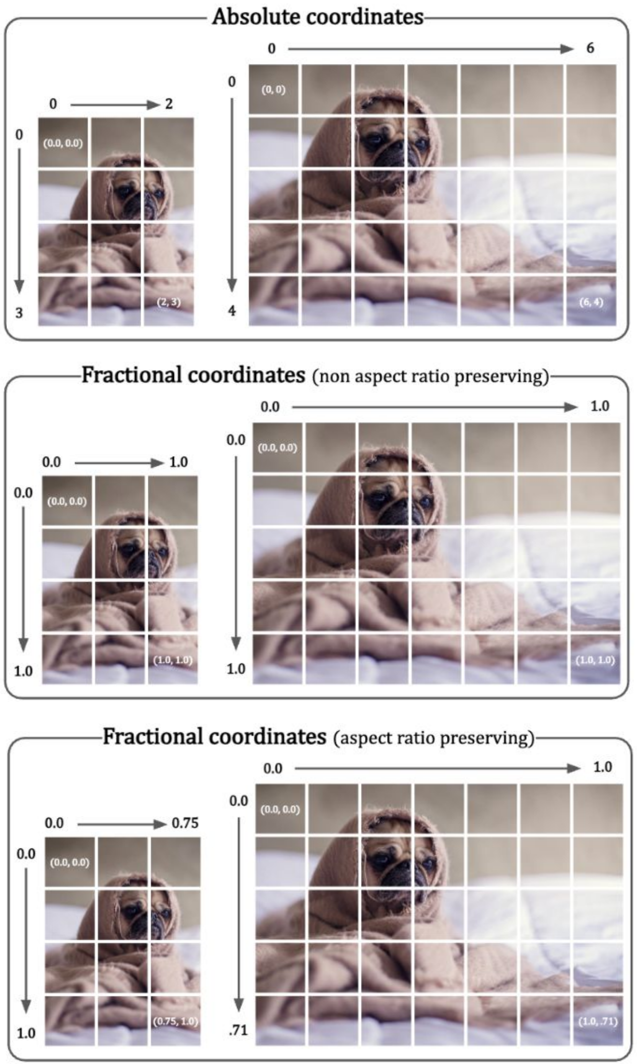


Native Resolution Vision Transformer (NaViT)

July 2023

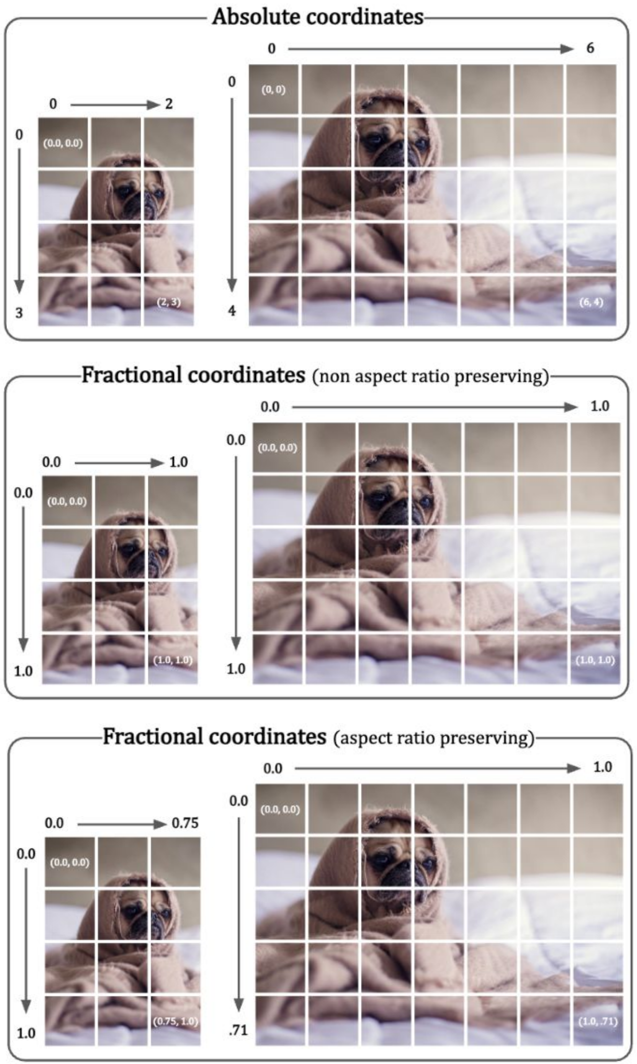
Positional Embeddings

- The original ViT paper used 1D positional embeddings, however it was only trained on a fixed resolution.
- NaViT explores several different techniques for positional embeddings.



Native Resolution Vision Transformer (NaViT)

July 2023



Positional Embeddings

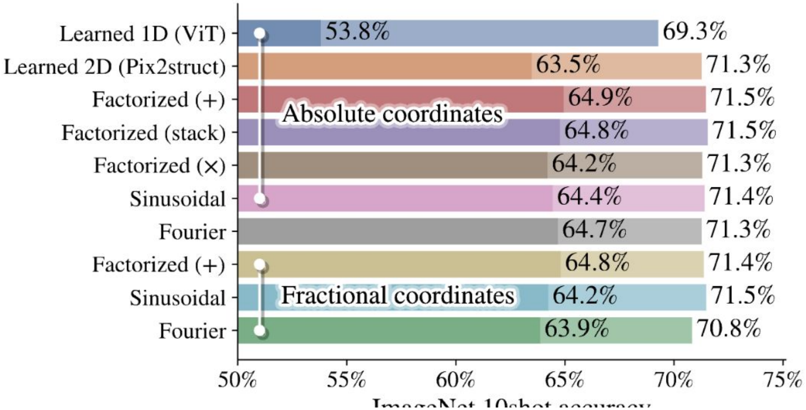
- The original ViT paper used 1D positional embeddings, however it was only trained on a fixed resolution.
- NaViT explores several different techniques for positional embeddings.
- Absolute coordinates assign integer coordinates (x, y) based on their original location within the image.
  - Limits model generalization.
- Fractional coordinates are normalized to the actual size of the input image and are obtained by dividing the absolute coordinates x and y above by the number number of columns and rows respectively.
  - Drops information about aspect ratio

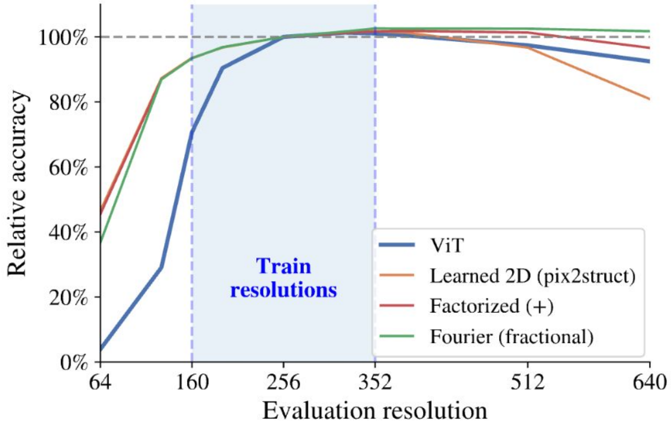
Native Resolution Vision Transformer (NaViT)

July 2023

Positional Embeddings

- Empirical studies found fractional coordinates to be better.
- Factorized (aka 2D) embeddings are best among fractional coordinate options.
- Fourier (fractional) embeddings generalize best to larger images but have worst absolute performance.



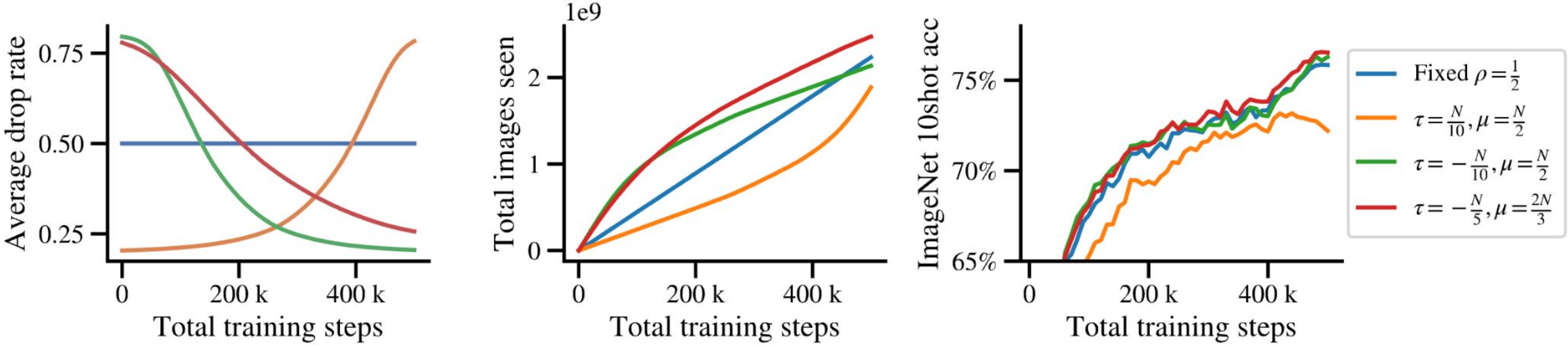


Native Resolution Vision Transformer (NaViT)

July 2023

Token Dropping

- Token dropping improves performance because the model sees more images.

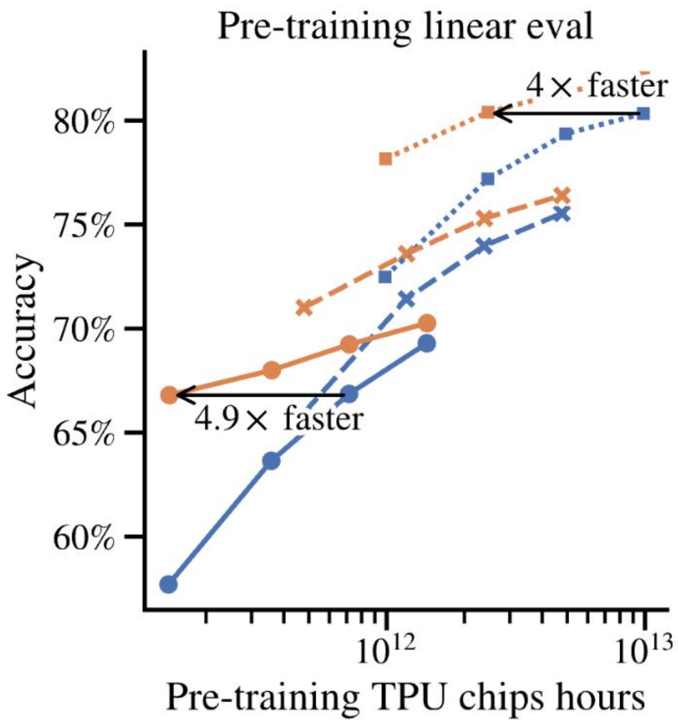


Native Resolution Vision Transformer (NaViT)

July 2023

Performance

- NaViT achieves equivalent performance as ViT with 4x fewer FLOPs.
- When trained for ~1e13 compute hours NaViT processes 4.75x as many training images

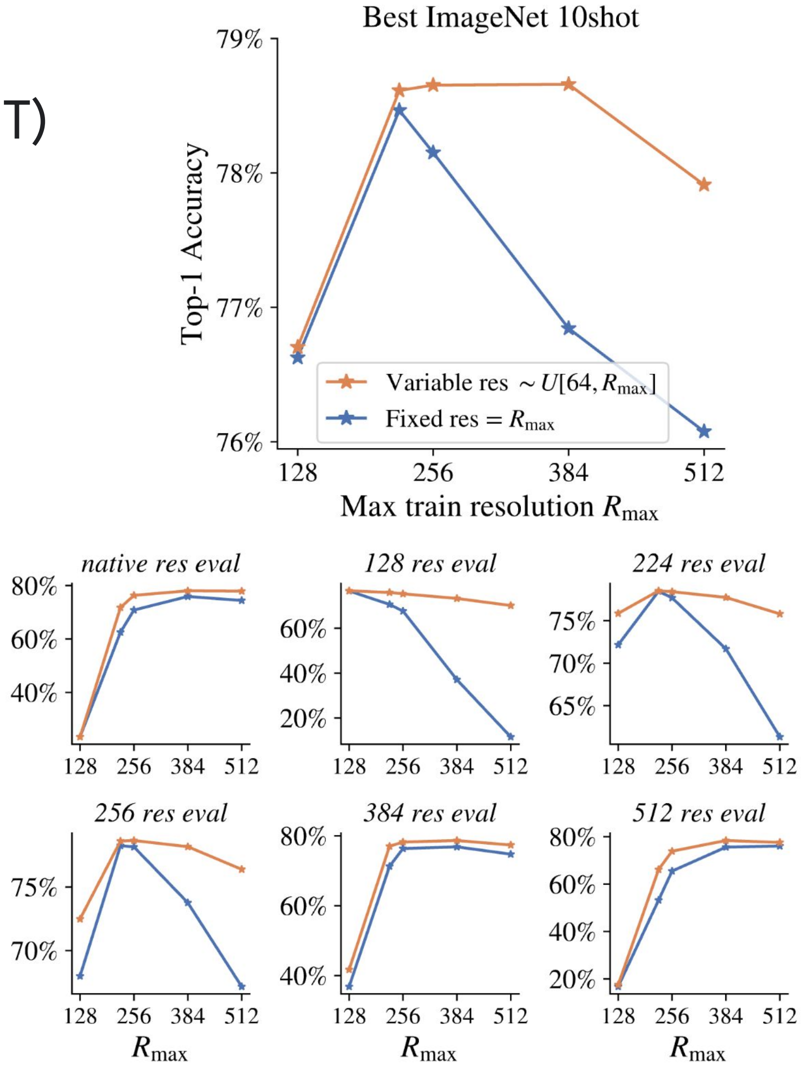


Native Resolution Vision Transformer (NaViT)

July 2023

Performance

- Increasing the maximum resolution at during training lowers throughput (images seen) but helps generalization to larger images.
- A solution is to randomly sample the maximum resolution at train time.



Native Resolution Vision Transformer (NaViT)

July 2023

In Summary

- Training on native image resolution is more compute efficient than resizing everything to one resolution.

A Recipe For Training Vision Encoders

Architecture

The Vision Transformer is a scalable architecture that is more compute efficient than CNNs.

Training Objective

CLIP is a general purpose pre-training objective that requires little labeled data.

Data and Compute

Use native image resolutions if possible.


Multimodal
Foundation
Models
Google DeepMind

Combining Modalities

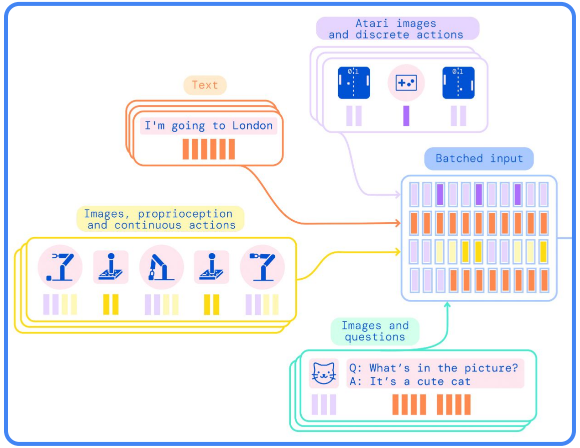
01 A Generalist Agent

02 Flamingo

03 Visual Instruction Tuning

Three Approaches to Combining Modalities

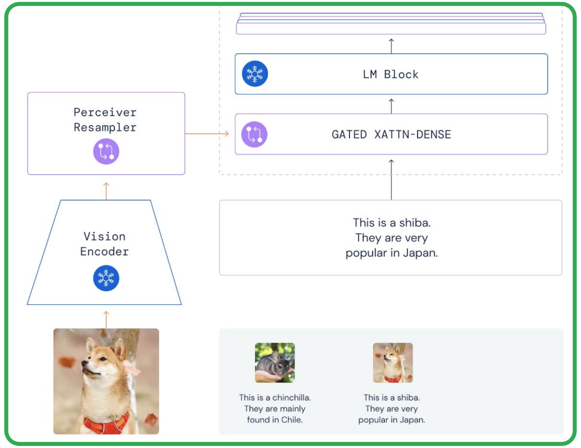
Merge Modalities at Input



Gato

1. Project all modalities to the same embedding space.
2. Combine data into a sequence of multimodal embeddings.
3. Feed the embeddings into a decoder-only transformer.

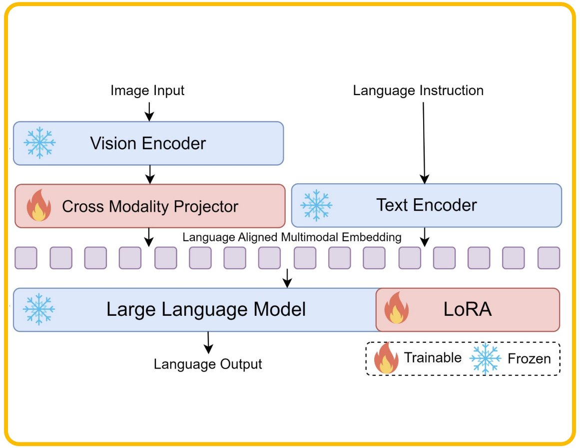
Merge Modalities with Cross-Attention



Flamingo

1. Put text through a frozen language model (transformer).
2. Encode images with a frozen vision encoder.
3. Feed image and text embeddings into new cross-attention layers.

Visual Instruction Tuning

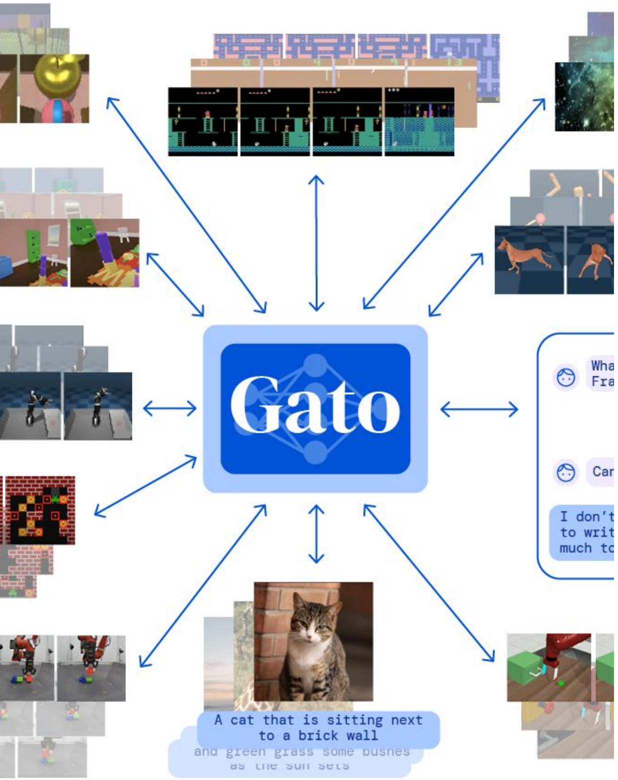


Visual Instruction Tuning

1. Frozen Vision Encoder and LLM.
2. Projects vision encoder outputs using a small trainable MLP.
3. Visual instruction tuning data.

Google DeepMind

A Generalist Agent



Scott Reed , Konrad Żołna , Emilio Parisotto , Sergio Gómez Colmenarejo , Alexander Novikov, Gabriel Barth-Maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar and Nando de Freitas

Gato Overview

Gato is a multi-modal, multi-task, multi-embodiment generalist policy.

The same network with the same weights can play Atari, caption images, chat, stack blocks with a real robot arm and much more, deciding based on its context whether to output text, joint torques, button presses, or other tokens.

Input Modalities

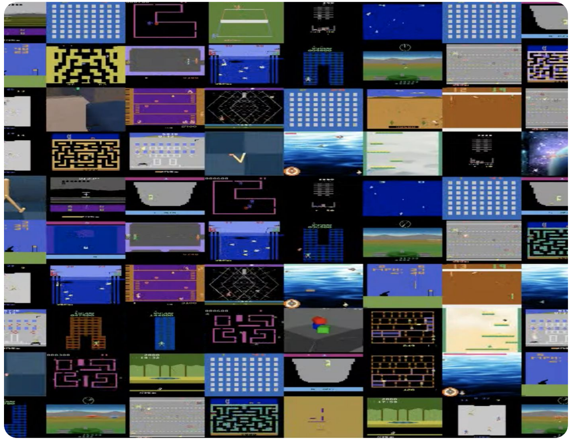
- Images
- Integers
- Floats
- Text

Output Modalities

- Text
- Integers
- Floats

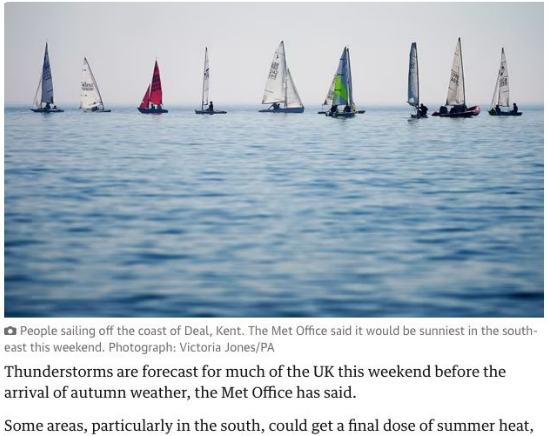
Next Token Prediction

Training Data



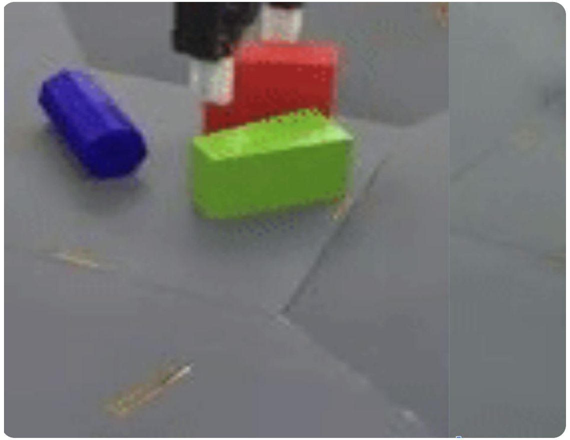
Simulated Control Tasks

Data is generated by training SoTA RL agents on different environments. We record a subset of the experience the agent generates during training.



Vision & Language

We use standard datasets that cover a range of tasks, from image classification to text generation.



Real Robot Block Stacking

We use the RGB-Stacking robotics benchmark for pre-training and fine-tuning.

Simulated Control Tasks

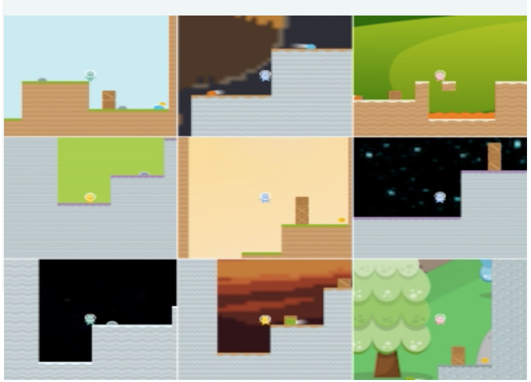
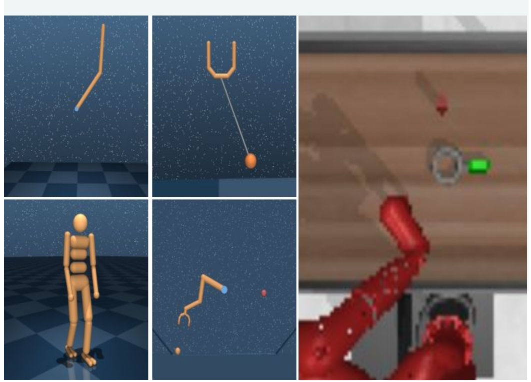


Image Observations

	Episodes	Tokens
DM Lab	16.4M	194B
Atari	90.8k	1.8B
Sokoban	27.2k	298M
Procgen Benchmark	1.6M	4.46B
DM Control Suite Pixels	485k	35.5B



Continuous Control

	Episodes	Tokens
DM Control Suite	395k	22.5B
Meta-World	94.6k	3.39B
Modular RL	843k	69.6B
DM Manipulation Playground	286k	6.58B
DM Control Suite Random	36.7M	1.1T

Text Instructions

	Episodes	Tokens
BabyAI	4.61M	22.8B
Playroom	829k	118B

We train on 62M episodes with 1.49T tokens

Vision & Language

Vision | Language

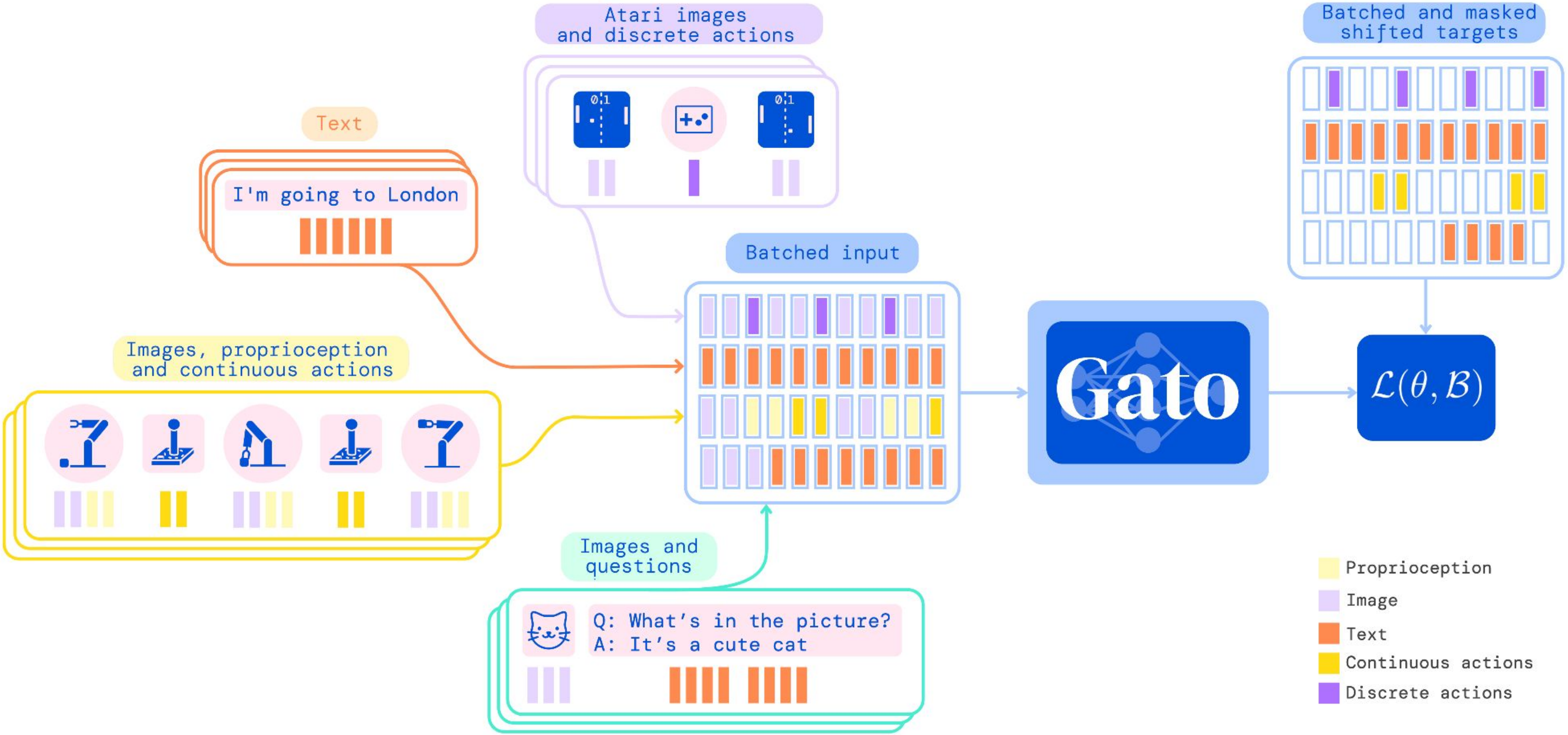
Image classification	Images	Text generation	Tokens
ImageNet	14M	MassiveWeb	506B

Vision & Language (VLM)

Captioning	*Examples*	Visual Querying	*Examples*
ALIGN	1.8B*	OKVQA	9K
MS-COCO Captions	*120K**	VQAV2	443K
Conceptual captions	3.3M*		
M3W	~10B**		
Stock images	320M*		

* image/text pairs
** webpages scrapped

Training Data and Masking Loss



Data Serialization: Text

Text

`The following is a conversation between a highly knowledgeable …`

SentencePiece Tokenization, Embed (Table)

$$\begin{bmatrix}2.3\\8.0\\\vdots\end{bmatrix}, \begin{bmatrix}2.8\\3.8\\\vdots\end{bmatrix}, \begin{bmatrix}3.4\\0.2\\\vdots\end{bmatrix}, \begin{bmatrix}9.0\\1.2\\\vdots\end{bmatrix}, \begin{bmatrix}3.9\\0.1\\\vdots\end{bmatrix}, \begin{bmatrix}2.9\\1.2\\\vdots\end{bmatrix}, \ldots$$

Data Serialization: Continuous Values

Proprioception

[cart_x, velocity_x, pole_x, pole_y]

[0.3, 1.5, 0.25, -0.4]

Mu-law Encode, Discretize & Embed (Table)

$$\overbrace{\begin{bmatrix}5.9\\8.4\\\vdots\end{bmatrix}}^{\textbf{Token Embedding}}, \begin{bmatrix}8.7\\2.6\\\vdots\end{bmatrix}, \begin{bmatrix}7.6\\5.3\\\vdots\end{bmatrix}, \begin{bmatrix}4.6\\3.4\\\vdots\end{bmatrix}$$

Continuous Action

[cart_move_x]

[-0.7]

Discretize & Embed (Table)

$$\begin{bmatrix}9.0\\3.3\\\vdots\end{bmatrix}$$

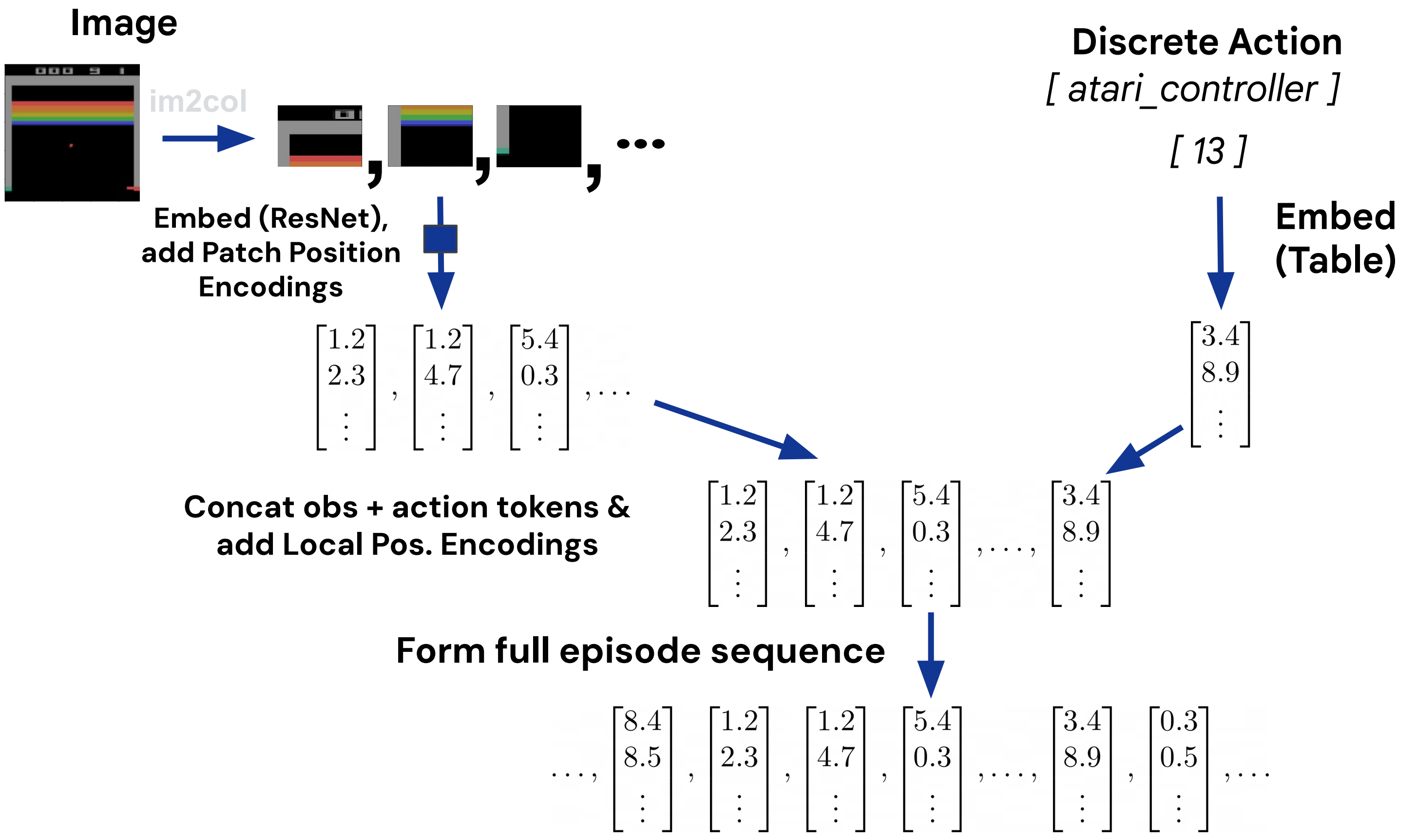
Concat obs + action tokens & add Local Pos. Encodings

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

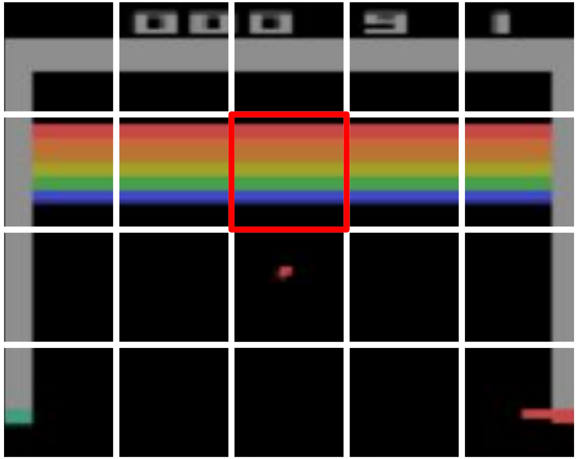
Form full episode sequence

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

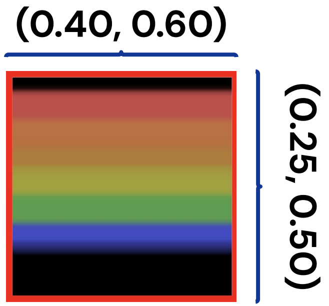
Data Serialization: Images



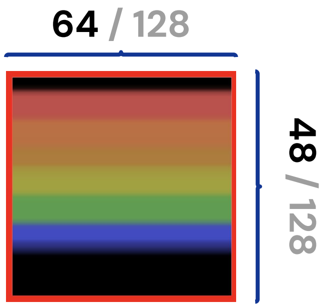
Data Serialization: Image Patch Positional Encodings



For each patch get its relative position



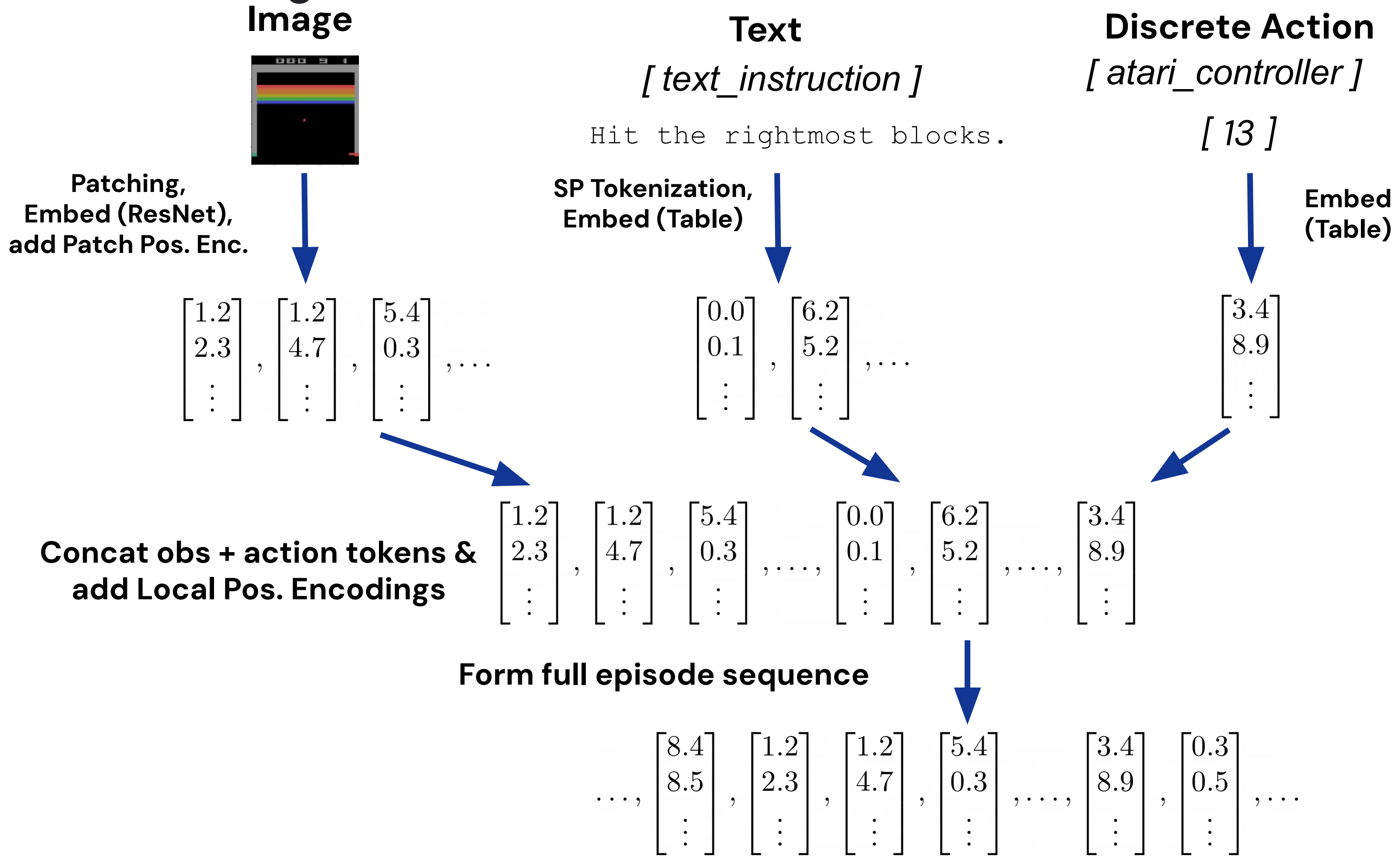
Discretize row and column values



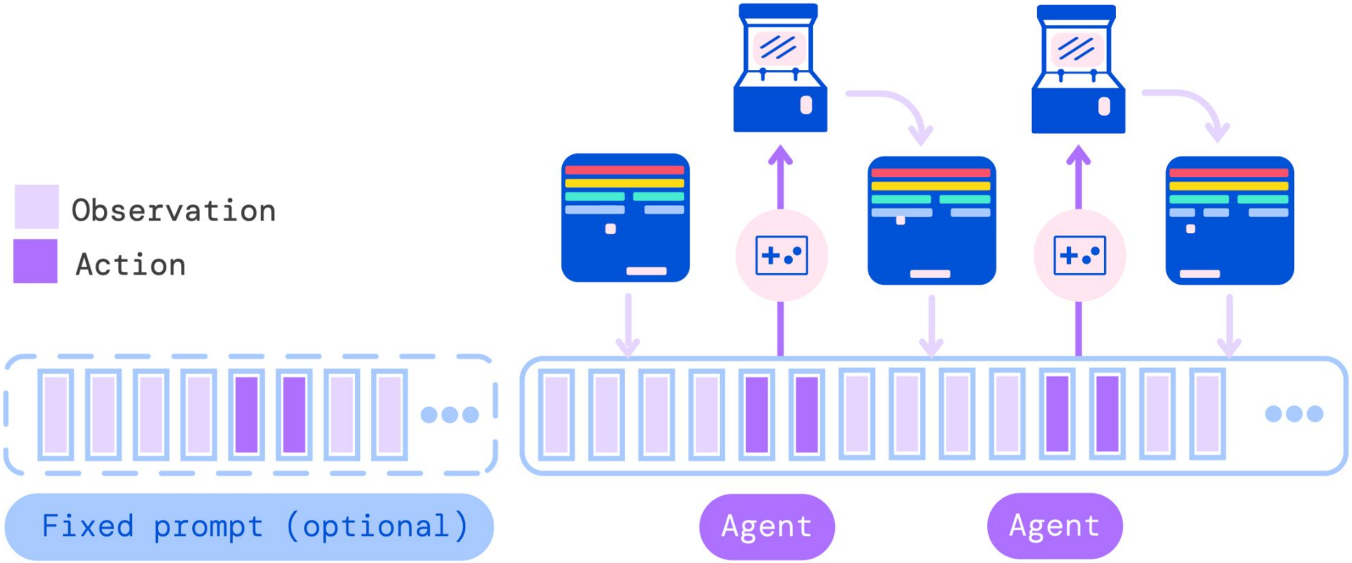
*We take the middle of the interval at test time, and a random point from the interval during training

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}

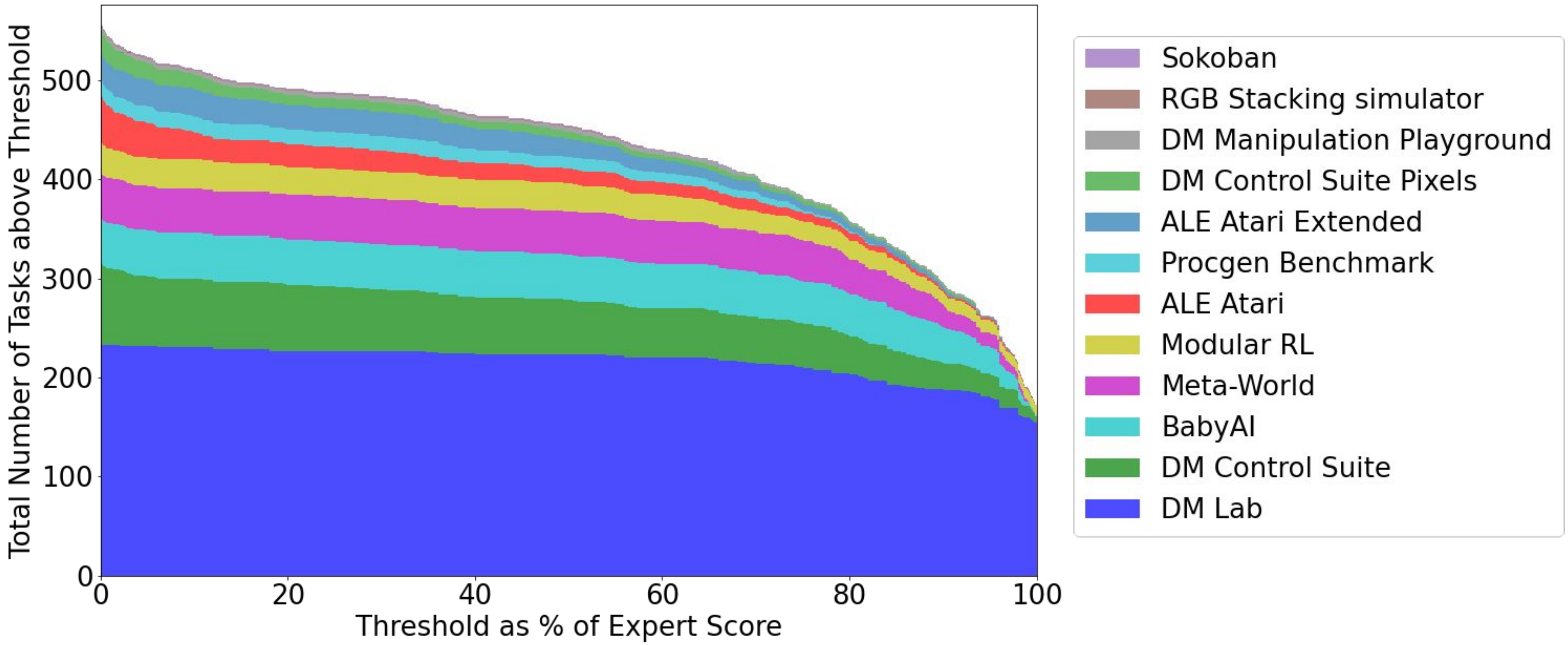
Combining Modalities



Interactive Evaluation

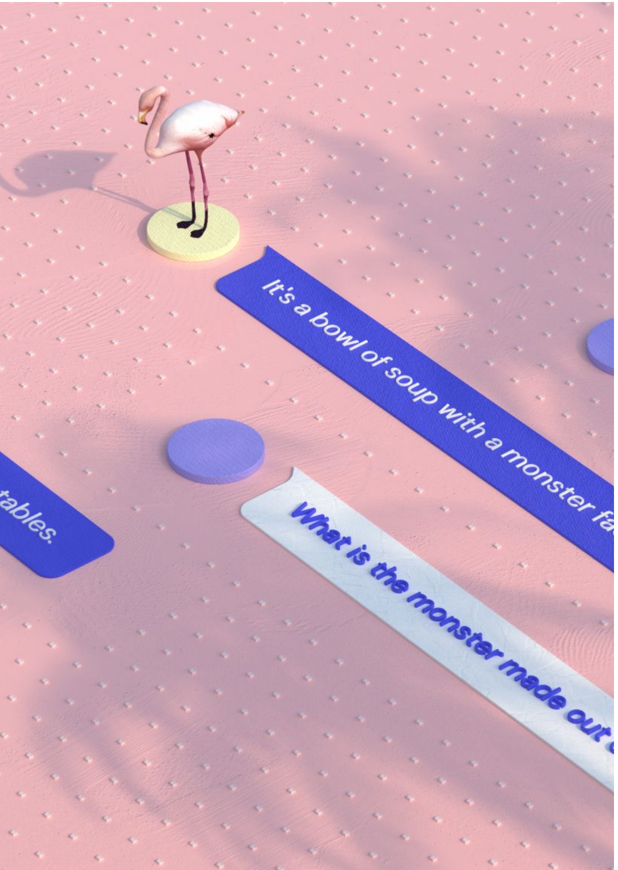


Results



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

Google DeepMind

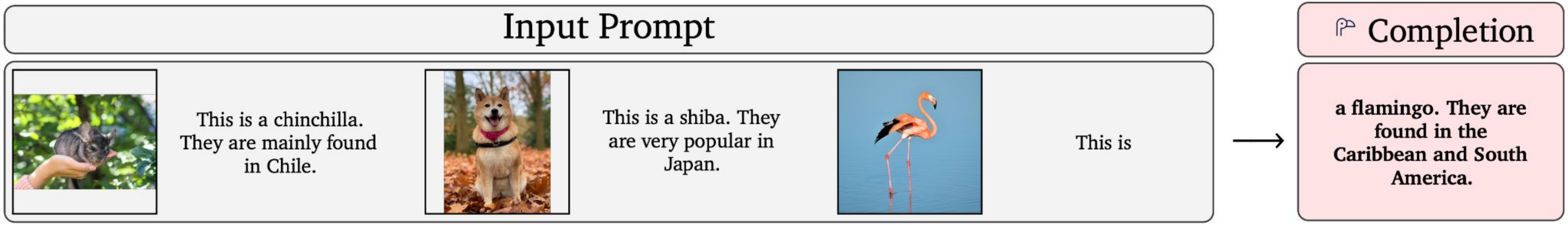


Flamingo

Jean-Baptiste Alayrac*, Jeff Donahue*, Pauline Luc*, Antoine Miech*, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, Karen Simonyan

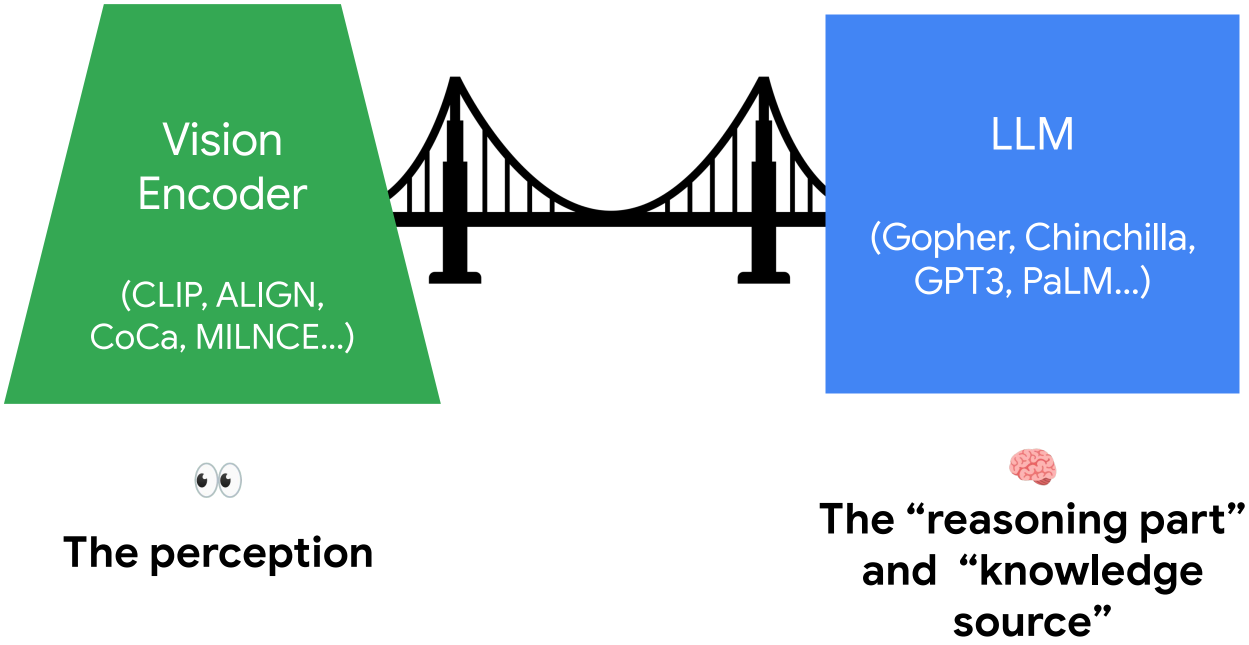
What is Flamingo?

Flamingo is a Visual Language Model that ingests visual data (images and/or videos) along with language input, and produces text output.



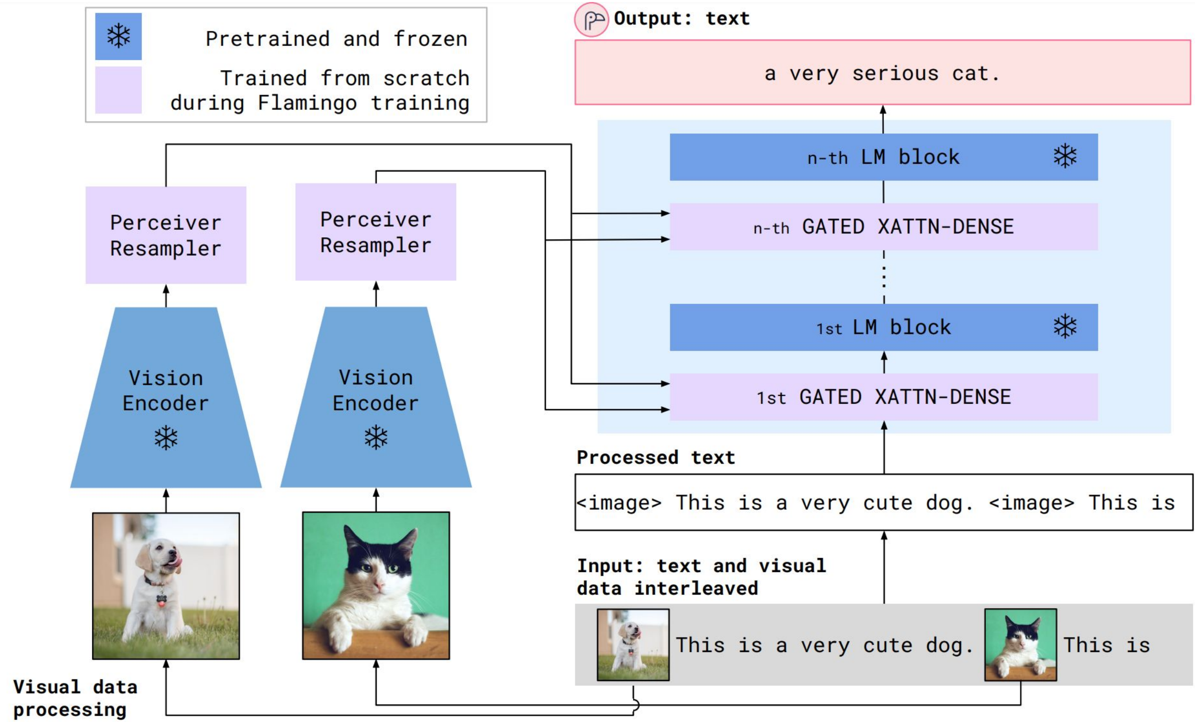
Model Overview

Pretrained parts of the model are frozen:
the Vision Encoder and the LLM.



Model Overview

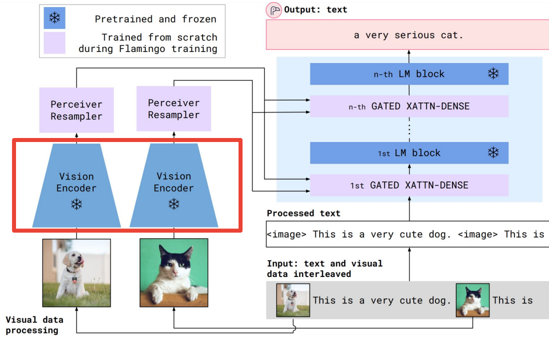
Pretrained parts of the model are frozen:
the Vision Encoder and the LLM.



Visual Processing

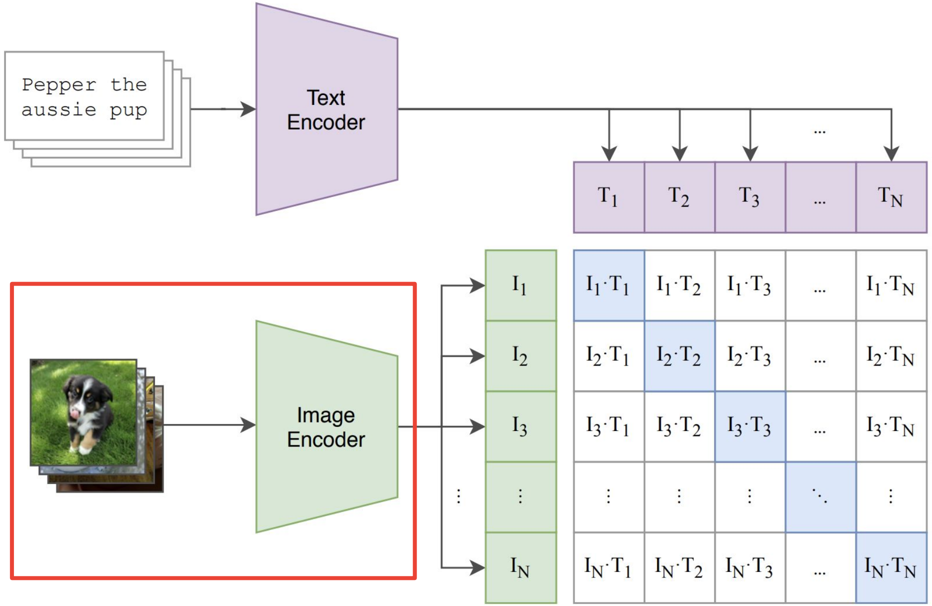
Vision Encoder

Pretrained with image-text contrastive training (CLIP-like) and kept frozen during Flamingo training.



Overall architecture

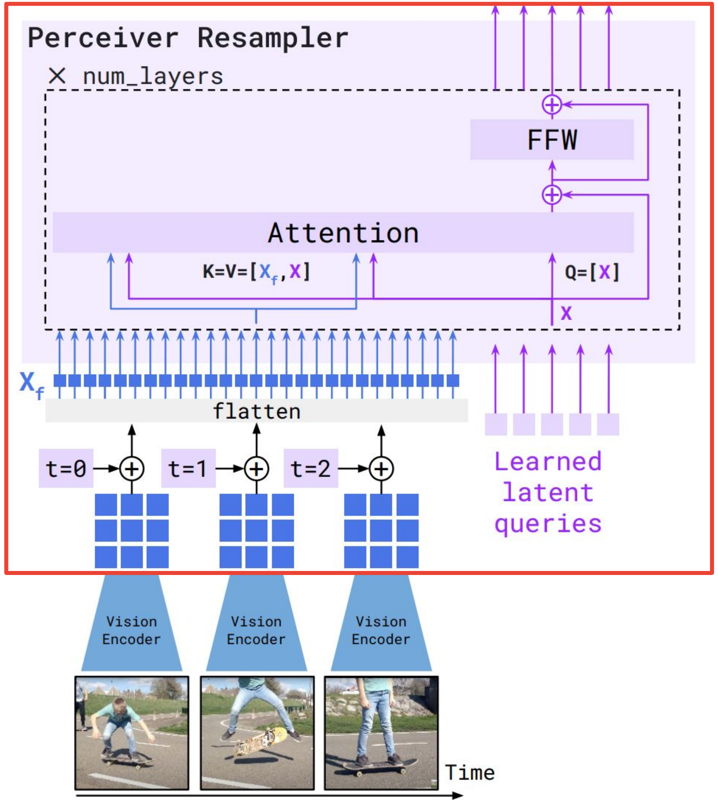
We only keep the vision encoder and discard the text encoder.

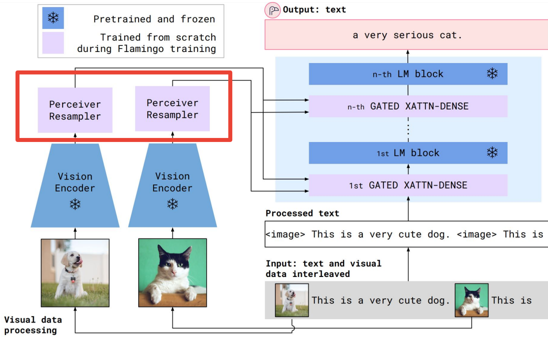


Visual Processing

Perceiver Resampler

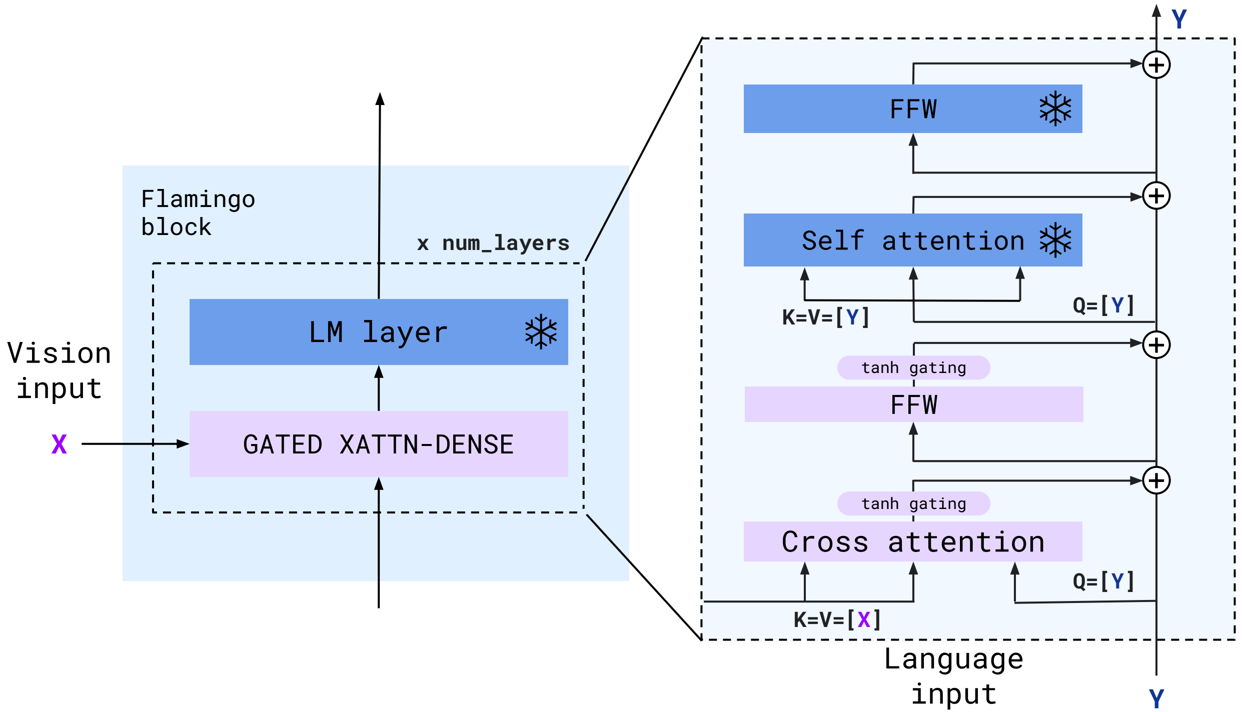
Takes as input a variable number of features (image or videos) and outputs a fixed number of “visual tokens”.

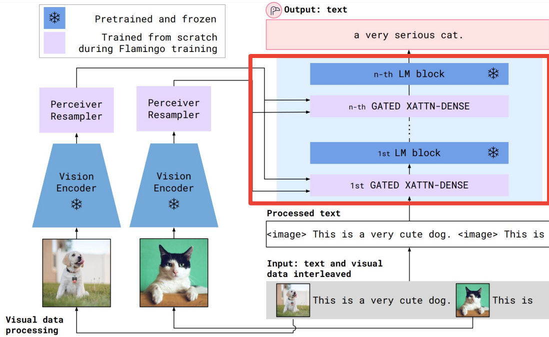




Overall architecture

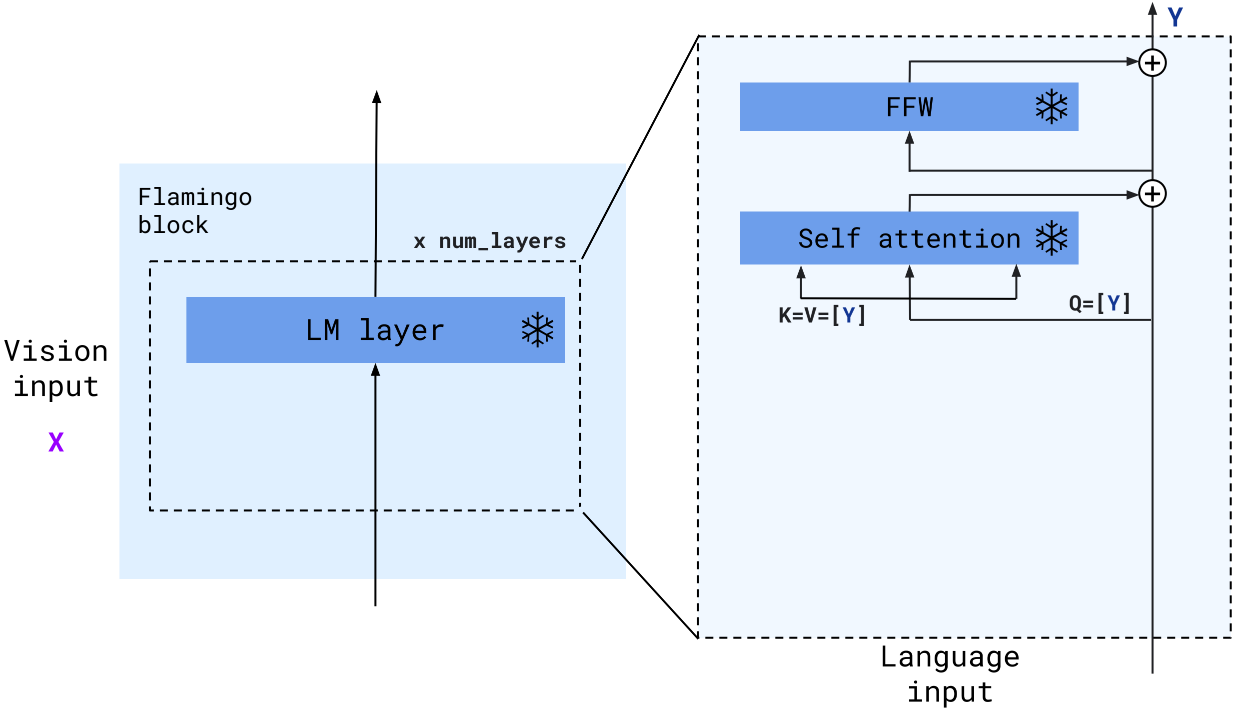
Leveraging an existing language model

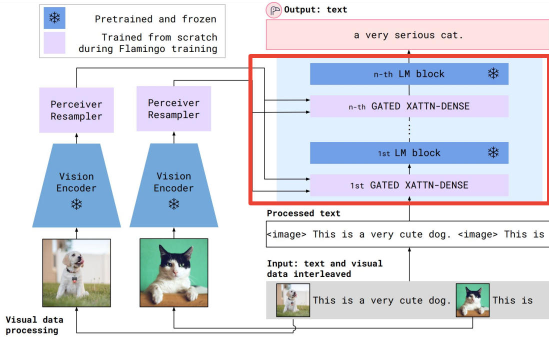




Overall architecture

Leveraging an existing language model



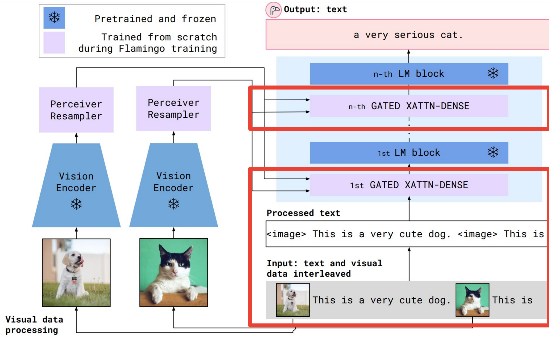


Overall architecture

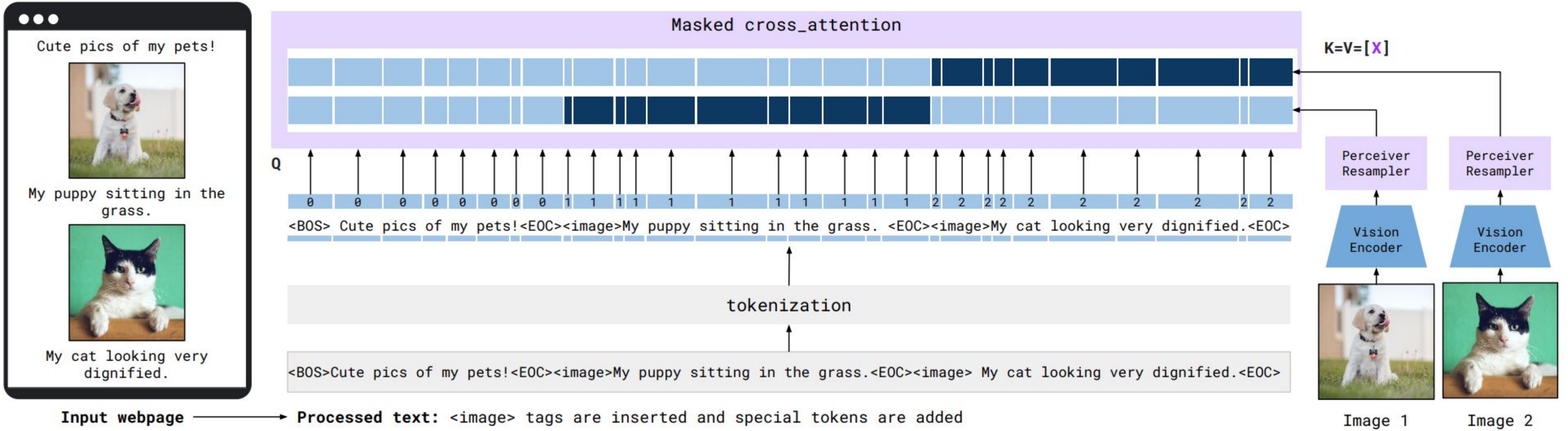
At initialisation, tanh gates are all 0.

Deal with interleaved visual and text sequence

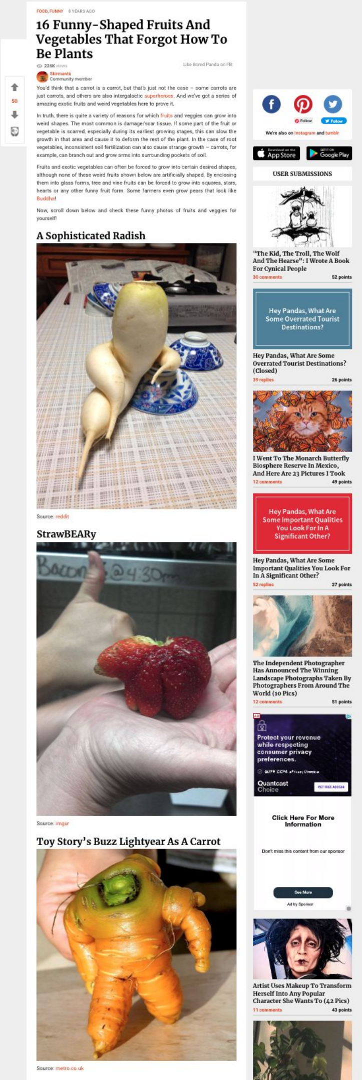
Each text token cross-attend to the image that precedes it in the interleaved sequence.



Overall architecture



M3W: Massive MultiModal Web Dataset



Processing

16 Funny-Shaped Fruits And Vegetables That Forgot How To Be Plants

You'd think that a carrot is a carrot, but that's just not the case – some carrots are just carrots, and others are also intergalactic superheroes. [...]. Some farmers even grow pears that look like Buddha!

Now, scroll down below and check these funny photos of fruits and veggies for yourself!

A Sophisticated Radish

<IMAGE PLACEHOLDER 1>

StrawBEARy

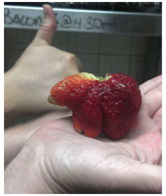
<IMAGE PLACEHOLDER 2>

Toy Story's Buzz Lightyear As A Carrot

<IMAGE PLACEHOLDER 3>

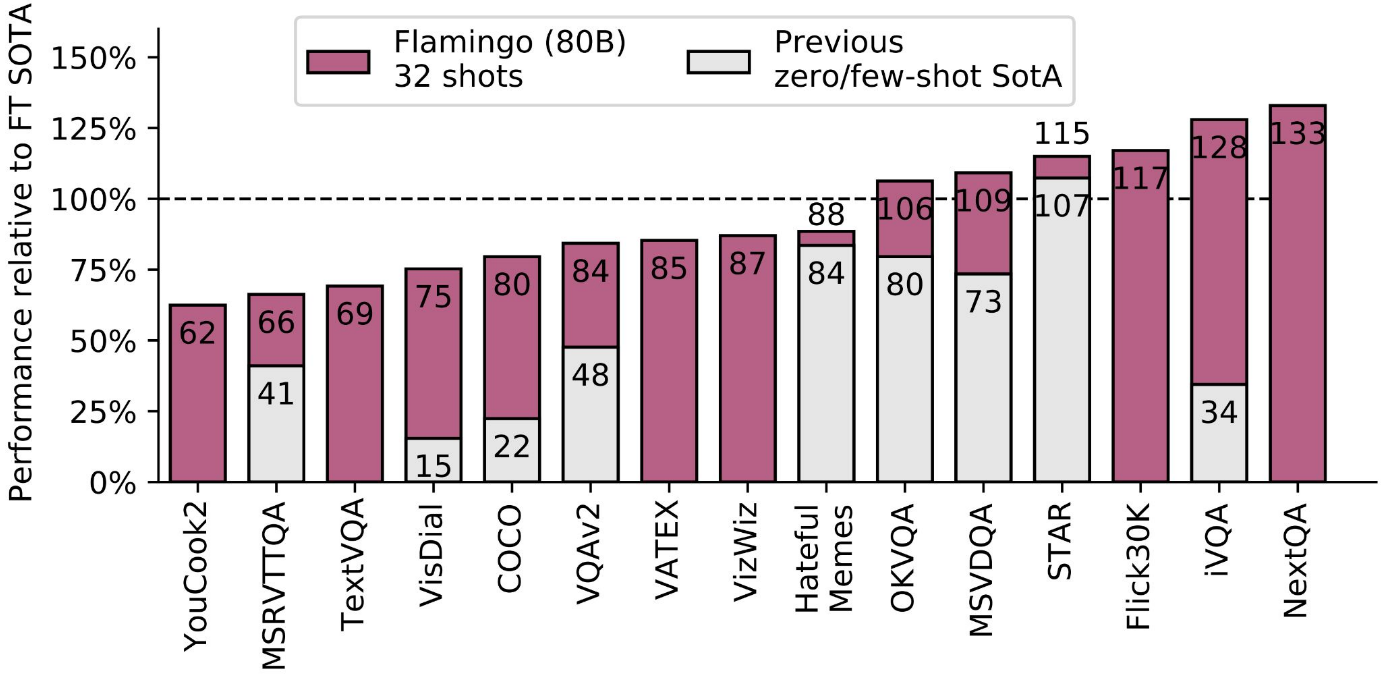
<IMAGE PLACEHOLDER 1>

<IMAGE PLACEHOLDER 2>



<IMAGE PLACEHOLDER 3>

Few Shot Results



Google DeepMind

Visual Instruction Tuning

He, Muyang, et al. "Efficient Multimodal Learning from Data-centric Perspective." arXiv preprint arXiv:2402.11530 (2024).

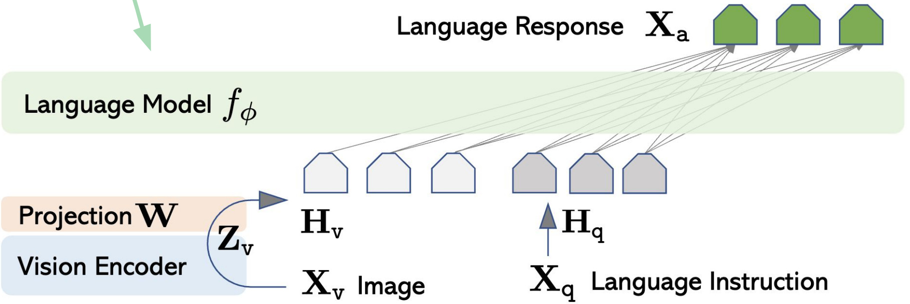
Liu, Haotian, et al. "Improved baselines with visual instruction tuning." arXiv preprint arXiv:2310.03744 (2023).

Liu, Haotian, et al. "Visual instruction tuning." Advances in neural information processing systems 36 (2024).

Visual Instruction Tuning (LLaVa)

Training: Stage 1

- ❄️Uses a frozen LLM (Vicuna-13B).
- ❄️Uses a frozen vision encoder (ViT-L/14).
- 🔥Trains a linear projection layer on the vision encoder outputs.
- Projected vision tokens are concatenated to language tokens.
- Trained on a subset of CC3M (Conceptual Captions) that was created for instruction tuning.



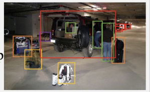
Trained on 8x A100s for 4hrs.

Visual Instruction Tuning (LLaVa)

Data

- Conceptual Captions 3M (CC3M) is a dataset of 3M images paired with natural language captions (from alt-text).
- Filtered CC3M down to 595K image-text pairs.
- Turned these image-text pairs into visual question answering (VQA) data by using GPT-4 (text-only) to generate answers to a fixed list of questions.

Context type 1: Captions
A group of people standing outside of a black vehicle with various luggage.
Luggage surrounds a vehicle in an underground parking area
People try to fit all of their luggage in an SUV.
The sport utility vehicle is parked in the public garage, being packed for a trip
Some people with luggage near a van that is transporting it.



Context type 2: Boxes
person: [0.681, 0.242, 0.774, 0.694], backpack: [0.384, 0.696, 0.485, 0.914], suitcase: ...<omitted>

Response type 1: conversation
Question: What type of vehicle is featured in the image?
Answer: The image features a black sport utility vehicle (SUV) ...<omitted>
Response type 2: detailed description
The image is an underground parking area with a black sport utility vehicle (SUV) parked. There are three people in the scene, with one person standing closer to the left side of the vehicle, another person in the middle, and the third person on the right side. They are all working together to pack their luggage into the SUV for a trip. ...<omitted>
Response type 3: complex reasoning
Question: What challenges do these people face?
Answer: In the image, a group of people is standing outside a black SUV in a parking area, surrounded by various pieces of luggage, including suitcases and backpacks. They are facing the challenge of fitting all their luggage into the black SUV. There are multiple suitcases and backpacks to be packed, which suggests that the group has a significant amount of belongings ...<omitted>

- "Describe the image concisely."
- "Provide a brief description of the given image."
- "Offer a succinct explanation of the picture presented."
- "Summarize the visual content of the image."
- "Give a short and clear explanation of the subsequent image."
- "Share a concise interpretation of the image provided."
- "Present a compact description of the photo's key features."
- "Relay a brief, clear account of the picture shown."
- "Render a clear and concise summary of the photo."
- "Write a terse but informative summary of the picture."
- "Create a compact narrative representing the image presented."

Table 11: The list of instructions for brief image description.

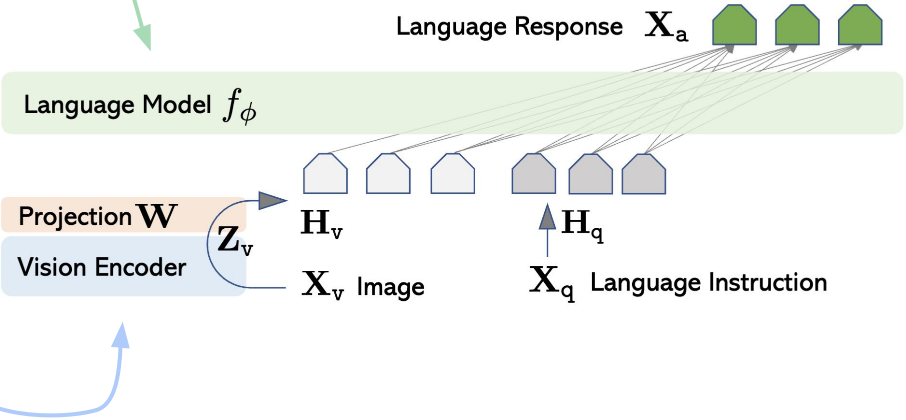
- "Describe the following image in detail"
- "Provide a detailed description of the given image"
- "Give an elaborate explanation of the image you see"
- "Share a comprehensive rundown of the presented image"
- "Offer a thorough analysis of the image"
- "Explain the various aspects of the image before you"
- "Clarify the contents of the displayed image with great detail"
- "Characterize the image using a well-detailed description"
- "Break down the elements of the image in a detailed manner"
- "Walk through the important details of the image"
- "Portray the image with a rich, descriptive narrative"
- "Narrate the contents of the image with precision"
- "Analyze the image in a comprehensive and detailed manner"
- "Illustrate the image through a descriptive explanation"
- "Examine the image closely and share its details"
- "Write an exhaustive depiction of the given image"

Table 12: The list of instructions for detailed image description.

Visual Instruction Tuning (LLaVa)

Training: Stage 2

- 🔥Trains LLM (Vicuna).
- ❄️Uses a frozen vision encoder (ViT-L/14).
- 🔥Trains a linear projection layer on the vision encoder outputs.
- Projected vision tokens are concatenated to language tokens.
- Adapt the original VQA data so it forms a sequence of instructions.
    - Find all (question, answer) pairs for an image.
    - Create a sequence of (question, image, answer) tuples and concatenate them.
    - LLaVa only predicts answers.



Trained on 8x A100s for 4hrs for ScienceQA or 10hrs for their Instruct-158k dataset.

Visual Instruction Tuning (LLaVa)

Results

- ScienceQA dataset.
- Sets of visual question answering problems involving different scientific fields.

Question

Is calcarenite a mineral or a rock?

Context

Calcarenite has the following properties: yellow-brown not made by organisms not a pure substance found in nature solid no fixed crystal structure

Choices

rock | mineral

Answer

rock

Visual Instruction Tuning (LLaVa)

Results

- ScienceQA dataset.
- Sets of visual question answering problems involving different scientific fields.
- Achieve close to SoTA on all subjects.
- GPT-4 achieves 70.75% accuracy on questions with image context!
    - Is this a useful multimodal benchmark?

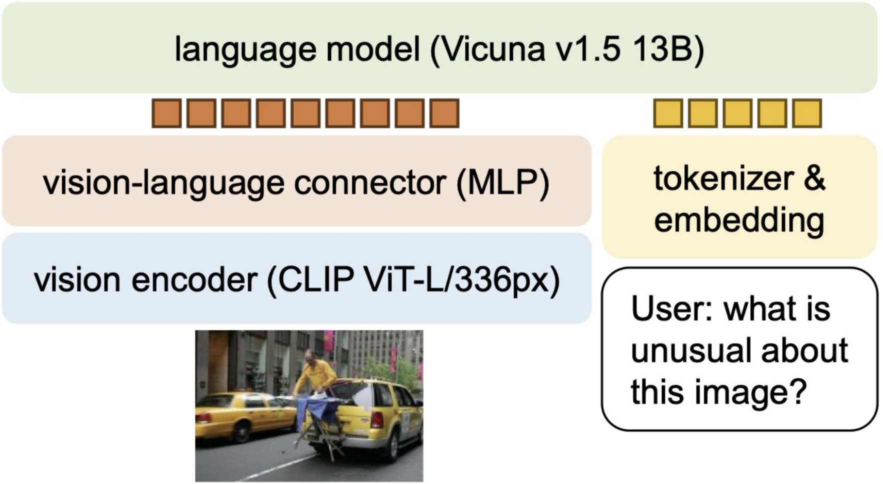
Method	Subject			Context Modality			Grade		Average
	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	
Representative & SoTA methods with numbers reported in the literature									
Human [34]	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
GPT-3.5 [34]	74.64	69.74	76.00	74.44	67.28	77.42	76.80	68.89	73.97
GPT-3.5 w/ CoT [34]	75.44	70.87	78.09	74.68	67.43	79.93	78.23	69.68	75.17
LLaMA-Adapter [59]	84.37	88.30	84.36	83.72	80.32	86.90	85.83	84.05	85.19
MM-CoT$_{Base}$ [61]	87.52	77.17	85.82	87.88	82.90	86.83	84.65	85.37	84.91
MM-CoT$_{Large}$ [61]	95.91	82.00	90.82	95.26	88.80	92.89	92.44	90.31	91.68
Results with our own experiment runs									
GPT-4†	84.06	73.45	87.36	81.87	70.75	90.73	84.69	79.10	82.69
LLaVA	90.36	95.95	88.00	89.49	88.00	90.66	90.93	90.90	90.92
LLaVA+GPT-4† (complement)	90.36	95.50	88.55	89.05	87.80	91.08	92.22	88.73	90.97
LLaVA+GPT-4† (judge)	91.56	96.74	91.09	90.62	88.99	93.52	92.73	92.16	**92.53**

Table 7: Accuracy (%) on Science QA dataset. Question categories: NAT = natural science, SOC = social science, LAN = language science, TXT = text context, IMG = image context, NO = no context, G1-6 = grades 1-6, G7-12 = grades 7-12. †Text-only GPT-4, our eval. Our novel model ensembling with the text-only GPT-4 consistently improves the model's performance under all categories, setting the new SoTA performance.

Improved Baselines with Visual Instruction Tuning (LLaVA-1.5)

Modifications

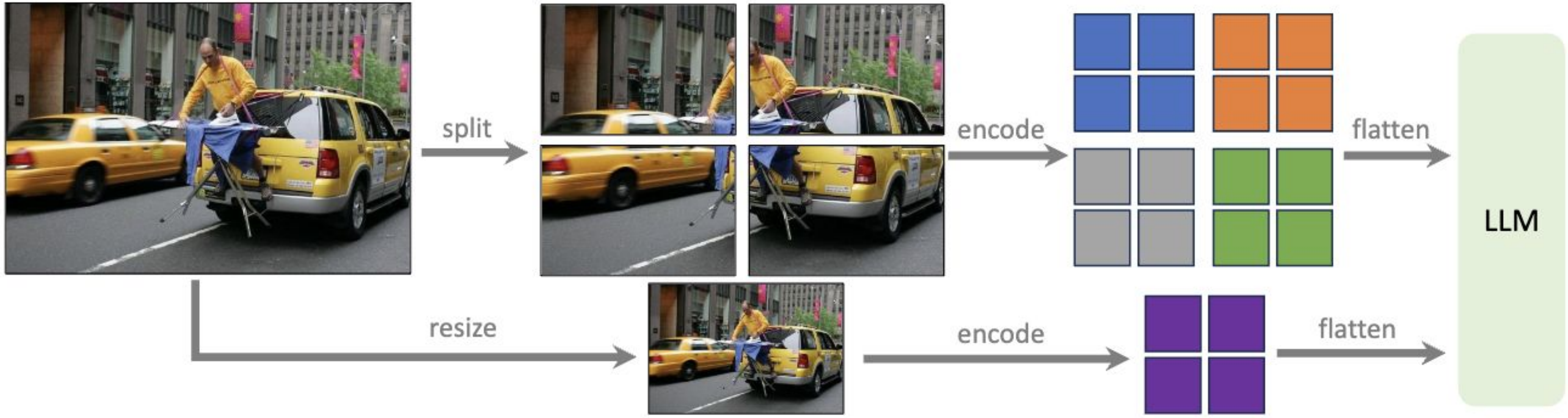
- Updated LLM (Vicuna v1.5 7/13B)
- High resolution vision encoder (CLIP ViT-L/336)
- More data!



Improved Baselines with Visual Instruction Tuning (LLaVA-1.5)

Higher Resolution

- Split images into grids and encode them independently.
- Concatenate encodings of the patches with the encoding of the resized (original) image.
- Scales to any resolution without adjusting positional embeddings.



Improved Baselines with Visual Instruction Tuning (LLaVA-1.5)

Modifications

- **MME** (total improvement = 721.7)
  14 subtasks: Coarse-Grained Recognition, Fine-Grained Recognition, OCR
    - Data: 89.5% (absolute = 646.2)
    - Model: 7.2% (absolute = 52)
    - Resolution: 3.3% (absolute = 23.5)
    - *GPT-4V: 1409.43*
- **MM-Vet** (total improvement = 10.6)
  6 core tasks: Recognition, Knowledge, OCR, Spatial awareness, Language generation, Math
    - Data: 44% (absolute = 4.6)
    - Model: 51.9% (absolute = 5.5)
    - Resolution: -4.7% (absolute = -0.5)
    - *GPT-4V: 67.7*

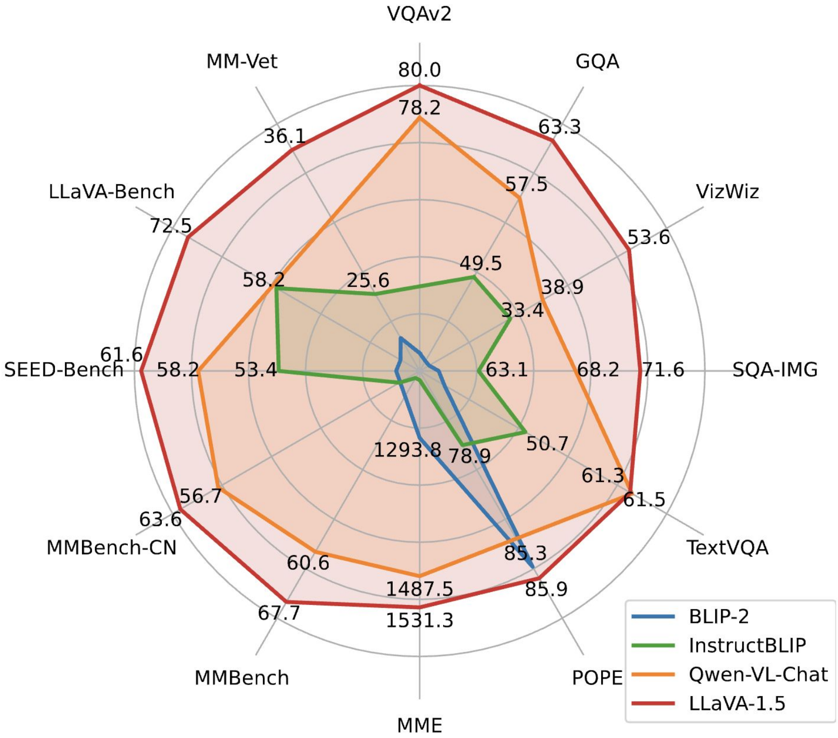
Method		LLM	Res.	GQA	MME	MM-Vet
InstructBLIP		14B	224	49.5	1212.8	25.6
Only using a subset of InstructBLIP training data						
0	**LLaVA**	7B	224	–	809.6	25.5
1	+VQA-v2	7B	224	47.0	1197.0	27.7
2	+Format prompt	7B	224	46.8	1323.8	26.3
3	+MLP VL connector	7B	224	47.3	1355.2	27.8
4	+OKVQA/OCR	7B	224	50.0	1377.6	29.6
Additional scaling						
5	+Region-level VQA	7B	224	50.3	1426.5	30.8
6	+Scale up resolution	7B	336	51.4	1450	30.3
7	+GQA	7B	336	62.0*	1469.2	30.7
8	+ShareGPT	7B	336	62.0*	1510.7	31.1
9	+Scale up LLM	13B	336	**63.3***	**1531.3**	**36.1**

Table 2. **Scaling results** on data, model, and resolution. We choose to conduct experiments on GQA [21], MME [17], and MM-Vet [55] to examine the representative capabilities of VQA with short answers, VQA with output formatting, and natural visual conversations, respectively. *Training images of GQA were observed during training.

Improved Baselines with Visual Instruction Tuning (LLaVA-1.5)

Results

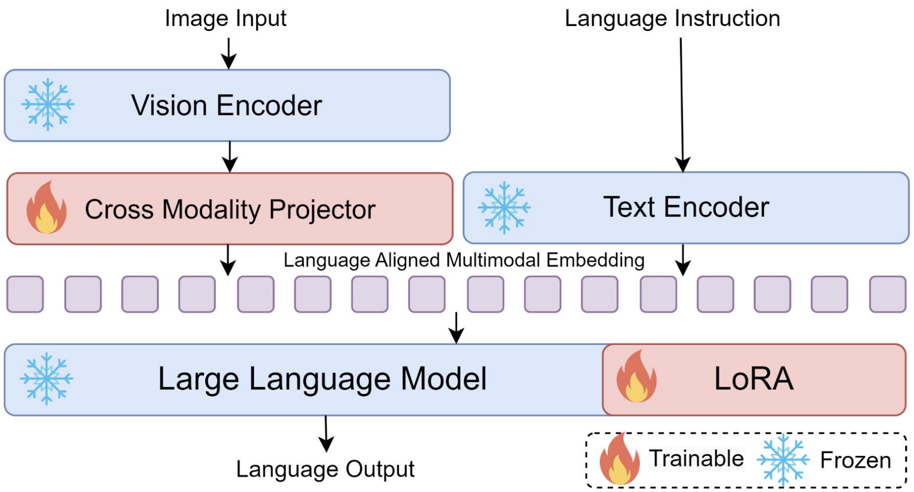
- ScienceQA dataset.
- Sets of visual question answering problems involving different scientific fields.
- Achieve close to SoTA on all subjects.
- GPT-4 achieves 70.75% accuracy on questions with image context!
    - Is this a useful multimodal benchmark?



Efficient Multimodal Learning from Data-centric Perspective (Bunny)

Overview

- Introduces Bunny-3B
- Architecture is almost identical to LLaVA.
- Uses Phi-2 (2.7B) LLM and SigLIP-SO (400M) vision encoder.
- LoRA for fine-tuning both the language model and cross modality projector.
- Curate high quality datasets for pre-training and fine-tuning.



Efficient Multimodal Learning from Data-centric Perspective (Bunny)

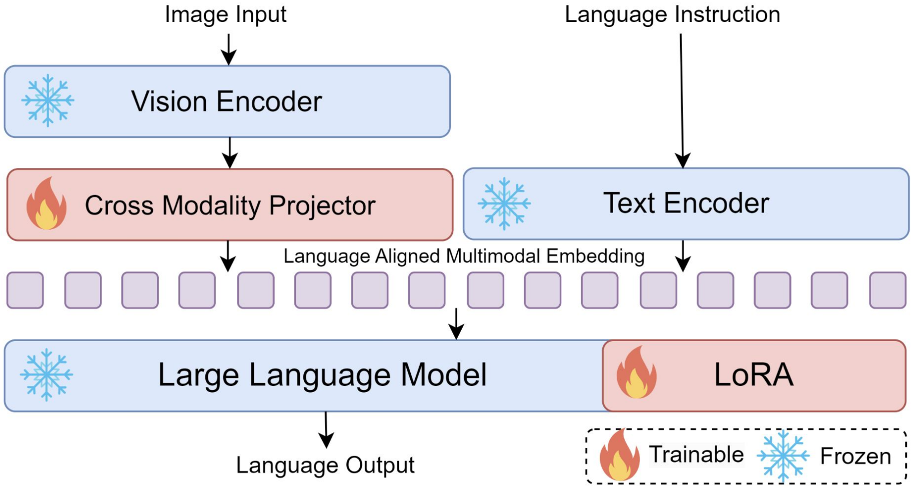
Data

Pre-training

- Filter LAION-2B to 2M image-text pairs by deduping and filtering on CLIP similarities.

Fine-tuning

- Start from SVIT-mix-665K, which is the LLaVA-1.5 dataset.
- Replace ShareGPT-40K with WizardLM-evol-instruct-70K



Efficient Multimodal Learning from Data-centric Perspective (Bunny)

Architecture Sweep

Table 2: Bunny with various language and vision backbones. The best performances are achieved by integrating SigLIP-SO [14] and Phi-2 [13].

Vision Encoder	LLM	MME^{P}	MME^{C}	MMB^{T}	MMB^{D}	SEED	$MMMU^{V}$	$MMMU^{T}$	VQA^{v2}	GQA	SQA^{I}	POPE
EVA02-CLIP-L (0.4B)	Phi-1.5 (1.3B)	1213.7	278.9	60.9	56.8	56.4	30.0	28.4	76.5	60.4	58.2	86.1
	StableLM-2 (1.6B)	1301.0	235.0	58.4	56.4	55.3	29.8	29.4	74.6	56.7	60.0	84.8
	Phi-2 (2.7B)	1421.0	285.4	68.6	67.4	62.2	35.9	32.6	78.9	62.3	69.1	87.1
SigLIP-SO (0.4B)	Phi-1.5 (1.3B)	1230.0	237.5	61.2	59.7	57.7	30.0	29.1	78.0	61.1	61.3	85.8
	StableLM-2 (1.6B)	1366.8	236.1	65.1	62.8	58.8	29.9	29.8	78.9	60.9	61.1	85.9
	Phi-2 (2.7B)	1488.8	289.3	69.2	68.6	62.5	38.2	33.0	79.8	62.5	70.9	86.8

Efficient Multimodal Learning from Data-centric Perspective (Bunny)

Architecture Sweep

Table 2: Bunny with various language and vision backbones. The best performances are achieved by integrating SigLIP-SO [14] and Phi-2 [13].

Vision Encoder	LLM	MME^P	MME^C	MMB^T	MMB^D	SEED	$MMMU^V$	$MMMU^T$	VQA^{v2}	GQA	SQA^I	POPE
EVA02-CLIP-L (0.4B)	Phi-1.5 (1.3B)	1213.7	278.9	60.9	56.8	56.4	30.0	28.4	76.5	60.4	58.2	86.1
	StableLM-2 (1.6B)	1301.0	235.0	58.4	56.4	55.3	29.8	29.4	74.6	56.7	60.0	84.8
	Phi-2 (2.7B)	1421.0	285.4	68.6	67.4	62.2	35.9	32.6	78.9	62.3	69.1	87.1
SigLIP-SO (0.4B)	Phi-1.5 (1.3B)	1230.0	237.5	61.2	59.7	57.7	30.0	29.1	78.0	61.1	61.3	85.8
	StableLM-2 (1.6B)	1366.8	236.1	65.1	62.8	58.8	29.9	29.8	78.9	60.9	61.1	85.9
	Phi-2 (2.7B)	1488.8	289.3	69.2	68.6	62.5	38.2	33.0	79.8	62.5	70.9	86.8

GPT-4V: 55.7
LLaVA-1.5-13B: 33.6
LLaMA-Adapter2-7B: 27.7

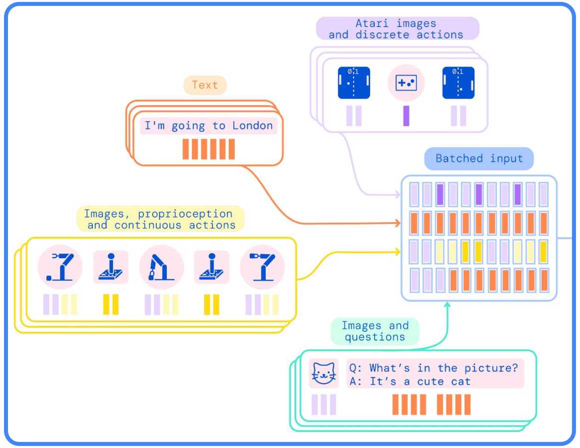
Efficient Multimodal Learning from Data-centric Perspective (Bunny)

Results

Model	Vision Encoder	LLM	MME^{P}	MME^{C}	MMB^{T}	MMB^{D}	SEED	MMMU^{V}	MMMU^{T}	VQA^{v2}	GQA	SQA^{I}	POPE
IDEFICS-80B [43]	OpenCLIP-H (1.0B)	LLaMA-65B	–	–	54.6	54.5	–	–	–	60.0	–	68.9	–
BLIP-2 [24]	EVA01-CLIP-G (1.0B)	Vicuna-13B	–	–	–	–	–	–	–	–	41.0	61.0	–
InstructBLIP [44]	EVA01-CLIP-G (1.0B)	Vicuna-13B	–	–	–	–	–	–	–	–	49.5	63.1	83.7
BLIP-2 [24]	EVA01-CLIP-G (1.0B)	Flan-T5-XXL (11B)	1293.8	290.0	–	–	–	35.4	34.0	65.0	44.6	64.5	–
InstructBLIP [44]	EVA01-CLIP-G (1.0B)	Flan-T5-XXL (11B)	1212.8	291.8	–	–	–	35.7	33.8	–	47.9	70.6	–
Shikra-13B [5]	CLIP-L (0.4B)	Vicuna-13B	–	–	–	–	–	–	–	77.4	–	–	–
LLaVA-v1.5-13B (LoRA) [26]	CLIP-L (0.4B)	Vicuna-13B	1541.7	300.4§	68.4§	68.5	61.3	40.0§	33.2§	80.0	63.3	71.2	86.7
InstructBLIP [44]	EVA01-CLIP-G (1.0B)	Vicuna-7B	–	–	33.9	36.0	53.4	–	–	–	49.2	60.5	–
MiniGPT-v2 [28]	EVA01-CLIP-G (1.0B)	LLaMA2-7B	–	–	–	–	–	–	–	–	60.3	–	–
IDEFICS-9B [43]	OpenCLIP-H (1.0B)	LLaMA-7B	–	–	45.3	48.2	–	–	–	50.9	–	44.2	–
LLaVA-v1.5-7B (LoRA) [26]	CLIP-L (0.4B)	Vicuna-7B	1476.9	267.9§	66.1§	66.1	60.1	34.4§	31.7§	79.1	**63.0**	68.4	86.4
mPLUG-Owl2 [45]	CLIP-L (0.4B)	LLaMA2-7B	1450.2	**313.2**	66.0	66.5	57.8	32.7	32.1	79.4	56.1	68.7	85.8
Shikra-7B [5]	CLIP-L (0.4B)	Vicuna-7B	–	–	60.2	58.8	–	–	–	–	–	–	–
TinyGPT-V [29]	EVA01-CLIP-G (1.0B)	Phi-2 (2.7B)	–	–	–	–	–	–	–	–	33.6	–	–
MobileVLM [15]	CLIP-L (0.4B)	MobileLLaMA (2.7B)	1288.9	–	–	59.6	–	–	–	–	59.0	61.0	84.9
LLaVA-Phi [9]	CLIP-L (0.4B)	Phi-2 (2.7B)	1335.1	–	–	59.8	–	–	–	71.4	–	68.4	85.0
MC-LLaVA [46]	SigLIP-SO (0.4B)	Dolphin 2.6 Phi-2 (2.7B)	–	–	–	–	–	–	–	64.2	49.6	–	80.6
Imp-v1 [10]	SigLIP-SO (0.4B)	Phi-2 (2.7B)	1434.0	–	–	66.5	–	–	–	79.5	58.6	70.0	**88.0**
MiniCPM-V [16]	SigLIP-SO (0.4B)	MiniCPM (2.4B)	1446.0	–	–	67.3	–	34.7	–	–	–	–	–
Moondream1 [47]	SigLIP-SO (0.4B)	Phi-1.5 (1.3B)	–	–	–	–	–	–	–	74.3	56.3	–	–
TinyLLaVA-v1 [48]	CLIP-L (0.4B)	TinyLlama (1.1B)	–	–	–	–	–	–	–	73.4	57.5	59.4	–
Bunny	SigLIP-SO (0.4B)	Phi-2 (2.7B)	**1488.8**	289.3	**69.2**	**68.6**	**62.5**	**38.2**	**33.0**	**79.8**	62.5	**70.9**	86.8

Summary

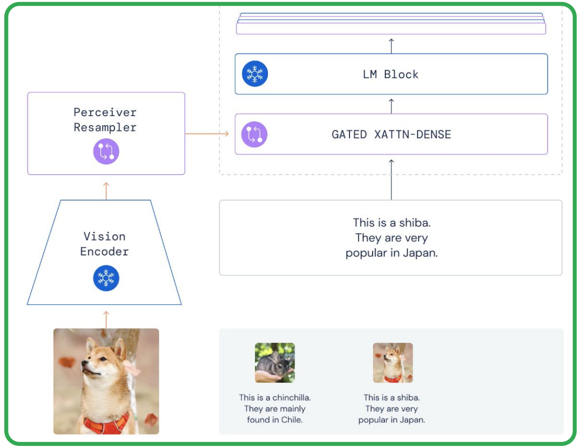
Merge Modalities at Input



Gato

- Flexible architecture.
- Requires most compute.

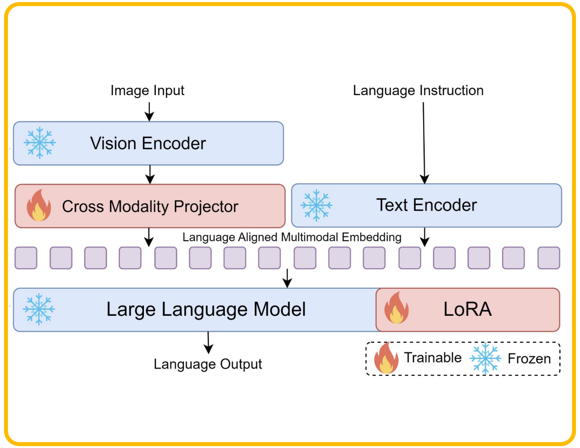
Merge Modalities with Cross-Attention



Flamingo

- Frozen vision encoder and LLM.
- Cross-attention layers and vision adapter are trained from scratch.

Visual Instruction Tuning



Visual Instruction Tuning

- Leverages pre-trained models.
- Low number of trained parameters.
- Relies on high quality instruction tuning datasets.

Multimodal Foundation Models

Generating Images

01 DALL-E 1

02 DALL-E 2

03 DALL-E 3

04 Diffusion Transformers

05 Recaptioning Ourselves

DALL-E

1

Zero-Shot Text-to-Image Generation

Ramesh, Aditya, et al., 2021

DALL-E

Vision Encoder Pre-Training

- Trains a Discrete Variational Autoencoder (dVAE).
- Input images are 256x256x3 RGB images.
- The encoder and decoder are a series of Conv2D layers.
- The encoder output is 32x32x8192, which represent 8192 logits for each position in the 32x32 grid of image tokens.
- Each 256x256 RGB input image can be represented as 32x32 = 1024 integers.

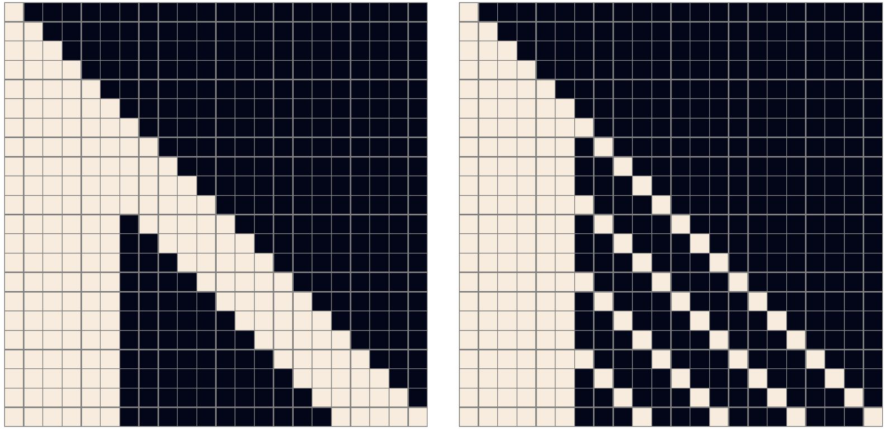
Original Images

Reconstructed Images

DALL-E

Text-to-Image Training

- Train a GPT-3 model with up to 12B parameters.
- Concatenate text caption (limited to 256 tokens) with the 1024 image tokens.
- Feed the 1280 tokens to the GPT-3 transformer decoder.
- Use next token prediction to generate text and image tokens.
- Text tokens use a causal mask. Image tokens attend to all tokens and use several different types of attention masks to attend to other image tokens.



(a) Row attention mask. (b) Column attention mask.

Zero-Shot Text-to-Image Generation (DALL-E)

Sampling

- Prompt the learned GPT-3 decoder with the input text and sample the 1024 image tokens.
- Pass these 1024 image tokens through the dVAE decoder.
- Do the previous two steps N times to obtain N samples for the text prompt.
- Use CLIP to rerank the N samples and select the top k.

DALL-E 2

2

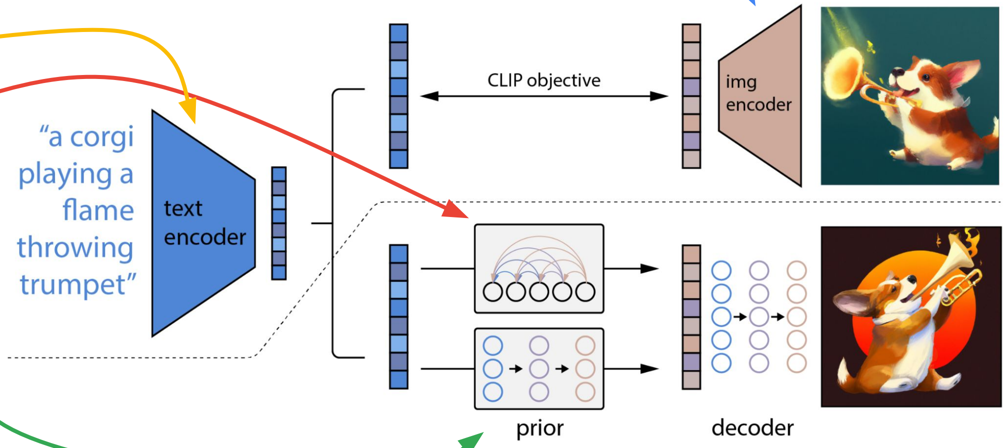
Hierarchical Text-Conditional
Image Generation with CLIP Latents

Ramesh, Aditya, et al., 2022

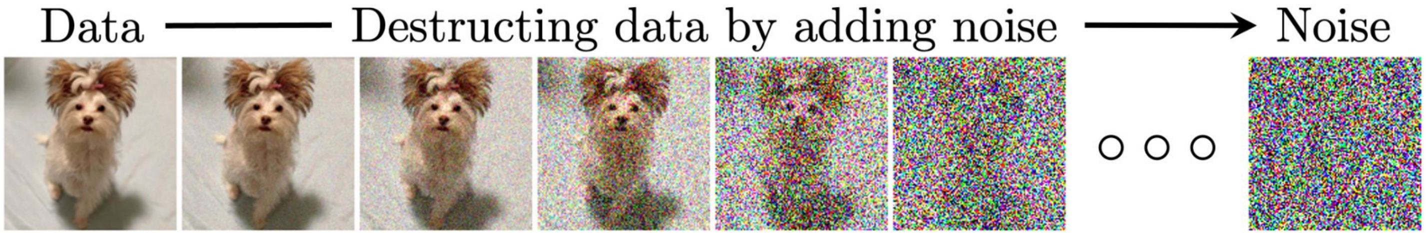
DALL-E 2 (aka unCLIP)

Overview

- Use a ViT-H/16 vision and text encoder trained with CLIP
- Train both an autoregressive an diffusion prior.
- Use a diffusion decoder and diffusion upsamplers for decoding and upsampling images.



A Very Short Overview of Diffusion

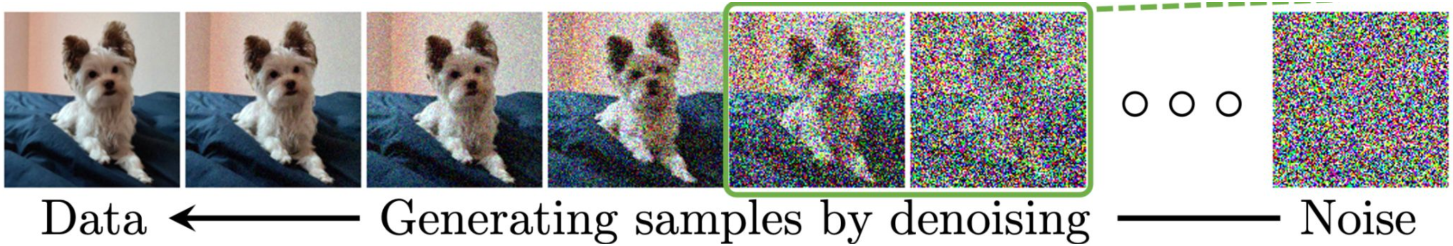


Yang, Ling, et al. "Diffusion models: A comprehensive survey of methods and applications." ACM Computing Surveys 56.4 (2023): 1-39.

Training

- Add gaussian noise to your input image.
- Predict the original image without noise.
- Typically we use a compact image representation (e.g. CLIP embeddings).
- Text information can be added to the image representation for text-to-image generation

A Very Short Overview of Diffusion

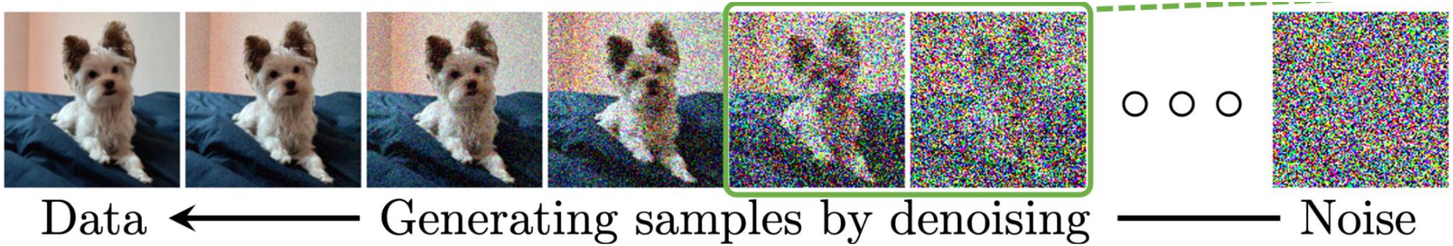


Yang, Ling, et al. "Diffusion models: A comprehensive survey of methods and applications." ACM Computing Surveys 56.4 (2023): 1-39.

Sampling

- Start with gaussian noise x.
- Take N steps:
  - Denoise the input noisy image by running it through the learned network.
  - Use the output of the network as input for the network in the next step.
- Classifier-free guidance: move your generation process away from unconditional generation.
  - During training use dropout on your conditioning signal (e.g. text).
  - During sampling denoising, draw two samples: one with the conditioning signal and another without.
  - Subtract the unconditional denoised output.

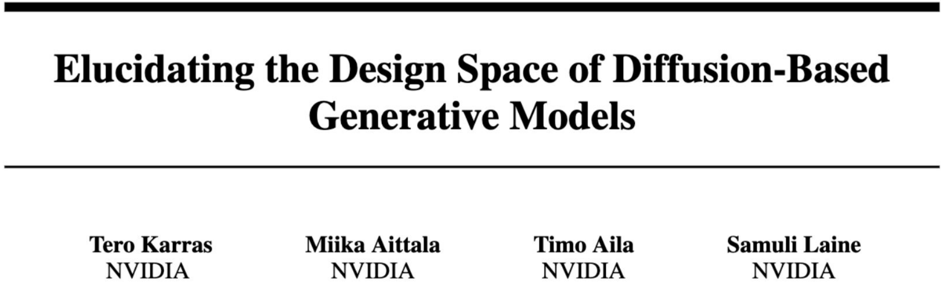
A Very Short Overview of Diffusion



Yang, Ling, et al. "Diffusion models: A comprehensive survey of methods and applications." ACM Computing Surveys 56.4 (2023): 1-39.

Sampling

- Start with gaussian noise x.
- Take N steps:
  - Denoise the input noisy image by running it through the learned network.
  - Use the output of the network as input for the network in the next step.
- Classifier-free guidance: move your generation process away from unconditional generation.
  - During training use dropout on your conditioning signal (e.g. text).
  - During sampling denoising, draw two samples: one with the conditioning signal and another without.
  - Subtract the unconditional denoised output.



Elucidating the Design Space of Diffusion-Based Generative Models

Tero Karras NVIDIA | Miika Aittala NVIDIA | Timo Aila NVIDIA | Samuli Laine NVIDIA

DALL-E 2 (aka unCLIP)

Diffusion Prior

- Train a decoder-only transformer with a causal attention mask on:
  - Encoded text
  - CLIP text embeddings
  - Embedding for diffusion timestep
  - Noised CLIP image embedding
- Two samples are drawn (CLIP image embeddings) from the diffusion prior.
- The sample with the highest dot-product with the CLIP text embeddings is kept.

unCLIP Prior	Photorealism	Caption Similarity	Diversity
AR	$47.1\% \pm 3.1\%$	$41.1\% \pm 3.0\%$	$62.6\% \pm 3.0\%$
Diffusion	$48.9\% \pm 3.1\%$	$45.3\% \pm 3.0\%$	$70.5\% \pm 2.8\%$

Human evals compared to GLIDE (text guided diffusion)

DALL-E 3

3

Betker, James, et al. "Improving image generation with better captions." Computer Science. https://cdn.openai. com/papers/dall-e-3. pdf 2.3 (2023): 8.

DALL-E 3

In a fantastical setting, a highly detailed furry humanoid skunk with piercing eyes confidently poses in a medium shot, wearing an animal hide jacket. The artist has masterfully rendered the character in digital art, capturing the intricate details of fur and clothing texture.

Overview

- Similar architecture to DALL-E 3…maybe?
- Synthetic data!

[5]DALL-E 3 has many improvements over DALL-E 2, many of which are not covered in this document and could not be ablated for time and compute reasons. The evaluation metrics discussed in this document should not be construed as a performance comparison resulting from simply training on synthetic captions.

DALL-E 3

Training a Captioning Model

- Start with a traditional causal decoder transformer.
- Modify it by adding CLIP image embeddings as a conditioning signal.
- Pre-train this language model on image-text pairs.
- Jointly train the captioning model with a CLIP and language modeling objective (Yu, Jiahui, et al 2022).
- Fine-tune this model with two human collected datasets
    - Short image descriptions
    - Long, detailed image descriptions
- Use this captioning model to re-caption image datasets

$$L(t) = \sum_j \log P(t_j | t_{j-k}, \ldots, t_{j-1}; \Theta)$$

$$L(t, i) = \sum_j \log P(t_j | t_{j-k}, \ldots, t_{j-1}; z_j; F(i); \Theta)$$

DALL-E 3

Training on Synthetically Generated Captions

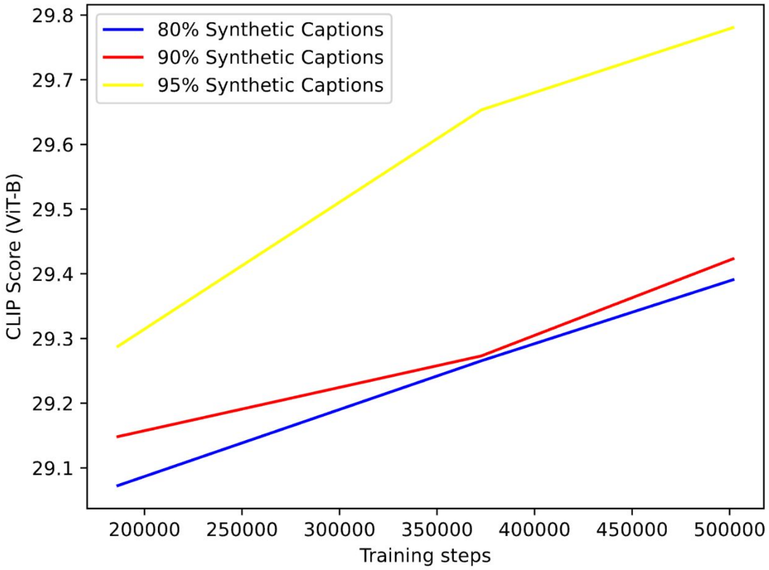


Figure 5 – CLIP scores for text-to-image models trained on various blending ratios of descriptive synthetic captions and ground-truth captions. Evaluation performed using ground truth captions.

DALL-E 3

Training on Synthetically Generated Captions

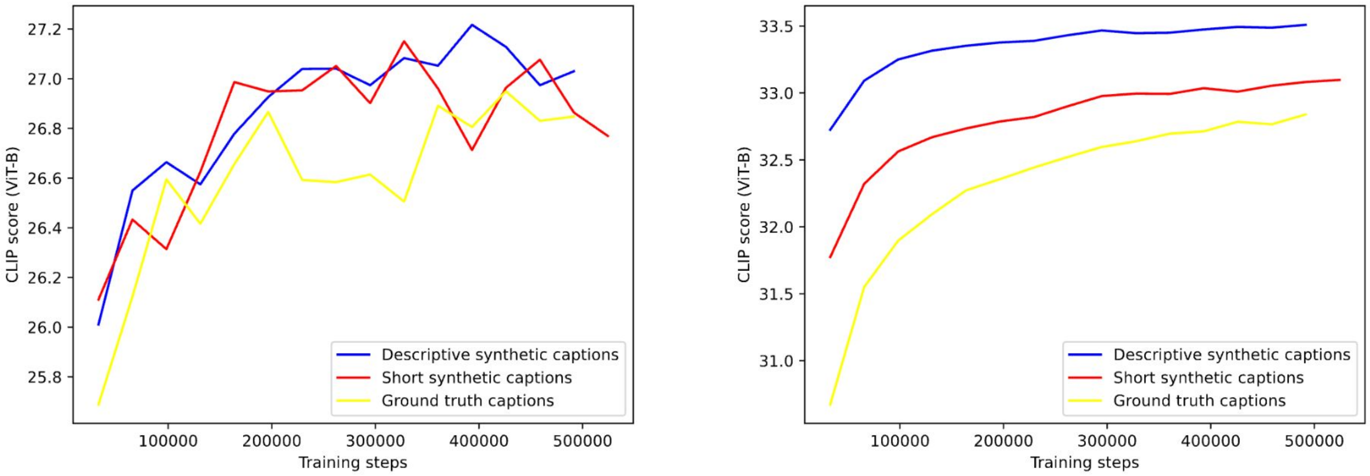


Figure 4 – CLIP scores for text-to-image models trained on different caption types. Left is evaluation results with ground truth captions on our evaluation dataset. Right uses the descriptive synthetic captions from the same dataset.

DALL-E 3

Training on Synthetically Generated Captions

- Used GPT-4 to “upsample” captions.

```
You are part of a team of bots that creates images. You work with an assistant bot that will draw anything
you say in square brackets. For example, outputting "a beautiful morning in the woods with the sun peaking
through the trees" will trigger your partner bot to output an image of a forest morning, as described.
You will be prompted by people looking to create detailed, amazing images. The way to accomplish this
is to take their short prompts and make them extremely detailed and descriptive.

There are a few rules to follow:
- You will only ever output a single image description per user request.
- Sometimes the user will request that you modify previous captions. In this case, you should refer to your
  previous conversations with the user and make the modifications requested.
- When modifications are requested, you should not simply make the description longer. You should refactor
  the entire description to integrate the suggestions.
- Other times the user will not want modifications, but instead want a new image. In this case, you should
  ignore your previous conversation with the user.
- Image descriptions must be between 15-80 words. Extra words will be ignored.

    {"role": "user", "content": "Create an imaginative image descriptive caption or modify an earlier
\
caption for the user input: "a man holding a sword\""},
    {"role": "assistant", "content": "a pale figure with long white hair stands in the center of a dark \
    forest, holding a sword high above his head.
the blade glows with a blue light, casting a soft glow on the trees and bushes surrounding him."},
    {"role": "user", "content": "Create an imaginative image descriptive caption or modify an earlier caption \
    for the user input: "make the light red"},
    {"role": "assistant", "content": "a pale figure with long white hair stands in the center of a dark forest, \
    holding a sword high above his head.
the blade glows with a red light, casting a warm glow on the trees and bushes surrounding him."},
    {"role": "user", "content": "Create an imaginative image descriptive caption or modify an earlier caption for the \
    user input: "draw a frog playing dominoes"},
    {"role": "assistant", "content": "a frog sits on a worn table playing a game of dominoes with an elderly raccoon.
the table is covered in a green cloth, and the frog is wearing a jacket and a pair of jeans. The scene is set in a forest,
with a large tree in the background."}
```

DALL-E 3

Training on Synthetically Generated Captions

- Used GPT-4 to “upsample” captions.

A bird scaring a scarecrow.

Paying for a quarter-sized pizza with a pizza-sized quarter.

A smafml vessef epropoeilled on watvewr by ors, sauls, or han engie.

A large, vibrant bird with an impressive wingspan swoops down from the sky, letting out a piercing call as it approaches a weathered scarecrow in a sunlit field. The scarecrow, dressed in tattered clothing and a straw hat, appears to tremble, almost as if it’s coming to life in fear of the approaching bird.

A person is standing at a pizza counter, holding a gigantic quarter the size of a pizza. The cashier, wide-eyed with astonishment, hands over a tiny, quarter-sized pizza in return. The background features various pizza toppings and other customers, all of them equally amazed by the unusual transaction.

A small vessel, propelled on water by oars, sails, or an engine, floats gracefully on a serene lake. the sun casts a warm glow on the water, reflecting the vibrant colors of the sky as birds fly overhead.

Diffusion Transformers (DiT)

4

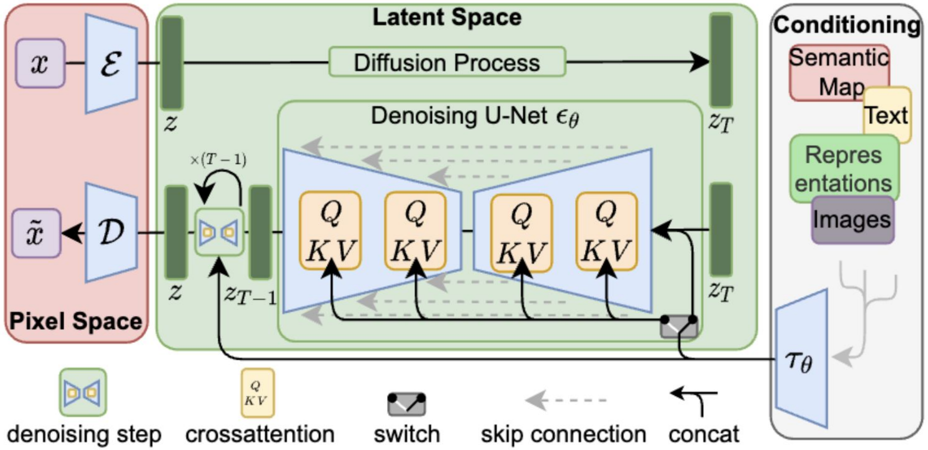
Peebles, William, and Saining Xie. "Scalable diffusion models with transformers." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023.

Diffusion Transformers

Latent Diffusion Model

- Diffusion is done in embedding, instead of pixel, space.
    - Used an off the shelf VAE from Stable Diffusion
    - 8x downsample factor: 256 x 256 x 3 → 32 x 32 x 4
- Embedding space is smaller than the original image.
- Training and samling both are more FLOP efficient.

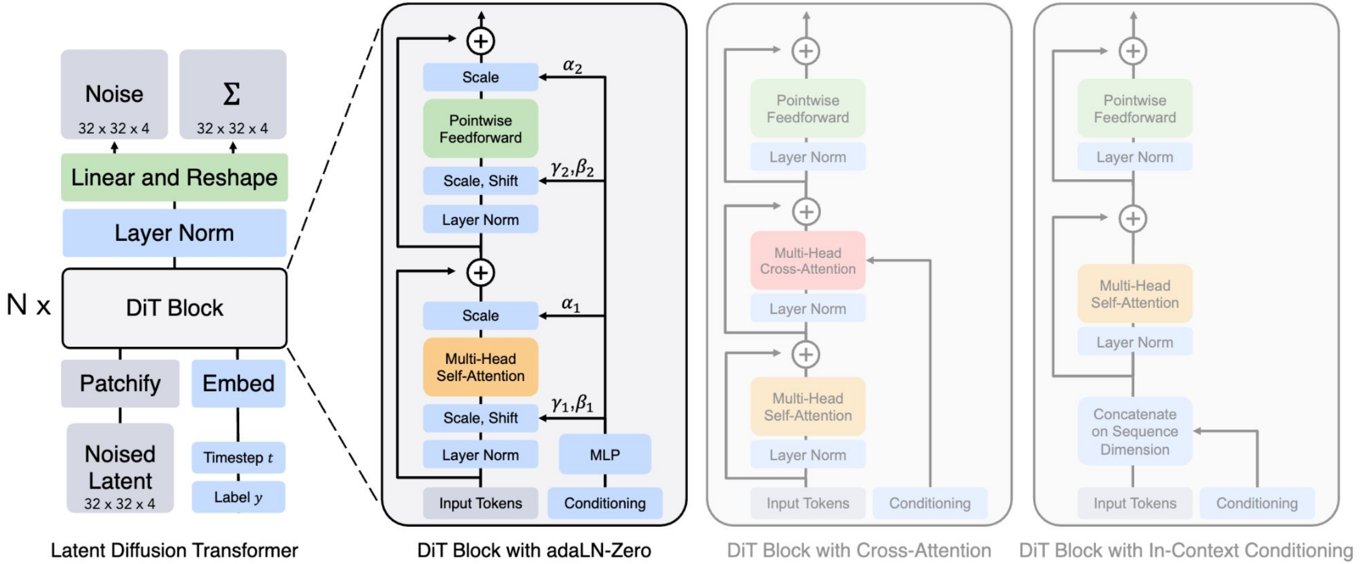
1. Pre-train encoders and decoders (typically an Autoencoder)
2. Encode the input image to get latents.
3. Add noise to those latents.
4. Train a network to predict the original latents given the noisy ones.
5. Decode the latents using the decoder.



Diffusion Transformers

Diffusion Transformer Architecture

- Uses a standard decoder transformer.
- Experimented with different ways to incorporate conditioning (text)



Diffusion Transformers

Diffusion Transformer Architecture

- Uses a standard decoder transformer.
- Experimented with different ways to incorporate conditioning (text)

FID Score

Model	Image Resolution	Flops (G)	Params (M)	Training Steps (K)	Batch Size	Learning Rate	DiT Block	FID-50K (no guidance)
DiT-XL/2	256×256	119.37	449	400	256	1×10^{-4}	in-context	35.24
DiT-XL/2	256×256	137.62	598	400	256	1×10^{-4}	cross-attention	26.14
DiT-XL/2	256×256	118.56	600	400	256	1×10^{-4}	adaLN	25.21

Compute per forward pass

Context mechanism

Diffusion Transformers

Diffusion Transformer Architecture

- Uses a standard decoder transformer.
- Experimented with different ways to incorporate conditioning (text)

FID Score

Model	Image Resolution	Flops (G)	Params (M)	Training Steps (K)	Batch Size	Learning Rate	DiT Block	FID-50K (no guidance)
DiT-XL/2	256×256	119.37	449	400	256	1×10^{-4}	in-context	35.24
DiT-XL/2	256×256	137.62	598	400	256	1×10^{-4}	cross-attention	26.14
DiT-XL/2	256×256	118.56	600	400	256	1×10^{-4}	adaLN	25.21

Compute per forward pass

Computer per gradient update
~=
Forward Flops * 3

Context mechanism

Diffusion Transformers

Scaling Diffusion Transformers

- Scaling laws for different model sizes.
- Larger models are more compute efficient.
  - Does not mean larger is better!

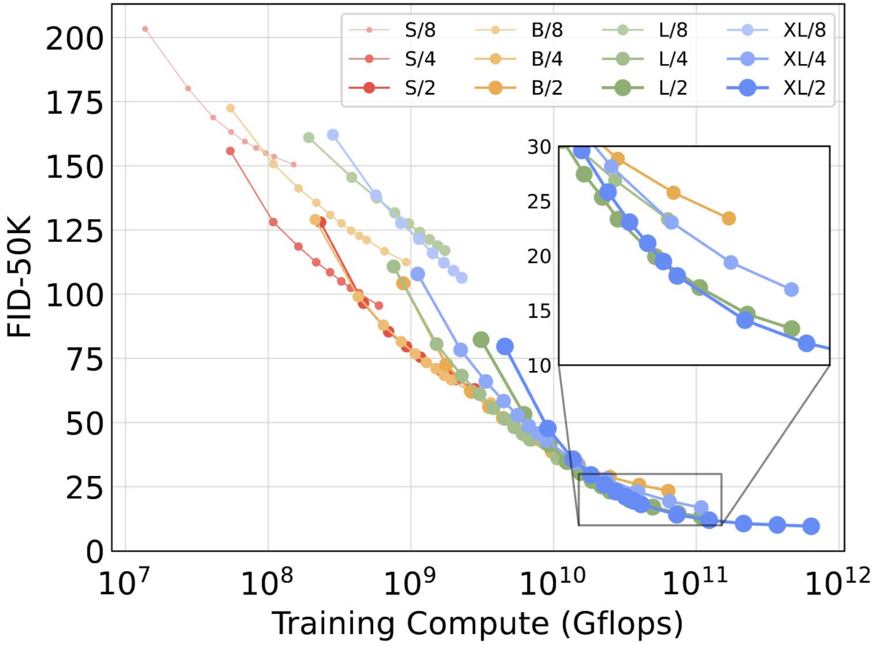


Figure 9. **Larger DiT models use large compute more efficiently.** We plot FID as a function of total training compute.

Recaptioning Ourselves

5

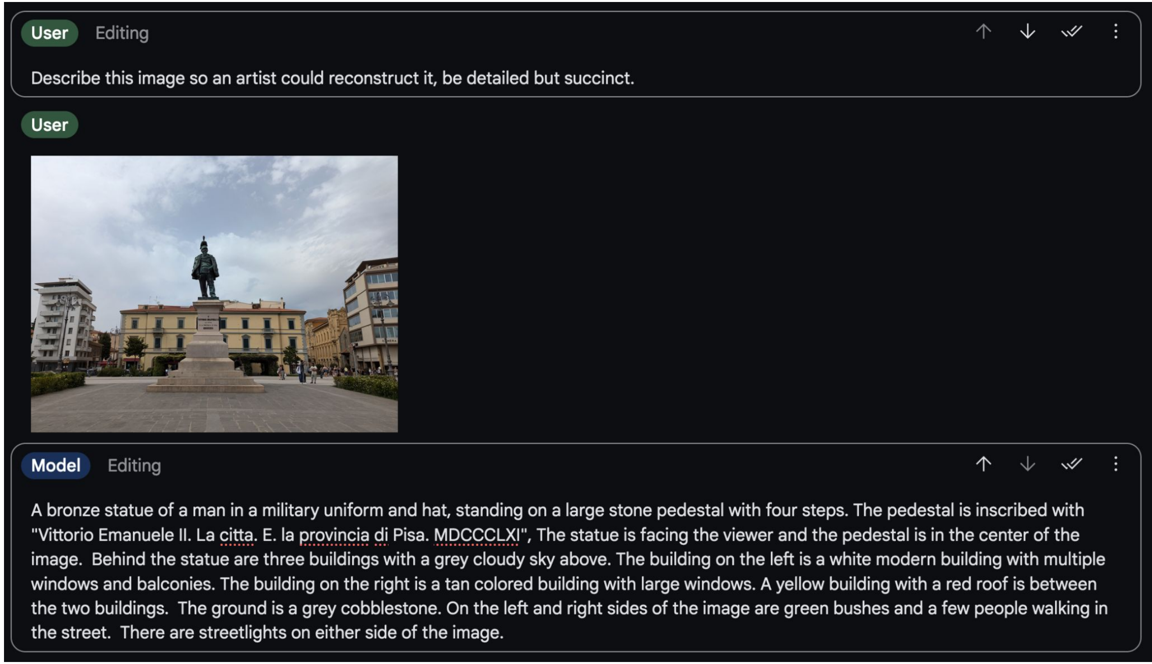
How Much Would it Cost to Recaption Yourself?

Assumptions

- Captioning prompt, e.g.
  “Describe this image in detail.” is 50 tokens.
- Output prompt is 256 tokens.

GPT-4o
- 1024x1024 image input = $0.003825
- 50 token input prompt = $0.00025
- 256 output tokens = $0.00384
- Captioning 1B images = $7,915,000

Gemini Flash 1.5
- 1024x1024 image input = $0.0000903
- 50 token input prompt = $0.0000175
- 256 output tokens = $0.0001792
- Captioning 1B images = $287,000



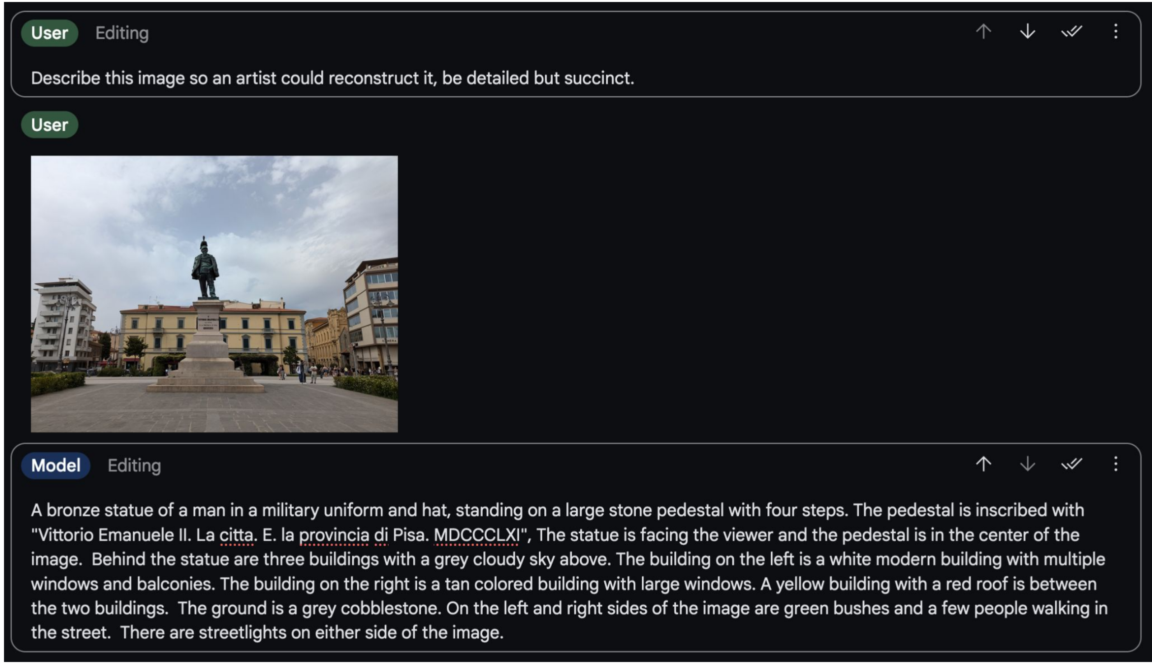
~~How Much Would it Cost to Recaption Yourself?~~
How long would it take to Recaption Yourself?

Assumptions

- Captioning prompt, e.g.
  “Describe this image in detail.” is 50 tokens.
- Output prompt is 256 tokens.

Gemini Flash 1.5 (Free Tier)

- 1,500 Requests per Day
- 1B images would take 666,667 days or
  1,826.5 years
- But you can do ~30k in a month for free :)



What If We Recaption Billions of Web Images with *LLaMA-3*?

Xianhang Li[⋆1] **Haoqin Tu**[⋆1] **Mude Hui**[⋆1] **Zeyu Wang**[⋆1] **Bingchen Zhao**[⋆2] **Junfei Xiao**[3]
Sucheng Ren[3] **Jieru Mei**[3] **Qing Liu**[4] **Huangjie Zheng**[5] **Yuyin Zhou**[1] **Cihang Xie**[1]

[⋆]equal technical contribution

[1]UC Santa Cruz [2]University of Edinburgh [3]JHU [4]Adobe [5]UT Austin

What If We Recaption Billions of Web Images with *LLaMA-3*?

Xianhang Li[⋆1] **Haoqin Tu**[⋆1] **Mude Hui**[⋆1] **Zeyu Wang**[⋆1] **Bingchen Zhao**[⋆2] **Junfei Xiao**[3]
Sucheng Ren[3] **Jieru Mei**[3] **Qing Liu**[4] **Huangjie Zheng**[5] **Yuyin Zhou**[1] **Cihang Xie**[1]

[⋆]equal technical contribution

[1]UC Santa Cruz [2]University of Edinburgh [3]JHU [4]Adobe [5]UT Austin

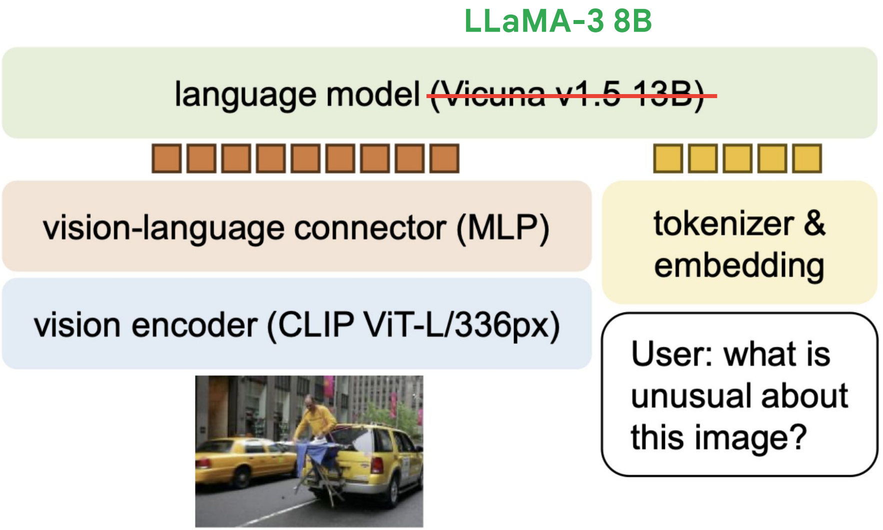
HOT OFF THE PRESS

June 12, 2024

What If We Recaption Billions of Web Images with LLaMA-3?

Overview

- Trained a modified LLaVA-1.5 for image captioning.
  - LLaMA-3 8B
  - CLIP ViT-L/14
- Used the same two-phase training strategy as LLaVA-1.5
- Similar data mixture as the original LLaVA-1.5
- Recaptioned DataComp-1B
  - Widely accessible, large-scale vision-language dataset comprising ~1.3 billion web-crawled image-text pairs.



What If We Recaption Billions of Web Images with LLaMA-3?

Caption Quality

- Re-captioned text is significantly longer.

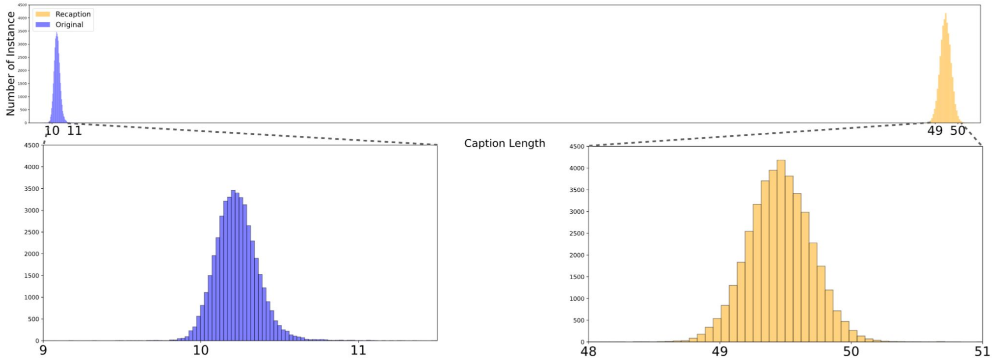


Figure 3: Average length distributions of both the original captions and our recaptioned data in DataComp-1B.

What If We Recaption Billions of Web Images with LLaMA-3?

Caption Quality

- Re-captioned text is significantly longer.
- Training on re-captioned data isn't great when evaluating on the original captions.
- Much better when evaluating on expanded prompts.
- This is consistent with the DALL-E 3 paper.

Table 7: Text-to-Image evaluation on COCO-30K results of DiT-BASE/4, trained with different mix ratios on Recap-DataComp-1B. Note for GPT-4V Score, we use a subset of 3K for the evaluation.

Training	Evaluation					
mixed ratio p	Raw		Our COCO-Recap			
	FID↓	CLIP Score↑	FID↓	CLIP Score↑	Recap-Clip Score↑	GPT-4V Score↑
0.00	37.6	29.2	$27.8_{-8.4}$	$32.5_{+3.1\%}$	$\mathbf{28.3}_{+8.4\%}$	$\mathbf{2.53}_{+1.1}$
0.05	38.5	29.1	27.9	32.5	28.0	2.51
0.10	36.0	29.7	**27.2**	32.7	28.2	2.51
0.15	35.8	29.9	28.2	**33.0**	28.1	2.45
0.20	35.8	29.8	28.4	32.7	28.0	**2.53**
0.50	35.3	29.3	30.2	31.9	26.7	2.13
0.75	31.3	29.4	32.7	31.2	25.8	1.89
1.00	32.5	28.9	36.2	29.3	19.9	1.40

What If We Recaption Billions of Web Images with LLaMA-3?

Dataset is Available

- On hugging face.
- Captions are typically pretty good.
- There are some common mistakes, e.g. "The second image…"
- Missing named entities.

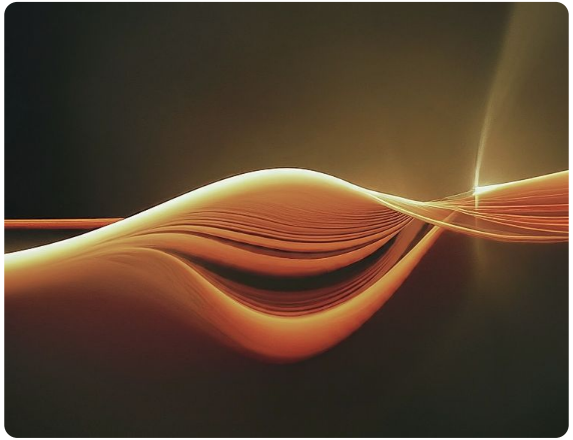
re_caption string · lengths 263↔323 20.6%	org_caption string · lengths 8↔121 88.3%
A modern coffee machine with a digital display and two white coffee cups filled with coffee is shown. The machine has a stainless steel finish and is accompanied by a milk frothing pitcher with a white liquid inside. The coffee machine is placed on a surface with a white background.	Saeco Xelsis Automatic Espresso Machine, SM7685/04, Stainless Steel

re_caption string · lengths 443↔503 3.8%	org_caption string · lengths 8↔121 88.3%
The second image shows the same scene with the two characters, but now they are facing each other. The man on the left is wearing a blue jacket and holding a gun, while the woman on the right is wearing a red jacket and also holding a gun. The background has changed to a darker, more ominous setting with a stormy sky and rain pouring down. The buildings and the	Resident Evil 2 Download

General Themes

Autoregressive Image Generation

Requires quantizing image embeddings, a lossy process.



Diffusion for Image Generation

Allows us to directly model the original continuous image latents.

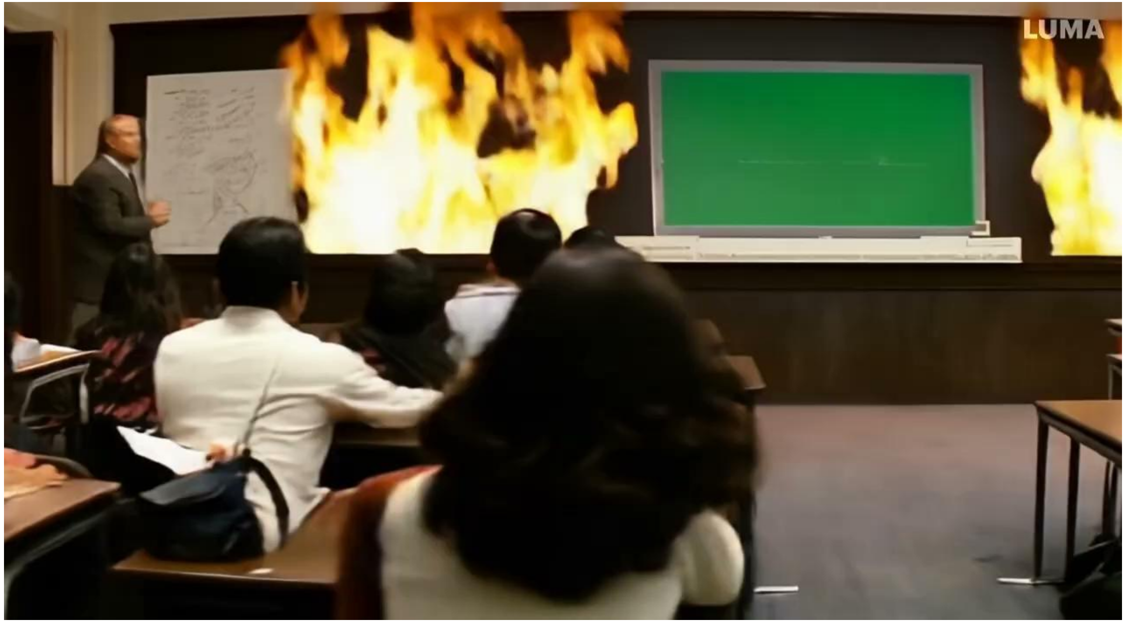
Re-captioning Data with LLMs

More detailed, richer captions at scale.

Multimodal Foundation Models

Google DeepMind

LumaLabs - DreamMachine



https://lumalabs.ai/dream-machine

Generating Videos

01 Stochastic Differential Editing

02 VideoPoet

03 WALT

04 Lumiere

05 Sora

06 Veo

07 Kling

Stochastic Differential Editing (SDEdit)

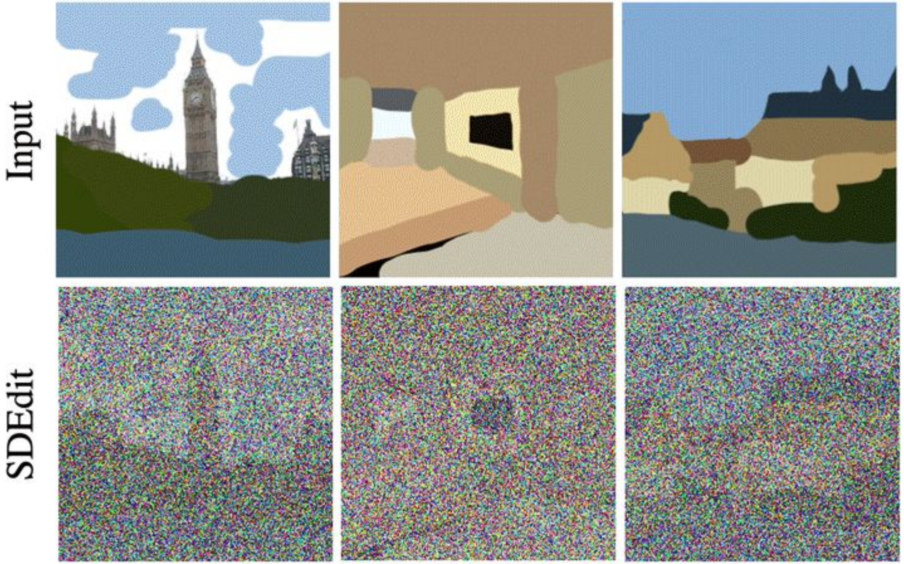
Meng, Chenlin, et al. "Sdedit: Guided image synthesis and editing with stochastic differential equations." arXiv preprint arXiv:2108.01073 (2021).

1

Guided Image Synthesis and Editing

Overview

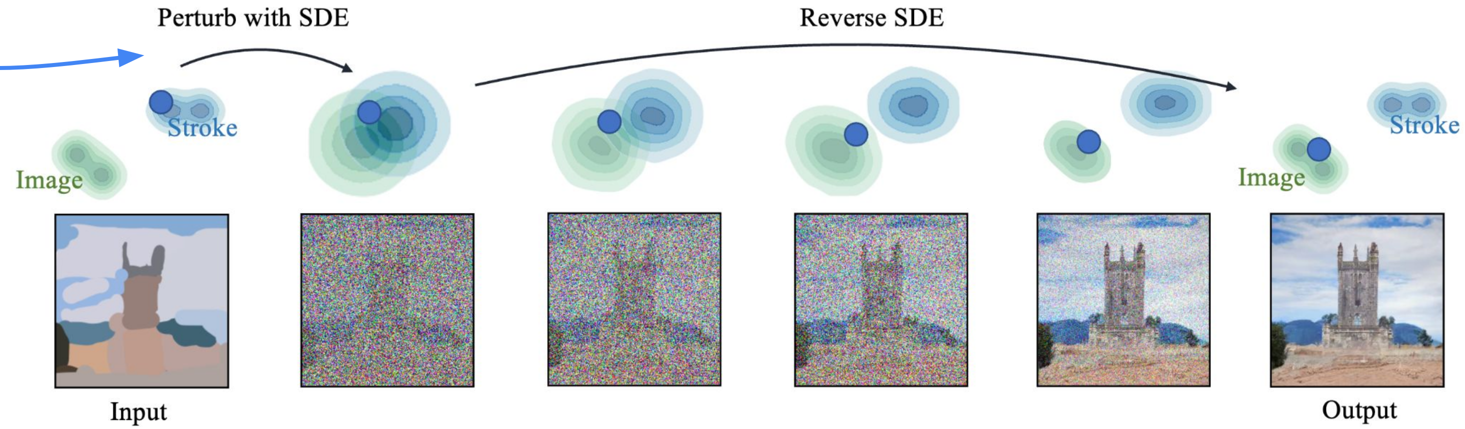
- Uses a pre-trained text-to-image diffusion model.
  - No fine-tuning required.
- Takes as input any image (could be strokes) or modified image.
- Creates an output image using the prior (aka output) distribution of the pre-trained text-to-image diffusion model.



Guided Image Synthesis and Editing

Specifics

- Add noise to the input sketch or modified image.
- Run the noisy input image through your pre-trained diffusion model.

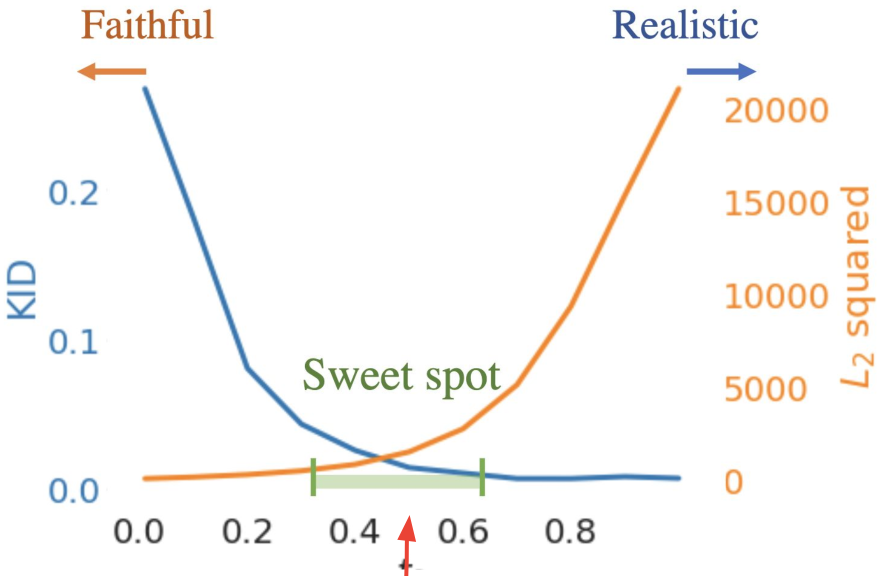


Guided Image Synthesis and Editing

How to add noise?

$$\mathbf{x}^{(g)}(t_0) \sim \mathcal{N}(\mathbf{x}^{(g)}; \sigma^2(t_0)\mathbf{I})$$

Add gaussian noise to the input, with stddev between 0.3 and 0.6

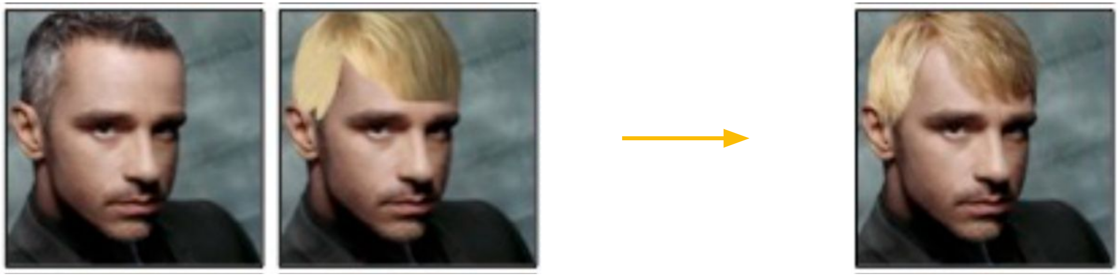


Guided Image Synthesis and Editing

Original Image

Modified Image (Input)

SDEdit Output Image



VideoPoet

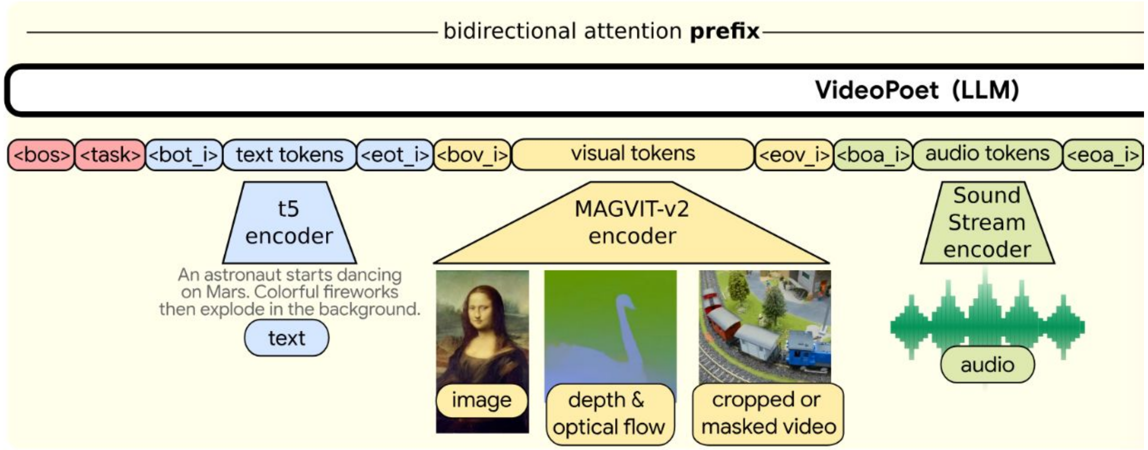
2

Kondratyuk, Dan, et al. "Videopoet: A large language model for zero-shot video generation." arXiv preprint arXiv:2312.14125 (2023).

VideoPoet

Overview

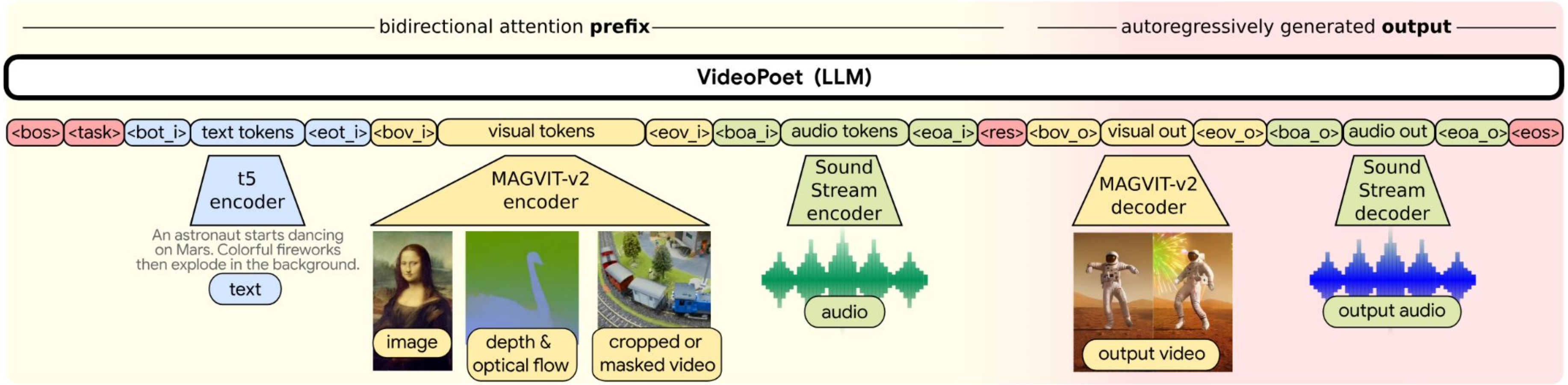
- Uses a Masked Language Model (MLM).
- Tokenizes each modality separately (like in Gato) and uses disjoint vocabularies to represent each.
- Concatenates text, image/video/other, and (optionally) audio, with task and modalities tokens.
- Autoregressively generates video and (optionally) audio tokens.



VideoPoet

Overview

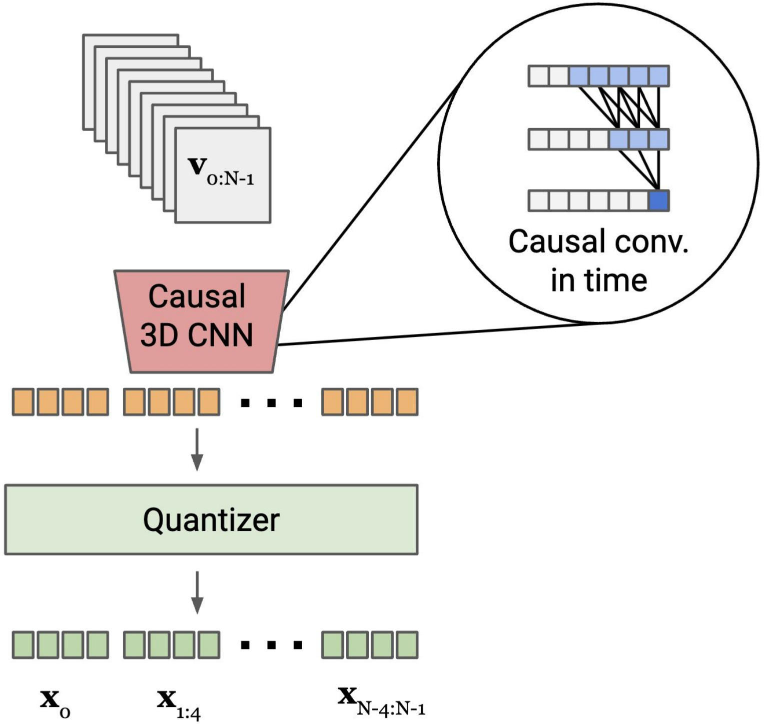
- Uses a Masked Language Model (MLM).
- Tokenizes each modality separately (like in Gato) and uses disjoint vocabularies to represent each.
- Concatenates text, image/video/other, and (optionally) audio, with task and modalities tokens.
- Autoregressively generates video and (optionally) audio tokens.



VideoPoet

Video Tokenizer

- Uses causal 3D CNNs
- Regular 3D CNNs attend to $\left\lfloor \frac{k_t - 1}{2} \right\rfloor$ frames before and $\left\lfloor \frac{k_t}{2} \right\rfloor$ frames after.
- Causal 3D CNNs prepend inputs with $k_t - 1$ empty frames.
- The effect is that the 3D CNN kernel only attends to frames before the current frame.
- Uses Lookup Free Quantization (LFQ) to quantize.



WALT

3

Gupta, Agrim, et al. "Photorealistic video generation with diffusion models." arXiv preprint arXiv:2312.06662 (2023).

WALT

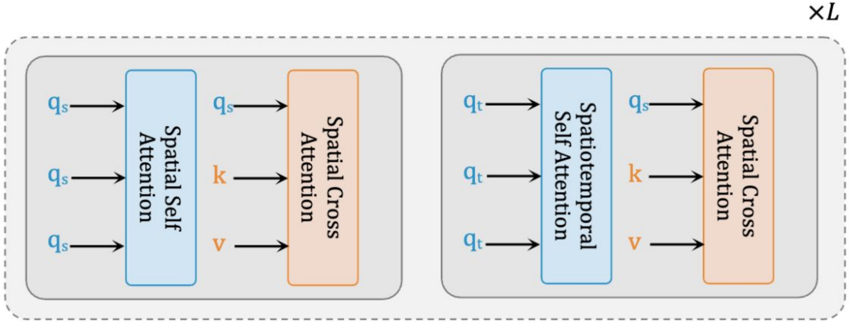
Overview

- Uses the same video tokenizer as VideoPoet (MagViT-v2) but without the LFQ quantization layer.
- Latent Diffusion Model with a specialized transformer that interleaves

WALT

Overview

- Uses the same video tokenizer as VideoPoet (MagViT-v2) but without the quantization layer.
- Latent Diffusion Model with a spatial and spatiotemporal windowed attention.
    - Spatial windowed (SW) attention allows each frame to attend only to the tokens for that frame.
    - Spatiotemporal windowed (STW) attention allows tokens in the video clip to attend to each other.
    - Found that SW and STW achieve similar performance to full self-attention but faster.



$q_s \in \mathbb{R}^{(B.(1+t))\times 1\times h_p\times w_p\times C}$

$q_t \in \mathbb{R}^{(B.W_n)\times(1+t)\times h'_p\times w'_p\times c}$

$k = v = \text{concat}[q_s, \text{t5\_xl (text)}]$

st window	FVD↓	IS↑	sps
5 × 4 × 4	56.9	87.3	2.24
5 × 8 × 8	59.6	87.4	2.00
5 × 16 × 16	55.3	87.4	1.75
full self attn.	59.9	87.8	1.20

WALT

Training

- Used a dataset of ~970M text-image pairs and ~89M text-video pairs from the public internet and internal sources.
- Trained a 3B parameter model and two cascading super-resolution models (1.3B and 419M parameters) to generate 512x896 videos.

Autoregressive Generation

- Generate long videos by using some previously generated frames instead of noise.
- To enable this, the model is conditioned on some past frames during training.

Method	IS (↑)	FVD (↓)
CogVideo (Chinese) [37]	23.6	751.3
CogVideo (English) [37]	25.3	701.6
MagicVideo [88]	-	699.0
Make-A-Video [66]	33.0	367.2
Video LDM [4]	33.5	550.6
PYoCo [24]	**47.8**	355.2
W.A.L.T *(Ours)* 419M (video only)	26.8	598.8
W.A.L.T *(Ours)* 419M (video + image)	31.7	344.5
W.A.L.T *(Ours)* 3B (video + image)	35.1	**258.1**

Table 5. **UCF-101 text-to-video generation**. Joint training on image and video datasets in conjunction with scaling the model parameters is essential for high quality video generation.

Lumiere

Bar-Tal, Omer, et al. "Lumiere: A space-time diffusion model for video generation." arXiv preprint arXiv:2401.12945 (2024).

4

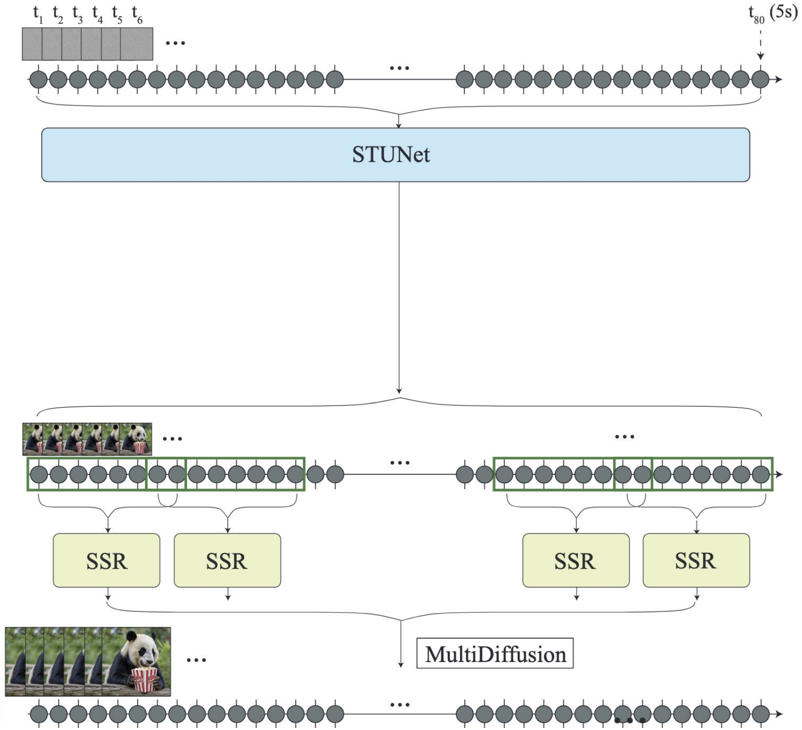
Lumiere

Overview

- Pixel Diffusion Model.
- Space-Time UNet (STUNet) receives noisy video input and outputs denoised video.
- Spatial Super Resolution (SSR) model is used to upsample the output denoised video.

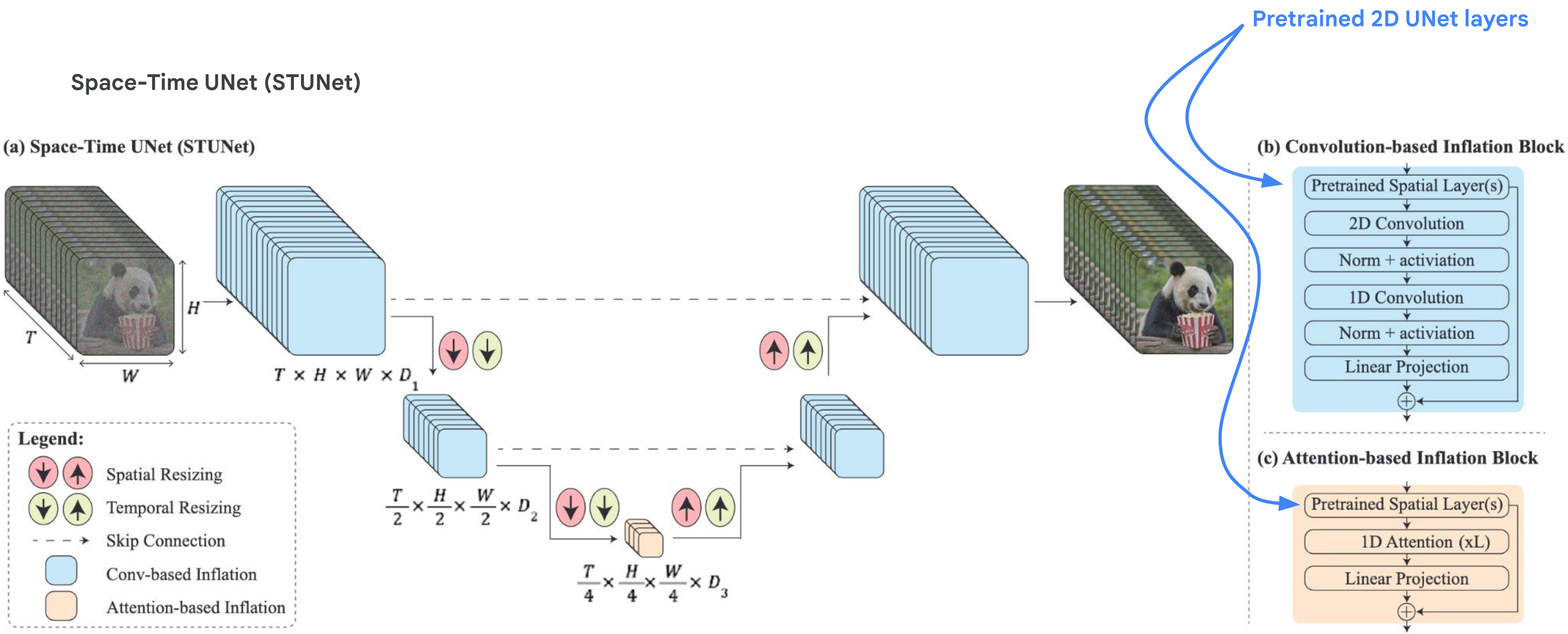
MultiDiffusion

- Due to memory constraints they can only apply SSR to short segments of the video.
- To avoid artifacts they break the video up into overlapping segments (2 frames overlap) and average the denoised predictions in the overlapped frames.



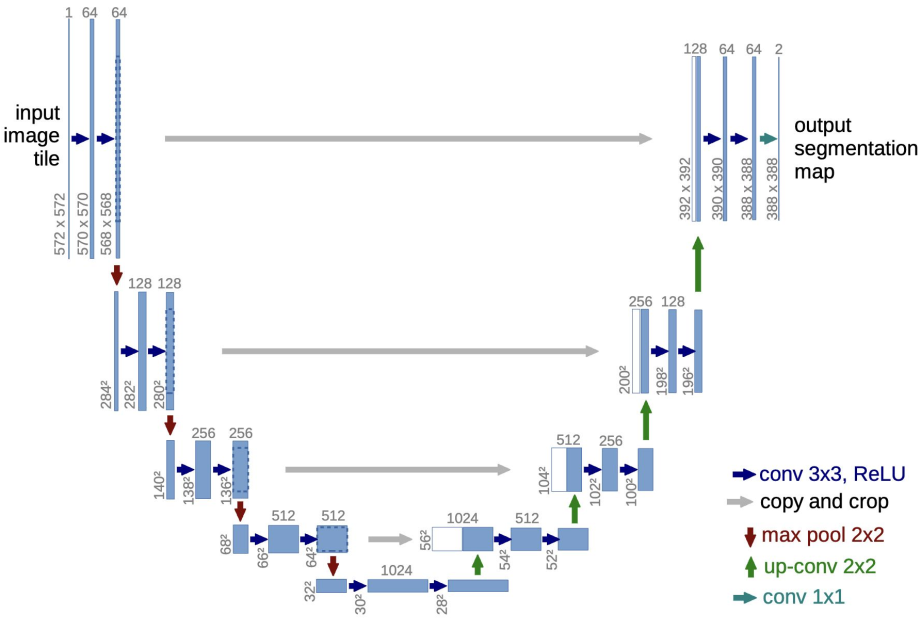
Lumiere

Space-Time UNet (STUNet)



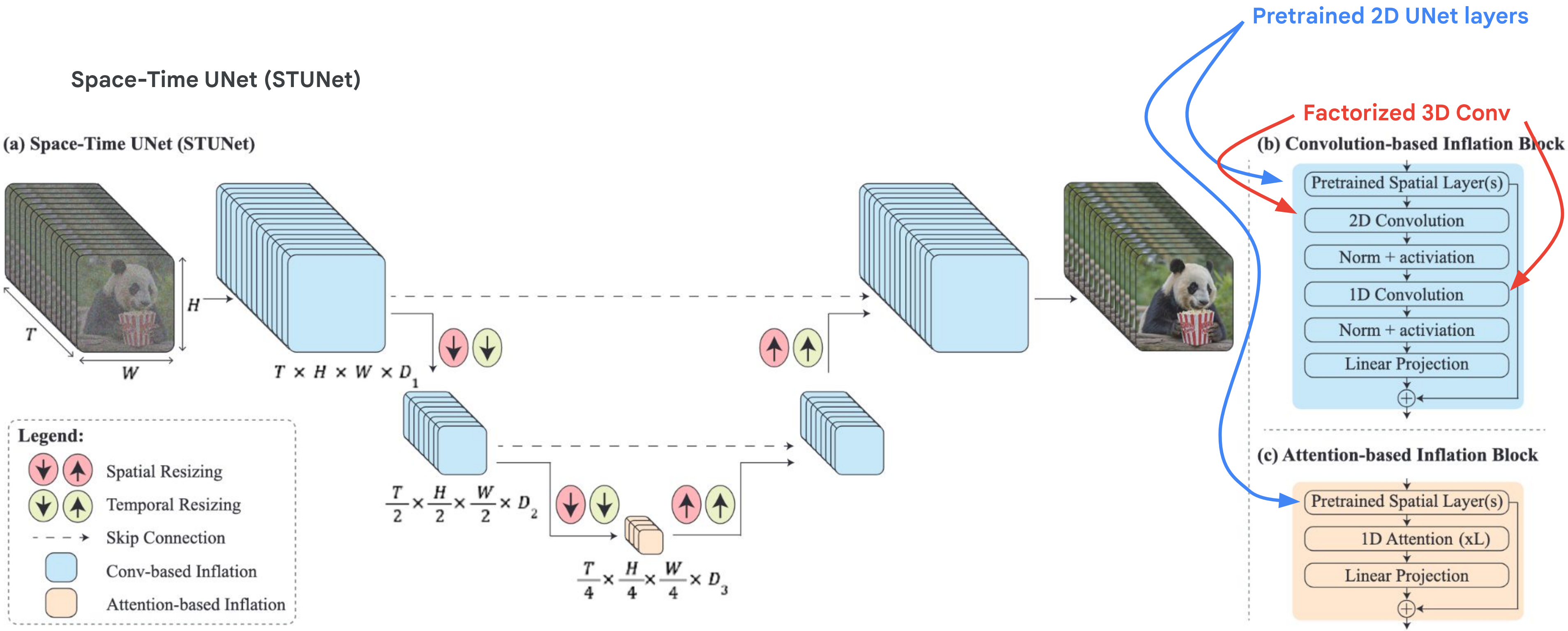
Lumiere

~~Space-Time UNet (STUNet)~~ UNet



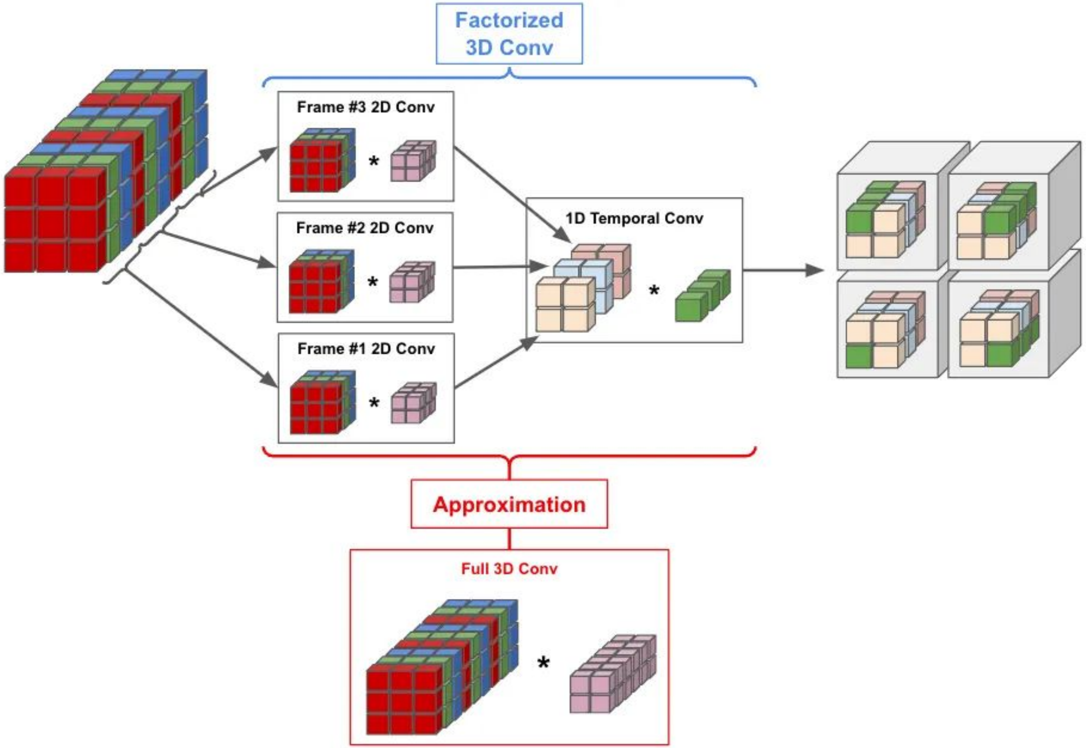
Lumiere

Space-Time UNet (STUNet)



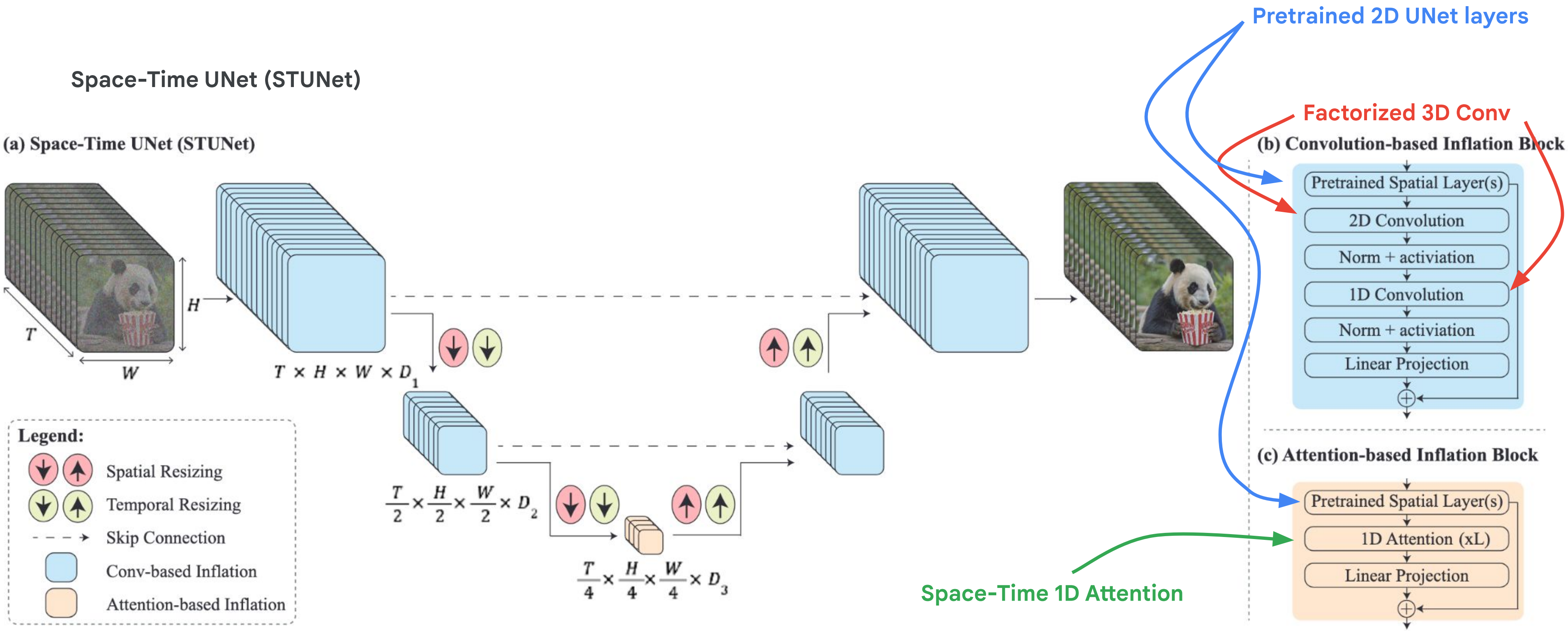
Lumiere

Factorized 3D Conv



Lumiere

Space-Time UNet (STUNet)



Lumiere

Video Editing with SDEdit

Source Video

"Made of wooden blocks"

"Origami folded paper art"

"Made of colorful toy bricks"

"Made of flowers"

Sora

https://openai.com/index/sora/
https://openai.com/index/video-generation-models-as-world-simulators/

5

Sora

Dissecting the blog post

- 1 minute videos at 1080p
- Latent Diffusion Model
    - Specifically a Diffusion Transformer
- VAE to encode/decode videos
- NaViT
- Prompt expansion
- SDEdit

Video compression network

We train a network that reduces the dimensionality of visual data.[20] This network takes raw video as input and outputs a latent representation that is compressed both temporally and spatially. Sora is trained on and subsequently generates videos within this compressed latent space. We also train a corresponding decoder model that maps generated latents back to pixel space.

Scaling transformers for video generation

Sora is a diffusion model[21, 22, 23, 24, 25]; given input noisy patches (and conditioning information like text prompts), it's trained to predict the original "clean" patches. Importantly, Sora is a diffusion *transformer*.[26] Transformers have demonstrated remarkable scaling properties across a variety of domains, including language modeling,[13, 14] computer vision,[15, 16, 17, 18] and image generation.[27, 28, 29]

Variable durations, resolutions, aspect ratios

Past approaches to image and video generation typically resize, crop or trim videos to a standard size—e.g., 4 second videos at 256×256 resolution. We find that instead training on data at its native size provides several benefits.

Similar to DALL·E 3, we also leverage GPT to turn short user prompts into longer detailed captions that are sent to the video model. This enables Sora to generate high quality videos that accurately follow user prompts.

Video-to-video editing

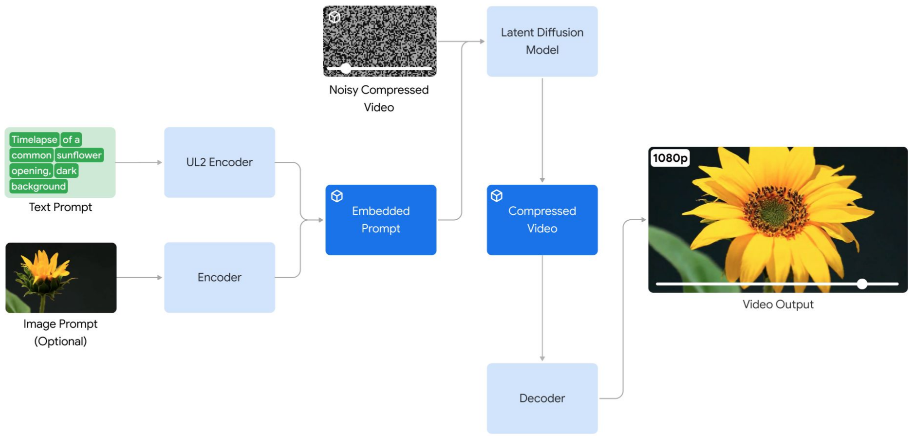
Diffusion models have enabled a plethora of methods for editing images and videos from text prompts. Below we apply one of these methods, SDEdit,[32] to Sora. This technique enables Sora to transform the styles and environments of input videos zero-shot.

Veo

6

https://deepmind.google/technologies/veo

Veo


Timelapse of a common sunflower opening, dark background
Text Prompt
UL2 Encoder
Image Prompt
(Optional)
Encoder
Noisy Compressed
Video
Embedded
Prompt
Latent Diffusion
Model
Compressed
Video
Decoder
1080p
Video Output

Kling

https://kling.kuaishou.com/

6

Kling

- Generates 1080p videos up to 2 minutes long.

Thanks!