VISION AND LANGUAGE MODELLING

Alessandro Bondielli

alessandro.bondielli@unipi.it

Assistant Professor

Dept. of Computer Science

& Dept. Of Philology, Literature and Linguistics

Today's Menu

- An overview of **current approaches to multimodality**, focusing on Vision (images) and Language (texts)
  - Discriminative and generative models
- A few technical aspects, but we can **keep it relatively high-level**
  - Otherwise we would need a far more than a couple hours…
  - Who's interested can find links in througout the presentation to research papers and more in-depth reads

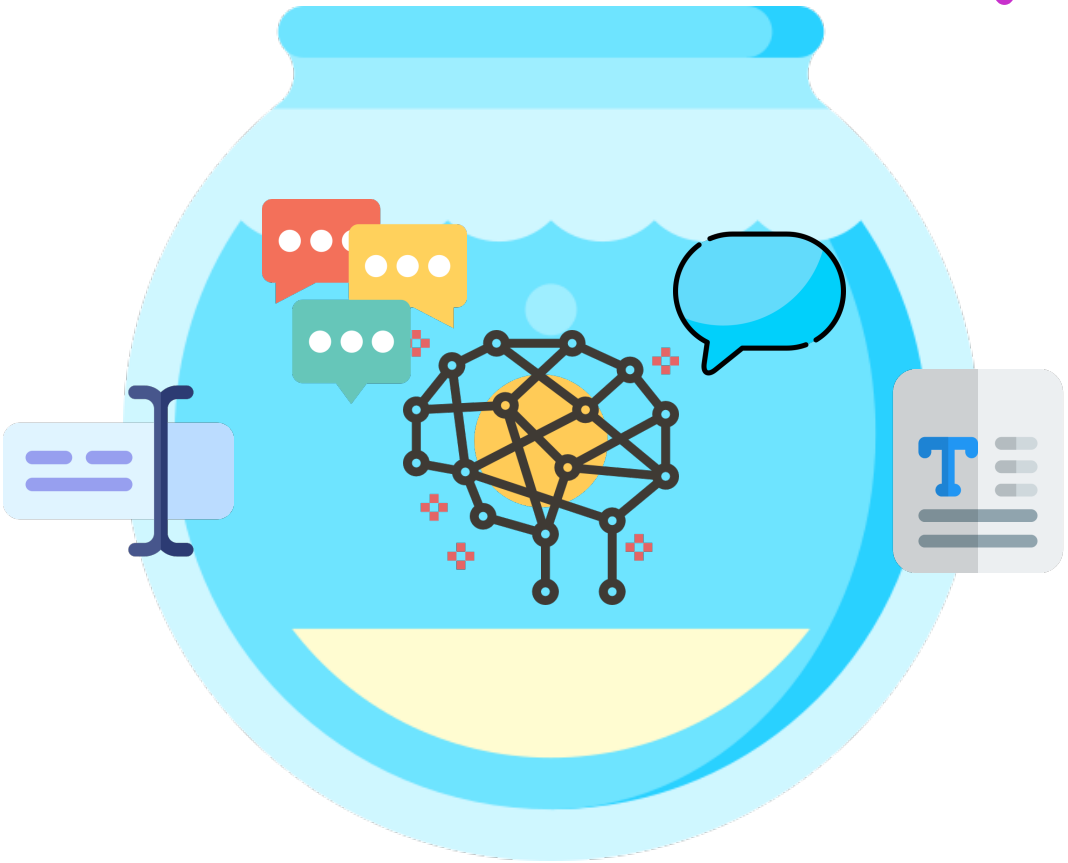
The slides are available here:

Introduction – Beyond NLP

(Large) Language Models are extremely powerful systems that can **interpret and produce coherent texts**

Nowadays we can interact with LLMs simply via **prompts**, often in a chat format and **solve many tasks**

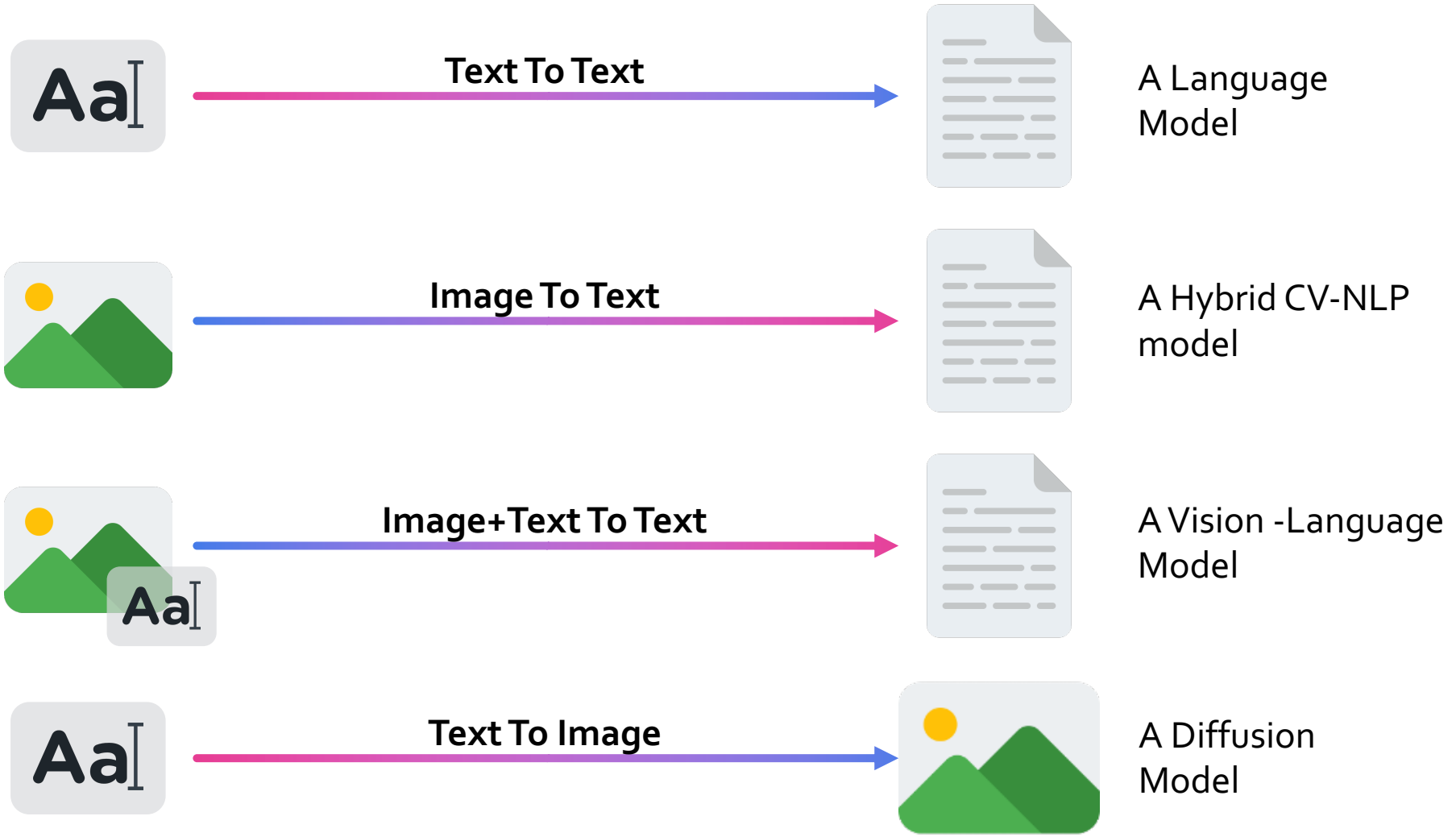
BUT: They are still limited by the boundaries of text, a single ***modality***



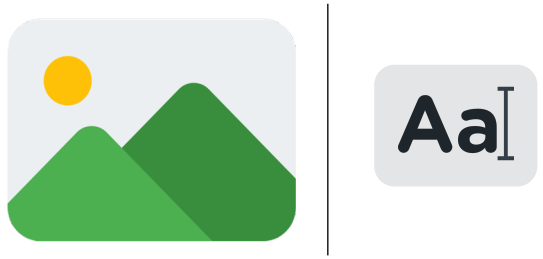
More than one modality

- *A lot* of recent interest on multi-modality
  - We are multi-modal creatures, after all
  - Many problems we are currently trying to face in AI, as we try to get to AGI, are inherently multi-modal
- How do we go from one modality to the other? **How do we combine them toghether?**

NLP	Multimodal
• Sequence classification • Token classification • Sequence-to-sequence • Generative models • IE • QA • …	• Multimodal Anything-NLP • Multimodal reasoning • Text-to-image and viceversa • …

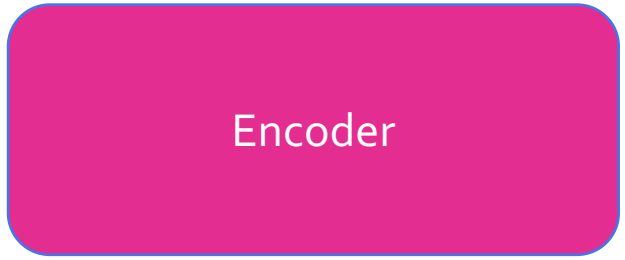

Text To Text
A Language Model
Image To Text
A Hybrid CV-NLP model
Image+Text To Text
A Vision -Language Model
Text To Image
A Diffusion Model

A Transformer-like architecture



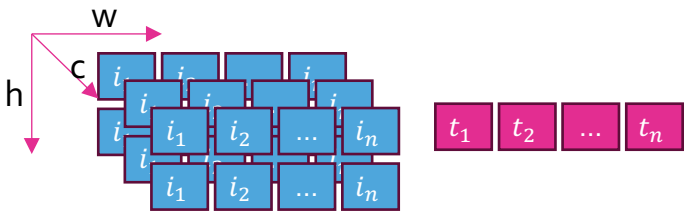
Encoder Only

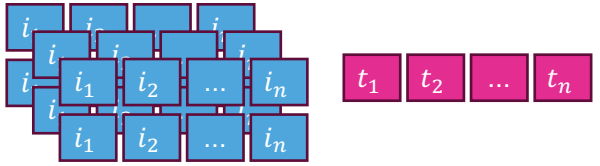
Jointly learn representations of images and texts



Decoder Only

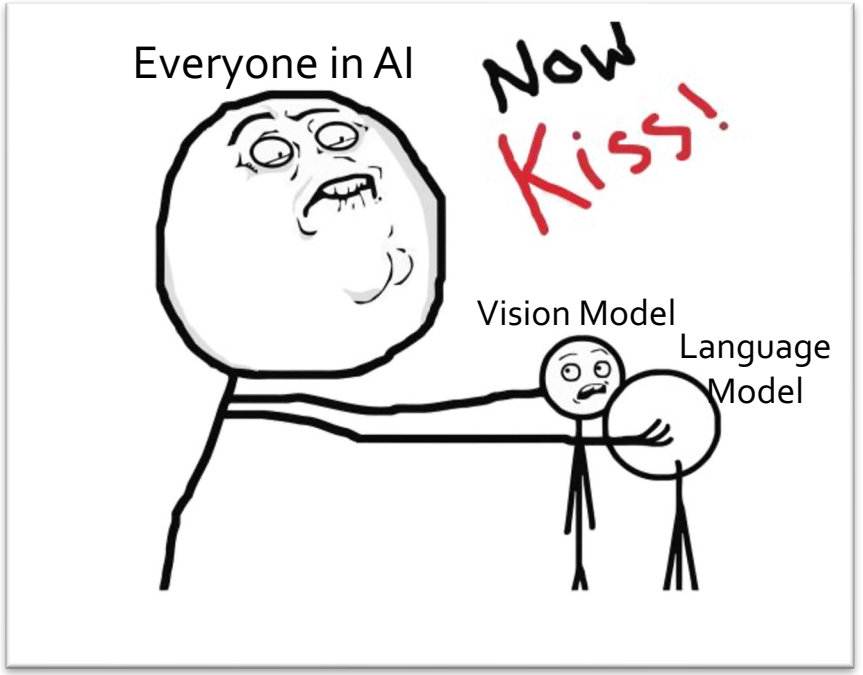
Leverage pre-trained encoders and learn to generate text or images from multimodal inputs





A few intuitions on VL models pre-training

- Vison-Language pre-training can be obtained in a few ways
  - Jointly learn a shared latent space for both texts and images
    - It works surprisingly well, given two powerful enough encoders
  - Combine vision and text pipelines
    - Independently, via specialized operations, layers
    - As a concatenation of inputs fed to a model further down the pipeline



Model architectures

2 Two-stream models (dual encoder)

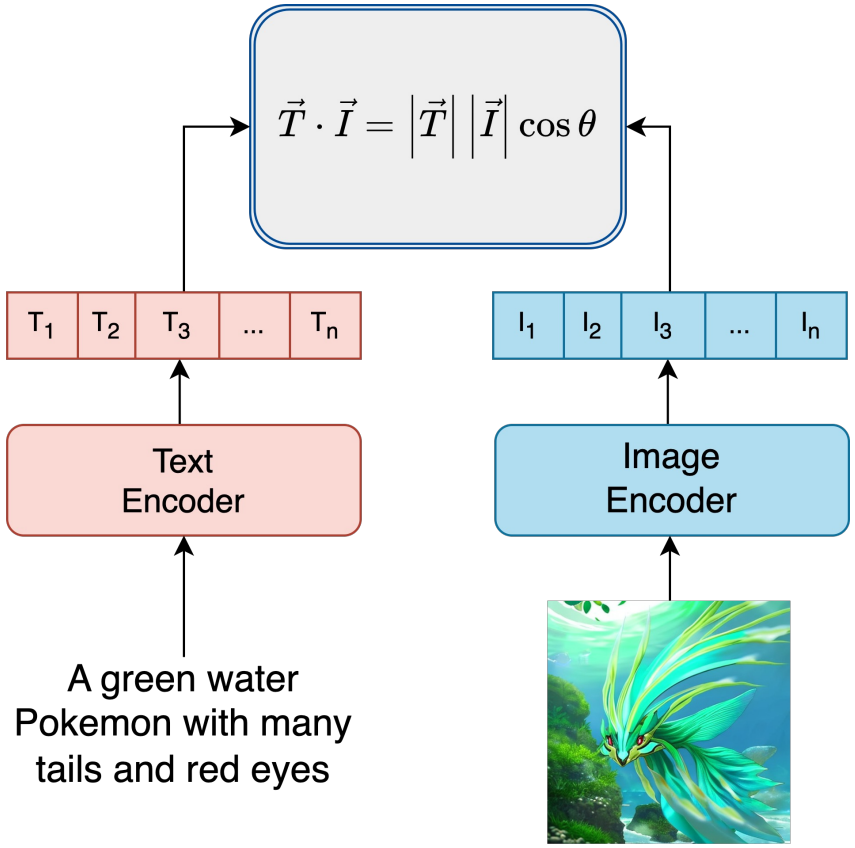
- Images and texts are encoded separately
- Interaction is handled e.g. by maximizing similarity between text and image representations
  - E.g., via cosine similarity
  - Still able to learn strong visual representation via large-scale pre-training
- Better for image-text matching tasks
  - Image retrieval
  - Open-domain image classification

1 Single-stream models (fusion encoder)

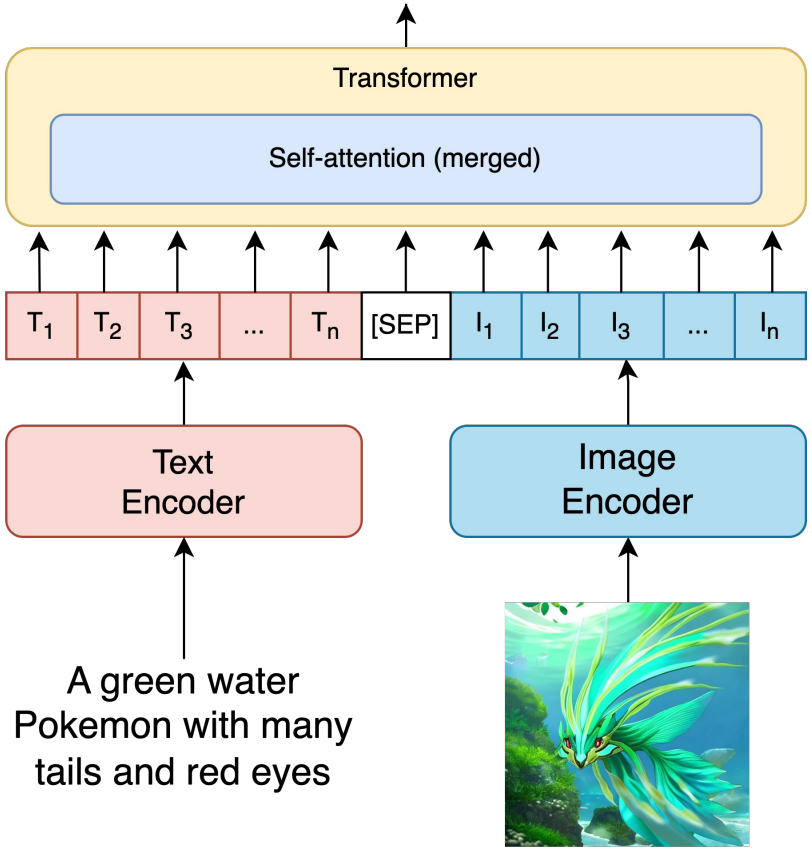
- The text and image encoders are fused togheter with additional transformer layers
- Aimed at modelling deeper interactions between texts and images
- Better for tasks that require both inputs to be modeled together
  - VQA, captioning, visual reasoning etc.

Model architectures

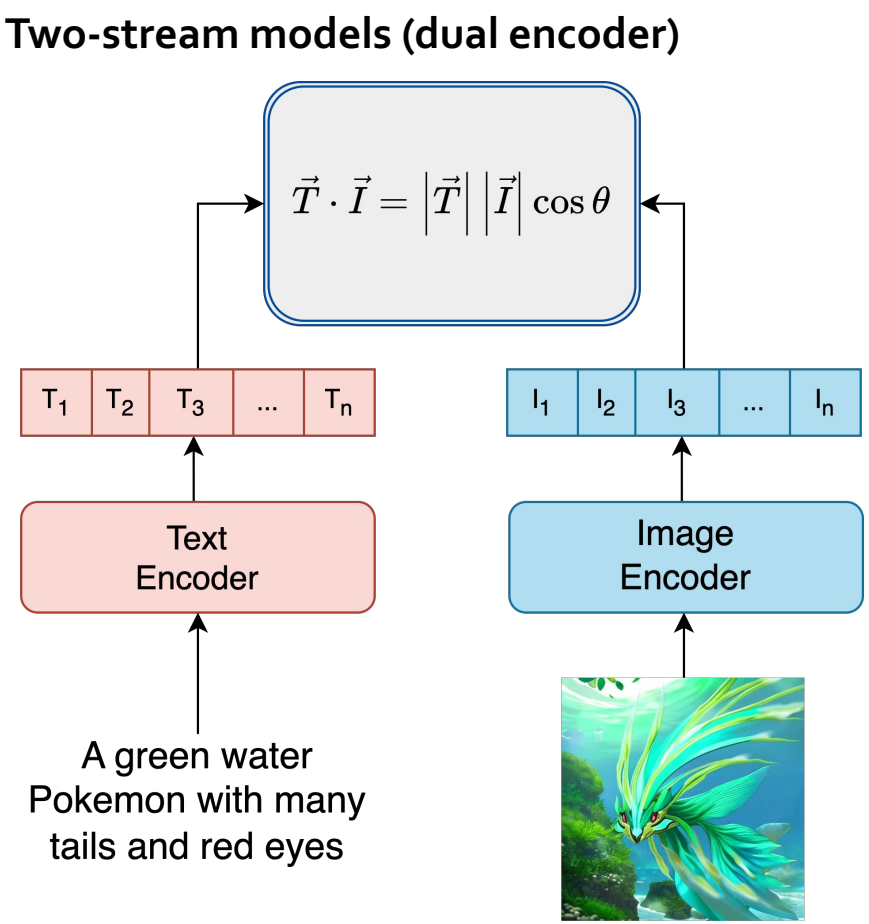
Two-stream models (dual encoder)

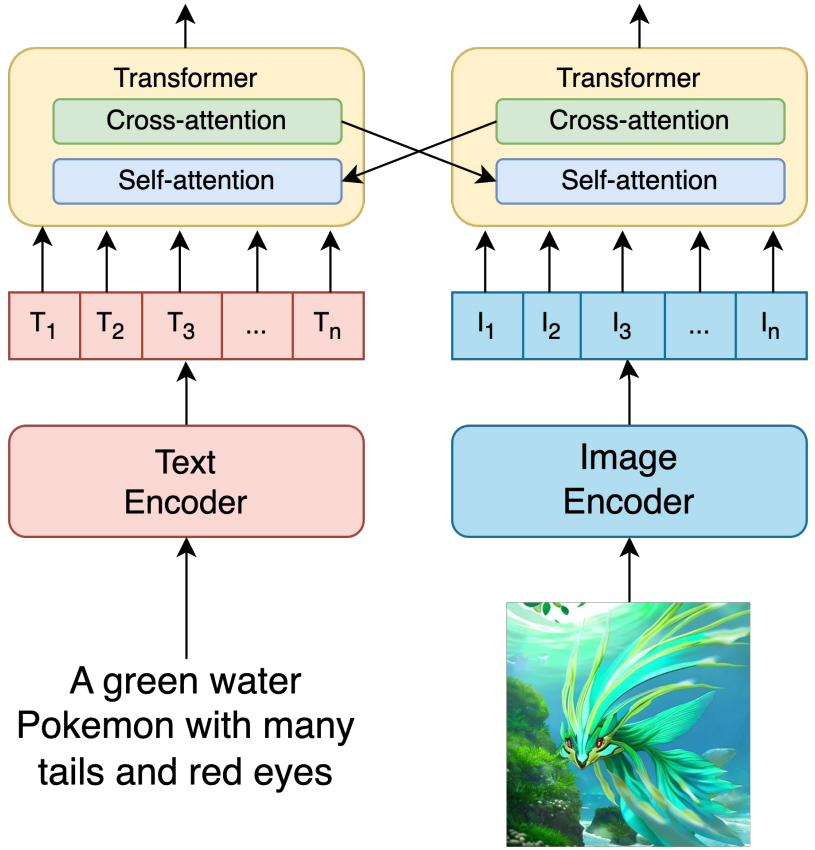


Single-stream models (fusion encoder)

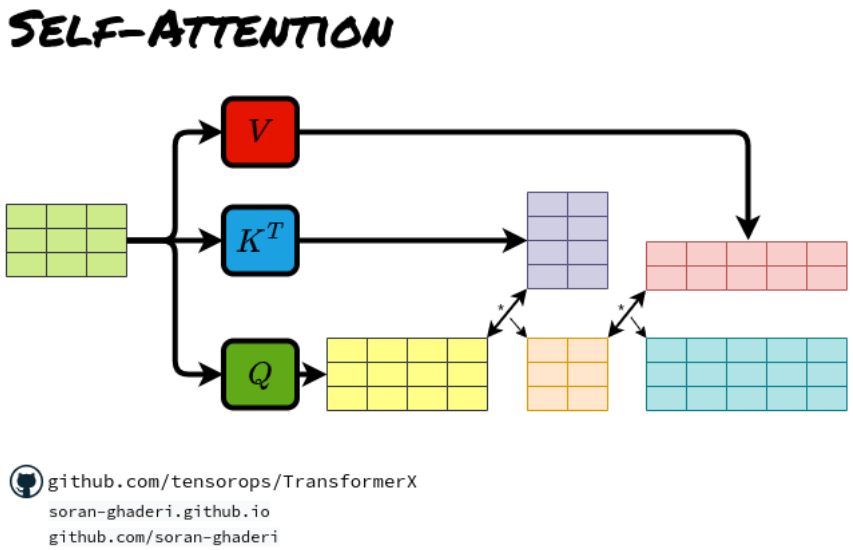


Model architectures

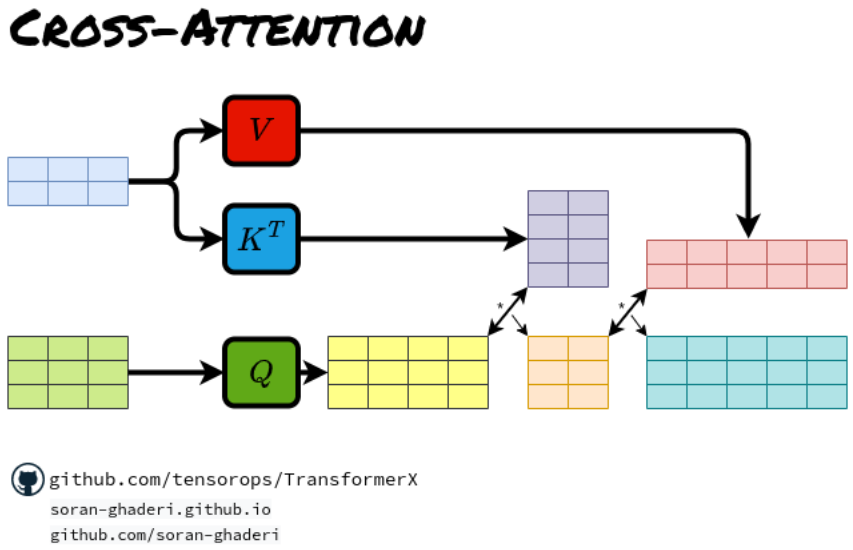




Self- and Cross-Attention



All keys, queries, and values vectors come from the same sequence, in the case of Transformer, the encoder's previous step outputs, allowing each position the encoder to simultaneously attend to all positions in its own previous layer



Cross-attention combines two different embedding sequences with the exact dimensions which derive its queries from one sequence and its keys and values from the other.

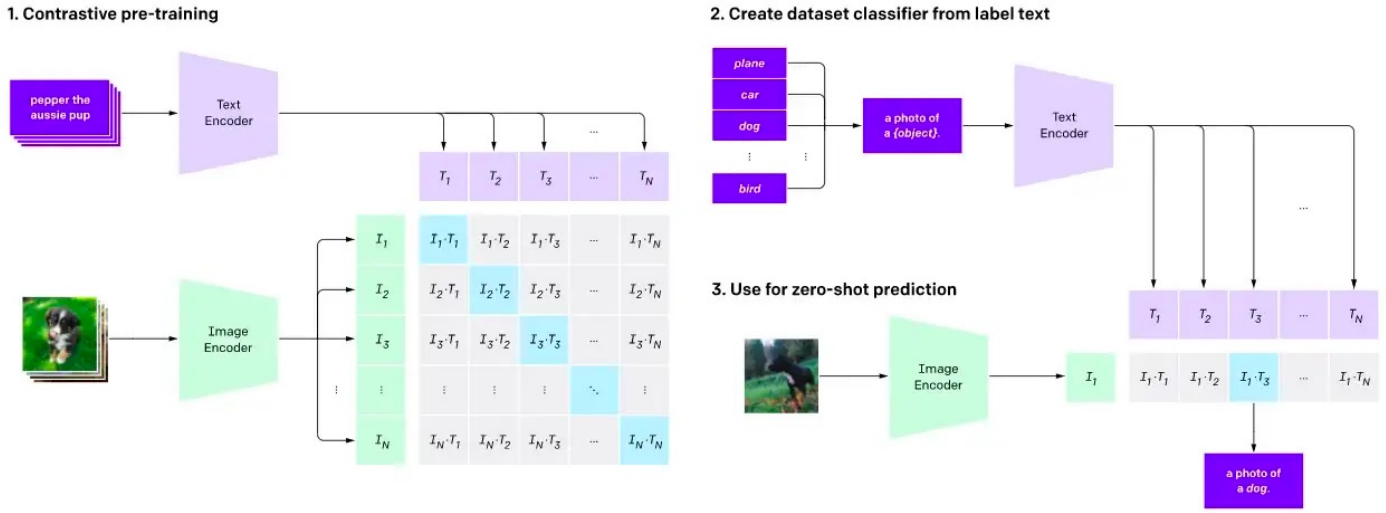
Basically the attention used in encoder-decoder attention for sequence-to-sequence tasks.

Training Strategies

- Few different ways to pre-train VL models
  - Adapt the Language Modelling task to attend also to visual inputs
    - E.g., predict the [MASK]ed token in the caption based on the rest of the caption *and* the image
  - Learn a shared model layer for both texts and images
    - Shared latent representations
    - E.g., training to match images and texts

Image-Text Contrastive Learning (ITC)

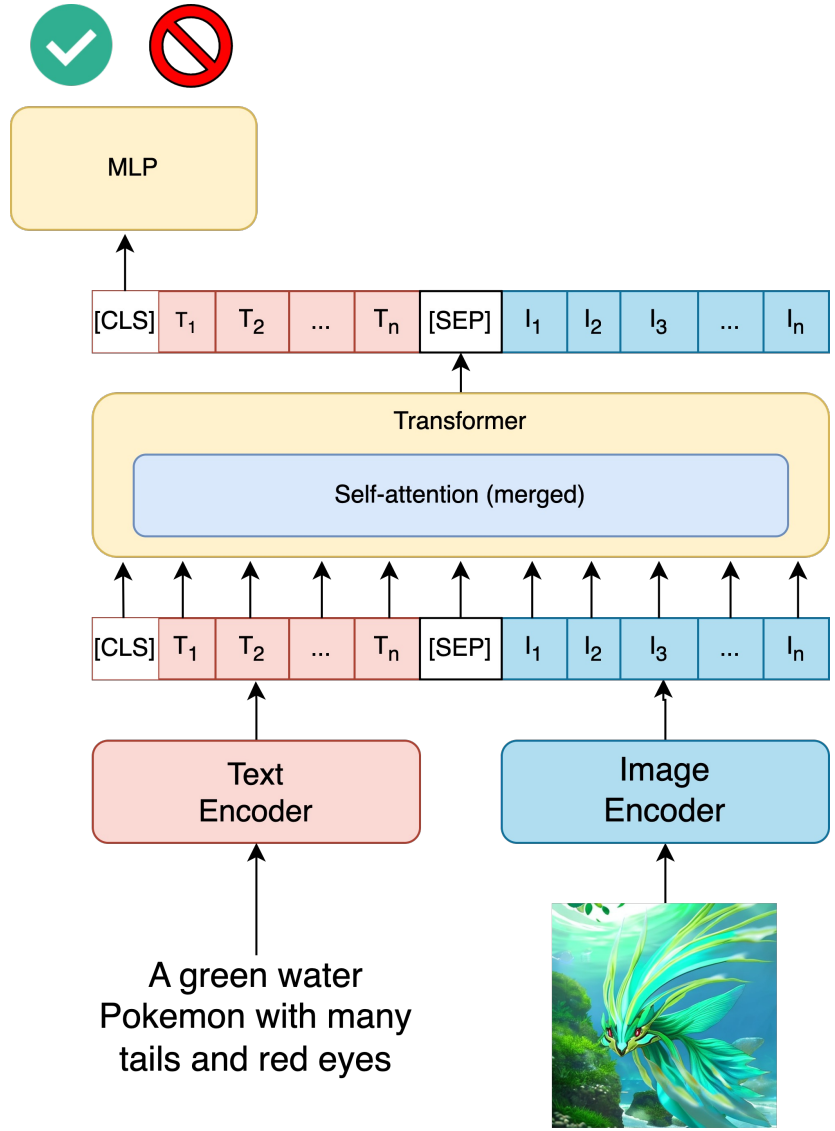
- Common in CV
- **Goal**: map input images and text to the same feature space
  - Minimizing the distance between matching $\{image, text\}$ pairs
- **How?** Contrastive loss on top of the dual encoders
  - Given a batch of N $\{image, text\}$ pairs, the model has to learn to predict the N matched pairs out of all the N^2 possibile ones
  - E.g., for **CLIP**, by minimizing the cosine distance between matching pairs



Available models	Is great for
• CLIP • CLOOB • ALIGN • DeCLIP	• Image-text matching • (Zero shot) image classification

Image Text Matching (ITM)

- Image Text-Matching as binary classification
- In a single stream model:
  - Use the [CLS] token for the entire sequence (text + image)
  - Add a binary classifier with *match* or *non match* outputs
  - Train the classifier on matched and mismatched $\{image, text\}$ pairs
    - With random negative pairs
    - Or stronger negative pairs (better on downstream tasks)

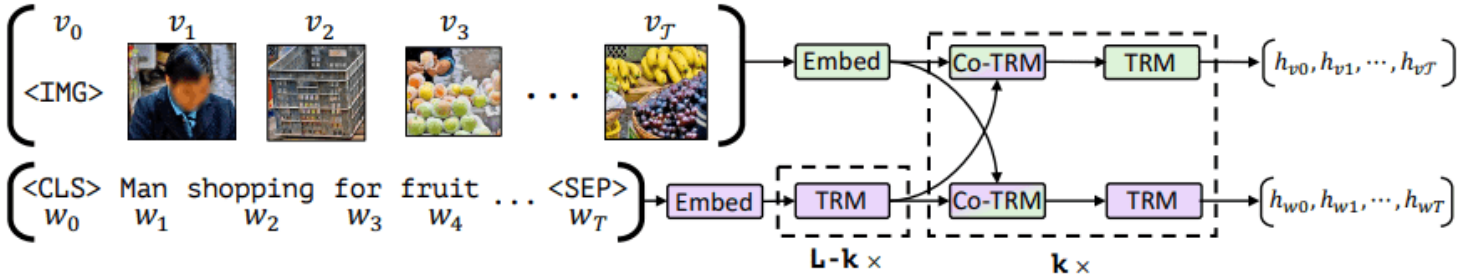


Masked LM / Image-Text Matching

- Masked LM: learning image-grounded representations of texts
- Image-Text Matching :align whole images and captions representations

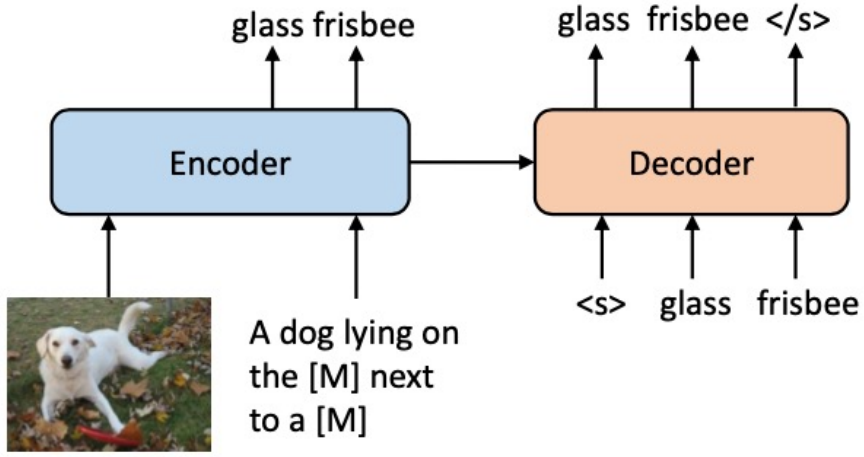
Available models	Is good for
• VisualBERT • FLAVA • ViLBERT • LXMERT • BridgeTower	• Visual QA • Visual commonsense reasoning • Text-based image retrieval • Text-guided object detection

- Train both objectives at the same time during pre-training
  - Use MLM to learn representation of objects or regions of the image (e.g., with bounding boxes)
  - Use ITM to align specific parts of the images with texts with self attention



What about decoders?

- An encoder-only architecture can be enough when dealing with representation-based tasks
  - Use MLP heads to generate the target output (e.g. for classification)
  - Perfect when dealing with non-generative tasks
- Recent literature in LLMs show a trend in favour of generative decoder-only Language Models
  - Larger, more powerful, more versatile generative models
  - Prompting *can* be used to solve both generative and non generative tasks via instruction fine-tuning
  - **Idea**: adapt a generative LLM to model visual inputs as well



Bare minimum requirements for a generative VLM

Text Decoder

- A standard **language model** trained on the next token prediction task

Image Encoder

- A **strong image encoder** model
- Trained with **multi-modal knowledge** (e.g., CLIP)

Multi-modal Projector

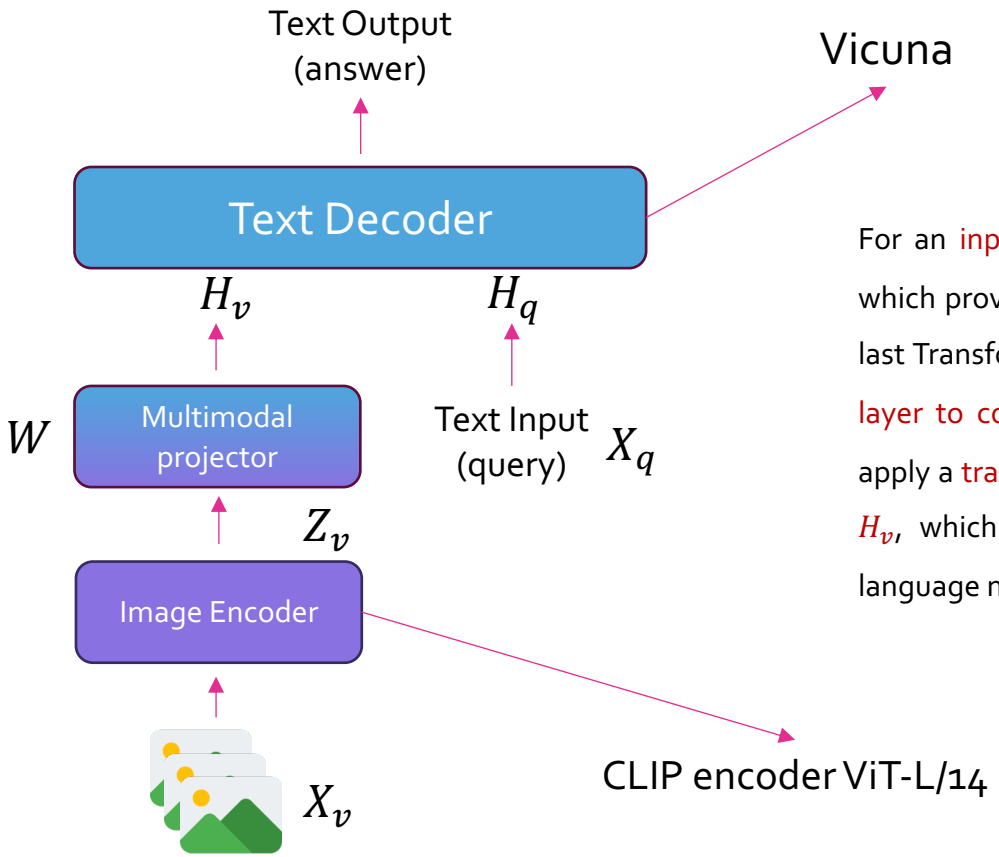
- A way to **pass image features** onto the **text decoder**
- **Input**: image features from the encoder
- **Output**: visual tokens in the embedding space of the decoder
- **Trainable**

The example of LLaVA (Liu et al. 2023)

Visual Instruction Tuning

Haotian Liu[1*], **Chunyuan Li**[2*], **Qingyang Wu**[3], **Yong Jae Lee**[1]
[1]University of Wisconsin–Madison [2]Microsoft Research [3]Columbia University
https://llava-vl.github.io

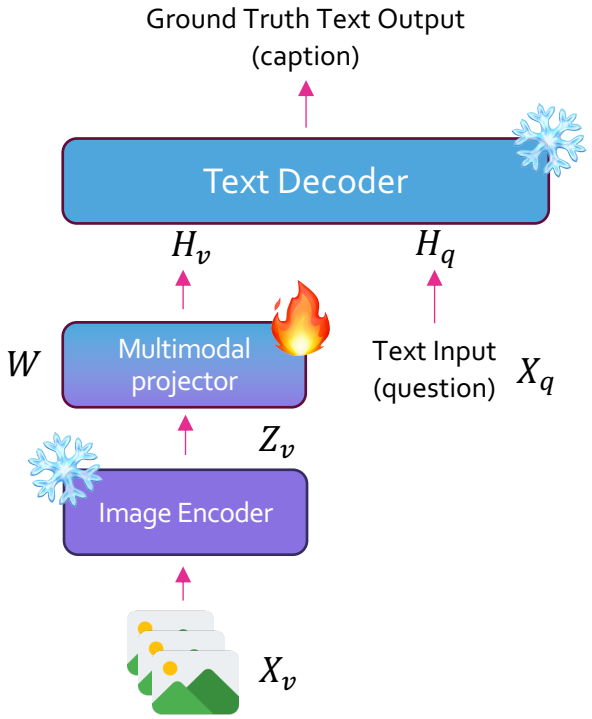
- A simple yet effective way to get an instruction tuned VLM



For an input image X_v, we consider the pre-trained CLIP visual encoder ViT-L/14, which provides the visual feature $Z_v = g(X_v)$. The grid features before and after the last Transformer layer are considered in our experiments. We consider a simple linear layer to connect image features into the word embedding space. Specifically, we apply a trainable projection matrix W to convert Z_v into language embedding tokens H_v, which have the same dimensionality as the word embedding space in the language model:

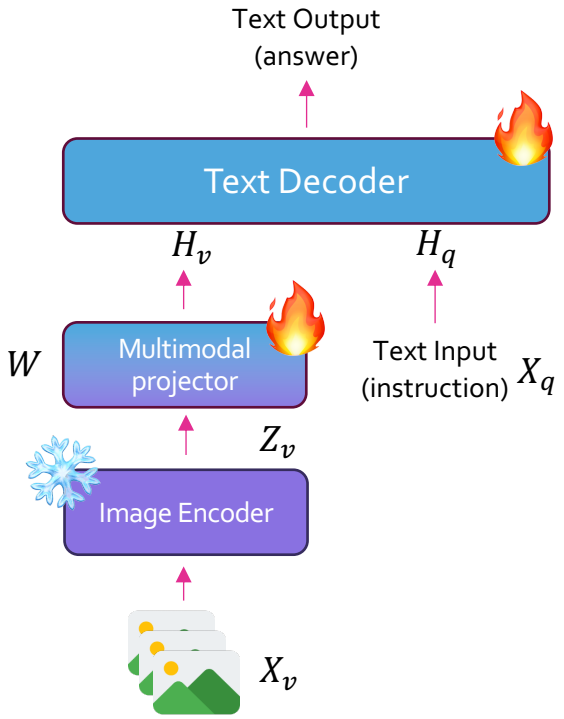
$$\boldsymbol{H_v} = \boldsymbol{W} \cdot \boldsymbol{Z_v} \text{ with } Z_v = g(X_v)$$

The example of LLaVA (Liu et al. 2023)



Pre Training

Train the multimodal projector on a image-caption dataset augmented with questions to generate the caption given the image and the question.



Fine Tuning

Train the projector + the LLM on multimodal instruction data.

LLaVA-Vicuna with Huggingface 🤗

Processor

Same function of the LLM tokenizer but for multimodal inputs:

- Tokenizes the prompt
- Encode & transform (project) images
- Decode generated output

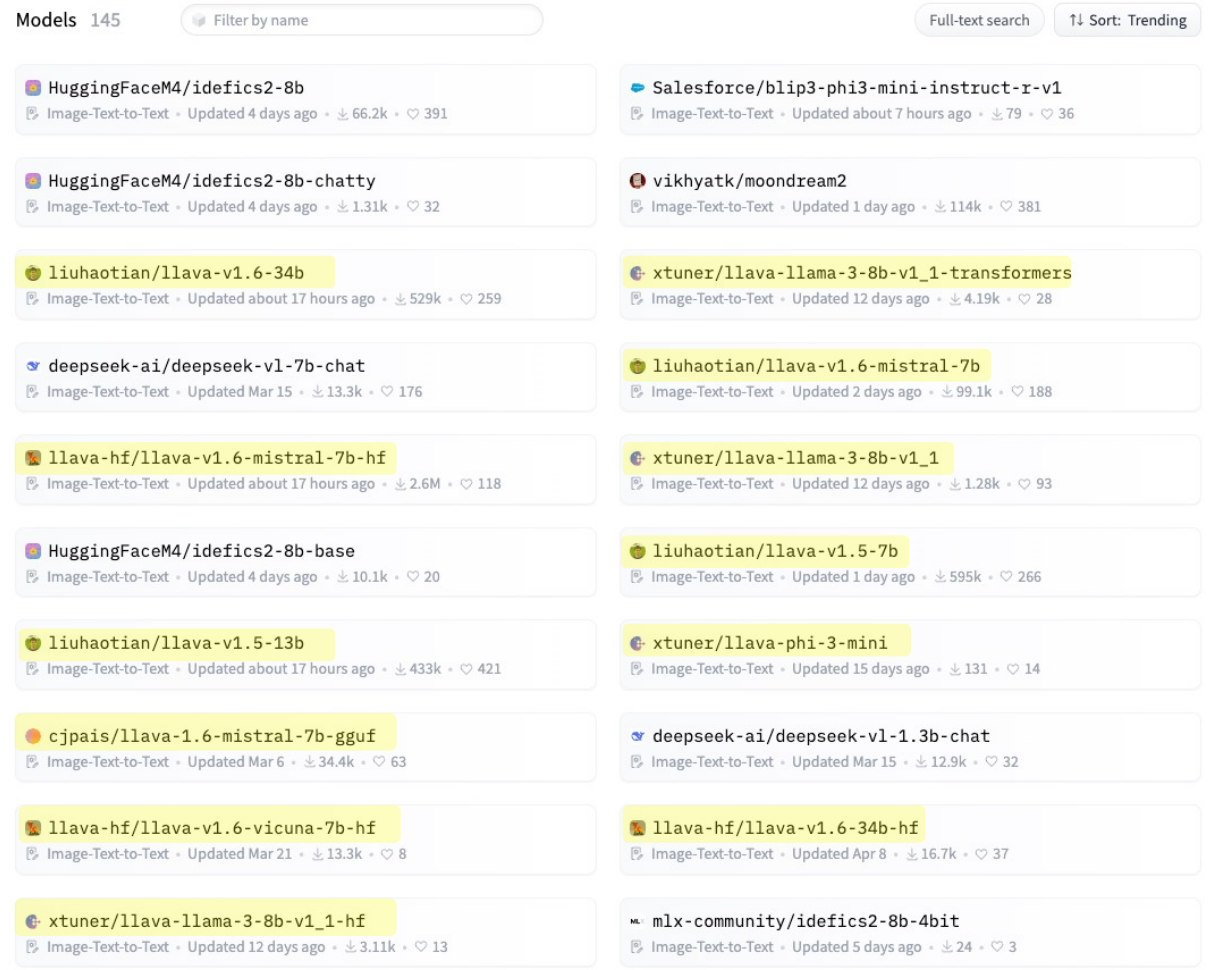
The **<image>** token will be "filled" with $\boldsymbol{H_v} = \boldsymbol{W} \cdot \boldsymbol{Z_v}$

LLaVA-Vicuna.py

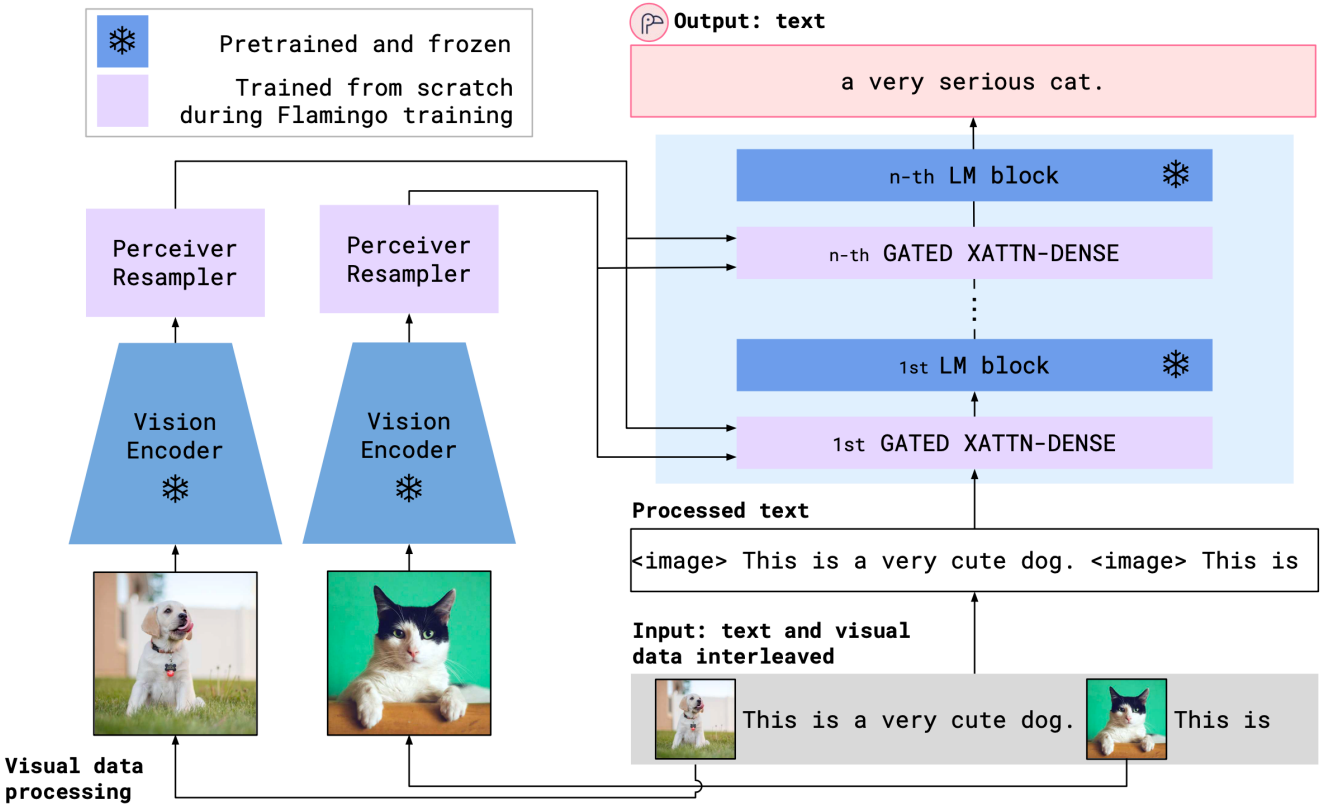
```python
from transformers import LlavaNextProcessor, LlavaNextForConditionalGeneration
import torch
from PIL import Image


processor = LlavaNextProcessor.from_pretrained("llava-v1.6-vicuna-7b-hf")

model = LlavaNextForConditionalGeneration.from_pretrained("llava-v1.6-vicuna-7b-hf",
                                                          torch_dtype=torch.float16,
                                                          low_cpu_mem_usage=True
)

model.to("cuda:0")

# prepare image and text prompt, using the appropriate prompt template
image_path = "/path/to/image.jpg"
image = Image.open(image_path)
prompt = "USER: <image>\nWhat is shown in this image? ASSISTANT:"


inputs = processor(prompt, image, return_tensors="pt").to("cuda:0")

# autoregressively complete prompt
output = model.generate(**inputs, max_new_tokens=100)

print(processor.decode(output[0], skip_special_tokens=True))

```

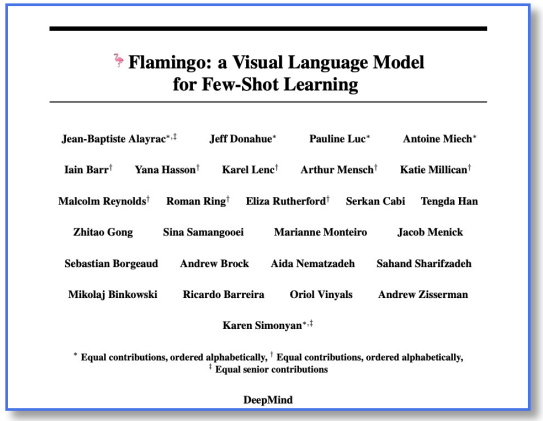
A simple and flexible architecture

- LLaVA is one of the most popular model on Huggingface
- The architecture has only 3 key components
- The **projection layer** needs to be trained from scratch
- The **Image encoder** and **LLM** are already pre-trained and in principle can be swapped with any other model with the same function
    - Many different version with various LLMs are available, e.g. LLaMA, Mistral, etc.



That can be made more complex/refined





Flamingo: a Visual Language Model for Few-Shot Learning

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*,‡

* Equal contributions, ordered alphabetically, † Equal contributions, ordered alphabetically, ‡ Equal senior contributions

DeepMind

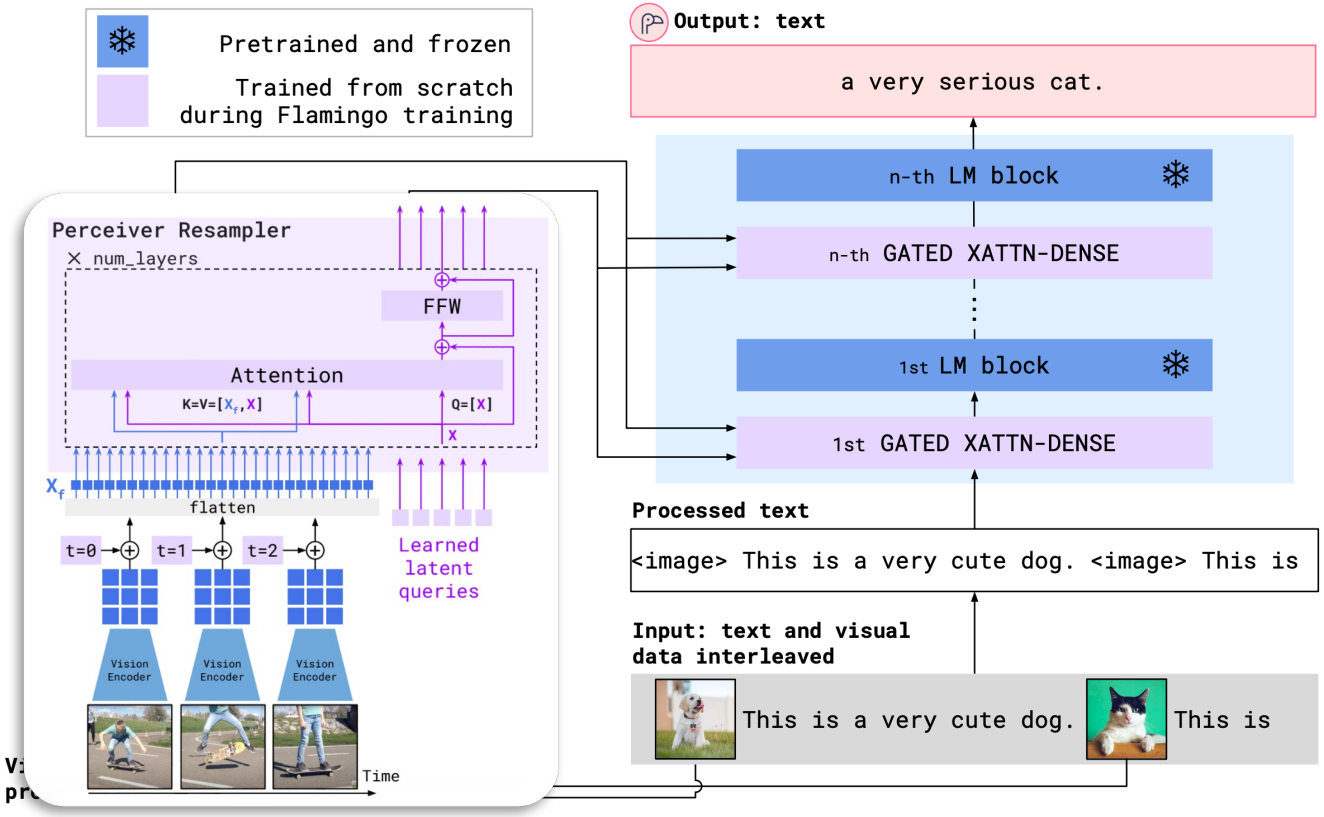
Flamingo (Alayrac et al., 2022) uses a similar idea with a few different design choices That allow the model to process interleaved text & image inputs.

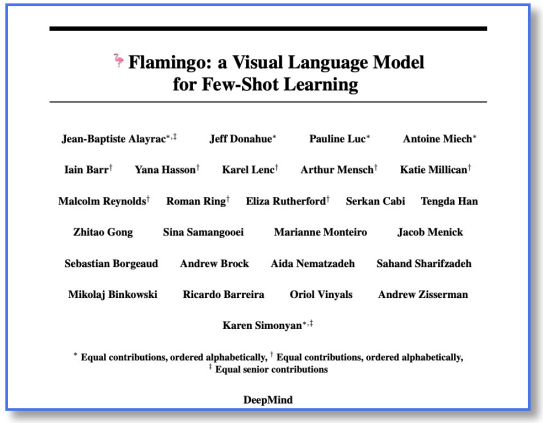
Both the encoder and LLM are pre-trained and kept frozen. The **visual encoder** can encode both images (2D) and videos (3D) and maps them to a 1D representation.

The **Perceiver Resampler** act as the projection layer. It takes as input a variable number of image/video features and map them to a fixed number of visual tokens.

The **GATED XATTN-DENSE** layers are interleaved to the LM layers. They are used to condition the next token generation task of the LM. Its outputs are multiplied by $\tanh(\alpha)$ with α initialized as 0.

That can be made more complex/refined





Flamingo: a Visual Language Model for Few-Shot Learning

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*,‡

* Equal contributions, ordered alphabetically, † Equal contributions, ordered alphabetically, ‡ Equal senior contributions

DeepMind

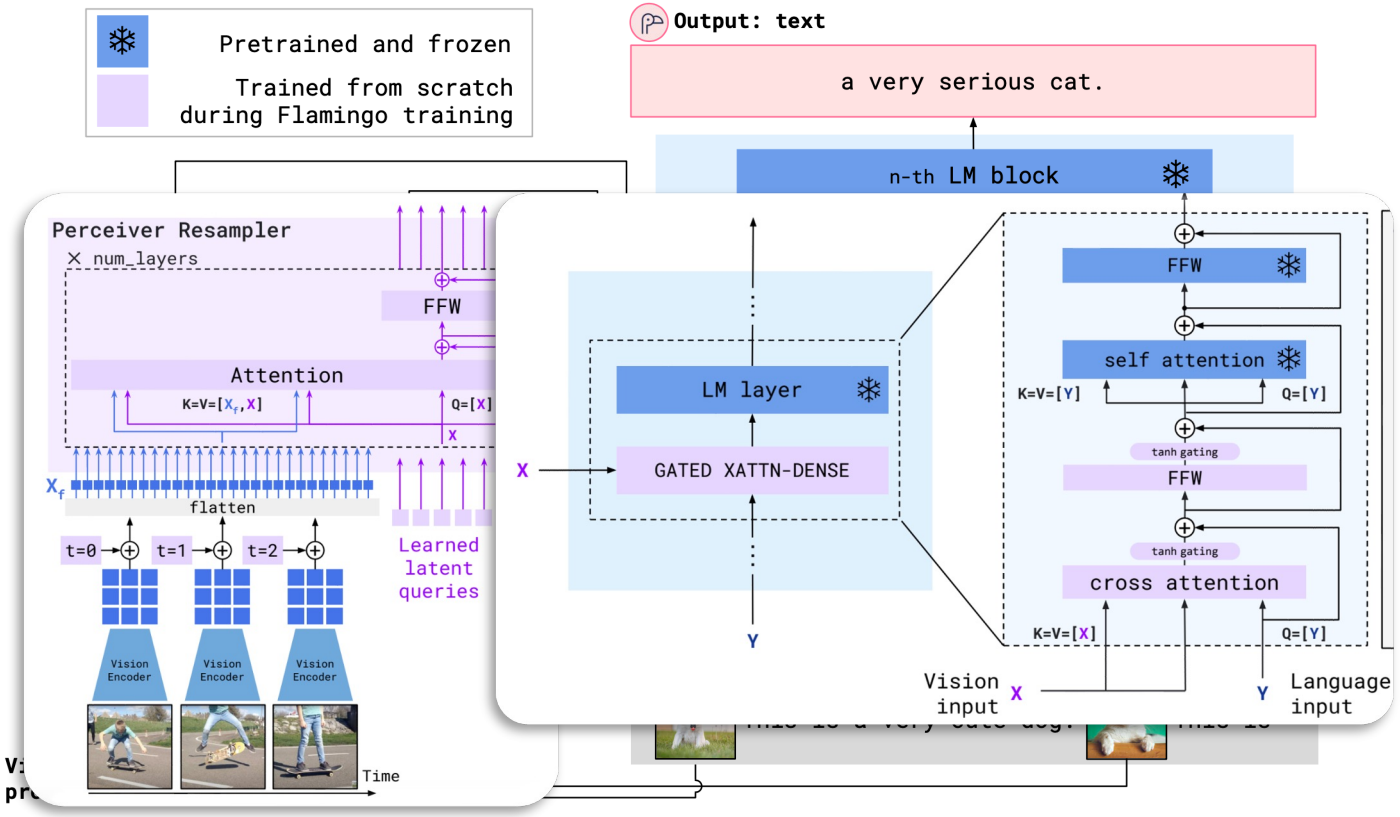
Flamingo (Alayrac et al., 2022) uses a similar idea with a few different design choices That allow the model to process interleaved text & image inputs.

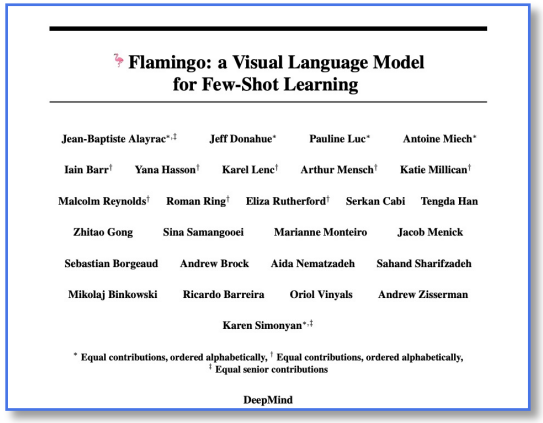
Both the encoder and LLM are pre-trained and kept frozen. The **visual encoder** can encode both images (2D) and videos (3D) and maps them to a 1D representation.

The **Perceiver Resampler** act as the projection layer. It takes as input a variable number of image/video features and map them to a fixed number of visual tokens.

The **GATED XATTN-DENSE** layers are interleaved to the LM layers. They are used to condition the next token generation task of the LM. Its outputs are multiplied by $\tanh(\alpha)$ with α initialized as 0.

That can be made more complex/refined





Flamingo: a Visual Language Model for Few-Shot Learning

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*,‡

* Equal contributions, ordered alphabetically, † Equal contributions, ordered alphabetically, ‡ Equal senior contributions

DeepMind

Flamingo (Alayrac et al., 2022) uses a similar idea with a few different design choices That allow the model to process interleaved text & image inputs.

Both the encoder and LLM are pre-trained and kept frozen. The **visual encoder** can encode both images (2D) and videos (3D) and maps them to a 1D representation.

The **Perceiver Resampler** act as the projection layer. It takes as input a variable number of image/video features and map them to a fixed number of visual tokens.

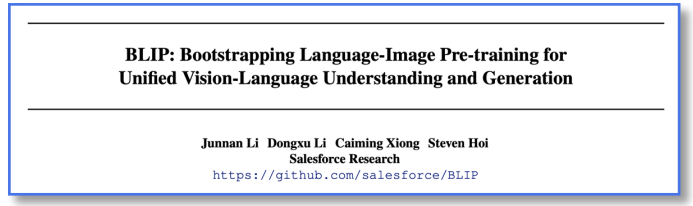
The **GATED XATTN-DENSE** layers are interleaved to the LM layers. They are used to condition the next token generation task of the LM. Its outputs are multiplied by $\tanh(\alpha)$ with α initialized as 0.

Idefics / Idefics2

- **Idefics** is Huggingface's reproduction of Flamingo
  - Image-aware Decoder Enhanced à la Flamingo with Interleaved Cross-attentionS
- Accepts arbitrary sequences of image and text inputs and produces text outputs
- Built solely on publicly available data and models
- Performances on par with the original closed-source model on various image-text benchmarks
- Fine-tuned version of the base models on a mixture of supervised and instruction fine-tuning datasets exist (e.g. idefics-9b-instruct)
- Idefics2 improves upon Idefics1, significantly enhancing capabilities around OCR, document understanding and visual reasoning.

BLIP (Li et al., 2022)

Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation

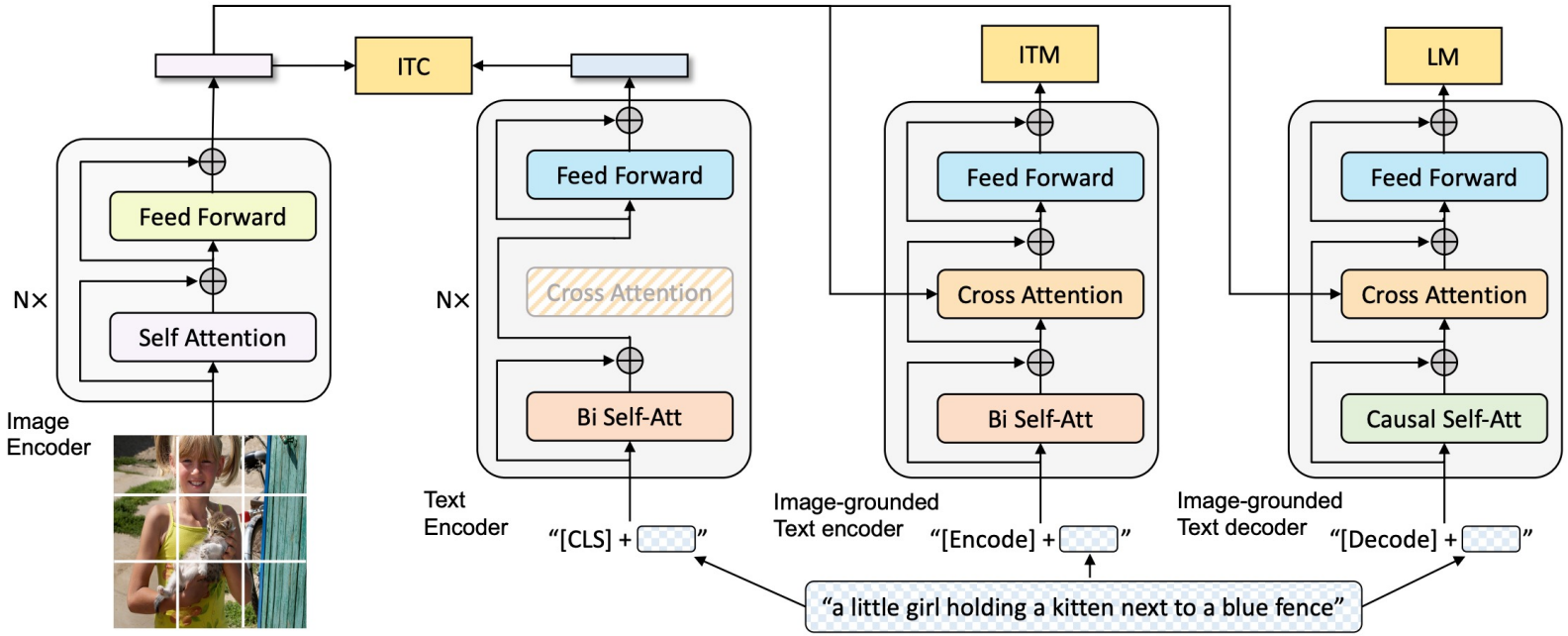

BLIP: Bootstrapping Language-Image Pre-training for
Unified Vision-Language Understanding and Generation

Junnan Li Dongxu Li Caiming Xiong Steven Hoi
Salesforce Research
https://github.com/salesforce/BLIP

Idea: a unified architecture for Vision-Language Pretraining.
Both understanding and generation tasks

How: an encoder-decoder model specifically developed for multi-task pre-training. It is jointly trained on three vision-language objectives: image-text **contrastive learning**, image-text **matching**, and image **conditioned language modeling**.
Each component can operate independently.

PROs: One model can solve many VL tasks
CONs: VL pre-training is very expensive the larger the models get



BLIP2 (Li et al., 2023)

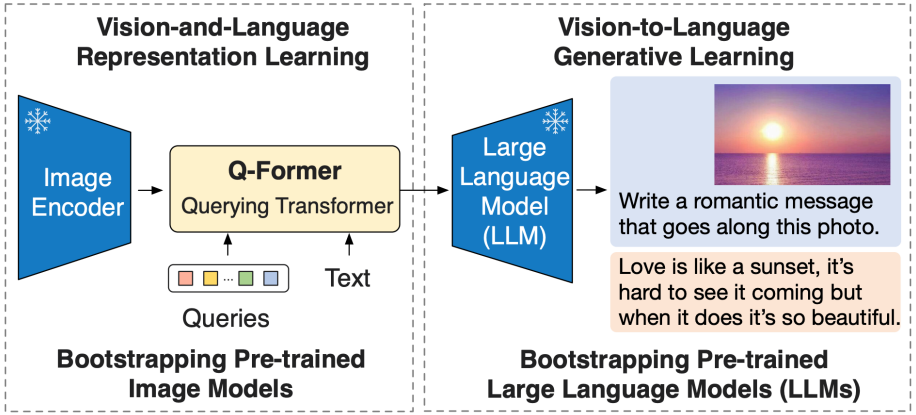
Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models

BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models

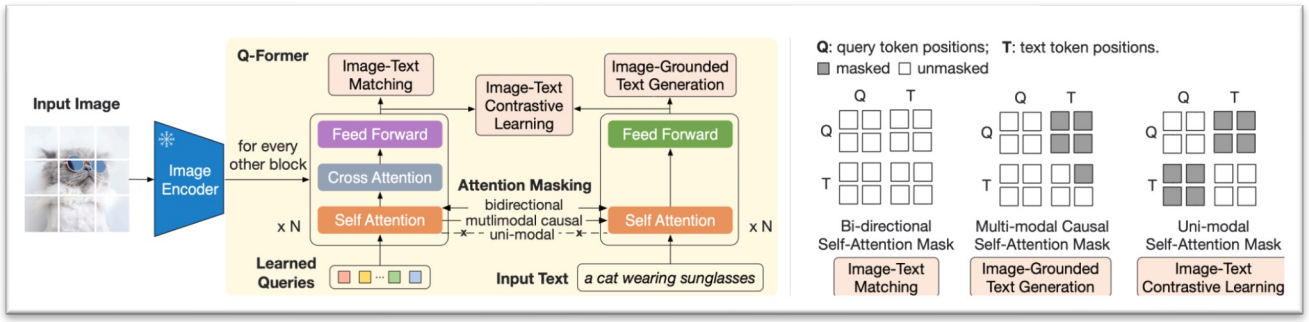
Junnan Li Dongxu Li Silvio Savarese Steven Hoi
Salesforce Research

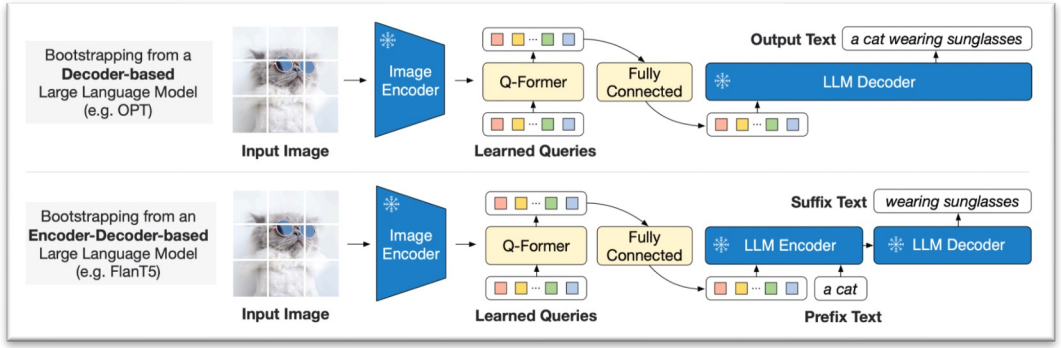
https://github.com/salesforce/LAVIS/tree/main/projects/blip2

Representations are learned similarly to BLIP, by joinlty optimizing 3 different VL tasks with attention, but this time with a much smaller and more efficient LM (i.e., BERT-base) to limit the number of trainable parameters.



Generation is learned by feeding the Q-former outputs to a pre-trained LLM, via the fully connected layer. The LM is then trained with a standar LM loss (for the decoder only) or a prefix LM loss (for the encoder-decoder)



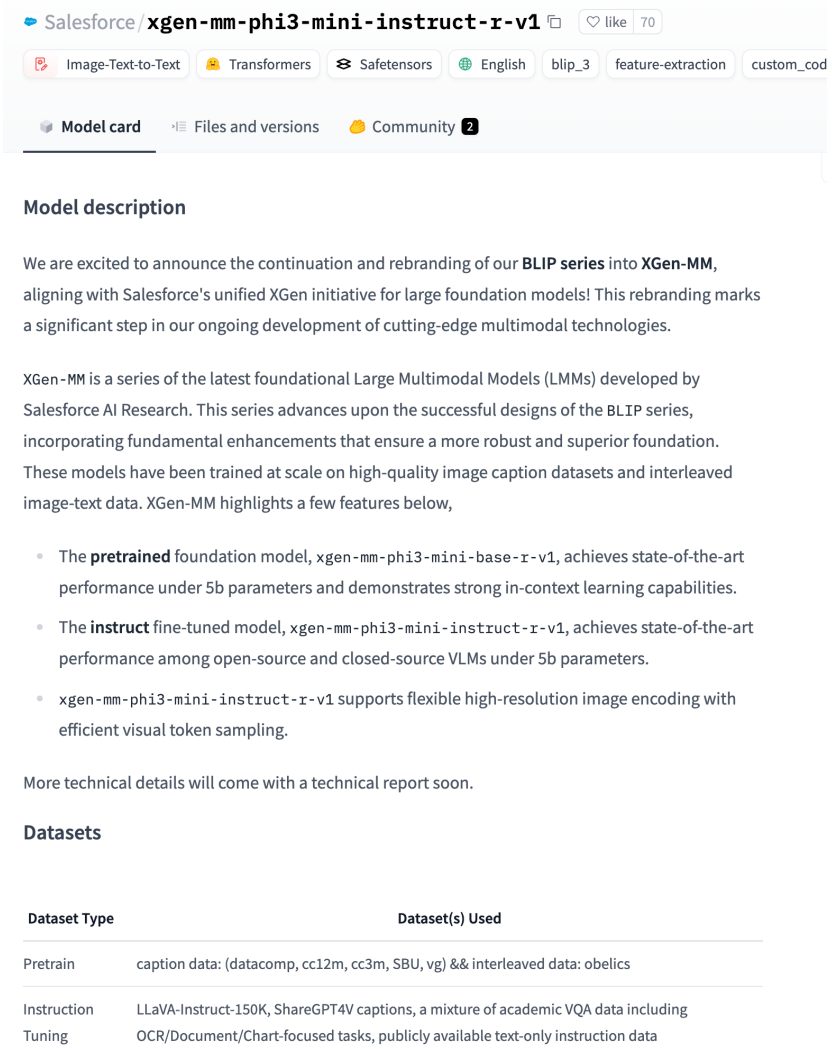


BLIP3?

Is now (from 2 days ago) XGen-MM

Model available on Huggingface but no technical report released yet.

The LLM appears to be Phi-3 Mini.
Phi-3 Mini is a 3.8B parameters, lightweight, state-of-the-art open model by Microsoft.



Salesforce / **xgen-mm-phi3-mini-instruct-r-v1** like 70

Image-Text-to-Text | Transformers | Safetensors | English | blip_3 | feature-extraction | custom_cod

Model card | Files and versions | Community 2

Model description

We are excited to announce the continuation and rebranding of our **BLIP series** into **XGen-MM**, aligning with Salesforce's unified XGen initiative for large foundation models! This rebranding marks a significant step in our ongoing development of cutting-edge multimodal technologies.

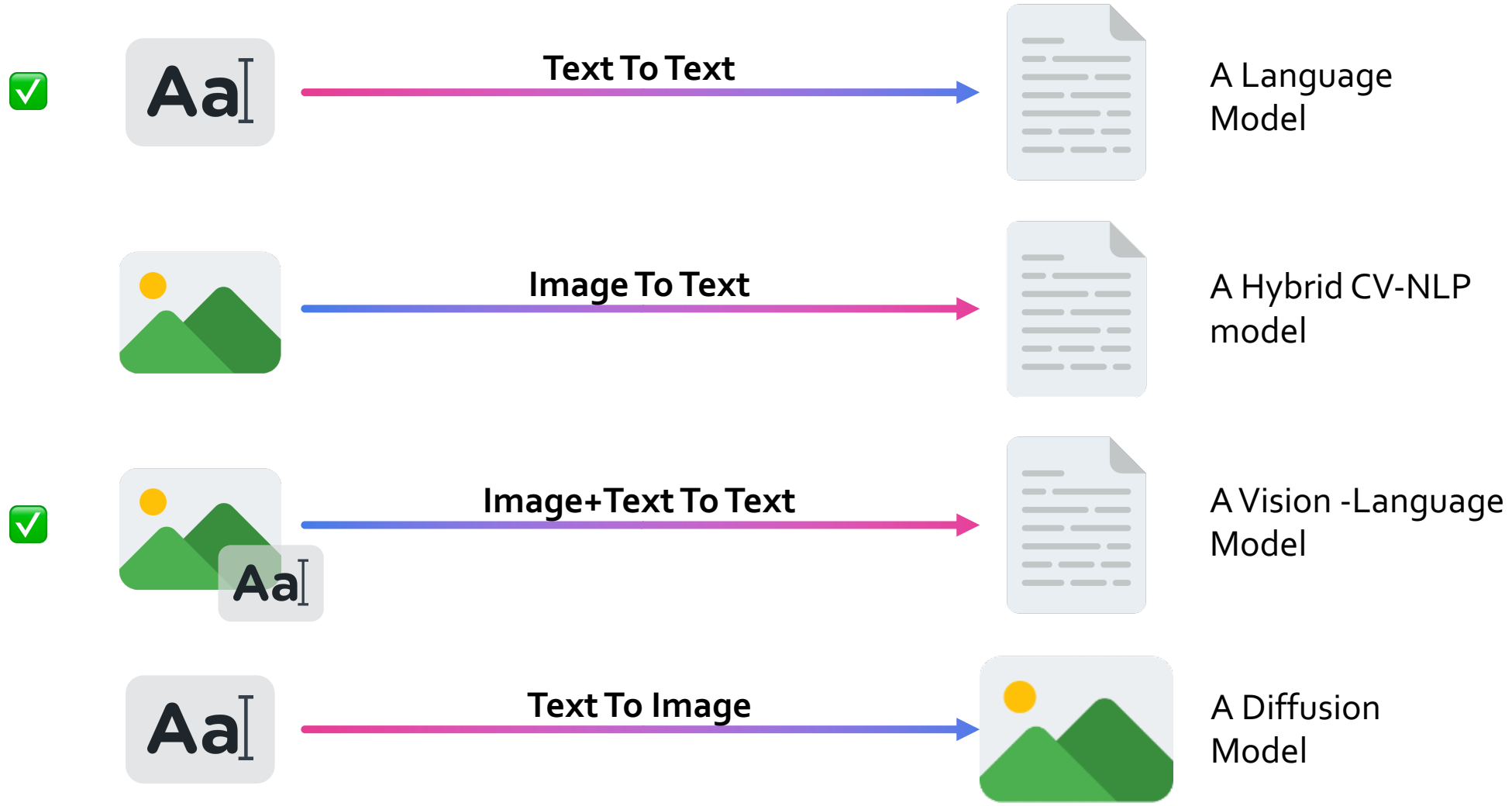
`XGen-MM` is a series of the latest foundational Large Multimodal Models (LMMs) developed by Salesforce AI Research. This series advances upon the successful designs of the `BLIP` series, incorporating fundamental enhancements that ensure a more robust and superior foundation. These models have been trained at scale on high-quality image caption datasets and interleaved image-text data. XGen-MM highlights a few features below,

- The **pretrained** foundation model, `xgen-mm-phi3-mini-base-r-v1`, achieves state-of-the-art performance under 5b parameters and demonstrates strong in-context learning capabilities.
- The **instruct** fine-tuned model, `xgen-mm-phi3-mini-instruct-r-v1`, achieves state-of-the-art performance among open-source and closed-source VLMs under 5b parameters.
- `xgen-mm-phi3-mini-instruct-r-v1` supports flexible high-resolution image encoding with efficient visual token sampling.

More technical details will come with a technical report soon.

Datasets

Dataset Type	Dataset(s) Used
Pretrain	caption data: (datacomp, cc12m, cc3m, SBU, vg) && interleaved data: obelics
Instruction Tuning	LLaVA-Instruct-150K, ShareGPT4V captions, a mixture of academic VQA data including OCR/Document/Chart-focused tasks, publicly available text-only instruction data


Text To Text
A Language Model
Image To Text
A Hybrid CV-NLP model
Image+Text To Text
A Vision -Language Model
Text To Image
A Diffusion Model

From texts to images – diffusion models

- **Goal**: automatically generate images/videos based on a textual description (aka the *prompt*)

Cute robot painter painting on a canvas, digital art, colorful, cartoon, drawing

DALL·E

S.

MIdjourney

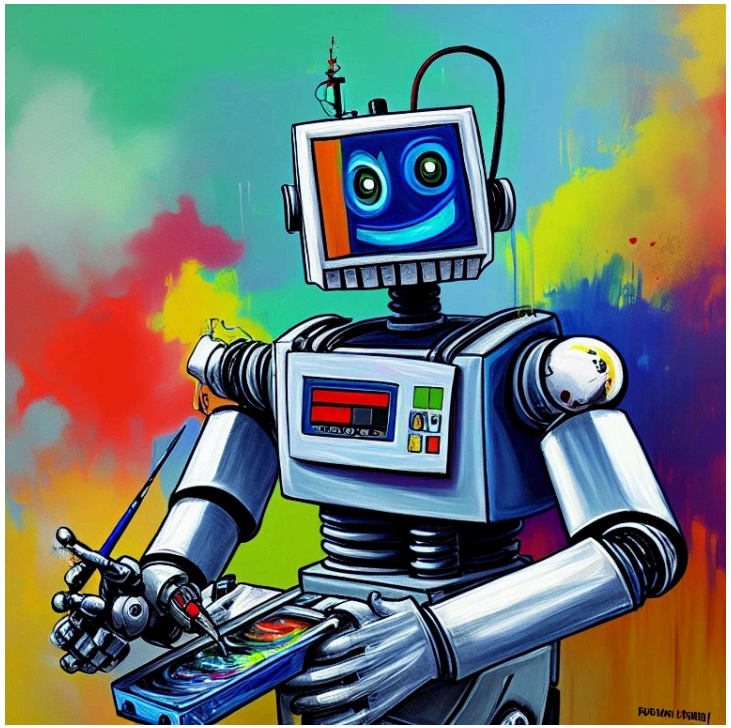
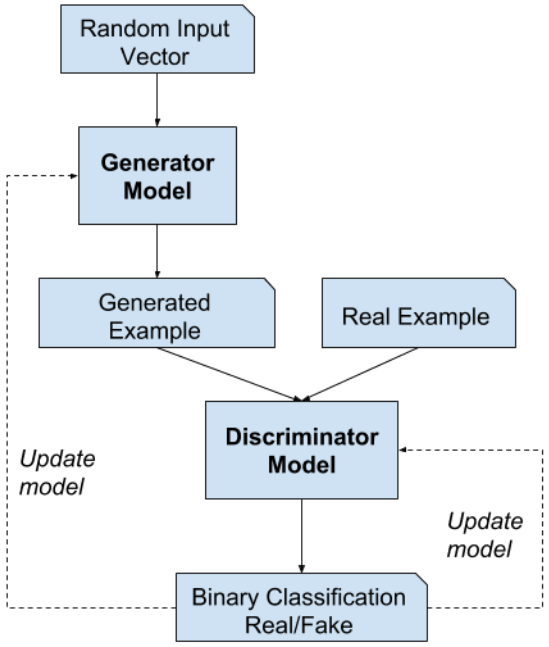
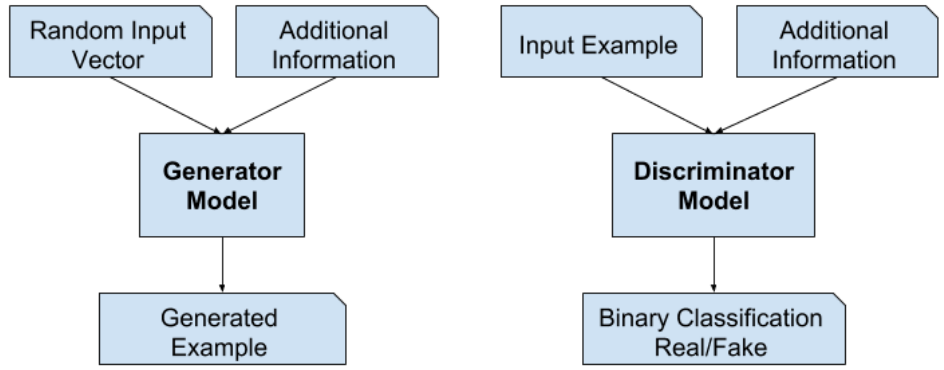


Image-conditioned generation

- AI image generation has been around for quite some time
  - GANs (Generative Adversarial Networks) have been first proposed by Goodfellow et al. (2014) and Radford et al. (2015).
  - The idea: train a generative model to produce an image from a random noise distribution (the *generator*), and exploit a discriminative model (a classifier) to guide the training process (the *discriminator*)
    - Trained to distinguish between real and fake images
    - The generator has to "fool" the discriminator into thinking that the generated image is actually real
  - By providing additional information to the generator and the discriminator the GAN can be conditioned to generate specific results





Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems* 27 (2014).

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015).

Generative AI today

- GANs are still widely studied and used in real world applications
  - E.g., they are *extremely* useful for data augmentation
    - They can produce similar looking images to those they were trained on
  - Some impressive models today leverage GANs for guided image generation and transformation
    - Just look at "Drag Your GAN" (Pan et al., 2023)

- But text-to-image models nowadays are almost exclusively based on **diffusion models**, and they have made a few huge leaps in the last couple years

Generative AI today

Midjourney progress

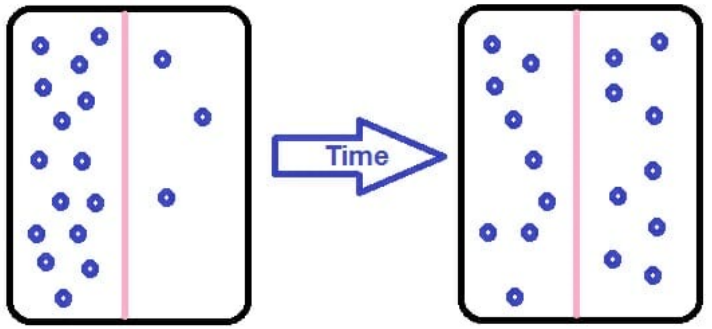
July 2022

December 2023

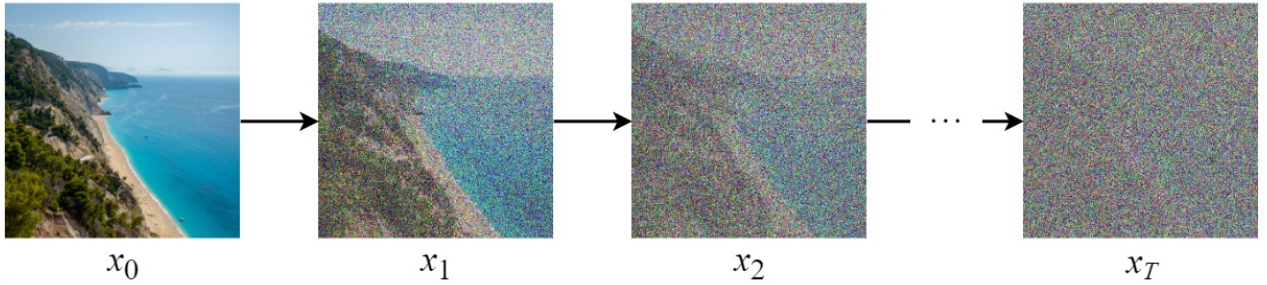
*Images generated on MidJourney by Henrique Centieiro and Bee Lee

Why diffusion

- In phisics, diffusion is the process for which particles move from high-concentration to low-concentration areas in the space, leading to an equal distribution



- In an image, we can imagine diffusion as the process that leads from a specific distribution of pixel values to a random, more uniform one.
  - I.e., **random noise**!

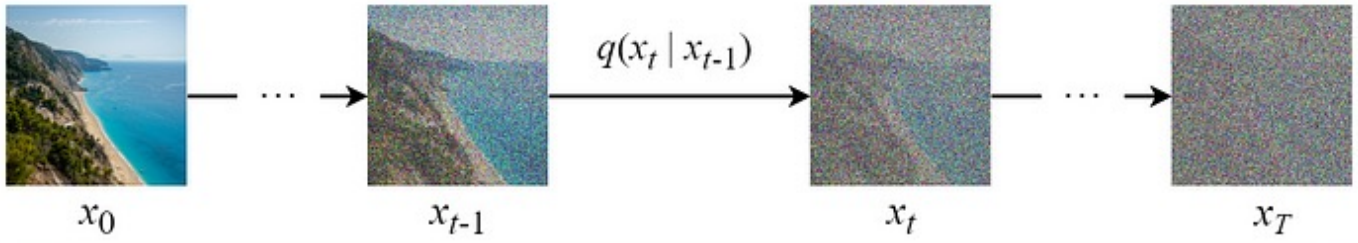


Read more here

How to diffuse

The diffusion process can be applied to a sample image by adding a small amount of Gaussian noise to the sample in T steps

- For $T \to \infty$, the final result will become a completely noisy image as if it is sampled from an isotropic Gaussian distribution.
- Image at x_t directly depends from x_{t-1}, but with a few math tricks we can directly sample noised images at any given t in T.



Distribution of the noised images — Output — Mean μ_t — Variance Σ_t

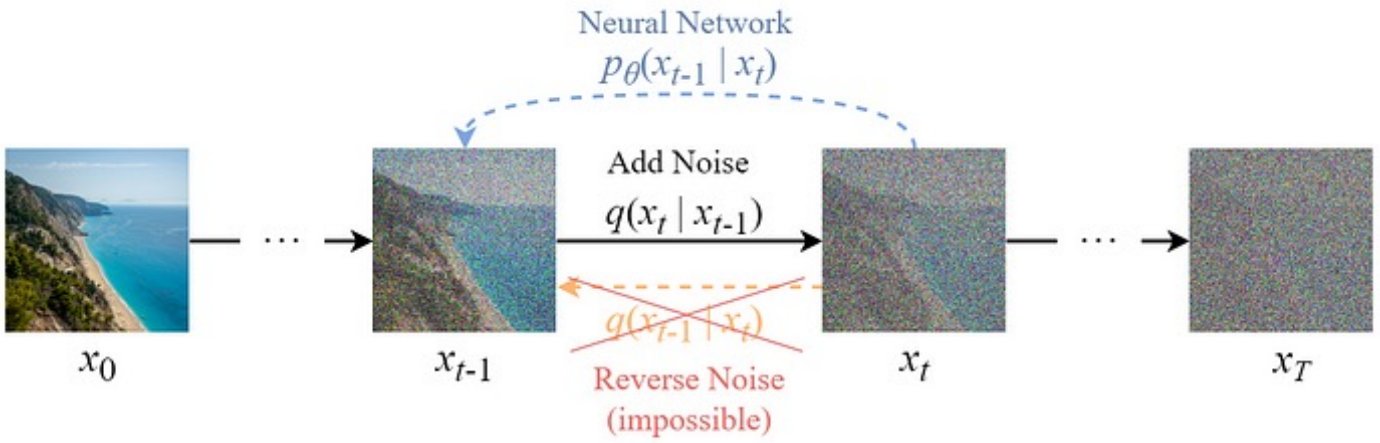
$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

Notations:
- t : time step (from 1 to T)
- x_0 : a data sampled from the real data distrition $q(x)$ (i.e. $x_0 \sim q(x)$)
- β_t : variance schedule ($0 \leq \beta_t \leq 1$, and β_0 = small number, β_T = large number)
- I : identity matrix

Forward diffusion

Read more here

And how to go back



Target Distribution $q(x_{t-1}|x_t) = \mathcal{N}(x_{t-1};\ \tilde{\mu}_t(x_t, x_0),\ \tilde{\beta}_t I)$

Approximated Distribution $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t))$

Learnable parameters
(Neural Network)

Going forward we can use x_{t-1} to get x_t.

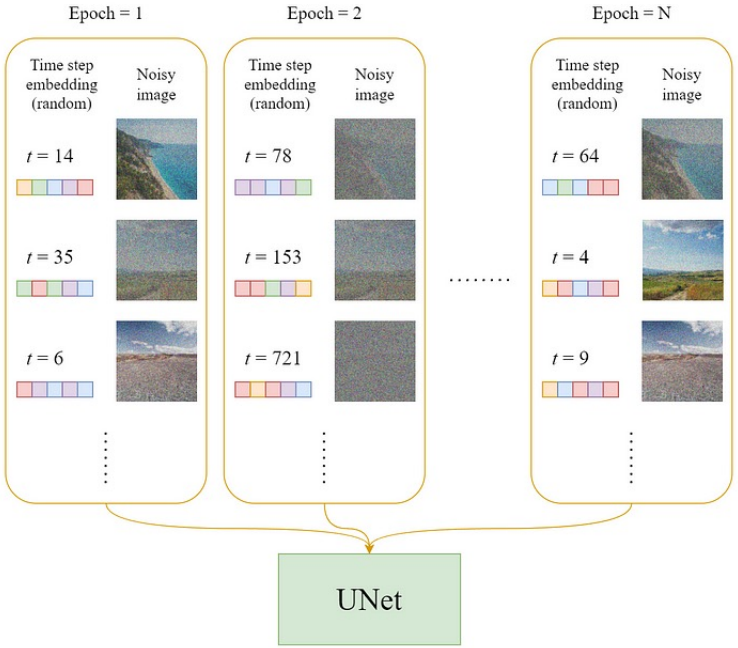
Going backwards, it's impossible.

But what we can do is approximate it by training a neural network to predict x_{t-1} given x_t

Basically, it's a neural network that should learn to predict the noise in a image.

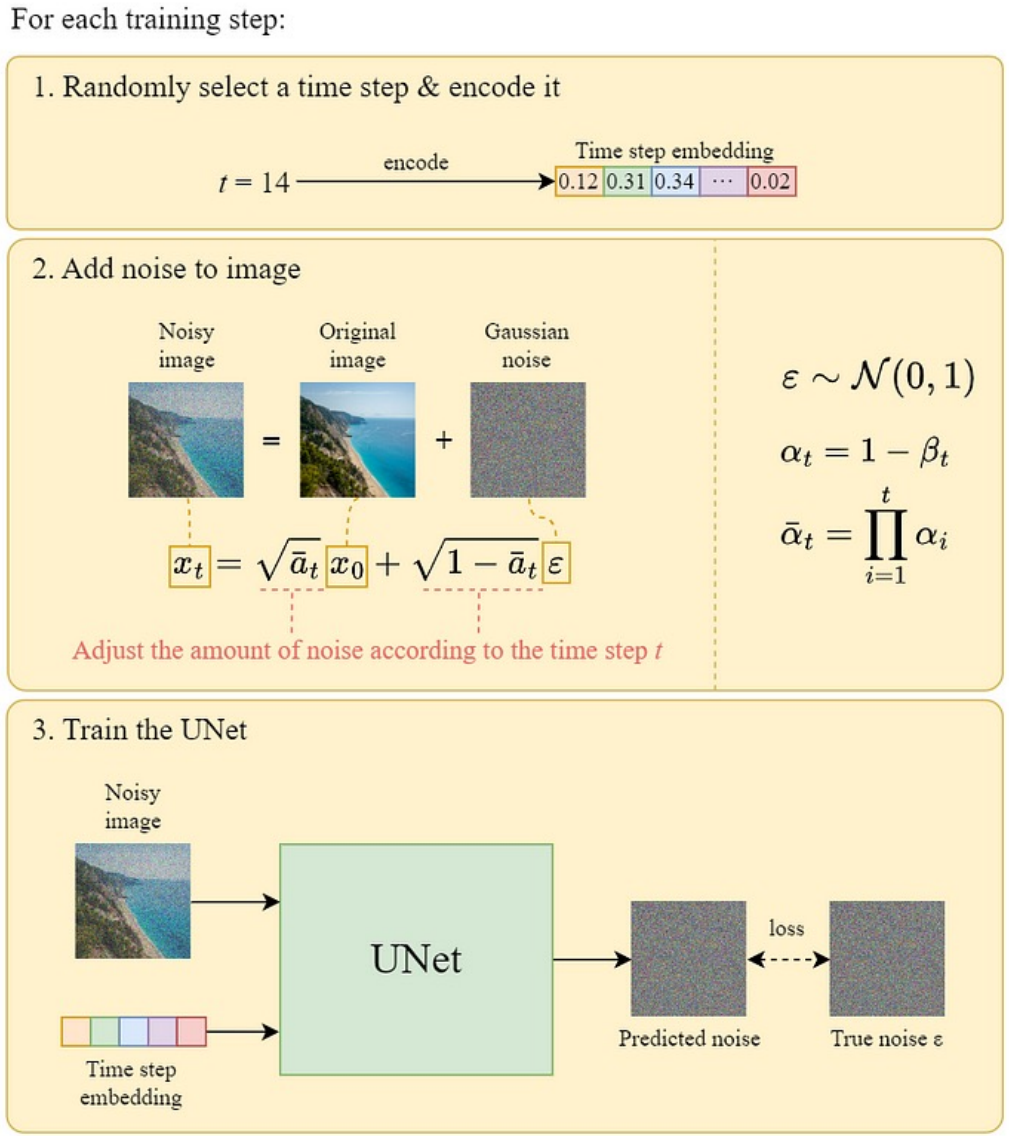
Read more here

Reverse diffusion training



For each epoch:

- A random time step *t* is selected for each training sample (image).
- The Gaussian noise for *t* is applied to each image.
- Time steps are converted to embeddings
- Model is fed the image and the time step embedding and has to predict the noise in it



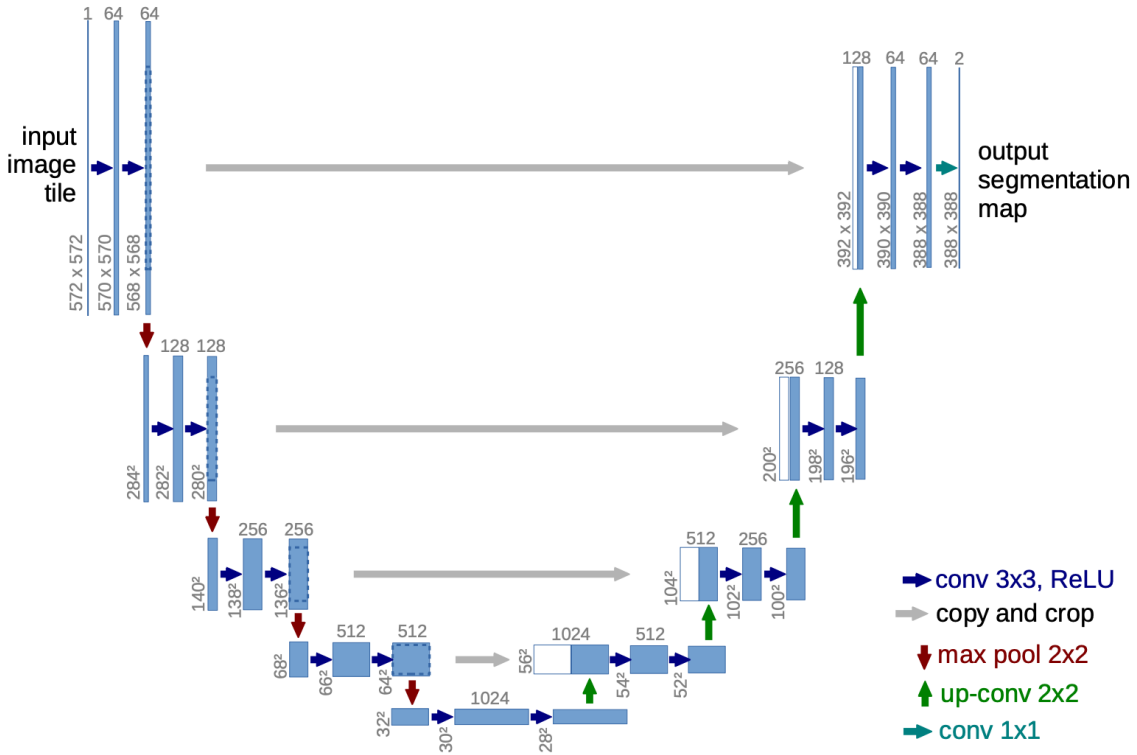
More details here

What kind of model

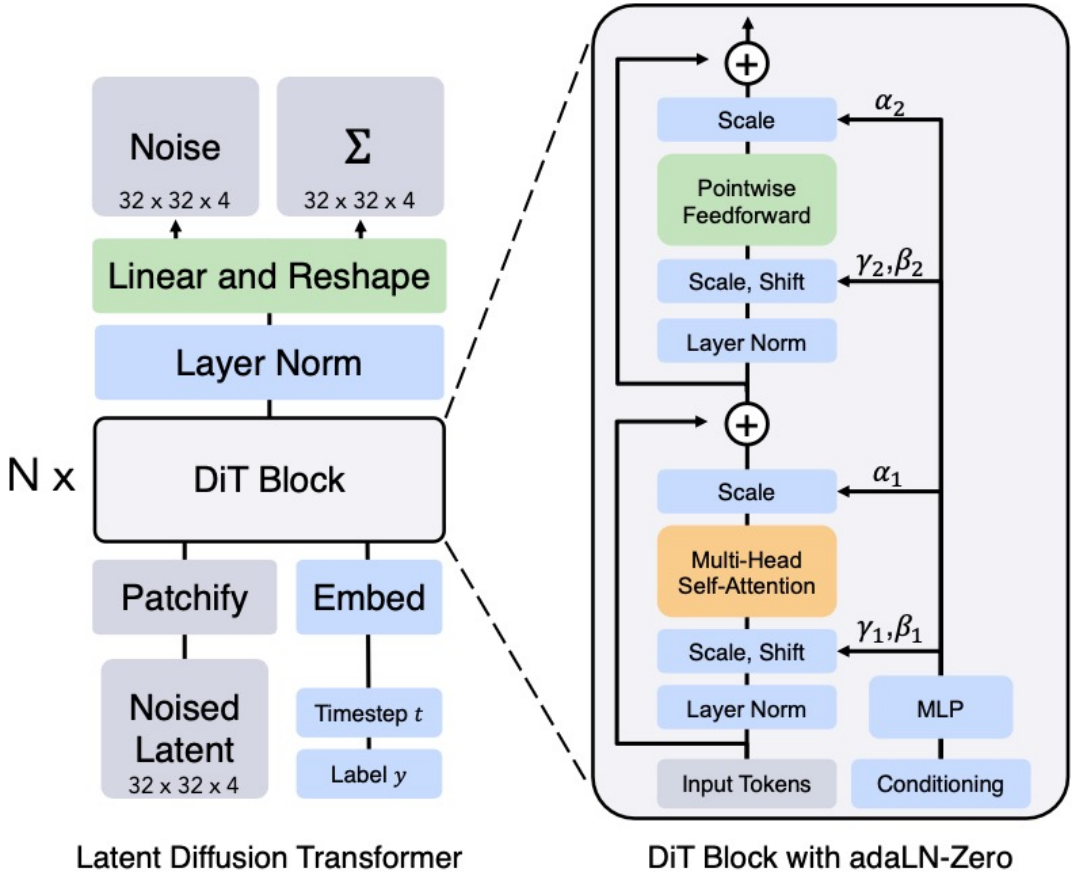
U-Net (Ronneberger, et al. 2015)

A U-shaped CNN architecture with a downsampling and an upsampling stack

- **Downsampling**: repeated convolutions + ReLU + max pooling. At each downsampling step, the number of feature channels is doubled.
- **Upsampling**: Repeated upsampling of the feature + convolution. Each step halves the number of feature channels.
- **Shortcuts**: connections with the corresponding layers of the downsampling stack to provide high-resolution features to the upsampling process.



What kind of model



Diffusion Transformer (DiT; Peebles & Xie, 2023)

A transformer model that operates on latent patches.

1. Take the latent representation of an input z as input to DiT.
2. "Patchify" the noise latent of size $I \; x \; I \; x \; C$ into patches of size p
3. and convert it into a sequence of patches of size $(I/p)^2$
4. Pass the sequence through the Transformer blocks
5. The transformer decoder outputs noise predictions and an output diagonal covariance prediction.

In the paper they explored different designs to do generation conditioned on some contextual info (e.g., timestamp or class label).

PROs: can easily scale up, and with size also performances scale better than with U-Net; can model ordered *sequences* of patches so not many modifications needed to model time-space relationships (for video generation).

But where is the text?

- For now we have only seen how to apply the idea of diffusion to reconstruct images from noise using different models.
- We want to be able condition this process based on other info, e.g. a text (but also other images, etc.)
- **Idea**: use a strong VL encoder (e.g., CLIP) and condition noise prediction with its output via attention!
  - Can make the model both creative and conditionable by following textual prompts

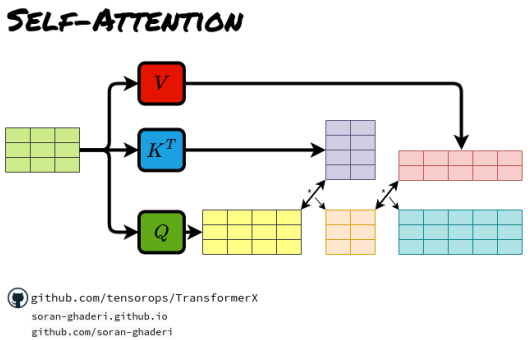


Image-to-self
Learns to produce coherent images

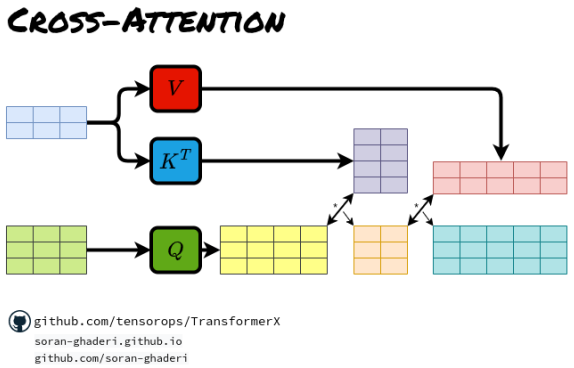
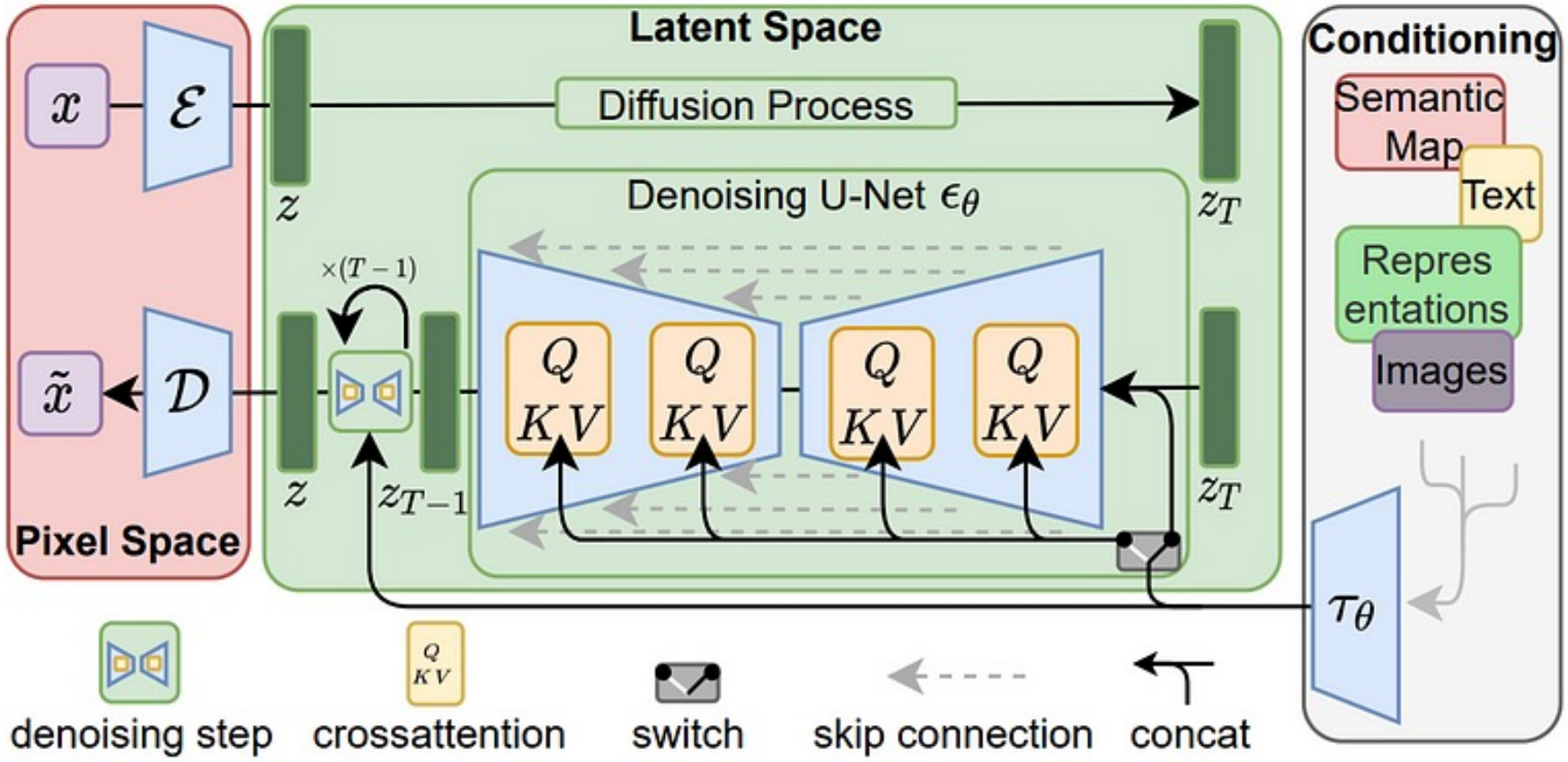


Image-to-text
Enables images inputs in the diffusion process to be modulated by elements of words

Putting it all togheter

Rombach et al., 2021

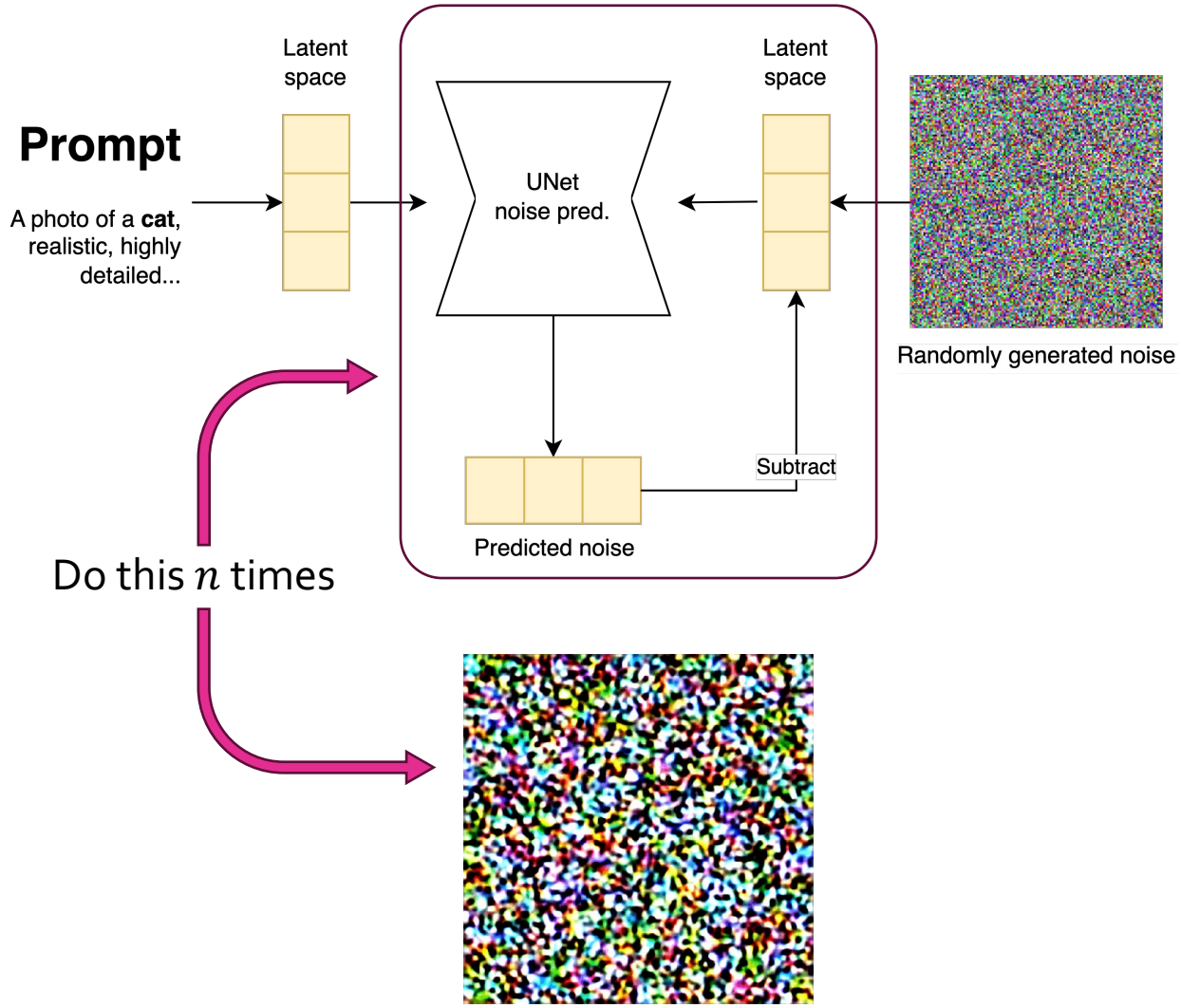


How diffusion models work? In practice

- During training we take a HUGE dataset of $\{image, caption\}$ pairs
- We apply noise to training images and learn the denoising function with the chosen model
  - Use the caption (encoded with a VLM) to condition the denoiser via attention
  - The denoiser gets conditioned by the captions
- The model learns a relationship between noise, images and texts
  - How to obtain the original image based on noise and text

How diffusion models work? In practice

- After training we feed the model a prompt + random noise distribution
  - + negative prompt, optionally
- The model knows how to iteratively denoise following a prompt and tries to do so on the random noise
  - It doesn't know (nor care) that the "source" image does not exist!
- After a few steps we get a (hopefully) beautiful image of what we asked for…



Adapting a diffusion model

- Diffusion models are usually trained on enormous **noisy** datasets
  - E.g., Laion 5B, where B stands for billions
- Great for zero-shot generalization
  - The model learns from a lot of different stuff
  - Visual cues in the prompt and negative prompts are an effective tool to steer the generation towards what we want
- We may still need to adapt the model to our specific needs

beautiful pale cyberpunk female with heavy black eyeliner, blue eyes, shaved side haircut, hyper detail, cinematic lighting, magic neon, dark red city

Poorly rendered eyes, poorly drawn hands, …

Pre-trained vs fine-tuned diffusion model

Prompt: *a pokemon that looks like a cute blue fox, with golden eyes and a long tail*

Negative prompt: *ugly, poorly rendered eyes, tiling, poorly drawn hands, poorly drawn feet, poorly drawn face, out of frame, extra limbs, disfigured, deformed, body out of frame, blurry, bad anatomy, blurred, watermark, grainy, signature, cut off, draft*

Stable-diffusion 2.1

Stable-diffusion 2.1
tuned on Pokémon w/ captions

Notebook

How to use and adapt a diffusion model

- Many available libraries and tools (not many completely free) to use and train diffusion models
- Huggingface Diffusers
  - Sibling of 🤗Transformers
    - Same principles and abstractions, including pipelines for generation
  - Many available models (pre-trained and fine tuned)
  - Easy-to-run scripts for training
    - A few different ways to fine-tune a model, e.g. with LoRA

Applications and limitations of Generative AI

- Generative AI provides amazing tools for dealing with images and video content
  - Generation, but also inpainting, upscaling, text-guided editing etc.
- But raise many issues
  - **Bias and fairness**
    - Noisy datasets are notorious for including stereotypical views and biases
    - Depending on the intended use this NEEDS to be addressed
    - True also for VL models in general
  - **Copyright**
    - Who is the owner of AI generated content? The prompt writer?
      The AI company that made the model?
      The artists on which the model was trained on?…
  - **Technical limitations**
    - Most recent models are quite good with photorealism, texts, and compositionality, but are not perfect yet
    - Consistency (in different images and in videos) is still a problem also for most advanced models

Useful tools/resources

For thinkering with VLMs and Diffusion Models

- https://huggingface.co/
- https://huggingface.co/models
- https://huggingface.co/blog/vision_language_pretraining

AILC Lectures on Computational Linguistics 2023

- https://github.com/luciacpassaro/LCL2023-Lab2

For prompting (LLM and VLM)

- https://learnprompting.org/

Online image generation tools

- https://github.com/AUTOMATIC1111/stable-diffusion-webui
- https://www.midjourney.com/home?callbackUrl=/explore

Final thoughts

- Reserch on multimodality is blowing up
  - Many new commercial models can be considered **Multimodal Language Models (MLMs)** rather than LLMs, e.g. GPT-4V, Gemini, etc.
  - Image and video generation have made giant leaps even only in the last year or so (see for example OpenAI Sora)
- Many unsolved questions to address. Just to name a few:
  - **Bias**, **Fairness** and **Copyright infringement** in text-to-image models
  - **Benchmarking** VLMs is way harder than benchmarking LLMs: what a VLM should be capable to do?
  - **Efficiency** is fundamental as models get larger and larger
- One of the "wild frontiers" of Generative AI today

VLMS

THANK YOU

Questions?

Feel free to hit me up via e-mail for further questions (& ideas)
alessandro.bondielli@unipi.it