EXPLORING MULTI-MODAL NEURAL MODELS AND THEIR APPLICATIONS

Lectures on Computational Linguistics

Pisa, May 30th 2023

LUCIA C. PASSARO

ALESSANDRO BONDIELLI

{lucia.passaro,alessandro.bondielli}@unipi.it

OUTLINE

MULTIMODALITY AND NLP(LAB 2)

Part II

Learning strategies for VL models

Specializing pre-trained models

Generative models

Vision-Language Models evolution

Mid to late '10s

- Smaller, task-specific models
- **Vision** models based on CNNs and GANs
- **Language** Models using word embeddings (Word2Vec, GloVE) or RNNs
- Focus: align modalities to solve a specific task
  - E.g. VQA, image captioning
  - Attending to pre-extracted visual and language features

Vision-Language Models evolution

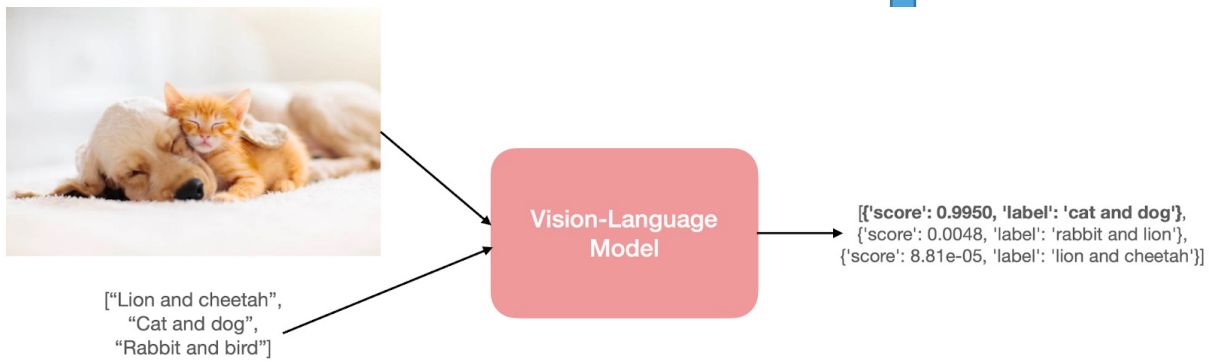
Late '10s - Early '20s

- Transformers and the attention mechanism for NLP and CV
- First Transformer-based VL models
  - *Attention* mechanism to connect modalities
  - Multi-modal pre-training objectives to align modalities
- Model scale goes up
  - Model parameters are in the (hundred of) millions
  - Datasets size are in the (tens of) millions
- Still needing fine-tuning on downstream tasks

Vision-Language Models evolution

Late 2021 – Now…

- Still (mostly) using Transformers
- Huge datasets and models
  - (tens of) Billions of parameters & pre-training examples
- Zero- and few-shot learning
  - With pre-training alone or prompt-based
- Mostly unfeasible with commercial hardware (GPUs etc.)



How to pre-train your VL model

- Multi-modal fusion can be achieved in a few different ways by pre-training Transformers
- We still need a few key components:

Data

Image-text pairs

Text Encoder

To represent the textual part of the input

Visual Encoder

To represent the visual part onf the input

Model Architecture

A way to combine encoders (and decoders)

Pre-training objective

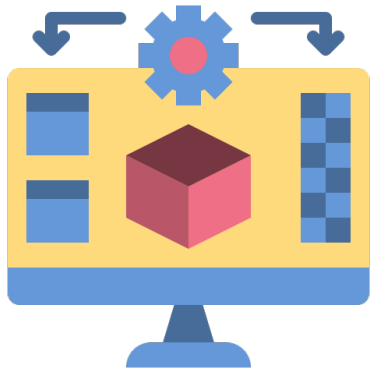
What the model has to learn during training

Hyperparameters

Learning rate, epochs etc.

Architectures and objectives

- A good model should be able to (Gan et al., 2022)
  - Obtain good performances on downstream tasks
  - Adapt to new tasks with minimal cost
    - Training samples, trainable parameters etc.
- Model architecture and pre-training are key for
  - Out-of-the-box zero and few-shot capabilities (larger models)
  - Performances when tuning on downstream tasks
- Different strategies for pre-training are available



Model architectures

2 Two-stream models (dual encoder)

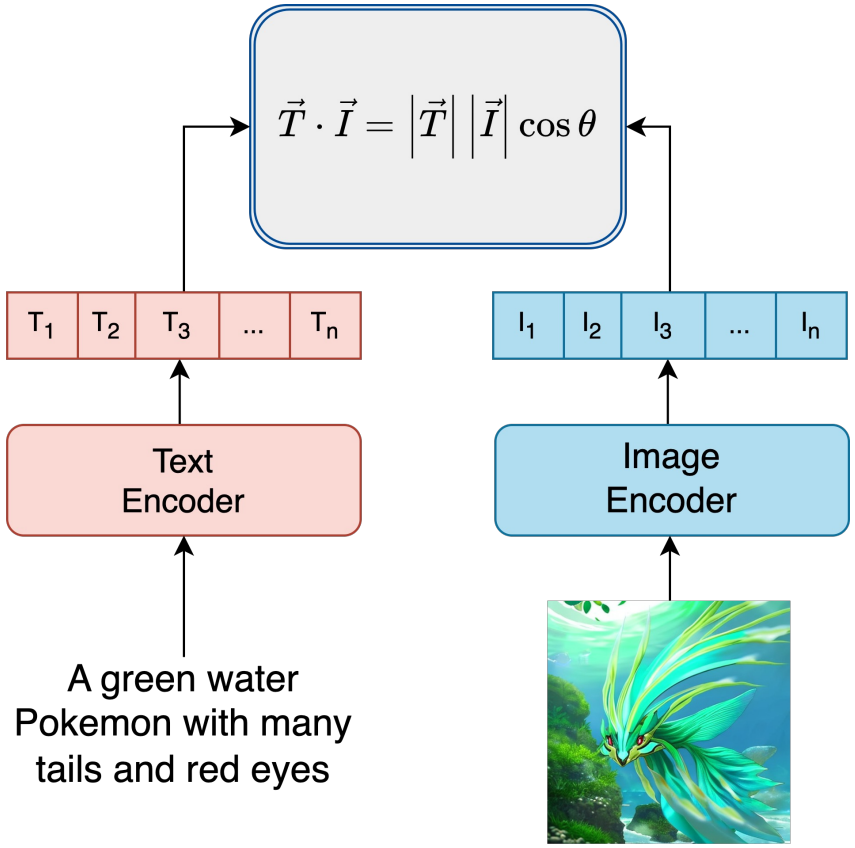
- Images and texts are encoded separately
- Interaction is handled by maximizing similarity between text and image representations
  - E.g., via cosine similarity
  - Still able to learn strong visual representation via large-scale pre-training
- Better for image-text matching tasks
  - Image retrieval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

1 Single-stream models (fusion encoder)

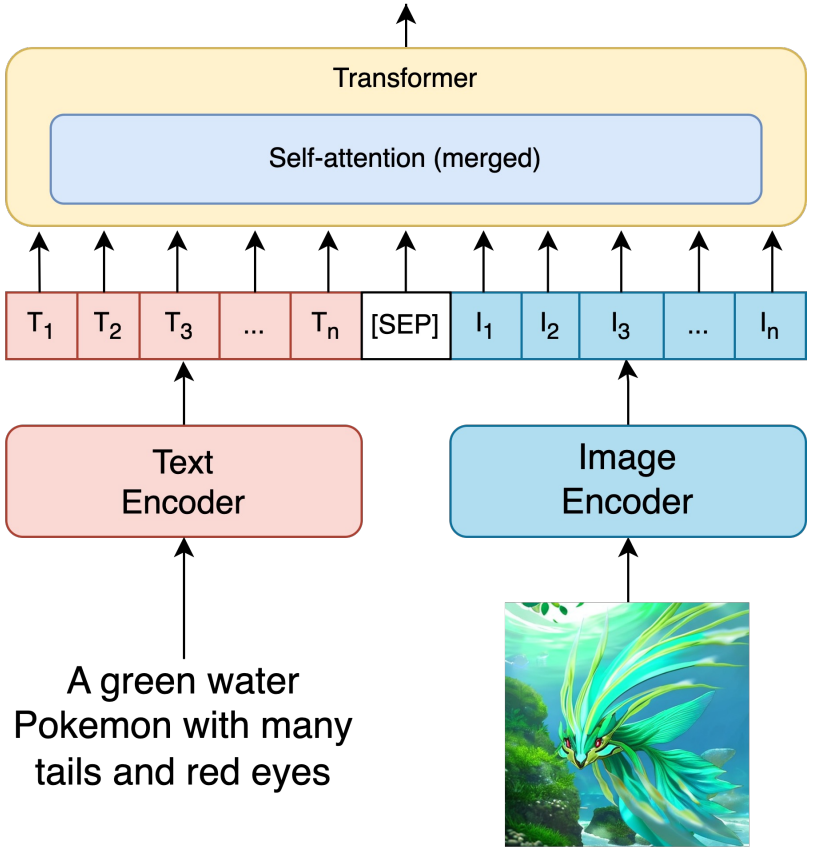
- The text and image encoders are fused togheter with additional transformer layers
- Aimed at modelling deeper interactions between texts and images
- Better for tasks that require both inputs to be modeled together
  - VQA, captioning, visual reasoning etc.

Model architectures

Two-stream models (dual encoder)

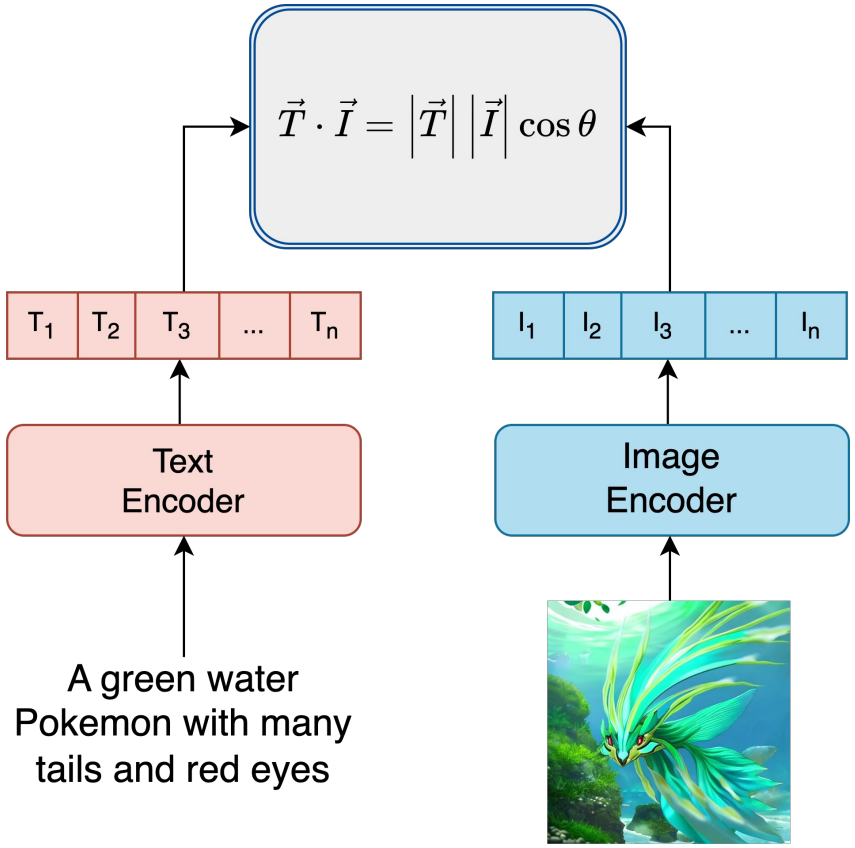


Single-stream models (fusion encoder)

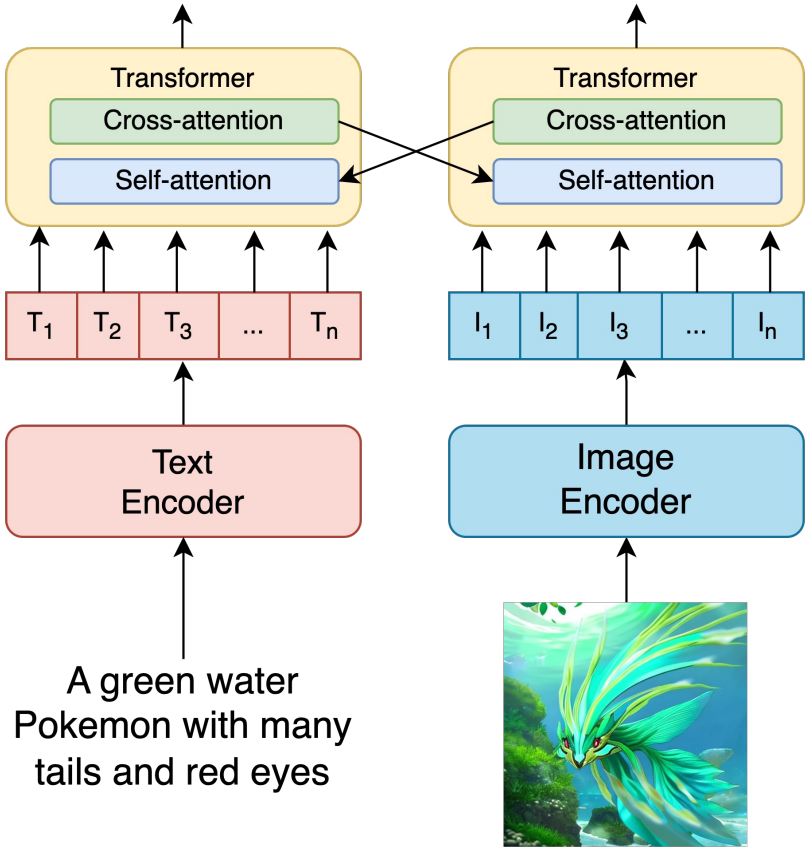


Model architectures

Two-stream models (dual encoder)

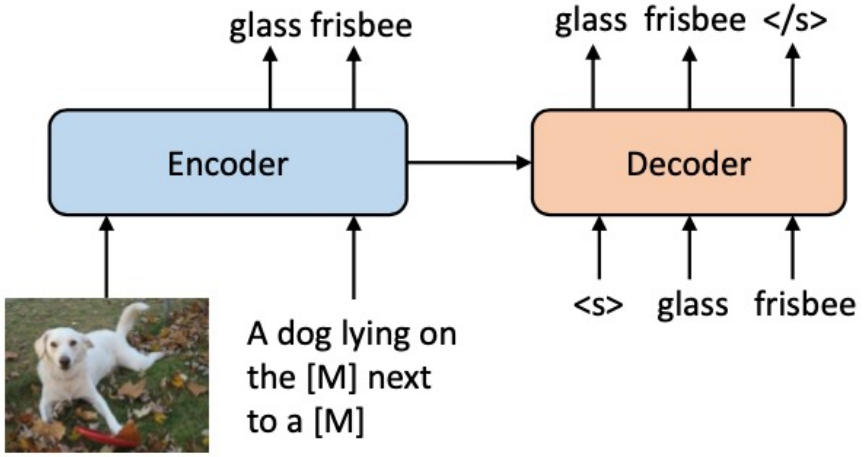


Single-stream models (fusion encoder)



What about decoders?

- An encoder-only architecture is sufficient
  - Use MLP heads to generate the target output (e.g. for classification)
  - Perfect when dealing with non-generative tasks
    - Still able to generate text via Causal Masking



- But some recent works advocate for an encoder-decoder architecture (Wang et al., 2021)
  - Outputs are generated autoregressively (i.e., by looking at the inputs & the previously generated output)
  - Natural fit for text generation tasks such as free-form VQA and captioning
  - Also used in vision-text encoder decoder models

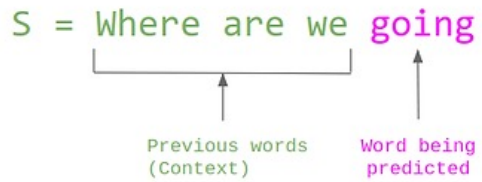
Learning objectives

- If we want to teach the model how to look at both inputs when providing the outputs
  - I.e., the relationship between texts and images
- We can start from uni-modal pre-training objectives and adapt them to a multi-modal scenario

- Common pre-training objectives in VL models comes from the NLP and CV literature
  - Especially from NLP

Language Modelling

- Transformer-based LMs are trained by predicting words in the inputs
  - BERT-style Masked Language Modelling (MLM)
    - Train the model to predict [MASK] tokens based on their context
  - GPT-style Causal Language Modelling (CLM)
    - Train the model to autoregressively predict the next token



P(S) = P(Where) x P(are | Where) x P(we | Where are) x P(going | Where are we)

Vision-Language Modelling

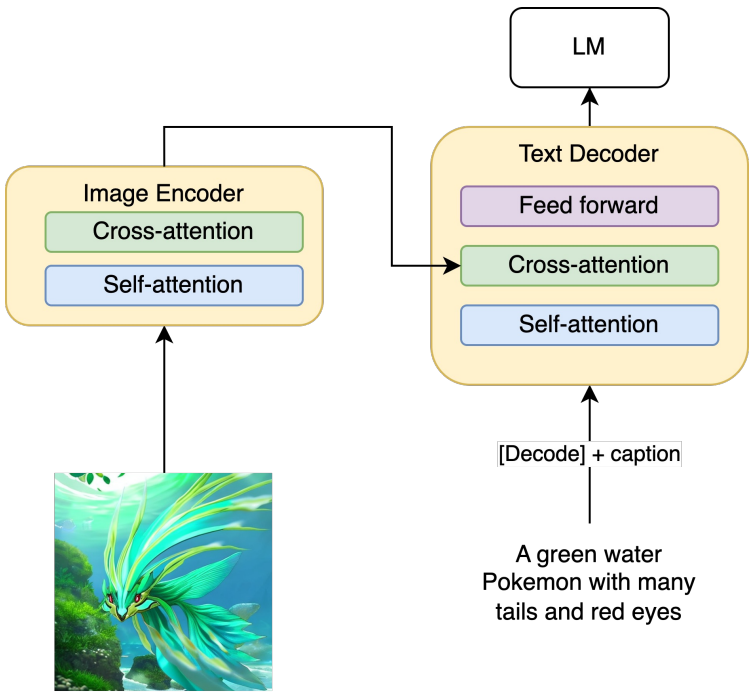
- Adapt the language modelling task to attend *also* to the visual input
  - E.g., learn to predict [MASK]ed parts of a caption by looking at the rest of the caption *and* its corresponding image
- Several effective pre-training strategies in Vision-Language modelling

Direct LM

- Generate a caption given an image (and optionally a prompt)
- **How**:
  - Concatenate the inputs in the encoder part of the model
  - Use the decoder to predict the caption token by token, autoregressively

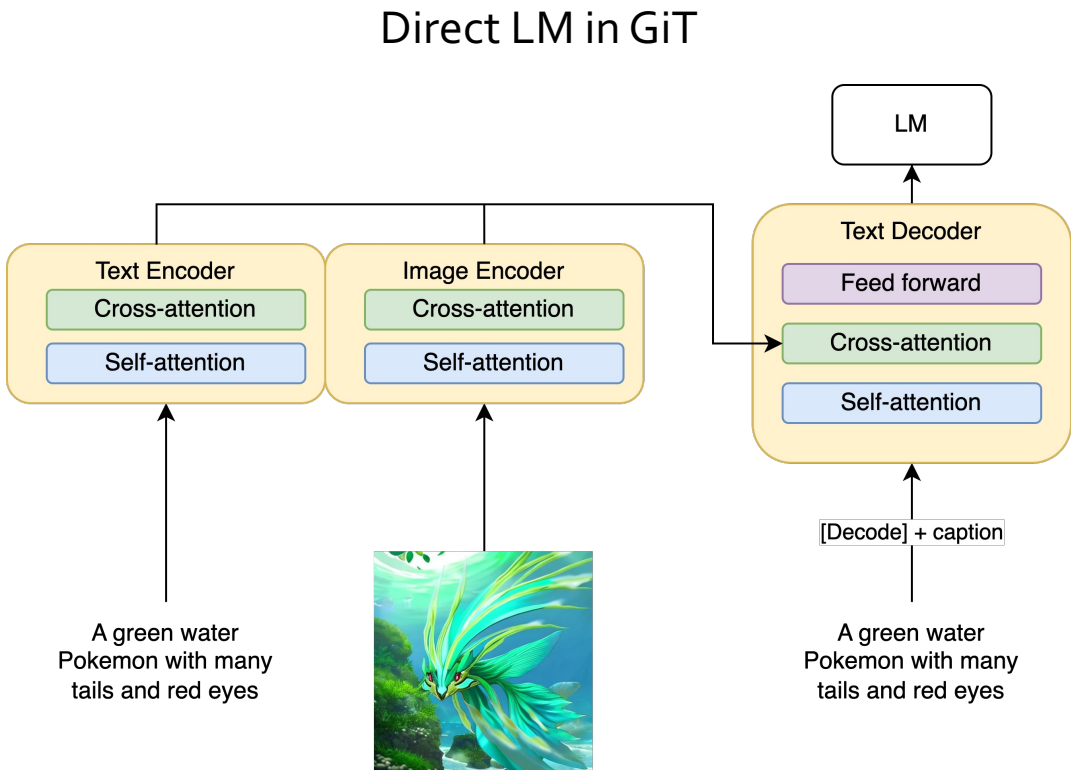
Available Models

Direct LM in BLIP



Direct LM

- Generate a caption given an image (and optionally a prompt)
- **How**:
  - Concatenate the inputs in the encoder part of the model
  - Use the decoder to predict the caption token by token, autoregressively



Available Models

Multi modal fusion with *Cross Attention*

- If we enable a more refined image encoder in the pipeline we can leverage the model for more tasks
  - E.g. by using object detection as inputs to the image encoder
  - Same Cross Attention mechanism
  - Learns word-object mapping as well

Available Models

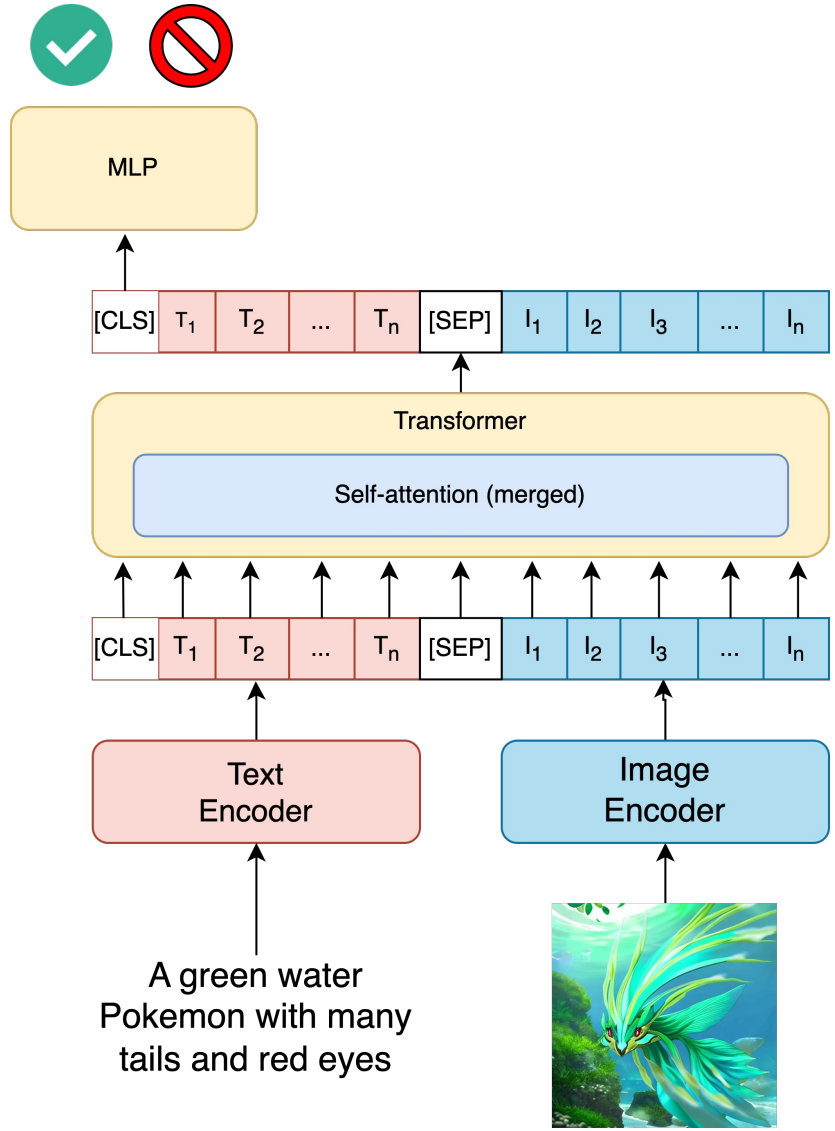
Prefix LM

- For Text-only Prefix Language Models
  - Consider part of the text as prefix and predict the rest based on it
  - Same idea for Vision Transformers, but with image patches
- For VL models
  - The prefix is the image patches + the first part of the text
  - Trained to predict the rest of the sequence
- Frozen Prefix LM
  - If we apply the same concept but freeze the parameters of the text encoder we can learn image embeddings aligned with the LM

Available models

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

Available models

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

- 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

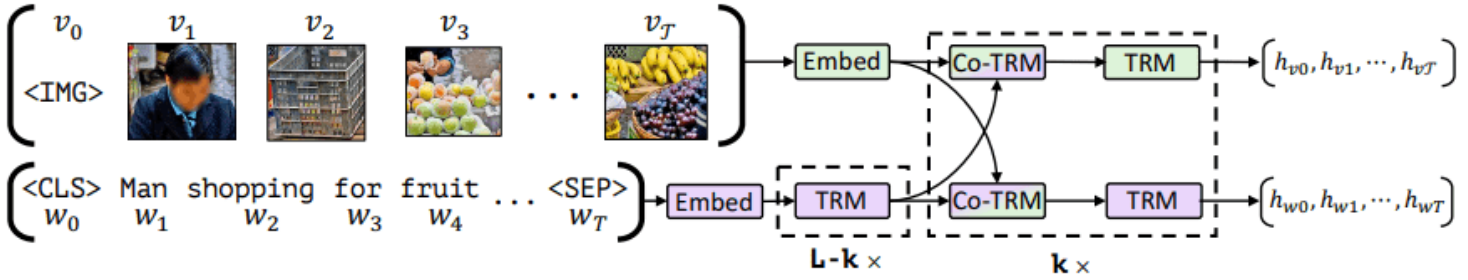


Image-Text Contrastive Learning (ITC)

- **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

Big models and pre-training

- Most larger models are trained with contrastive learning, generative pre-training, or both
  - Around 1B parameters and 1-10B training samples
  - Best of both worlds: fast image-text retrieval/matching and text generation capabilities
- Two key benefits of scale:
  - Zero-shot generalization: pre-training objectives resemble real-world tasks and allow for zero-shot capabilities
    - Contrastive learning → zero-shot retrieval
    - Generative pre-training → zero-shot captioning
  - In-context few-shot learning: large enough models can be adapted to specific tasks or domains with only a few in-context examples rather than full fine tuning

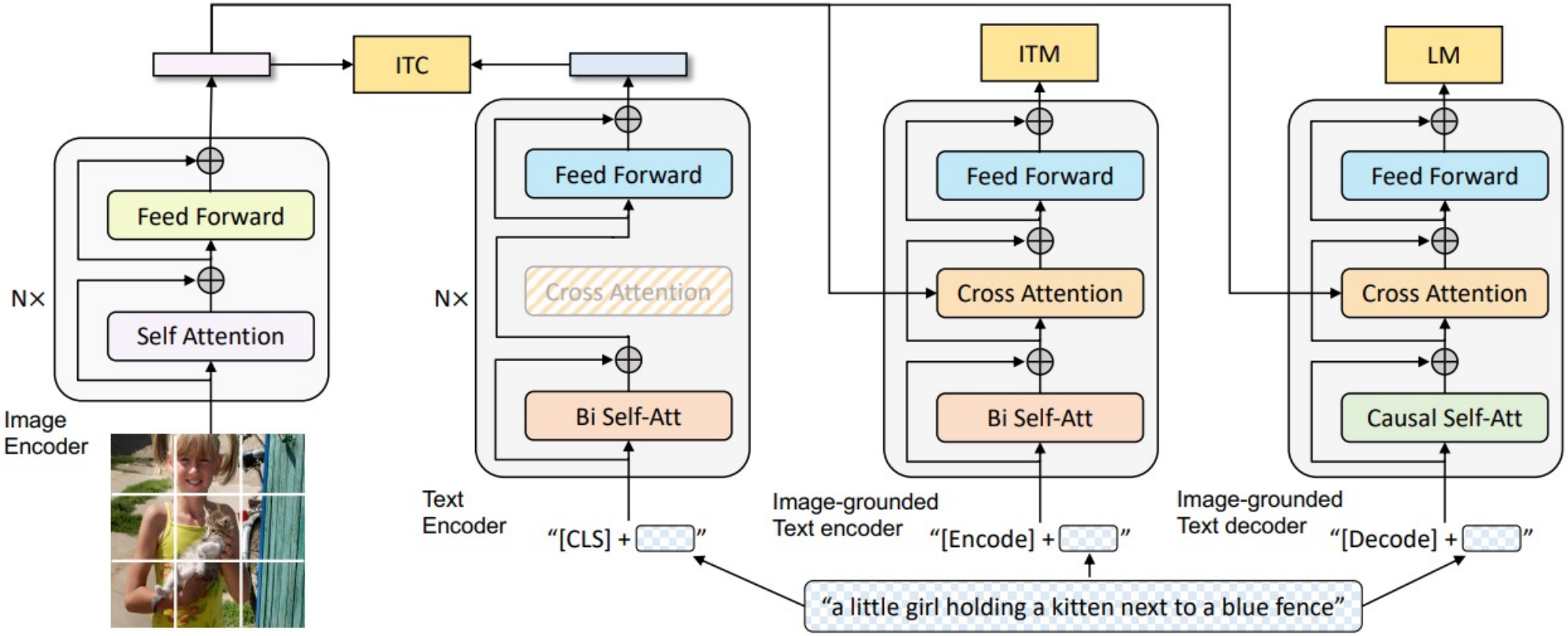
All-in-one models

- Four macro-areas of tasks in VL models

Close-set classification

- Several approaches aimed at tackling at least 2 of them in the same model
  - E.g., BLIP (Li et al., 2022) unifies Close-set classification and open-ended text generation

The example of BLIP



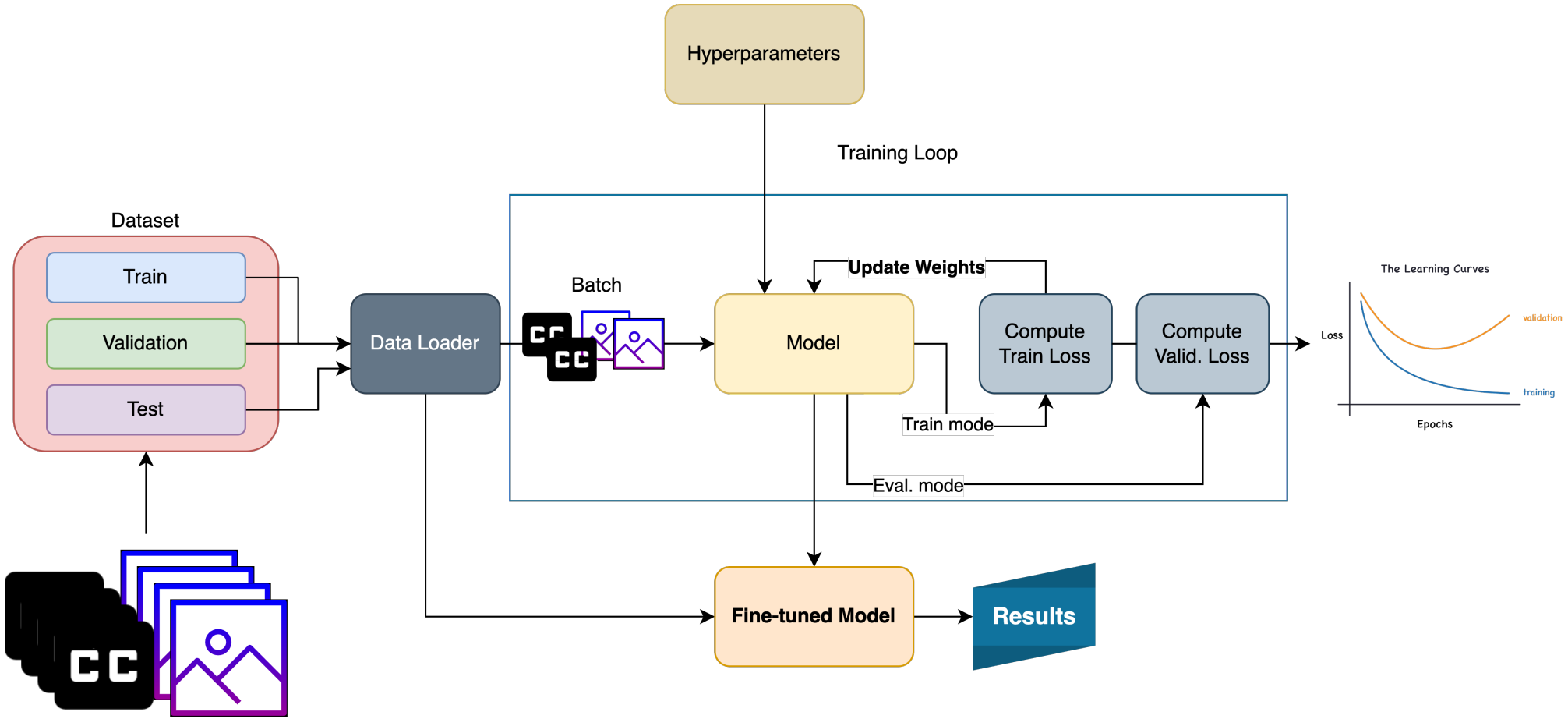
Model fine-tuning

- Zero-shot generalization and few-shot learning are feasible for large enough foundational models
- BUT!...
  - A LOT of power and compute to run
    - Not enough in consumer-grade GPUs
  - Fine-tuning mid-to-large sized VL models is not dead
    - For adapting models to specific tasks and/or domains
    - A fraction of the cost of running a ultra-large sized VL model
    - More control over the task, the data and the training process
    - Specific problems calls for specific knowledge!

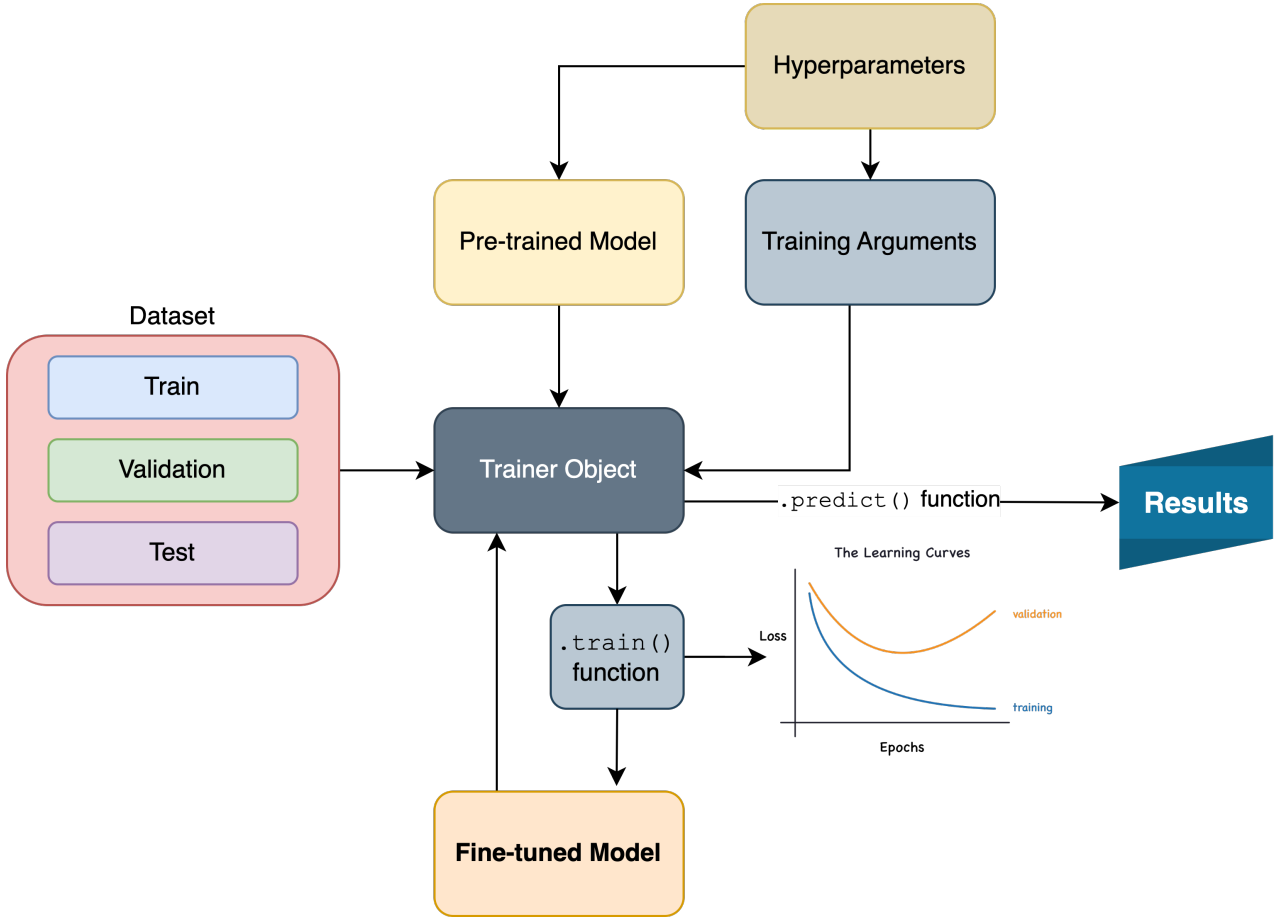
What we need

- To fine-tune a model, we need to
  - Choose the model
    - Ideally, an already effective one for the task
  - Prepare our dataset
    - Or find a suitable one for our task
  - Write few lines of code code to train and evaluate it

Fine-tuning


Hyperparameters
Training Loop
Dataset
Train
Validation
Test
Data Loader
Batch
Update Weights
Model
Compute Train Loss
Compute Valid. Loss
Train mode
Eval. mode
The Learning Curves
Loss
validation
training
Epochs
Fine-tuned Model
Results

Fine-tuning (with 🤗 Trainer)



Let's get our hands dirty

Open-domain image classification

- Pre-trained model: **CLIP**
  - A strong contrastive learning-based model for image-text matching
  - Very good zero-shot performances on image-oriented tasks (classification, object detection etc.)
- Fine-tuning
  - human actions recognition

Image captioning

- Pre-trained model: **BLIP**
  - A foundational model for image-text matching, captioning and VQA
- Fine-tuning
  - Character drawings dataset

What we saw so far

- VL models enable us to go beyond uni-modal problems
  - Image-oriented tasks
    - E.g., open-domain classification, direct captioning
  - Text-oriented tasks
    - E.g., VQA, image-guided generation of texts, free form captioning
- Fine-tuning for specific tasks is (quite) easy and cost effective

About generative models

Generative VL models can be effectively applied to image-to-text tasks

- Given an image and possibly a prompt, provide the rest
  - As a caption, as an answer, etc.

But what about text-to-image?

- Given a textual prompt, can we generate a corresponding image?

Text-to-image models

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

Goal: automatically generate images/videos

based on a textual description (aka the *prompt*)

Text-to-image models

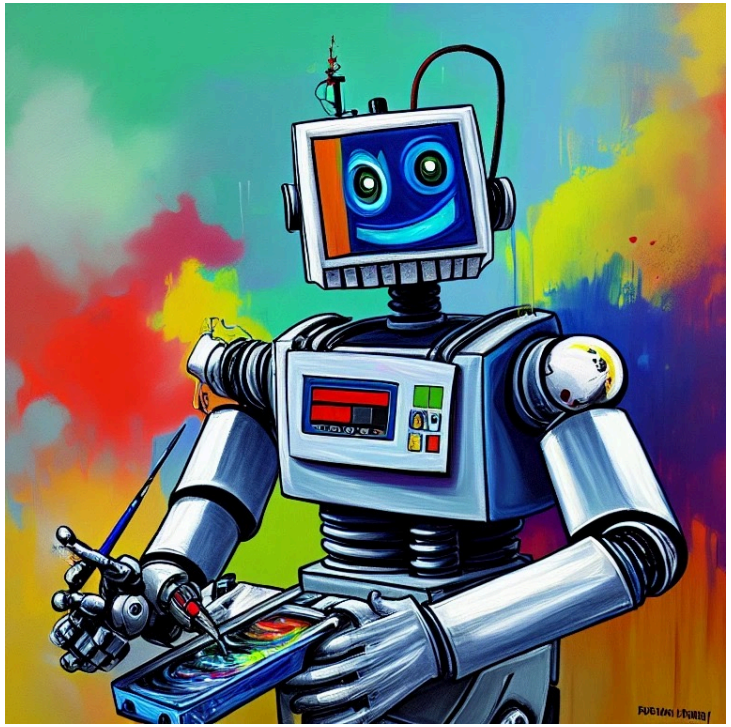
a cowboy gunslinger walking the neon lit streets and alleys of a futuristic tokyo covered in a dense fog

Goal: automatically generate images/videos

based on a textual description (aka the *prompt*)

Text-to-image models

Cute robot painter painting on a canvas, digital art

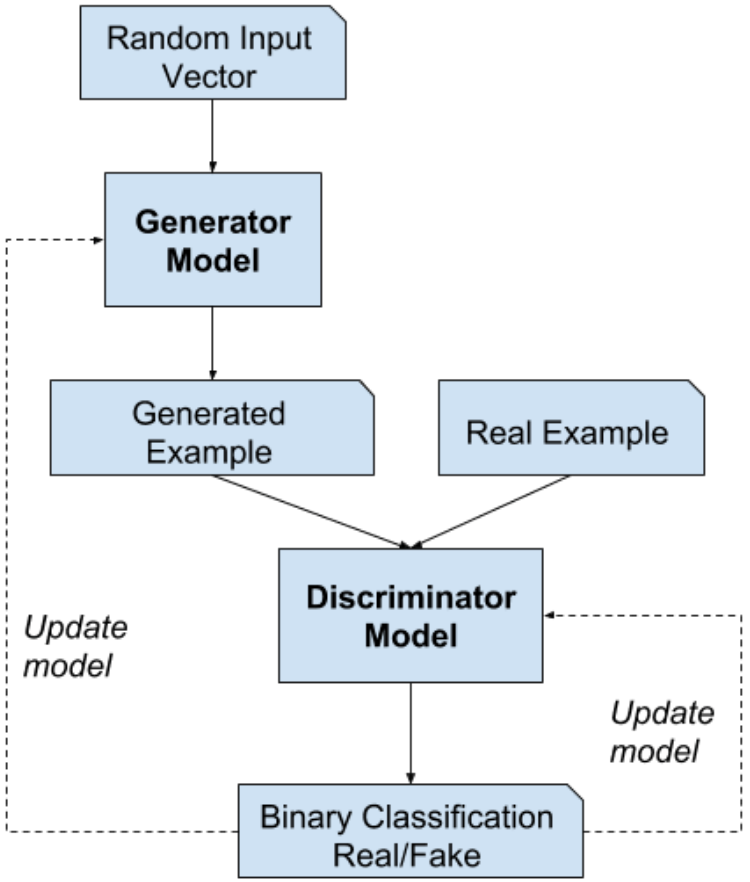


Goal: automatically generate images/videos

based on a textual description (aka the *prompt*)

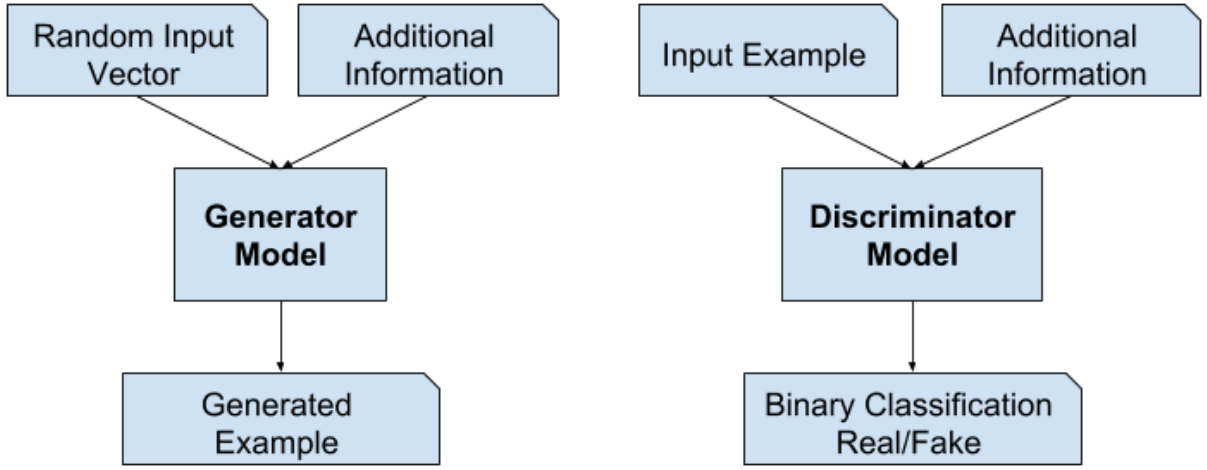
First steps into generative AI

- GANs (Generative Adversarial Networks)
  - Goodfellow et al., 2014
  - Radford et al., 2015 (DCGAN, Deep Convolutional GANs)
- The basic idea:
  - Train a generative model to produce an image from a random noise distribution (the *generator*)
  - 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



Conditioned GANs

- By providing additional information to the generator and the discriminator we can condition the GAN to generate specific results



Generative AI today

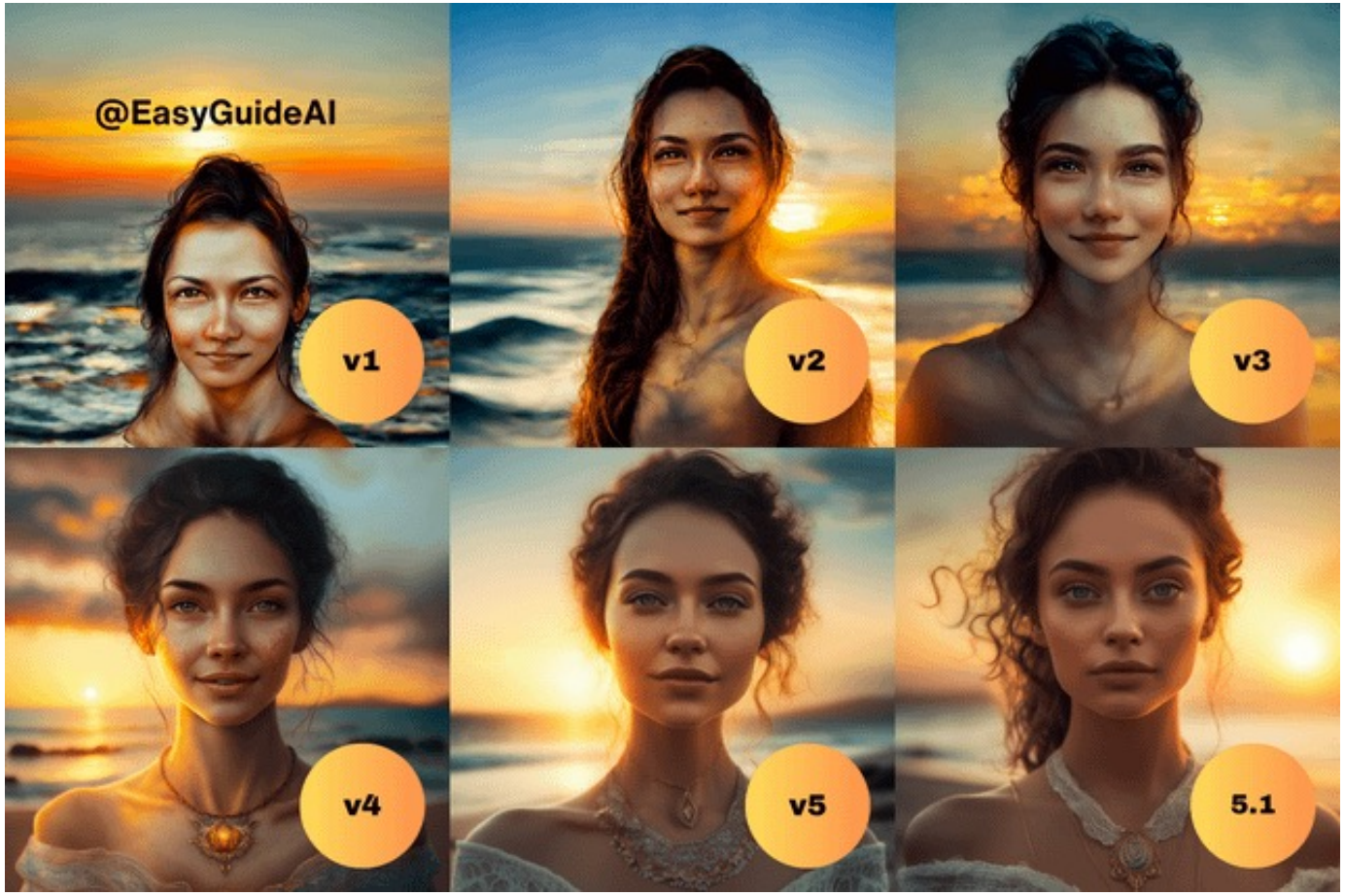
- GANs are still widely studied and used in real world applications
  - E.g., they are *extremely* useful for data augmentation
    - They can produce similar looking images to those they were trained on
  - Some impressive models today leverage GANs for guided image generation and transformation
  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

- But there is a new kid on the block that made headlines for its zero-shot text-to-image capabilities with photorealistic results and almost weekly jumps in quality…

Generative AI today

Midjourney progress

July 2022



May 2023

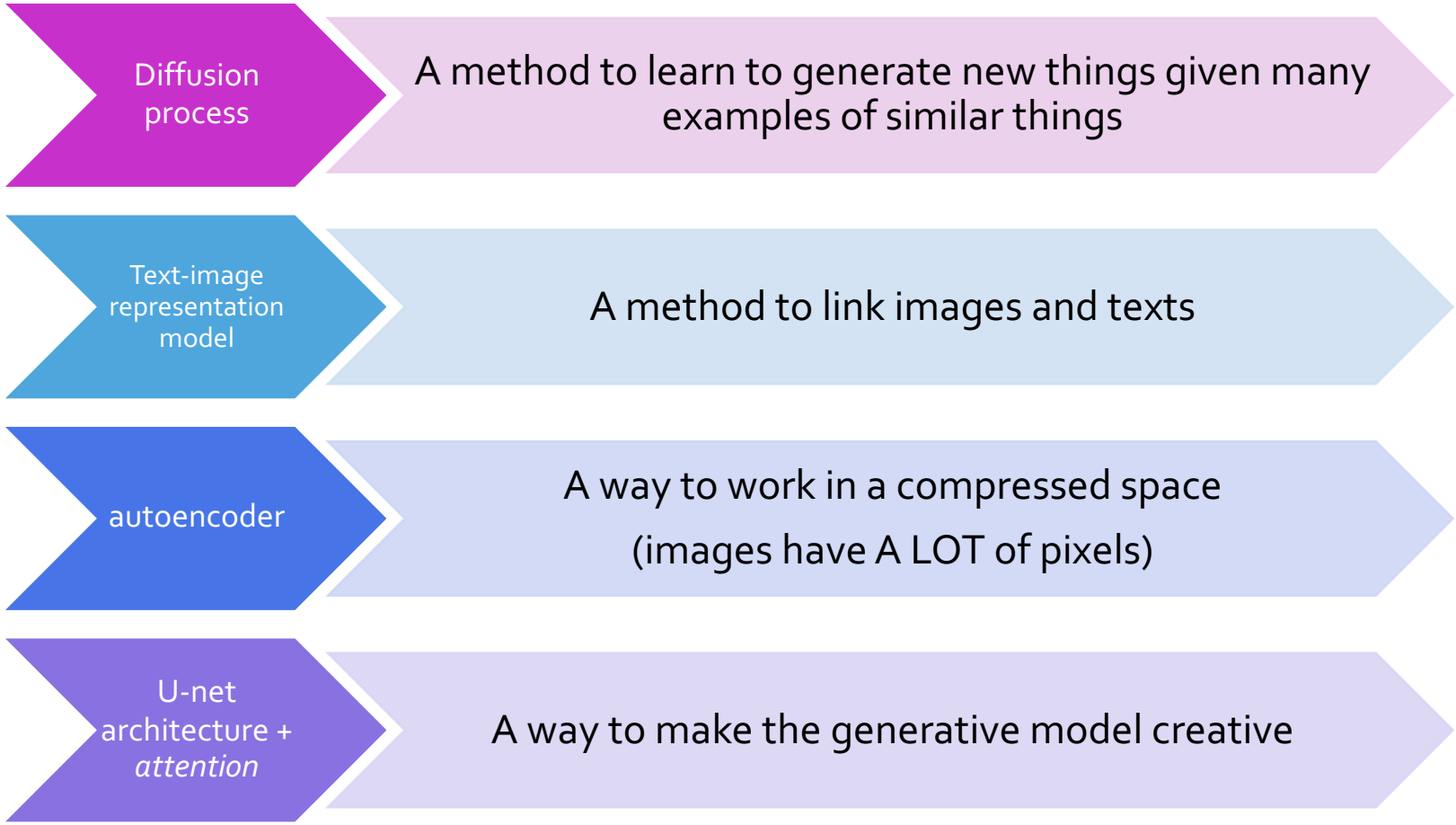
Diffusion Models

- DALL-E
- Stable Diffusion
- Midjourney
- …

How do diffusion models work?

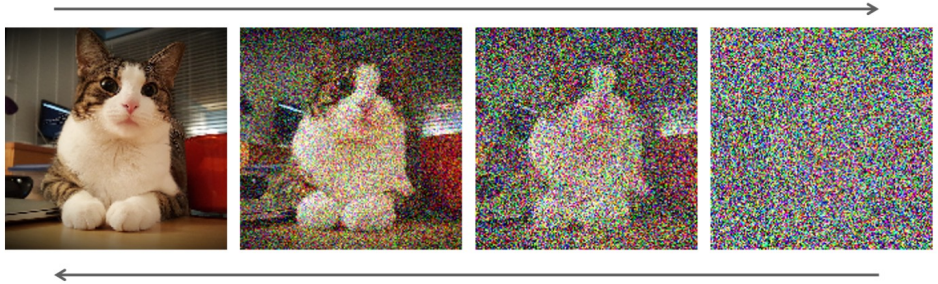
- They are complex, definitely
- But we can have a high-level idea of how they work
- Let's first look at their key components
  - The question is: "what do we need to generate a good image from text?"

Key components of diffusion models



The diffusion process

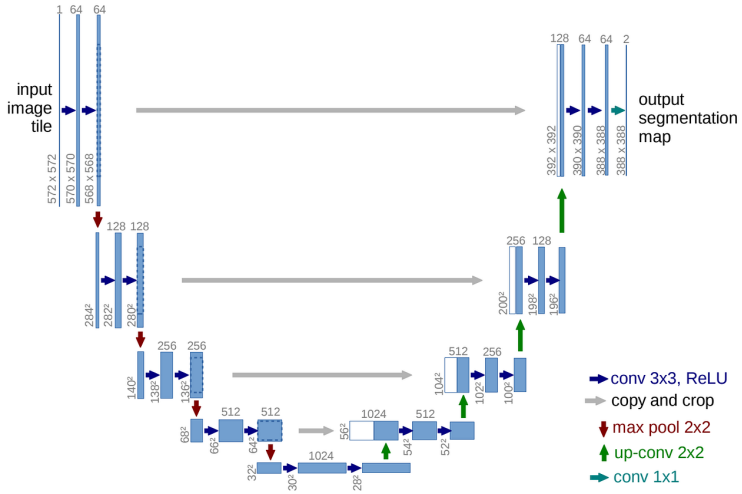
- Training a diffusion model = Train a model to denoise images
- A forward and a backward pass
  - **Forward diffusion** (fixed): images are applied increasing levels of noise



  - **Reverse diffusion** (Trained): sample from the noise distribution and generate data from it
- If we can learn a score function at each point in the diffusion process
- We can train the model to infer noise from a noised sample (and then remove the inferred noise)

Conditional diffusion

- We can use a U-Net architecture to learn to denoise
  - Based on the diffusion process
  - CNN-based Image-to-image model



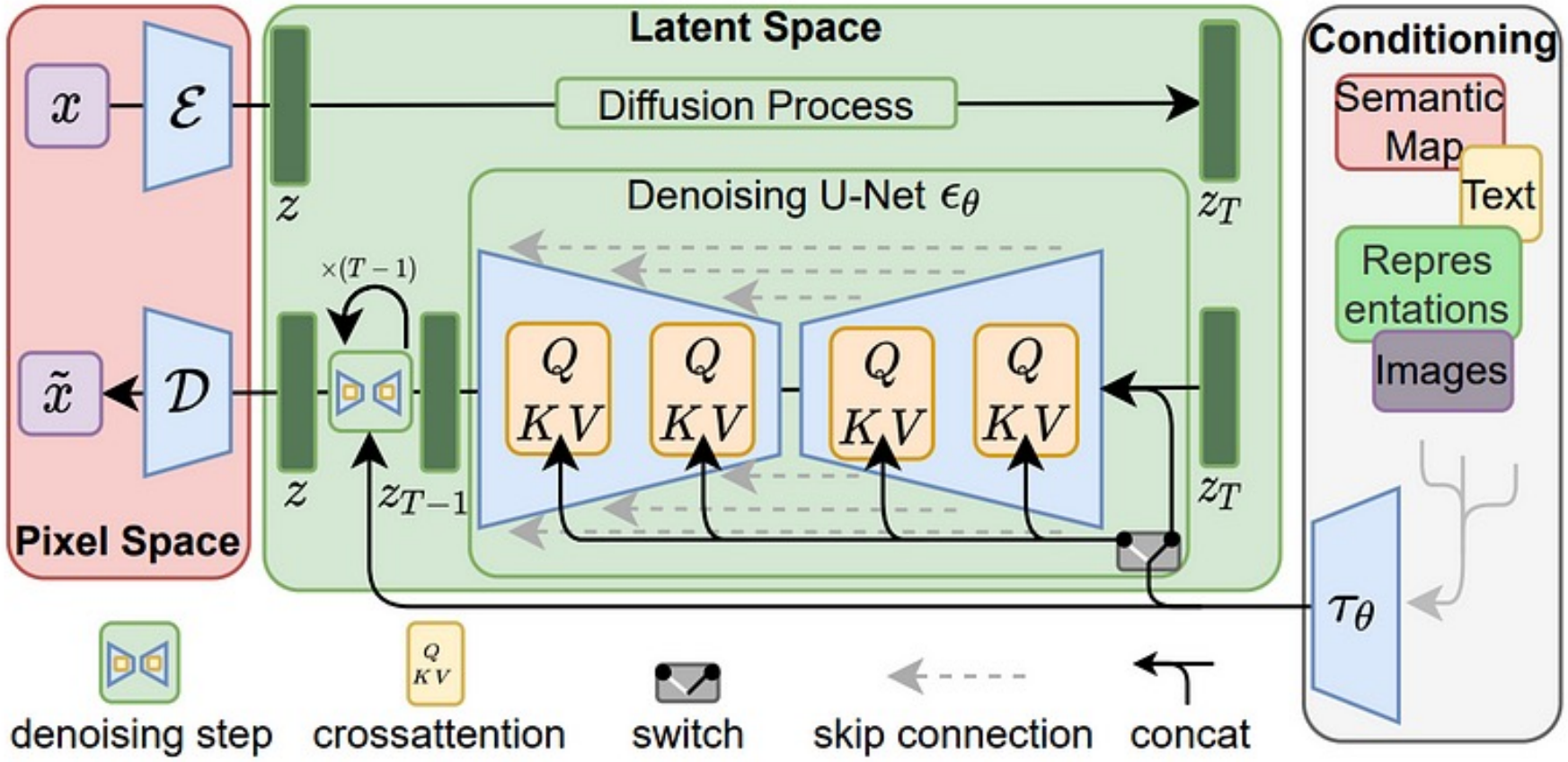
- If we add a condition to the equation, we can try to infer the noise from a noised sample, *based on* the condition
  - We can use text representation learned from a VL model as the condition!
    - We know that they are grounded in vision
    - E.g., in Stable-Diffusion the VLM used is CLIP

The missing link - attention

- To effectively condition the U-Net model on text we can use attention
  - **Self-attention**: image to self
    - Learn to produce coherent images
  - **Cross-attention**: image to words
    - Learn to let words modulate the diffusion
- Somewhat like in machine translation:
  - Tokens in one language attend to tokens in the same language AND tokens in the other language

Putting it all togheter

Rombach et al., 2021

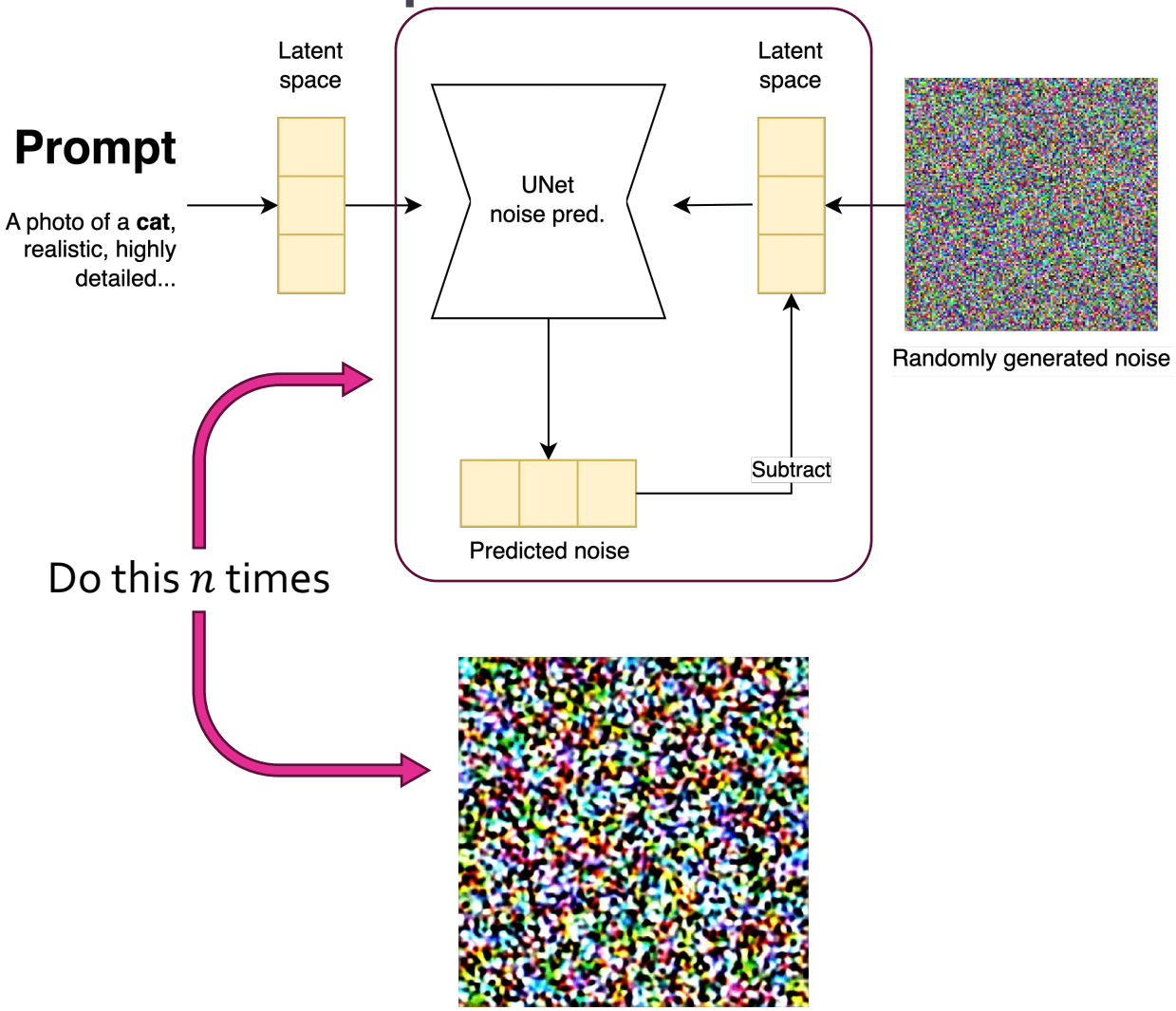


How diffusion models work? In practice

- During training
  - Take a HUGE dataset of $\{image, caption\}$ pairs
  - Apply noise to training images and learn the denoising function (U-Net)
    - Conditioned on the respective captions (with a VLM model)
  - The model learns a relationship between noise, images and texts
    - Learning how to obtain the original image based on noise and text

How diffusion models work? In practice

- After training
  - Prompt + random noise distribution
    - + negative prompt, optionally
  - The model tries to iteratively denoise the random noise following the prompt
    - 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 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

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

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…

Unconditional image generation (training script)

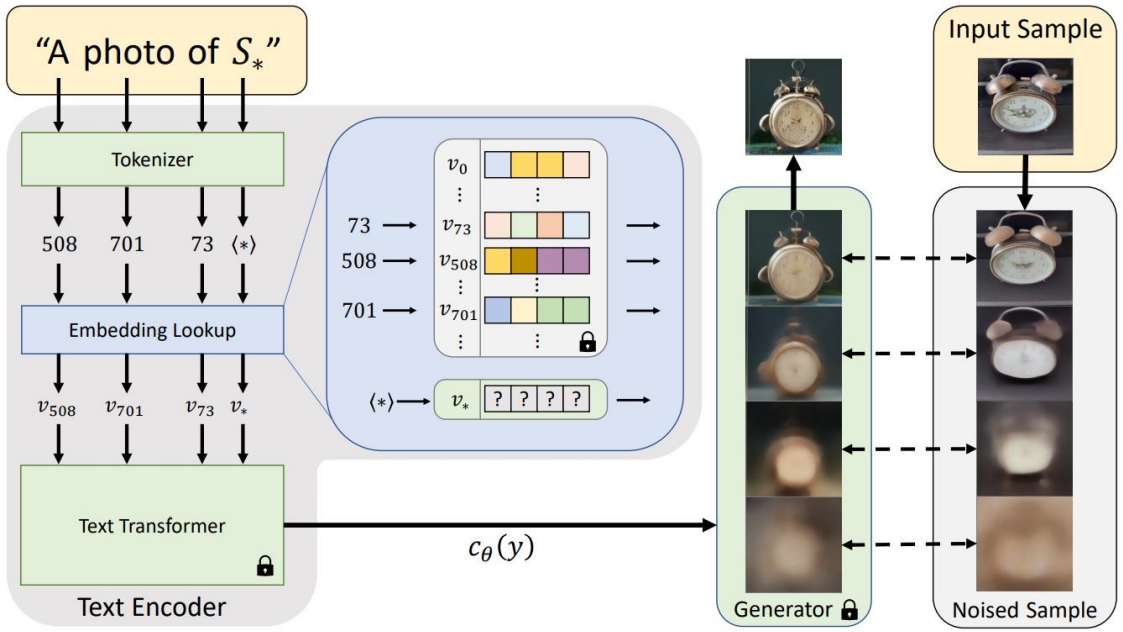
- Plain and simple, with no text involved
- The model simply generates samples that resemble the training distribution
  - E.g., it can be trained on a flower dataset to generate new flowers

Text to Image (training script)

- Same training objective of the pre-training
- Just like starting from a checkpoint and keep training the model a bit more
- Model parameters are not frozen
  - High sensitivity to hyperparameters
  - Very prone to overfitting and even catastrophic forgetting
  - Very high training cost
    - Best case scenario is at least 24-30GB VRAM depending on parameters (w/ batch size = 1)

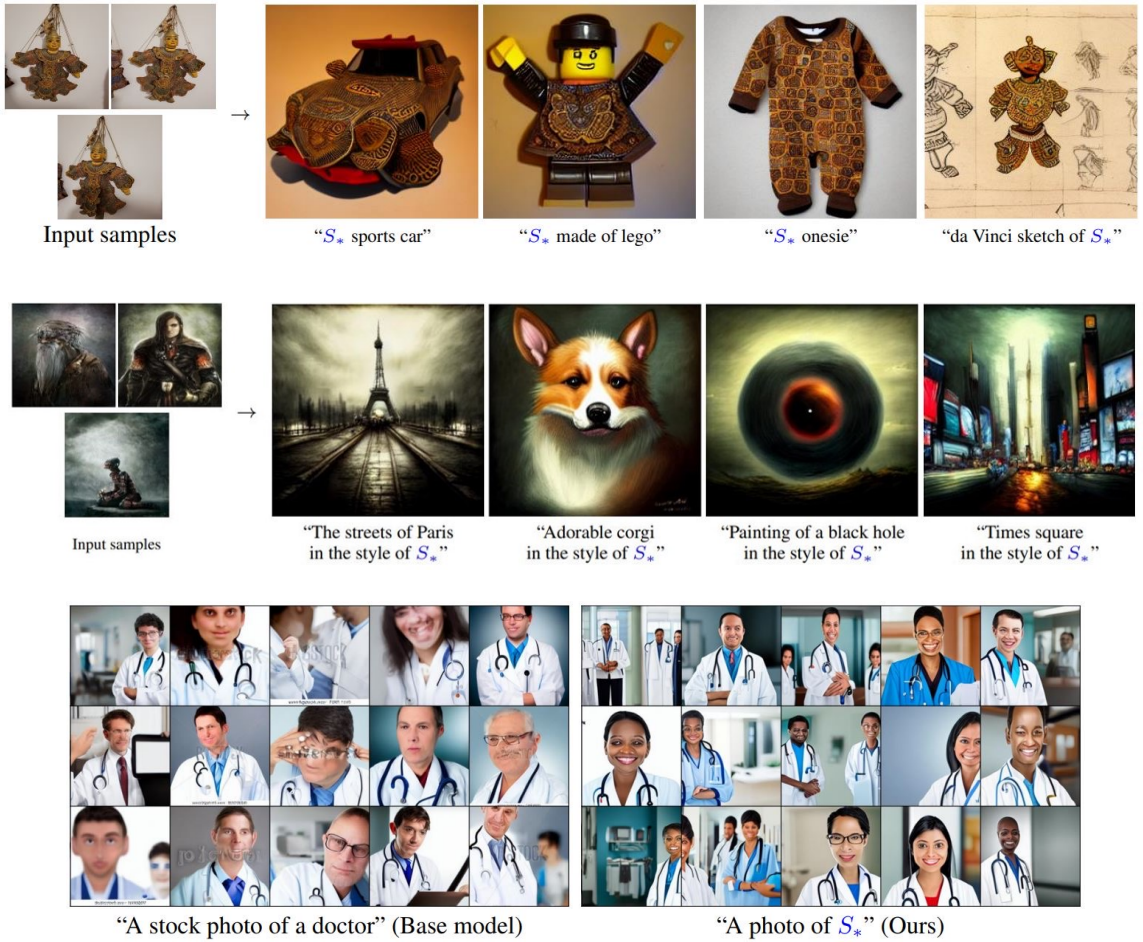
Textual inversion (Gal et al., 2022) (training script)

- A technique to learn new concepts from few samples
  - Get sample images representing the concept
  - Link the concept to an existing token in the model
  - Train the generator on the "new" concept
    - With image + "a photo of [concept]"-like captions



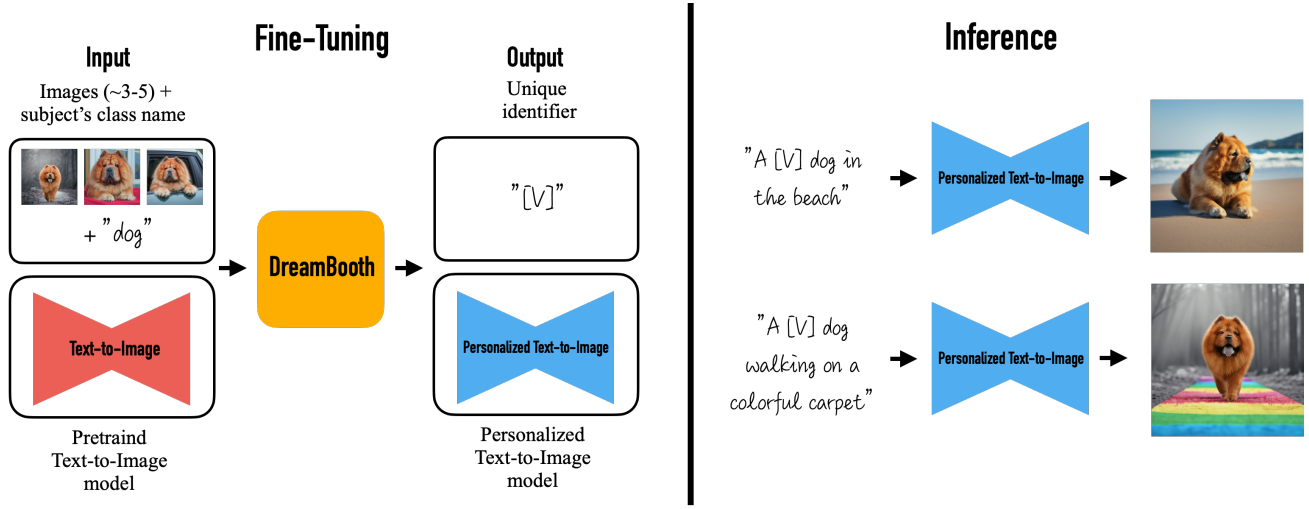
Textual inversion (Gal et al., 2022) (training script)

- Can learn to generate specific concepts
  - personal objects or art styles
- Authors claim it can also be used to reduce the bias of the model (Gal et al., 2022)
  - Associating different embeddings to potentially biased concepts



Dreambooth (Ruiz et al., 2023) (training script)

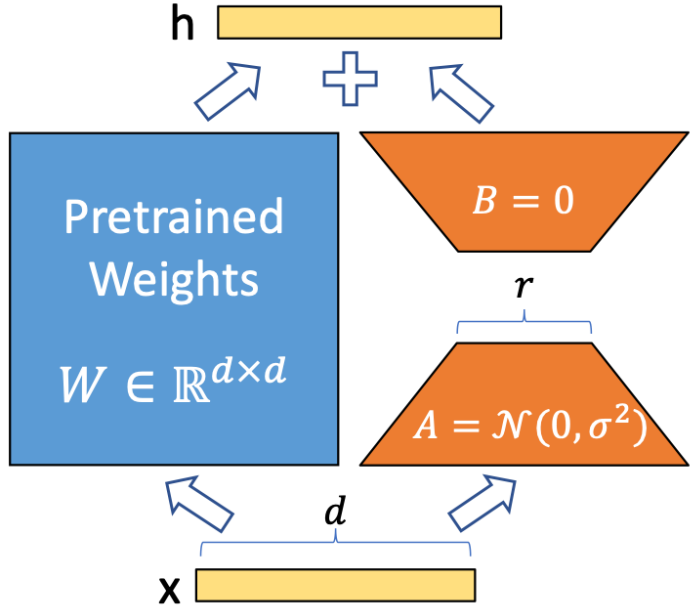
- Similar to Textual Inversion, but with a key difference
  - In Textual Inversion
    - Find a latent space description to express a complex concept that looks like the training images
    - Assigns that latent to a keyword
  - In Dreambooth
    - Train a model N steps to learn a new keyword given training images



- Extremely expensive to train, just like text-to-image

LoRA

- First introduced for memory-efficient adaptation of LMs (Hu et al., 2022)
- A rank decomposition matrix is added to each layer's attention in the LM
  - Update matrix
- Only the matrices are trained on new data, the rest of the model is kept frozen
  - Just have to train a few million parameters



LoRA for stable-diffusion

- Update matrices are injected into the cross-attention modules

- During training
  - only the matrices are updated
  - Training is waaay faster
  - It can be done on consumer-grade hardware (~11GB VRAM should be enough)
  - Trained weights are smaller
    - ~3MB files

- At inference
  - Load the original model
  - Load the LoRA weights
  - Choose the scale of cross-attention weights
    - 0.0 uses the original model, 1.0 uses only the decomposition matrices
  - Prompt the model

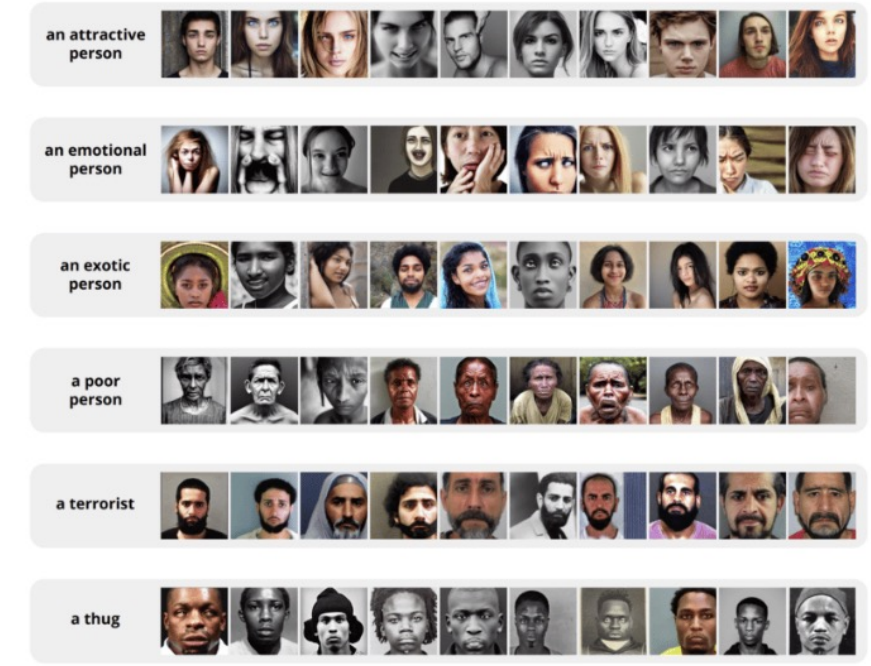
LoRA demo

- Base model: Stable-Diffusion 2.1
- Dataset: lambdalabs/pokemon-blip-captions
  - Pokémon images and captions (automatically generated with BLIP)
- Training results: *"baby red dragon with yellow tail"* (epoch 1-100)

- Let's see how we got there

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 art? The prompt writer? The AI company that made the model? The artists on which the model was trained on?…
  - **Technical limitations**
    - Photorealism, text generation, compositionality…



Conclusions

- The growth of VL models in the last few years is staggering
  - Model sizes and performances
- Not only V & L
  - Also audio, video, etc…
- Open the door to many research opportunities for AI practitioners
  - Core NLP tasks
  - Core Vision tasks

  } as VL tasks
- They give NLP many new challenges to face
  - E.g., how to address compositionality in generation, how to improve on the L of VLMs in general, how to do it and stay open at the same time… just to name a few!

THANK YOU!

Notebooks and Slides here +

Questions?

References

- Gan, Zhe, et al. "Vision-language pre-training: Basics, recent advances, and future trends." *Foundations and Trends® in Computer Graphics and Vision* 14.3–4 (2022): 163-352.
- Wang, Zirui, et al. "Simvlm: Simple visual language model pretraining with weak supervision." *arXiv preprint arXiv:2108.10904* (2021).
- Li, Junnan, et al. "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation." *International Conference on Machine Learning*. PMLR, 2022.
- Li, Junnan, et al. "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models." *arXiv preprint arXiv:2301.12597* (2023).
- Wang, Jianfeng, et al. "Git: A generative image-to-text transformer for vision and language." *arXiv preprint arXiv:2205.14100* (2022).
- Yu, Jiahui, et al. "Coca: Contrastive captioners are image-text foundation models." *arXiv preprint arXiv:2205.01917* (2022).
- Alayrac, Jean-Baptiste, et al. "Flamingo: a visual language model for few-shot learning." *Advances in Neural Information Processing Systems* 35 (2022): 23716-23736.

References (cont.d)

- Chen, Jun, et al. "Visualgpt: Data-efficient adaptation of pretrained language models for image captioning." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
- Luo, Ziyang, et al. "VC-GPT: Visual Conditioned GPT for End-to-End Generative Vision-and-Language Pre-training." *arXiv preprint arXiv:2201.12723* (2022).
- Dou, Zi-Yi, et al. "Coarse-to-fine vision-language pre-training with fusion in the backbone." *arXiv preprint arXiv:2206.07643* (2022).
- Desai, Karan, and Justin Johnson. "Virtex: Learning visual representations from textual annotations." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.
- Tsimpoukelli, Maria, et al. "Multimodal few-shot learning with frozen language models." *Advances in Neural Information Processing Systems* 34 (2021): 200-212.
- Mañas, Oscar, et al. "MAPL: Parameter-Efficient Adaptation of Unimodal Pre-Trained Models for Vision-Language Few-Shot Prompting." *arXiv preprint arXiv:2210.07179* (2022).
- Mokady, Ron, Amir Hertz, and Amit H. Bermano. "Clipcap: Clip prefix for image captioning." *arXiv preprint arXiv:2111.09734* (2021).

References (cont.d)

- Li, Liunian Harold, et al. "Visualbert: A simple and performant baseline for vision and language." arXiv preprint arXiv:1908.03557 (2019).
- Singh, Amanpreet, et al. "Flava: A foundational language and vision alignment model." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
- Lu, Jiasen, et al. "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks." Advances in neural information processing systems 32 (2019).
- Xu, Xiao, et al. "Bridge-Tower: Building Bridges Between Encoders in Vision-Language Representation Learning." arXiv preprint arXiv:2206.08657 (2022).
- Tan, Hao, and Mohit Bansal. "Lxmert: Learning cross-modality encoder representations from transformers." arXiv preprint arXiv:1908.07490 (2019).
- Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021).

References (cont.d)

- Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.
- Fürst, Andreas, et al. "Cloob: Modern hopfield networks with infoloob outperform clip." Advances in neural information processing systems 35 (2022): 20450-20468.
- Jia, Chao, et al. "Scaling up visual and vision-language representation learning with noisy text supervision." International Conference on Machine Learning. PMLR, 2021.
- Li, Yangguang, et al. "Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm." arXiv preprint arXiv:2110.05208 (2021).
- Goodfellow, Ian, et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.
- Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015).

References (cont.d)

- Pan, Xingang, et al. "Drag Your GAN: Interactive Point-based Manipulation on the Generative Image Manifold." *arXiv preprint arXiv:2305.10973* (2023).
- Ramesh, Aditya, et al. "Zero-shot text-to-image generation." *International Conference on Machine Learning*. PMLR, 2021.
- Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer International Publishing, 2015.
- Gal, Rinon, et al. "An image is worth one word: Personalizing text-to-image generation using textual inversion." *arXiv preprint arXiv:2208.01618* (2022).
- Ruiz, Nataniel, et al. "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.