Computer Graphics: Image from text

The problem: Image from text

- We want to generate an entire image from a text description.
  - As opposed to generating an image from another image, which we did for style transfer.
  - Or filling in holes in an image, which we did for inpainting.

Style transfer

- **Inputs**:
  - Text description of the image to generate
- **Outputs**:
  - An image matching the description
- **Performance measure**:
  - User satisfaction
  - Image quality (resolution, realism, etc.)
  - Relevance to the text description

How would we hack a solution?

- Search for images matching the text description
  - "a black cat sitting on a red sofa"
  - search for "black cat" and "red sofa" and combine the results
- You need to find really close matches for this to work, and even then it is not guaranteed to produce a good result.

How it works: diffusion models

- Most current image generation models are so called "diffusion models"
- They work by starting with random noise and iteratively refining it to produce an image that matches the text description
- The model is trained on a large dataset of images and their corresponding text descriptions, learning to generate images that match the descriptions
- The iterative refinement process makes these expensive **at inference time**

Diffusion vs flow matching

- Actually there are some newer models that use a different approach called "flow matching",
  - It is more efficient at inference time
  - But the general idea of starting with noise and refining it is similar
- Some of most popular models include:
  - DALL-E 2 (OpenAI)
  - Stable Diffusion (Stability AI)
    - Although it has diffusion in the name, in newer versions it uses flow matching
  - Midjourney (Midjourney Inc.)

How it works: steering image generation

- When we described diffusion, we said that it basically samples from a distribution of training images.
- But that is a very wide range. How do we steer it towards the specific image we want?
- Another problem, is that the image we want might not be in the training set:
  - "a black cat sitting on a red sofa" might in the training set
  - "a six legged cat performing a circus act on a bull on planet Mongo" is not likely to be in the training set

How it works: steering image generation with a classifier

- Let us think about how we would do it if we were to hack a solution:
  - During our iterative refinement process, we always generate several images.
  - We build a classifier that can evaluate how well each image matches the text description.
  - We select the image that best matches the description and use it as the starting point for the next iteration of refinement.

How it works: classifier free guidance

- classifier guidance is effective but it requires training a separate classifier, which can be expensive and time-consuming to train.
- A more efficient approach is called "classifier free guidance", which does not require a separate classifier
- Instead, the model is trained to generate images both with and without conditioning on the text description.
- During inference, we can control the strength of the conditioning by adjusting a parameter called the "guidance scale".
- A higher guidance scale means that the model will generate images that are more closely aligned with the text description, while a lower guidance scale allows for more creativity and diversity in the generated images.

Vision-Language Models

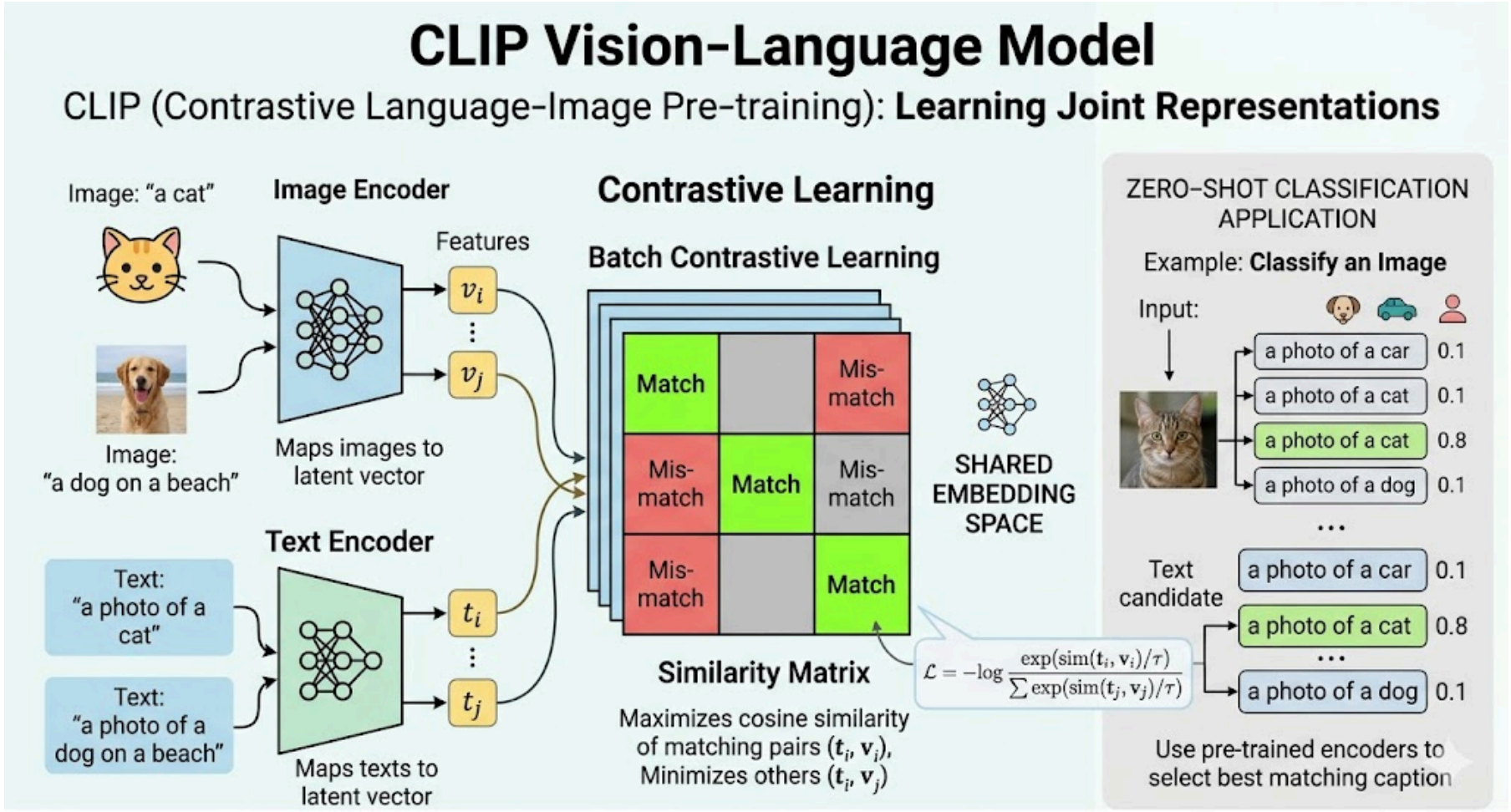
- The models we have discussed so far are primarily focused on generating images from text descriptions.
- However, there are also models that are designed to understand and generate both images and text, known as vision-language models.

Encoders and latent spaces

- In this class we often talked about "encoders". These are models that take some input (like an image or a text description) and convert it into a vector representation in a **latent space**.
- Usually we have a latent space for images and a latent space for text etc. But in vision-language models, we can have a shared latent space where both images and text are represented.
- The idea is that the text "dog" images of "dog" sit close to each other in the latent space, and far from images of "cat" or "car".

Vision-Language Models: CLIP

- CLIP (Contrastive Language-Image Pretraining) is a vision-language model developed by OpenAI that can understand and generate both images and text.
- It is trained on a large dataset of images and their corresponding text descriptions, learning to associate images with their descriptions and vice versa.
- CLIP can be used for a variety of tasks, including image classification, image generation, and zero-shot learning, where the model can recognize and generate images for concepts it has never seen during training.


CLIP Vision-Language Model
CLIP (Contrastive Language-Image Pre-training): Learning Joint Representations
Image: "a cat"
Image Encoder
Features
Image:
"a dog on a beach"
Maps images to
latent vector
Text Encoder
Text:
"a photo of a
cat"
Text:
"a photo of a
dog on a beach"
Maps texts to
latent vector
Contrastive Learning
Batch Contrastive Learning
Match
Mis-
match
Mis-
match
Match
Mis-
match
Mis-
match
Match
SHARED
EMBEDDING
SPACE
Similarity Matrix
Maximizes cosine similarity
of matching pairs (t_i, v_i),
Minimizes others (t_i, v_j)
ZERO-SHOT CLASSIFICATION
APPLICATION
Example: Classify an Image
Input:
a photo of a car 0.1
a photo of a cat 0.1
a photo of a cat 0.8
a photo of a dog 0.1
Text
candidate
a photo of a car 0.1
a photo of a cat 0.8
a photo of a dog 0.1
Use pre-trained encoders to
select best matching caption

Zero-shot learning with CLIP

- You have trained an image classifier on a dataset of cats and dogs and extended it to a different animal.
- This required a certain amount of data (eg. 50 pictures of the new animal).
- We mentioned that there are techniques that can learn from very few examples (eg. 5 pictures of the new animal or even just one).
    - These techniques are called **few-shot learning** and **one-shot learning**.
- What about **zero-shot learning**? Can we recognize a new animal **without any examples**?

Zero-shot learning with CLIP

- CLIP can do zero-shot learning because it has a shared latent space for images and text.
- If we want to recognize a new animal, we can simply provide a text description of the animal (eg. "a picture of a green giraffe")
- Pass the picture of the new animal through the image encoder to get its latent representation, and pass the text description through the text encoder to get its latent representation.
- If the two representations are close to each other in the latent space, we can conclude that the image matches the text description, even if the model has never seen that specific animal during training.

Application: generate pictures to illustrate an article

- We can use image generation models to create illustrations for articles, blog posts, or social media content.
- This can be especially useful for creating custom images that are not available in stock photo libraries.
- The picture of on the class website was generated like this.

Future directions: painting and drawing

- Even if we generate pictures from text that look like paintings or drawings, they are still generated using the same underlying techniques as photorealistic images.
    - They start from denoising random noise, not actually "painting" or "drawing" in the way that a human artist would.
- There is ongoing research into developing models that can mimic the process of painting or drawing
    - It might involve generating images in a step-by-step manner, where the model decides what to draw or paint at each step, rather than generating the entire image in one go.
    - Deciding on every line or brush stroke, rather than generating the entire image at once.
    - It might even be executed by a robotic arm holding a paintbrush.

Pitfalls and dangers: Generating harmful images

- Image generation models can be used to create harmful or inappropriate content, such as violent or sexually explicit images.
- This can be a concern for both users and developers of these models, as it can lead to misuse and ethical issues.
- Developers of image generation models often implement content filters and moderation systems to prevent the generation of harmful content, but these systems are not perfect and can sometimes fail to catch all inappropriate images.
    - And sometimes they can also mistakenly filter out harmless content, leading to false positives and censorship concerns.

Try it out: generating images on given subjects