Johns Hopkins
Whiting School of Engineering

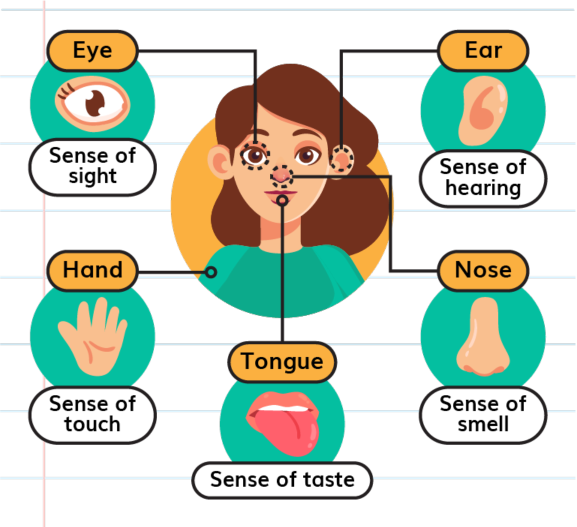
Connecting Language to the World

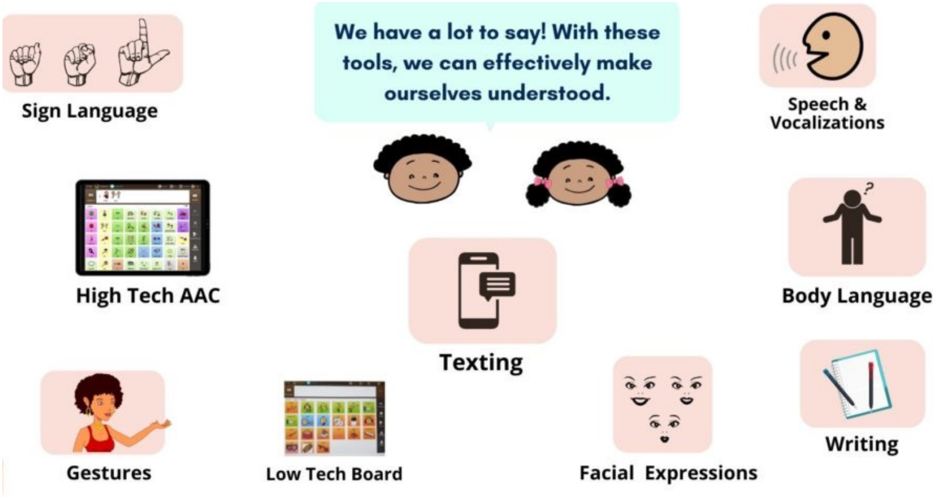
CSCI 601-471/671 (NLP: Self-Supervised Models)

https://self-supervised.cs.jhu.edu/sp2024/

Connecting Language to the World

- So far, we have focused on only “language” as our primary focus.
- But there is there are other modalities in which humans communicate with the world.





Connecting LMs to the World: Chapter Plan

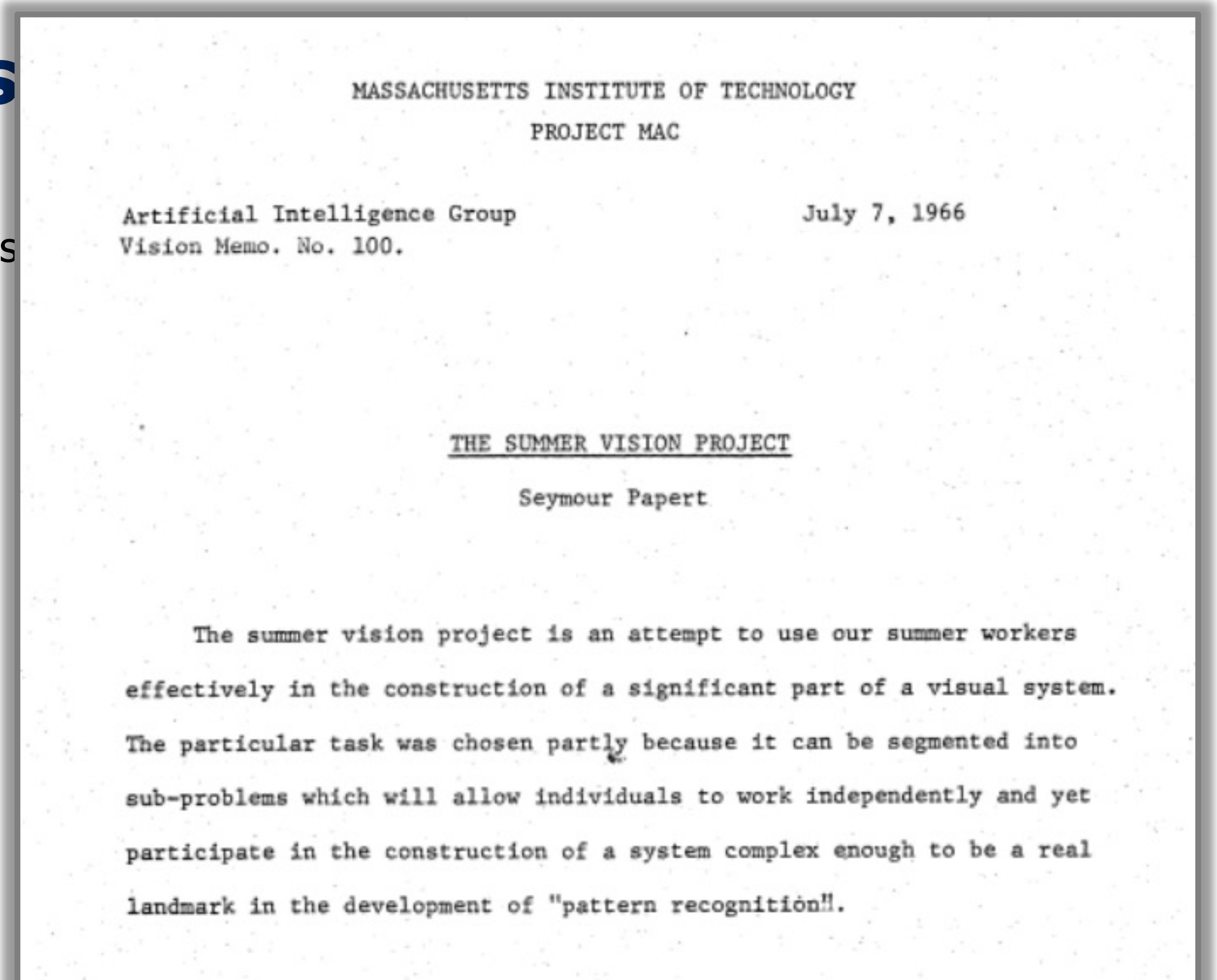
1. Connecting vision and language
2. Generative vision-language models
3. Other modalities [speech, audio, …]
4. From language to code
5. From language to action

Chapter goal: Getting comfortable with thinking about extending LLMs to other modalities of the world and their limits.

Connecting vision and language

Computer Vis

- 1960s - First computer vis



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group   July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Computer Vision Abridged History

- 1960s - First computer vision projects (MIT summer project)
- 2000s - Emergence of “tasks” and benchmarking in computer vision

PASCAL Visual Object Classes (2005-2012)

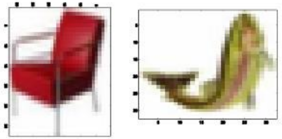
Caltech-101 (2003)

Caltech-256 (2007)

ImageNet (2009)

ImageNet (2009)

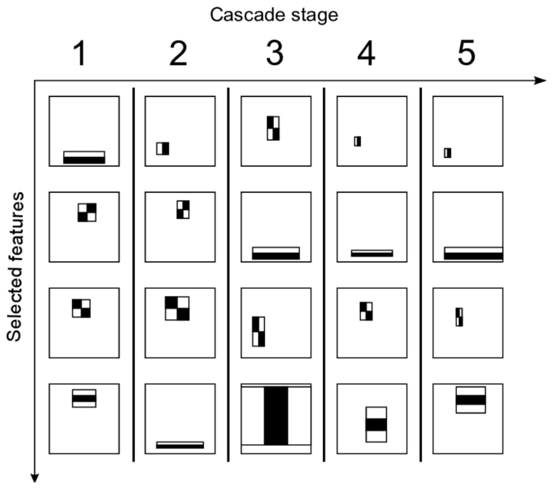
(subset of tiny images)

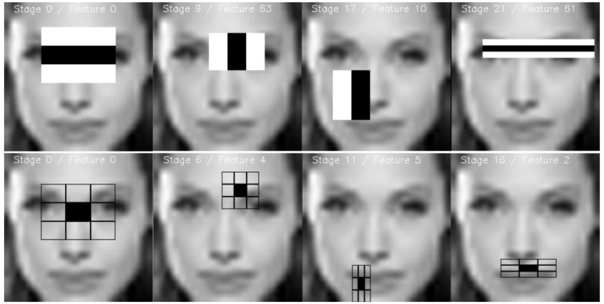


https://en.wikipedia.org/wiki/List_of_datasets_in_computer_vision_and_image_processing

Computer Vision Abridged History

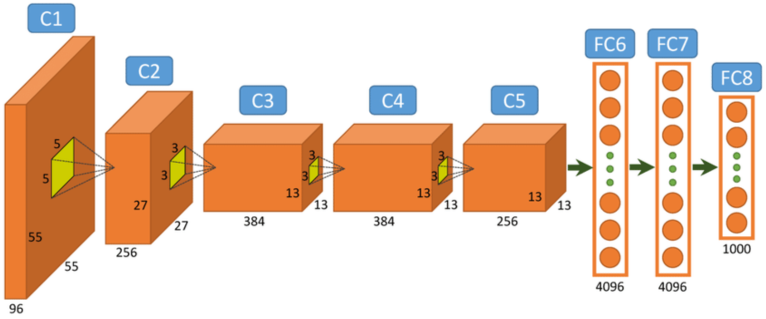
- 1960s - First computer vision projects (MIT summer project)
- 2000s - Emergence of “tasks” and benchmarking in computer vision
- 2000s - Shallow classifiers and feature engineering (e.g., Viola & Jones algorithm)



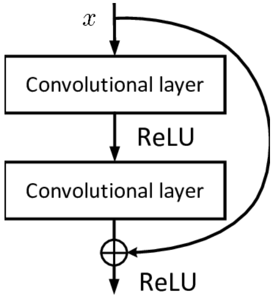


Computer Vision Abridged History

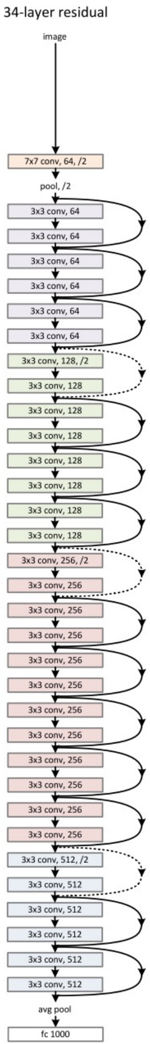
- 1960s - First computer vision projects (MIT summer project)
- 2000s - Emergence of "tasks" and benchmarking in computer vision
- 2000s - Shallow classifiers and feature engineering
- 2012 - Deep Learning revolution:
  - Success of Convolutional neural nets in ImageNet



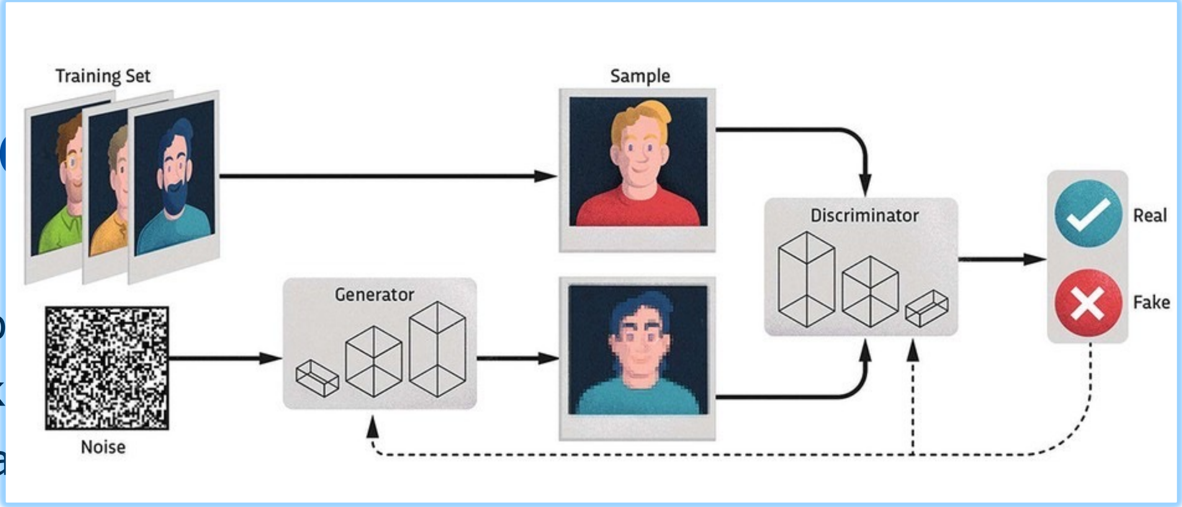
ImageNet Classification with Deep Convolutional Neural Networks, 2012



Deep Residual Learning for Image Recognition, 2015



Computer Visi



- 1960s - First computer visio
- 2000s - Emergence of “task
- 2000s - Shallow classifiers a
- 2012 - Deep Learning revolution:
  - Success of Convolutional neural nets in ImageNet
  - Unification of architectures
  - Rise of image generation (VAEs, GANs, etc.)

Kingma et al. Auto-Encoding Variational Bayes, 2013
Goodfellow et al. Generative Adversarial Networks, 2014

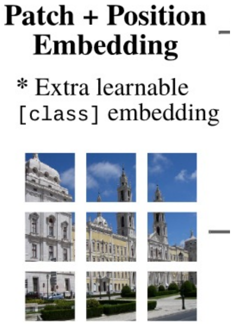
Computer Vision Abridged History

- 1960s - First computer vision projects (MIT summer project)
- 2000s - Emergence of “tasks” and benchmarking in computer vision
- 2000s - Shallow classifiers and feature engineering
- 2012 - Deep Learning revolution:
  - Success of Convolutional neural nets in ImageNet
  - Unification of architectures
  - Rise of image generation (VAEs, GANs, etc.)
- 2020s - Era of Vision Transformer
  - Stronger connection to language
  - Better generative models
  - Further unification of models and tasks

This is where we begin!

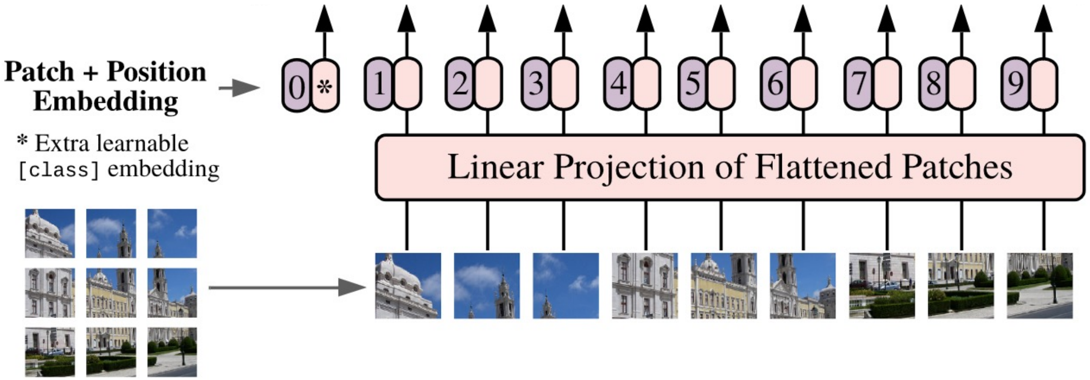
Let's Consider Images – How to Encode?

Vision Transformers

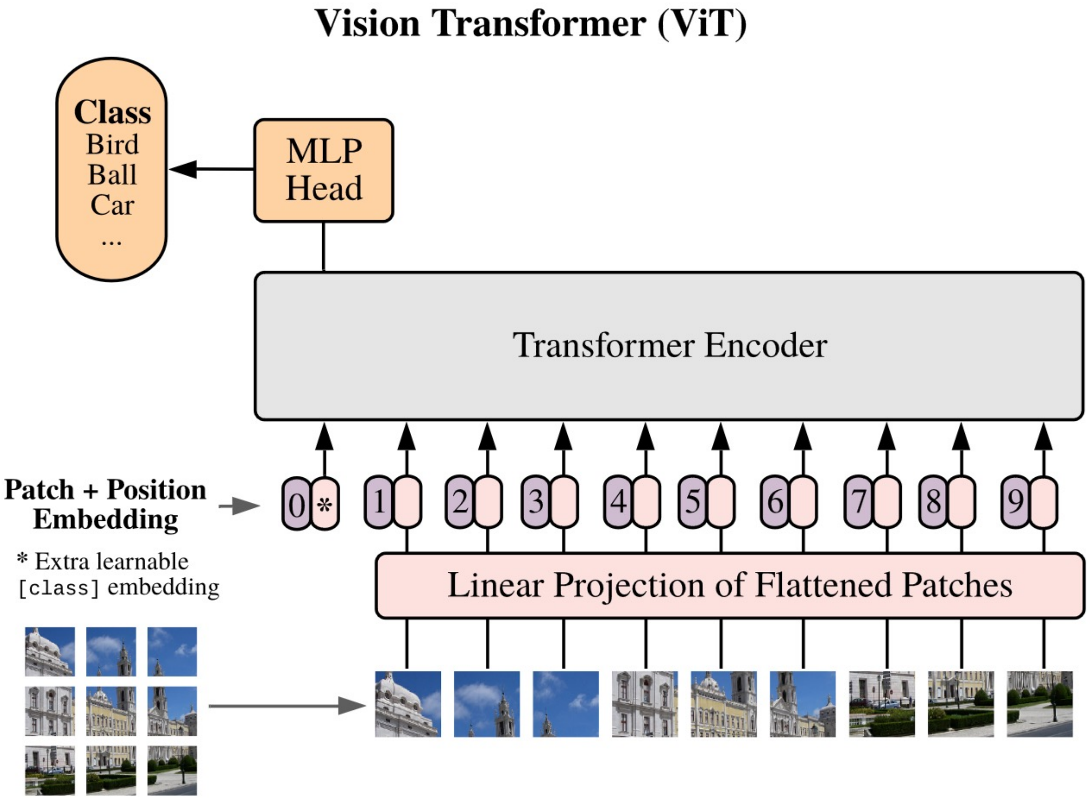


tokenize" the image by cutting it into patches of 16px², and treating each patch as a token, e.g. embedding it into input space

Vision Transformers



Vision Transformers



[An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2020]

What about paired image-text – How to Encode?

Basilica of St. John Lateran

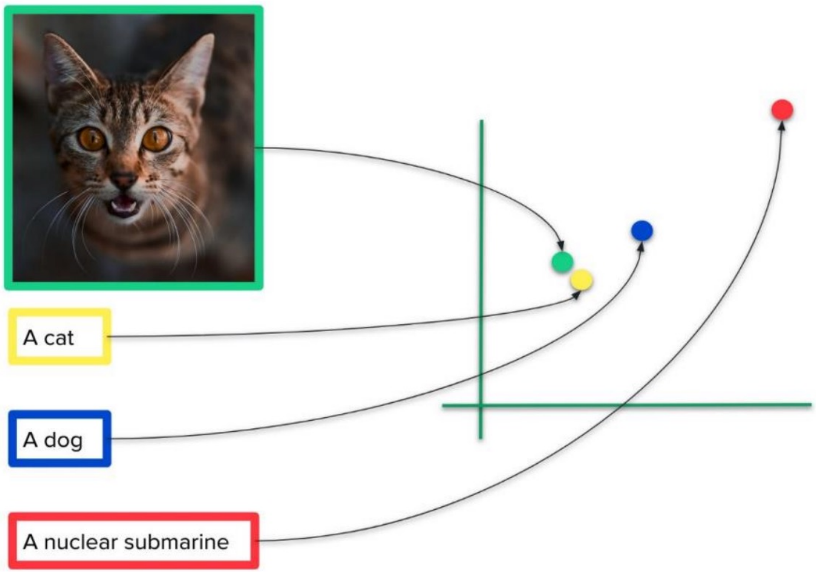
House with Chimeras – Kiev

Johns Hopkins
Whiting School
of Engineering

What about paired image-text – How to Encode?

- The idea is to create a space to represent both semantics of language and image.

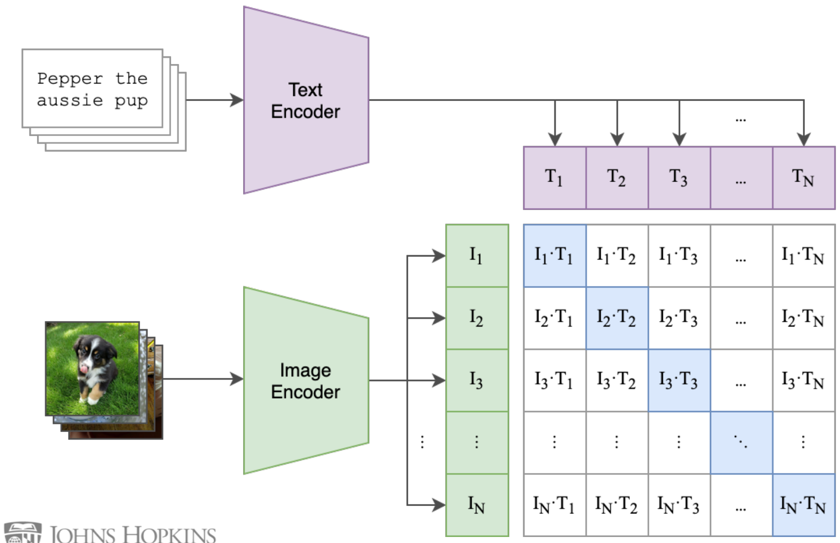
- Then, create a model that can align semantically-equivalent text and images nearby.



Contrastive Language–Image Pre-training (CLIP)

- Training: simple contrastive loss between image-text pairs
  - Learning image representations from web-scale noisy text supervision

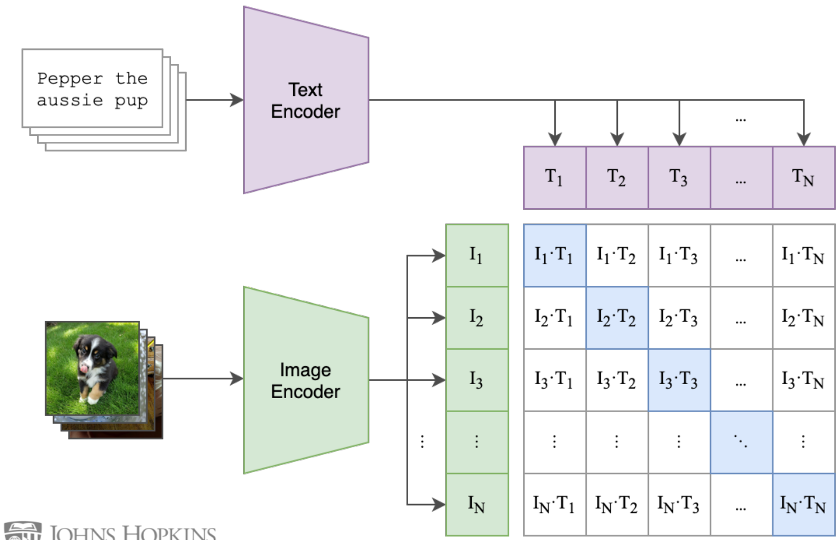
(1) Contrastive pre-training



Contrastive Language–Image Pre-training (CLIP)

- Given a batch of N (image, text) pairs, predict which of the N × N possible (image, text) pairings across a batch occurred.

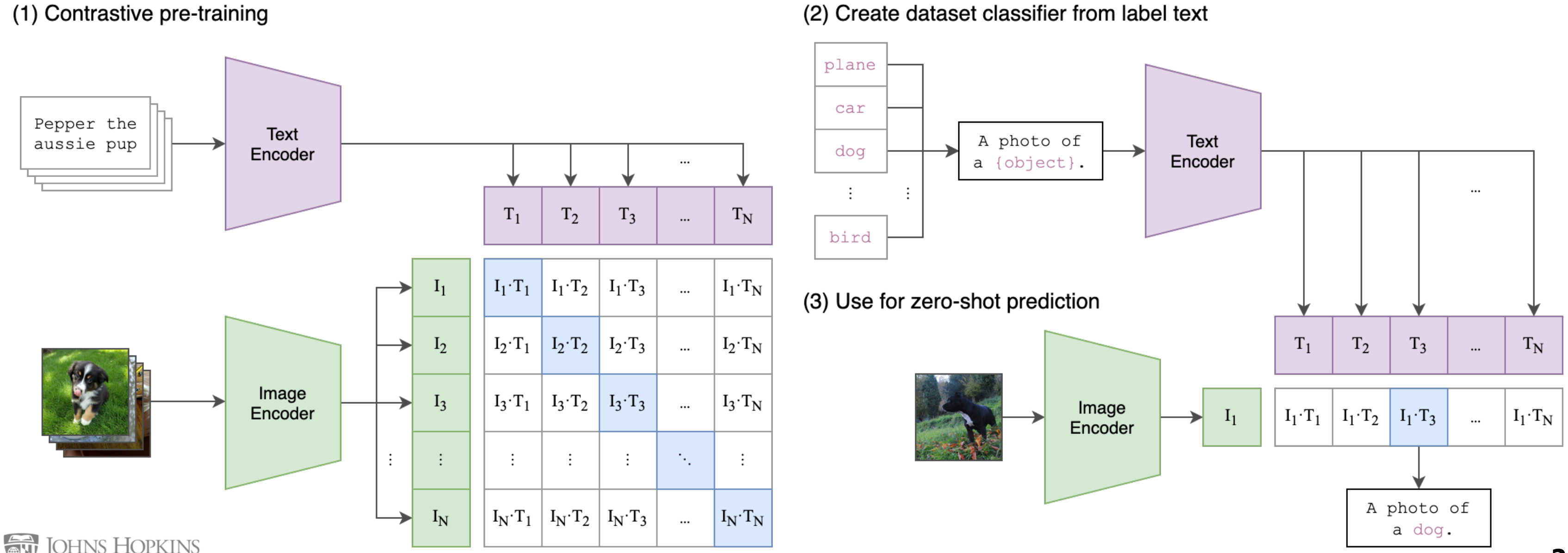
(1) Contrastive pre-training




```
# 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
```

What can CLIP do?

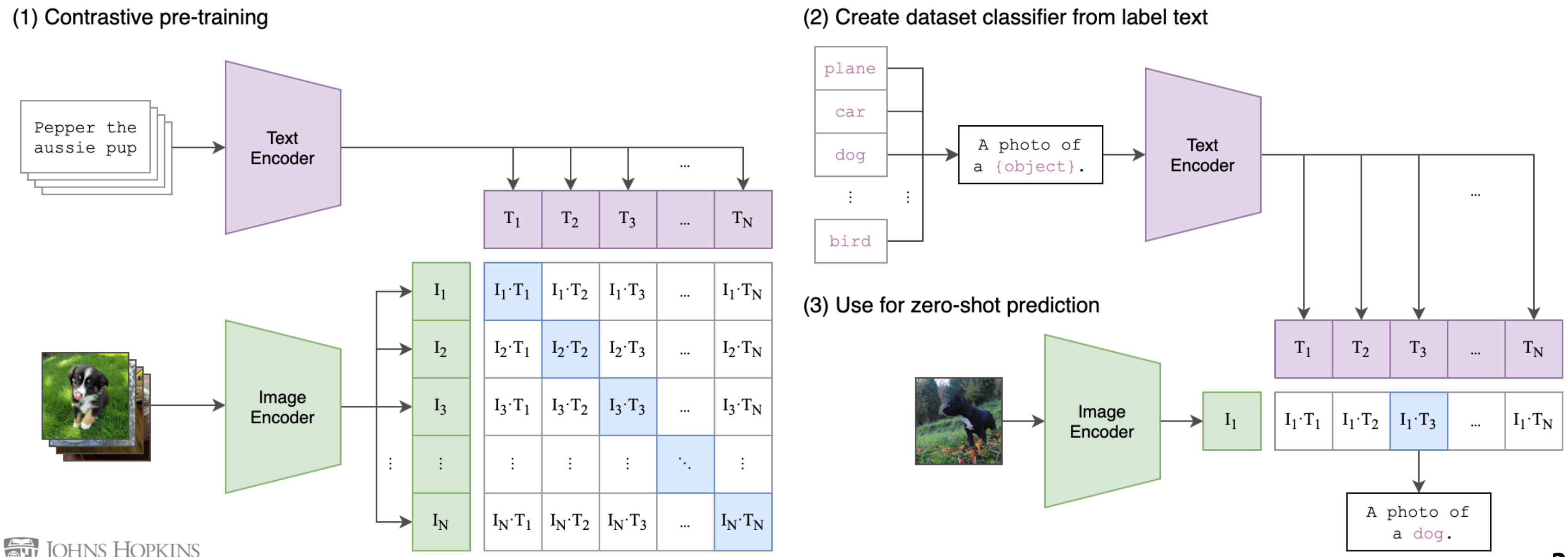
- Downstream: zero-shot image classification and image-text retrieval
  - Image classification can be reformatted as a retrieval task via considering the semantics behind label name.



[Learning Transferable Visual Models From Natural Language Supervision, 2021]

What can CLIP do?

- Consider classifying photos of “dogs” vs “cats”—for each image, check if CLIP predicts text description “a photo of a dog” or “a photo of a cat” is more likely to be paired with it.

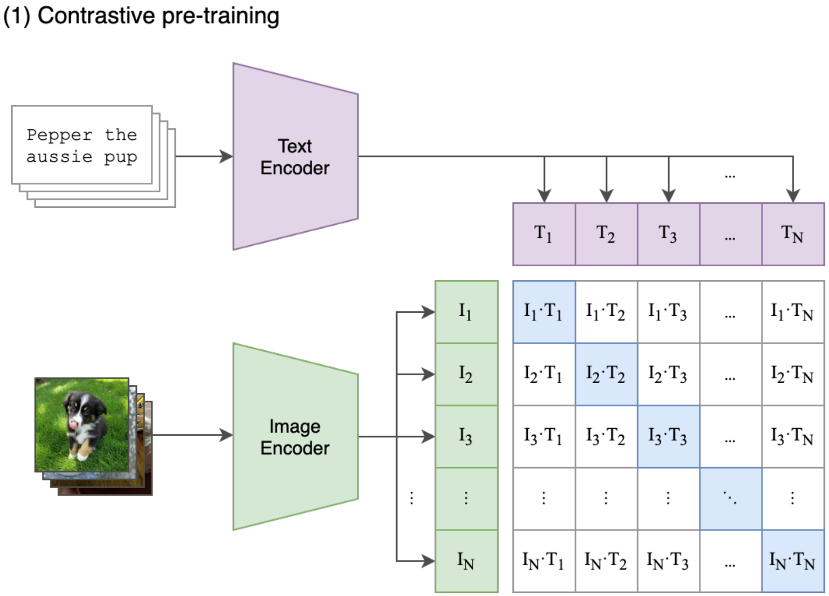


[Learning Transferable Visual Models From Natural Language Supervision, 2021]

What can CLIP do?

CLIP evaluates associations between image-text pairs:

- Image Classification
- Image Searching
- ...



What can CLIP do?

CLIP evaluates associations between image-text pairs:

- Image Classification
- Image Searching
- …

https://huggingface.co/openai/clip-vit-large-patch14

Possible class names (comma-separated)

Johns Hopkins, Stanford, Berkeley, UPenn

Compute

Computation time on cpu: cached

Johns Hopkins	0.992
UPenn	0.008
Stanford	0.000
Berkeley	0.000

What can CLIP do?

CLIP evaluates associations between image-text pairs:

- Image Classification
- Image Searching
- ...

https://huggingface.co/openai/clip-vit-large-patch14

Possible class names (comma-separated)

Johns Hopkins, Stanford, Berkeley, UPenn

Compute

Computation time on cpu: 0.623 s

Johns Hopkins	0.729
UPenn	0.268
Stanford	0.003
Berkeley	0.001

JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

What can CLIP do?

CLIP evaluates associations between image-text pairs:

- Image Classification
- Image Searching
- …

https://huggingface.co/openai/clip-vit-large-patch14

Possible class names (comma-separated)

Johns Hopkins, Stanford, Berkeley, UPenn

Compute

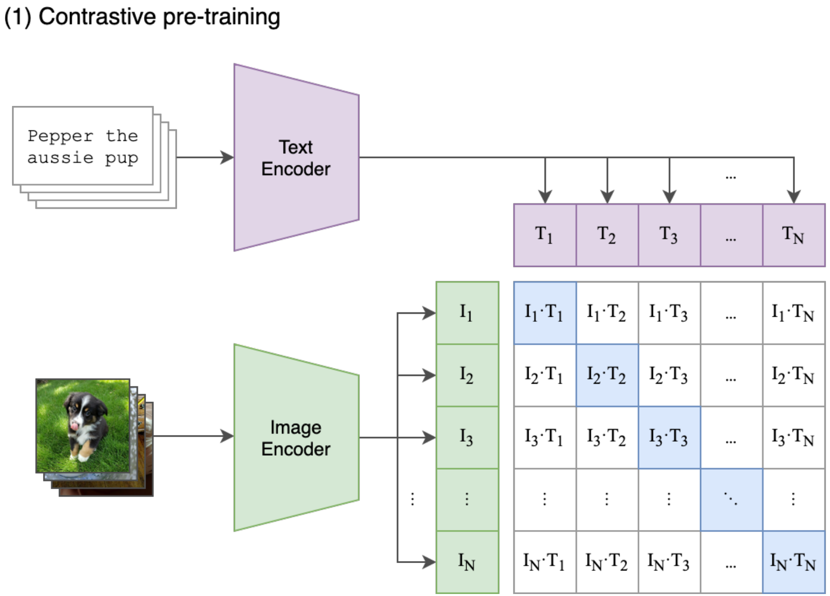
Computation time on cpu: 0.891 s

UPenn	0.928
Johns Hopkins	0.071
Stanford	0.000
Berkeley	0.000

What can't CLIP do?

- No generation capabilities

- Prompting / In-Context Learning
  - Few-shot captioning



What can't CLIP do?

- No generation capabilities

- Prompting / In-Context Learning
    - Few-shot captioning

- CLIP is not perfect.

What can't CLIP do?

https://huggingface.co/openai/clip-vit-large-patch14

- No generation capabilities

- Prompting / In-Context Learning
  - Few-shot captioning

- CLIP is not perfect.

Possible class names (comma-separated)

too many fingers, not enough fingers, just the right amount of fingers

Compute

not enough fingers	0.461
just the right amount of fingers	0.339
too many fingers	0.200

JSON Output   Maximize

What can't CLIP do?

https://huggingface.co/openai/clip-vit-large-patch14

- No generation capabilities
- Prompting / In-Context Learning
  - Few-shot captioning
- CLIP is not perfect.

Possible class names (comma-separated)

too many fingers, not enough fingers, just the right amount of fingers

Compute

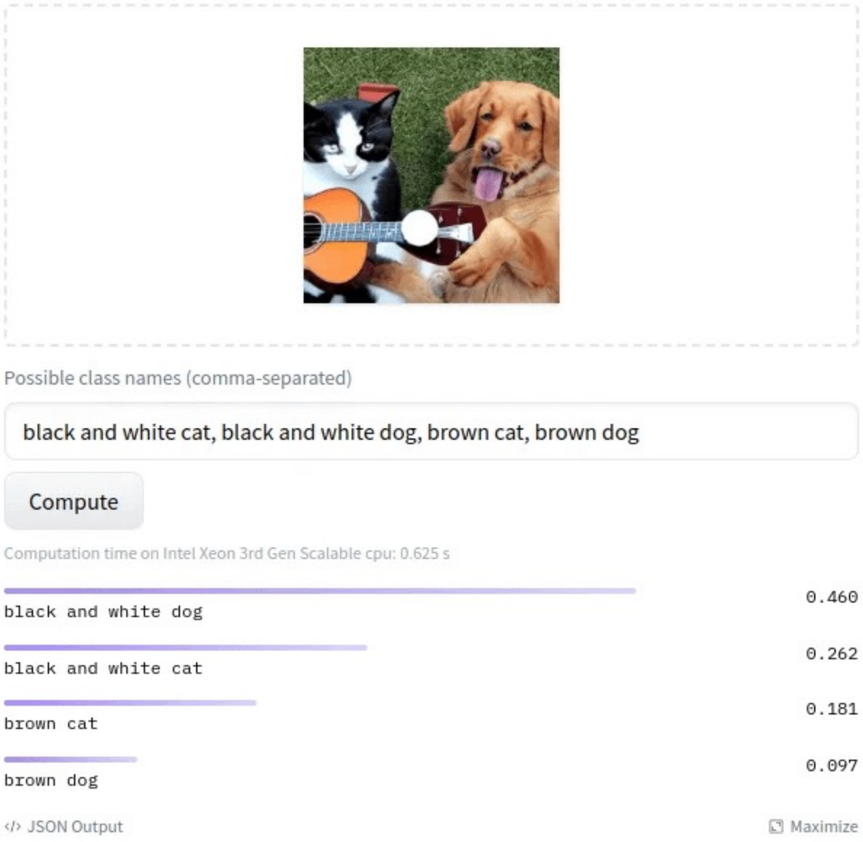
Computation time on cpu: 0.251 s

just the right amount of fingers	0.412
too many fingers	0.316
not enough fingers	0.272

What can't CLIP do?

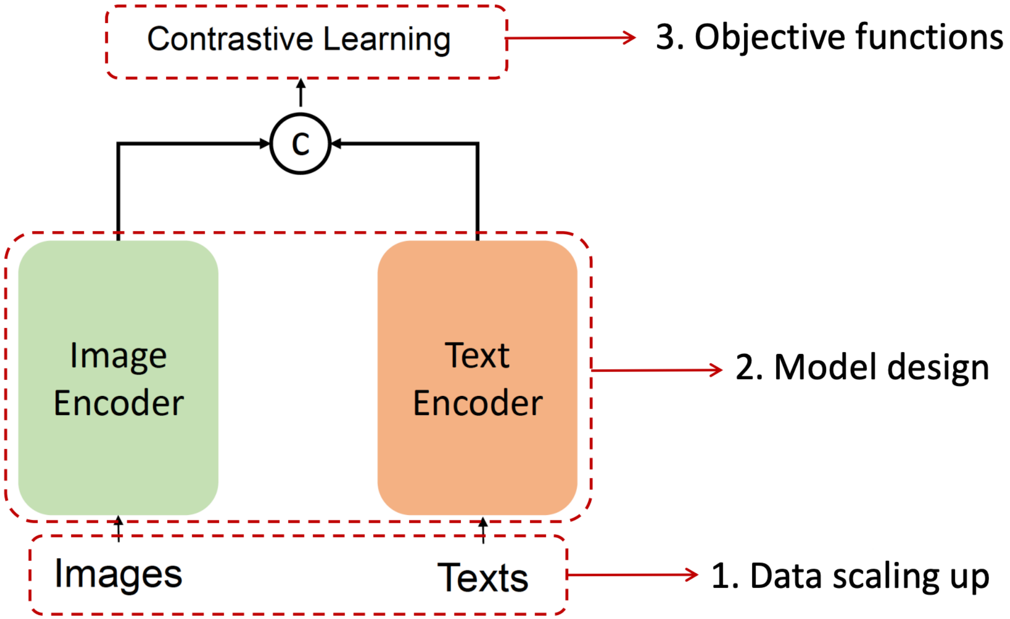
- No generation capabilities
- Prompting / In-Context Learning
  - Few-shot captioning
- CLIP is not perfect.

https://huggingface.co/openai/clip-vit-large-patch14



What happened after CLIP?

- Ton of follow-up work on improve its design:



What happened after CLIP?

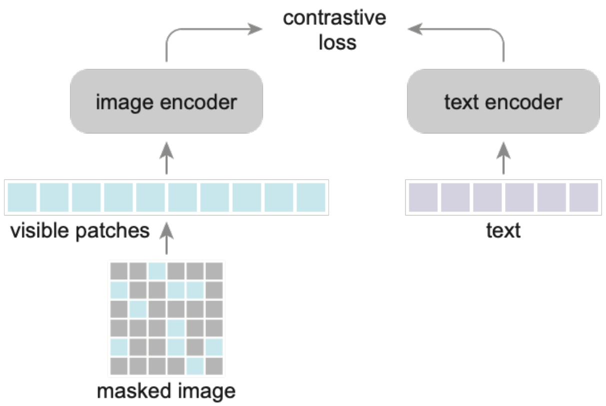
- Open-source reproduction and scaling up
  - Open-source model: OpenCLIP
  - Pre-training on LAION-5B dataset

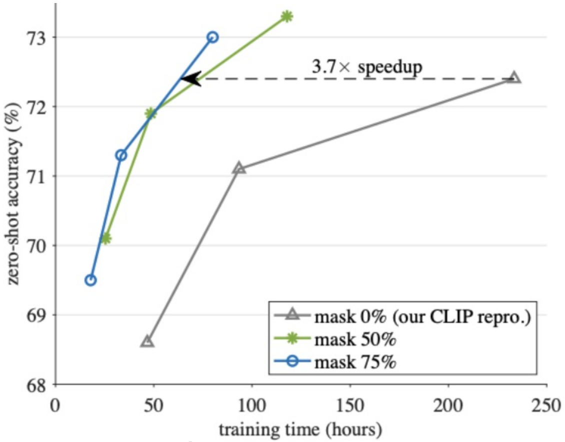
	Data	Arch.	ImageNet	VTAB+	COCO
CLIP [55]	WIT-400M	L/14	75.5	55.8	61.1
Ours	LAION-2B	L/14	75.2	54.6	71.1
Ours	LAION-2B	H/14	78.0	56.4	73.4

Ilharco et al., OpenCLIP, 2021
https://github.com/mlfoundations/open_clip

What happened after CLIP?

- Open-source reproduction and scaling up
- Modifying the training process
  - A notable example is FLIP
    - Randomly masking out image patches
    - Does not hurt the performance, yet it improves the training efficiency



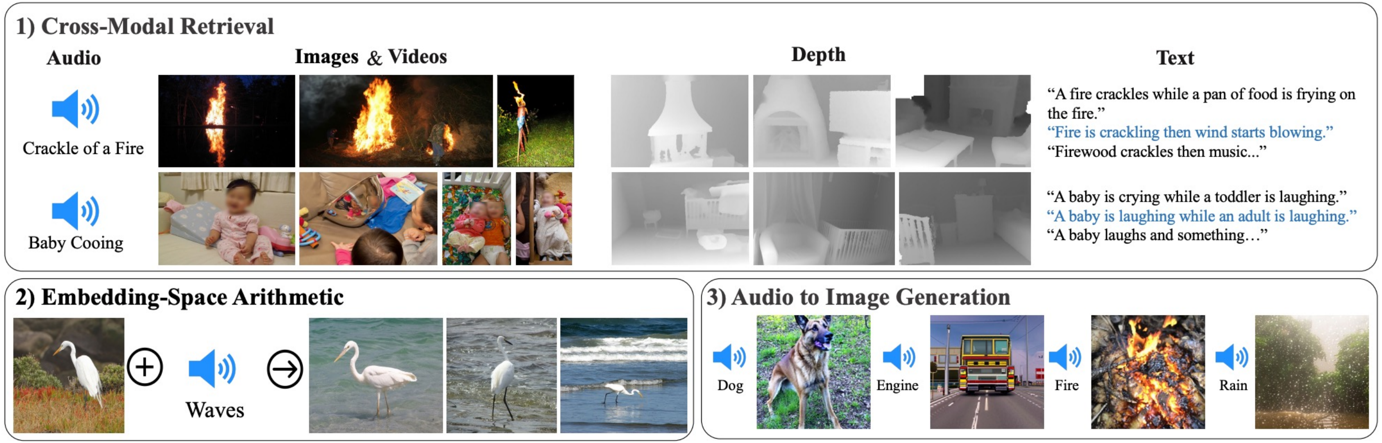


Scaling Language-Image Pre-training via Masking, 2022

What happened after CLIP?

- Open-source reproduction and scaling up
- Modifying the training process
- Adding more modalities:
  - ImageBind: linking 7 modalities

https://facebookresearch.github.io/ImageBind
ImageBind: One Embedding Space To Bind Them All, 2023



What happened after CLIP?

- Open-source reproduction and scaling up
- Modifying the training process
- Adding more modalities:
- Generative models:
    - Text → Image
    - Image, Text → Image
    - Image, Image → Text
    - …

Forthcoming

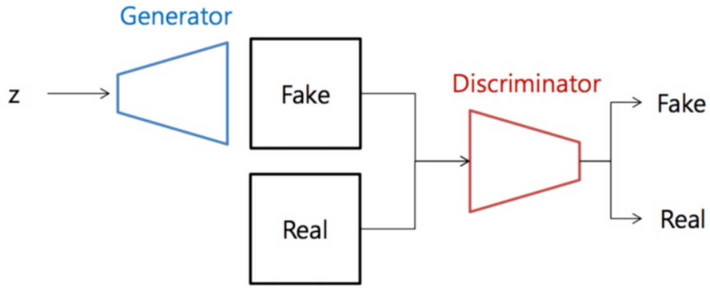
Summary

- The same computational architecture (e.g., Transformer) can represent different modalities.
- One can build models that embed different modalities in the same space.
- Next: generative vision-language models.

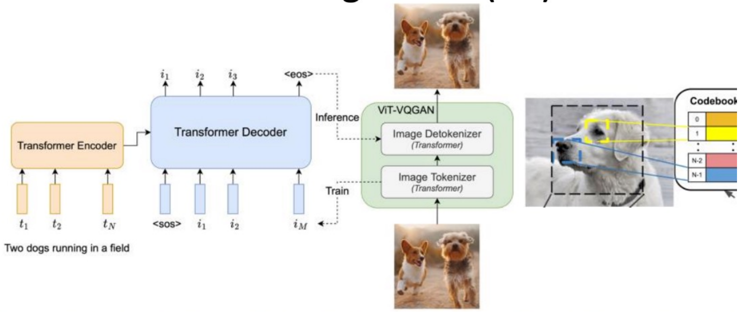
Generative Vision-Language Models

Image Generation Toolkit

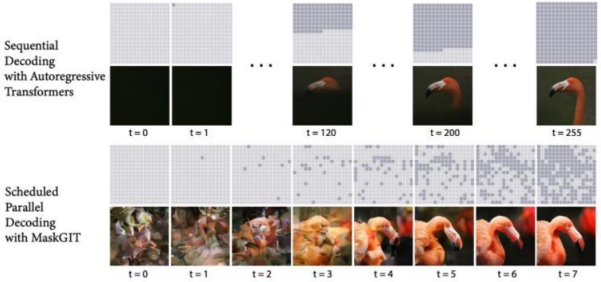
Generative Adversarial Networks (GAN)



Auto-regressive (AR)



Non-AR Transformer



Diffusion

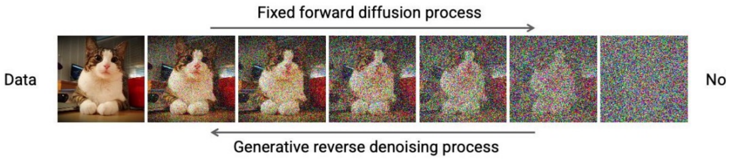


Image Generation Toolkit: Diffusions

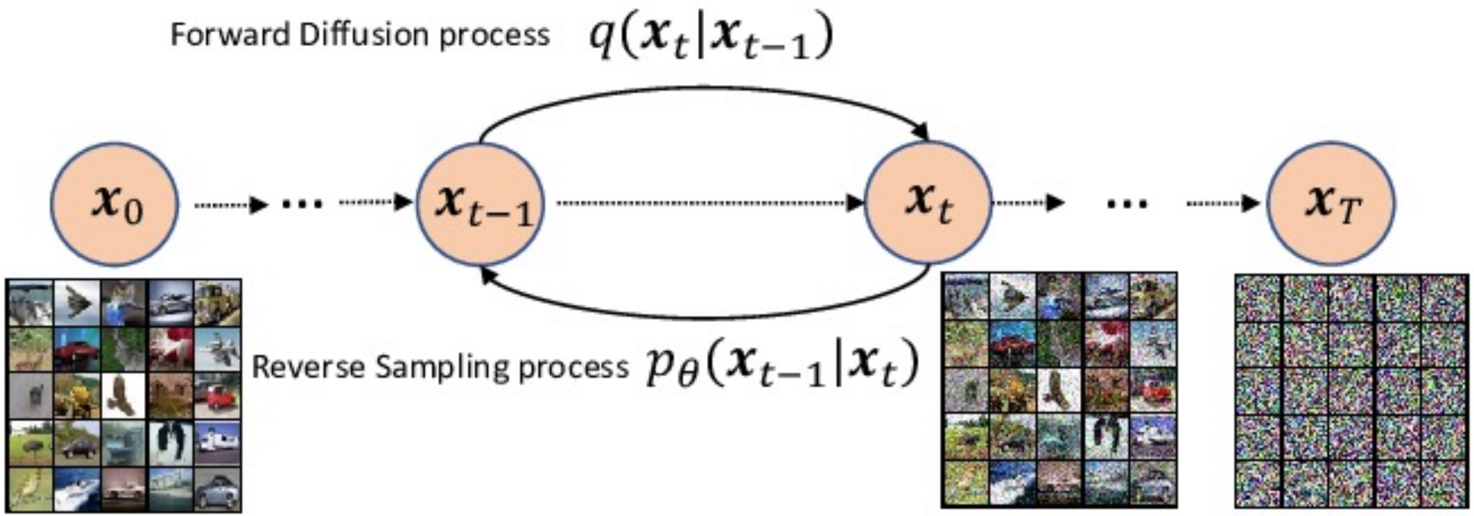
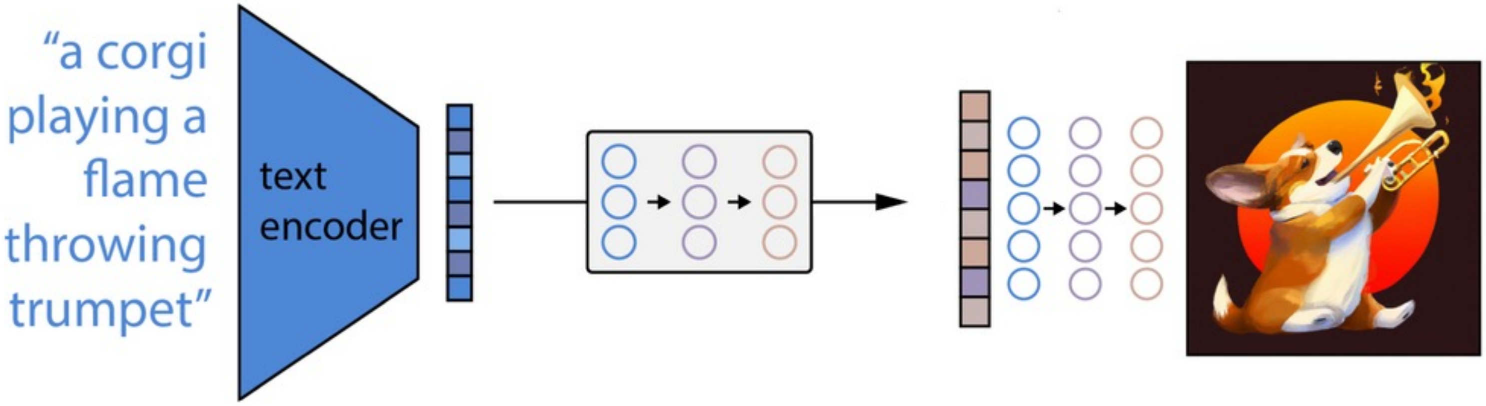
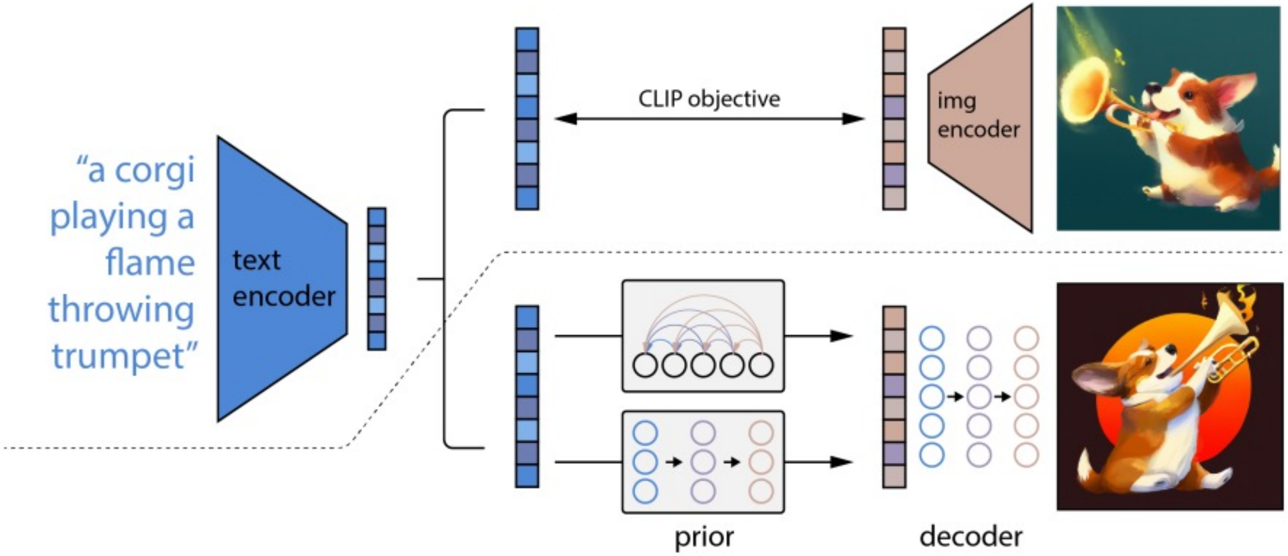


Image generation is out of scope for us. You can learn more by taking a computer vision class or watching the online tutorials. For example: https://cvpr2022-tutorial-diffusion-models.github.io/

Text to Image Generation



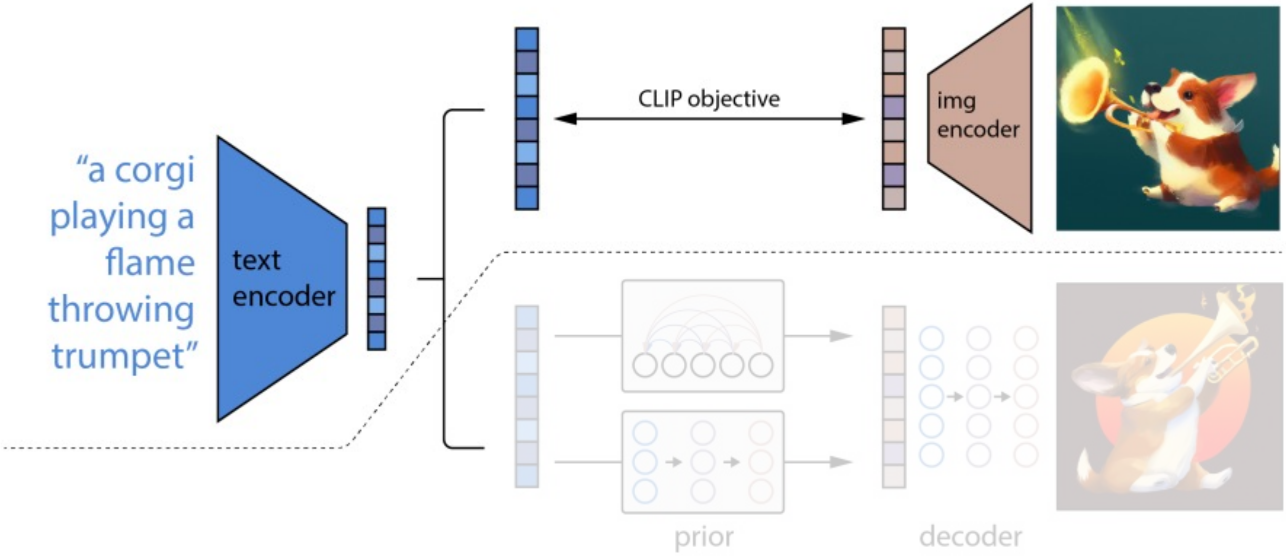
DALL-E



Zero-Shot Text-to-Image Generation, 2021
Hierarchical Text-Conditional Image Generation with CLIP Latents. 2022

DALL-E

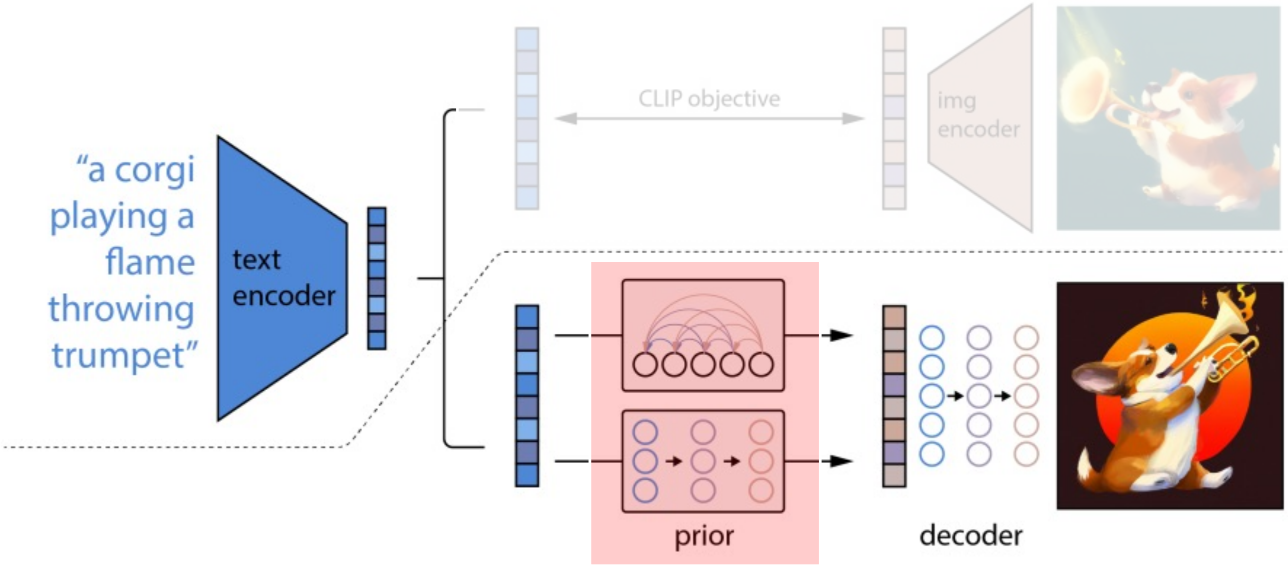
- DALL-E is built on top of a pre-trained CLIP model.
  - This frozen model provides the representations of text and images.



Zero-Shot Text-to-Image Generation, 2021
Hierarchical Text-Conditional Image Generation with CLIP Latents. 2022

DALL-E

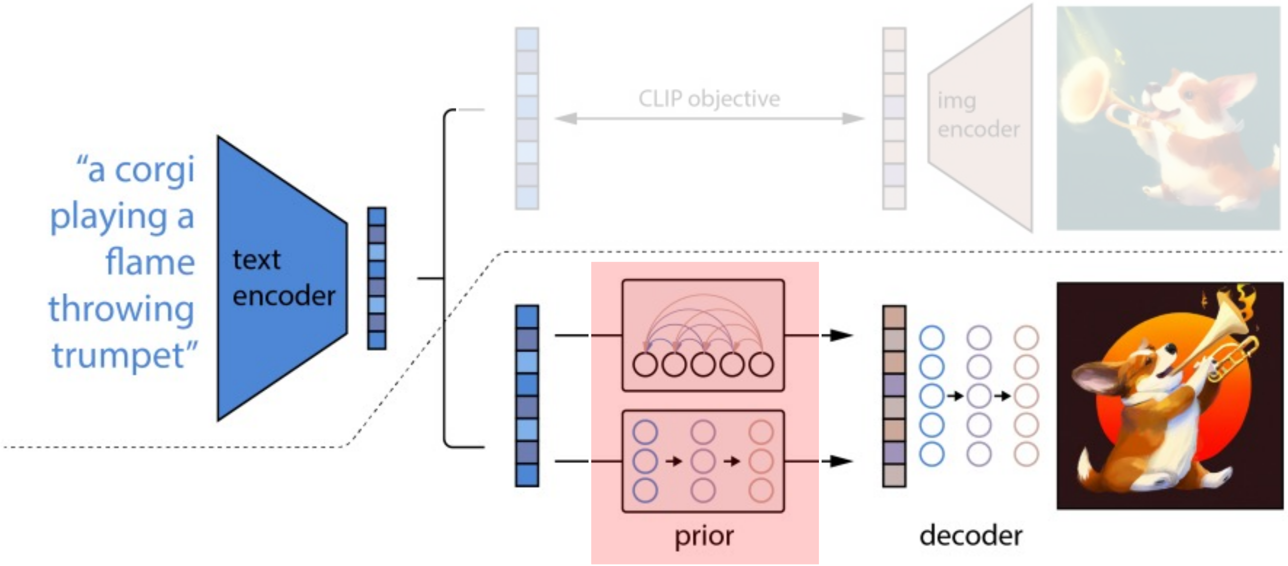
- First, a text prompt is input into a **text encoder** that is trained to map the prompt to a representation space.



Zero-Shot Text-to-Image Generation, 2021
Hierarchical Text-Conditional Image Generation with CLIP Latents. 2022

DALL-E

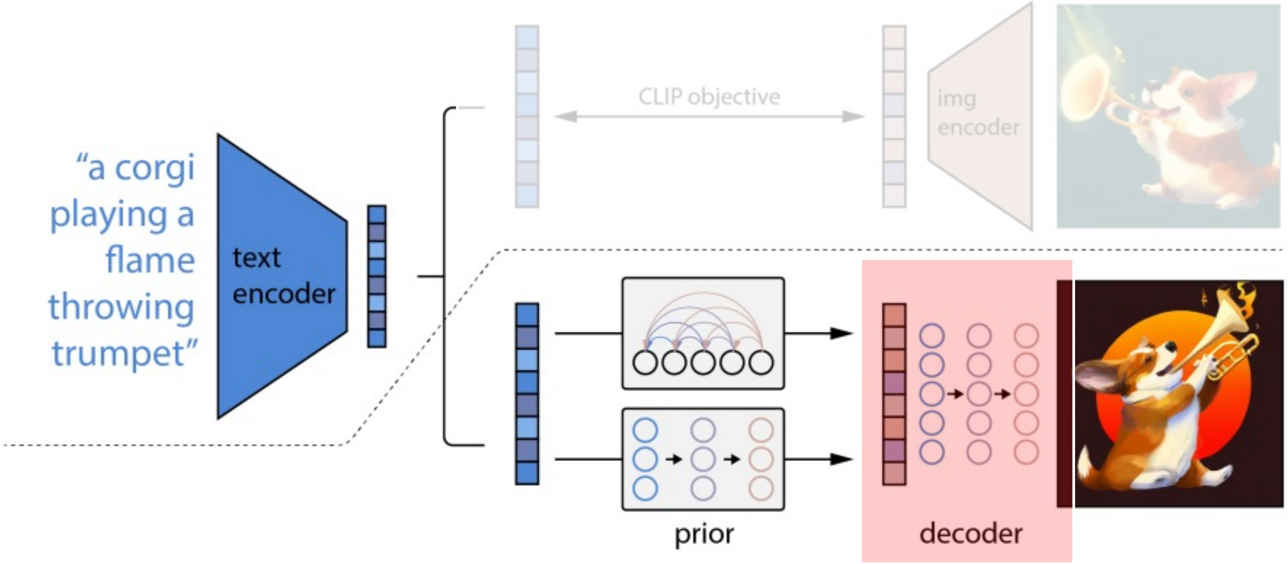
- A **prior** maps the text encoding to a corresponding **image encoding** that captures the semantic information of the prompt contained in the text encoding.



Zero-Shot Text-to-Image Generation, 2021
Hierarchical Text-Conditional Image Generation with CLIP Latents. 2022

DALL-E

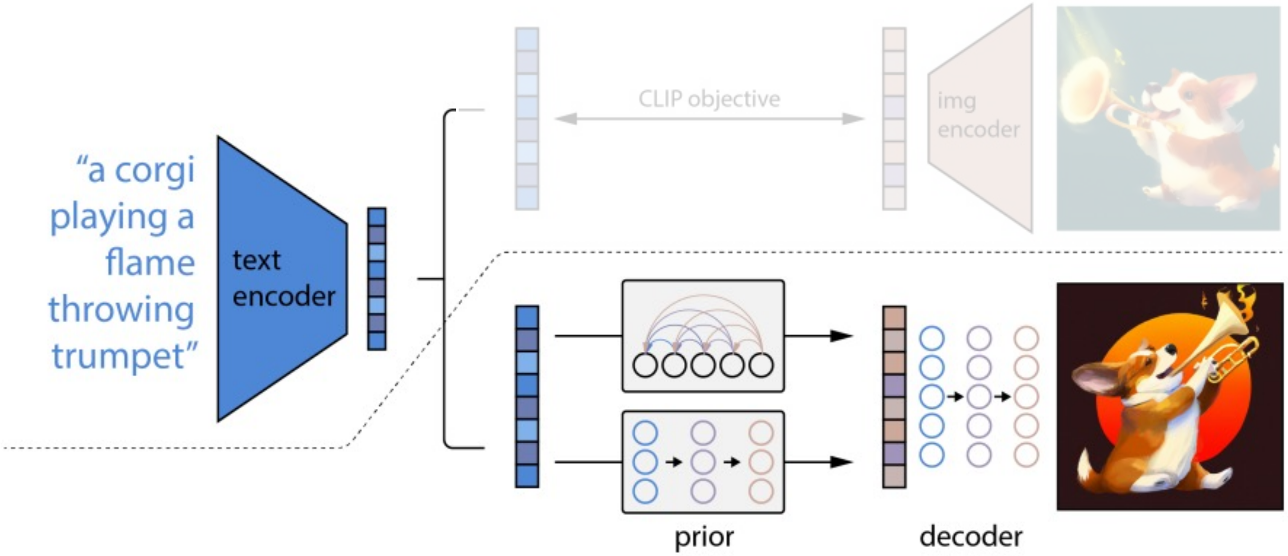
- Finally, an **image decoder** stochastically generates an image which is a visual manifestation of this semantic information.



Zero-Shot Text-to-Image Generation, 2021
Hierarchical Text-Conditional Image Generation with CLIP Latents. 2022

DALL-E

- [Ignoring some details] Both modules are implemented using "diffusion models".



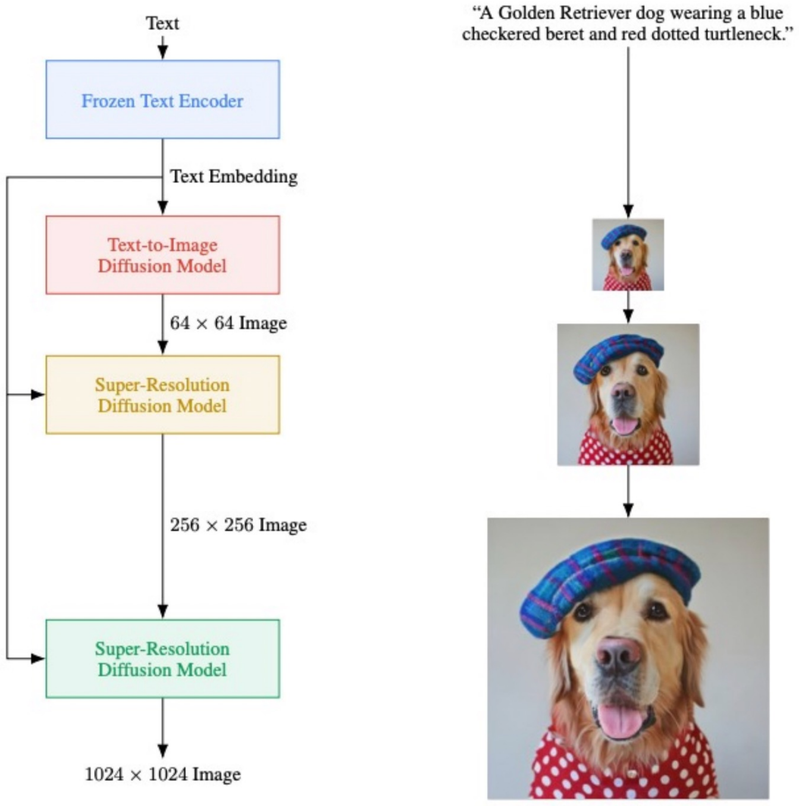
Zero-Shot Text-to-Image Generation, 2021
Hierarchical Text-Conditional Image Generation with CLIP Latents. 2022

Imagen

- Simpler than DALL-E

- Key components:
  - Frozen language model providing text embeddings to all diffusion models.
    - Human raters prefer T5-XXL as the text encoder over CLIP encoder.

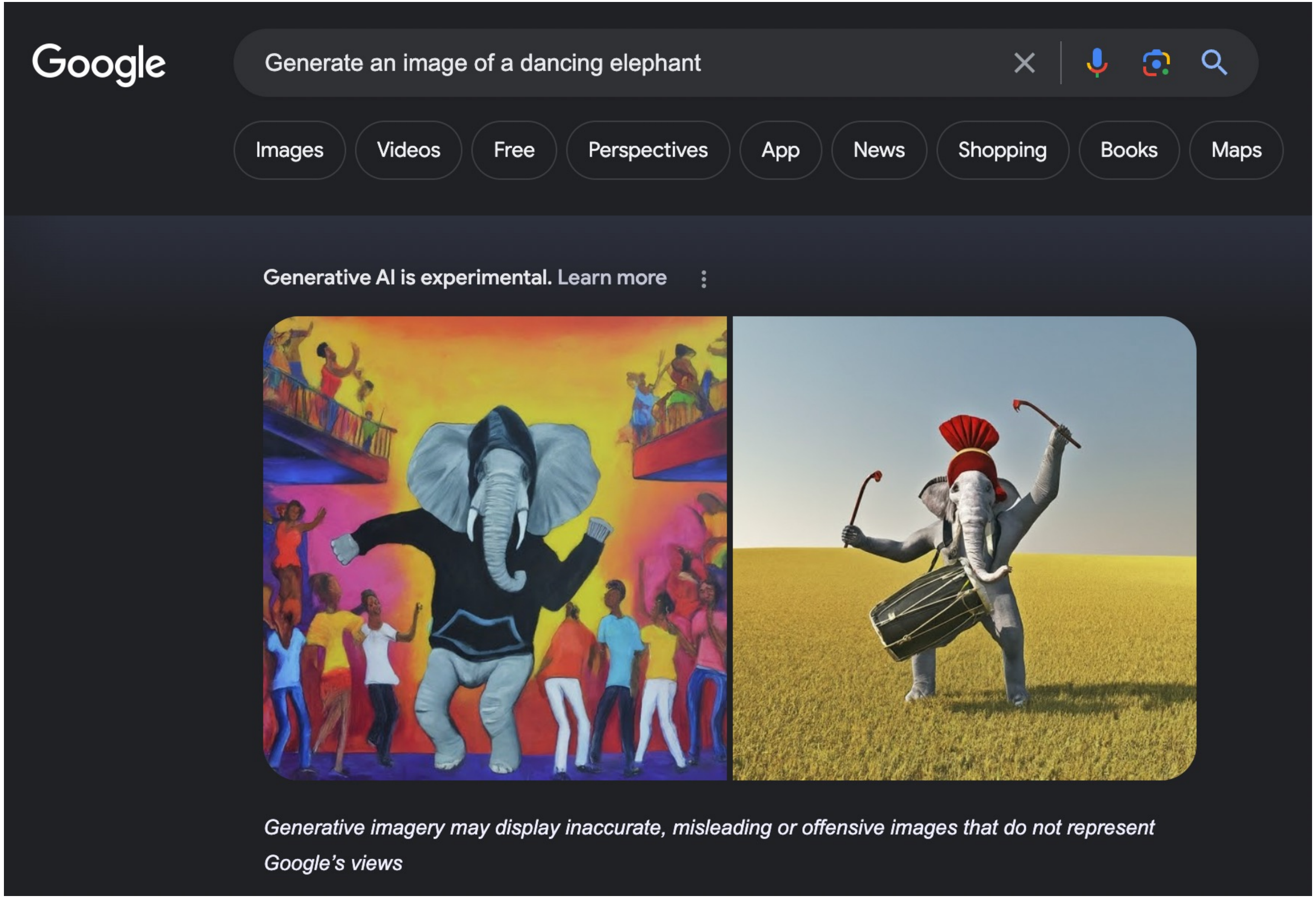
  - Cascaded diffusion models providing effective way to generate high-quality images.



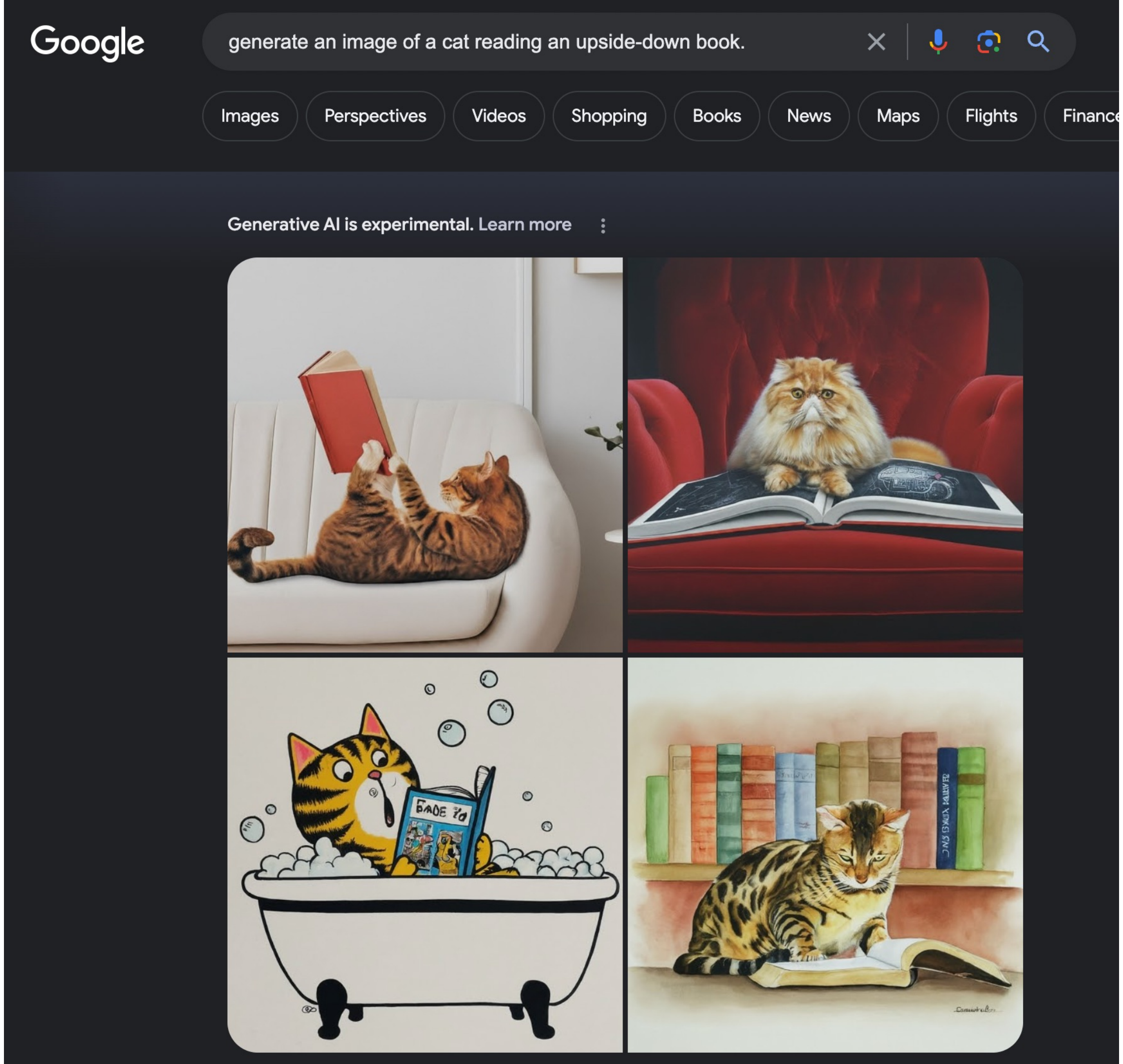
Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, 2022

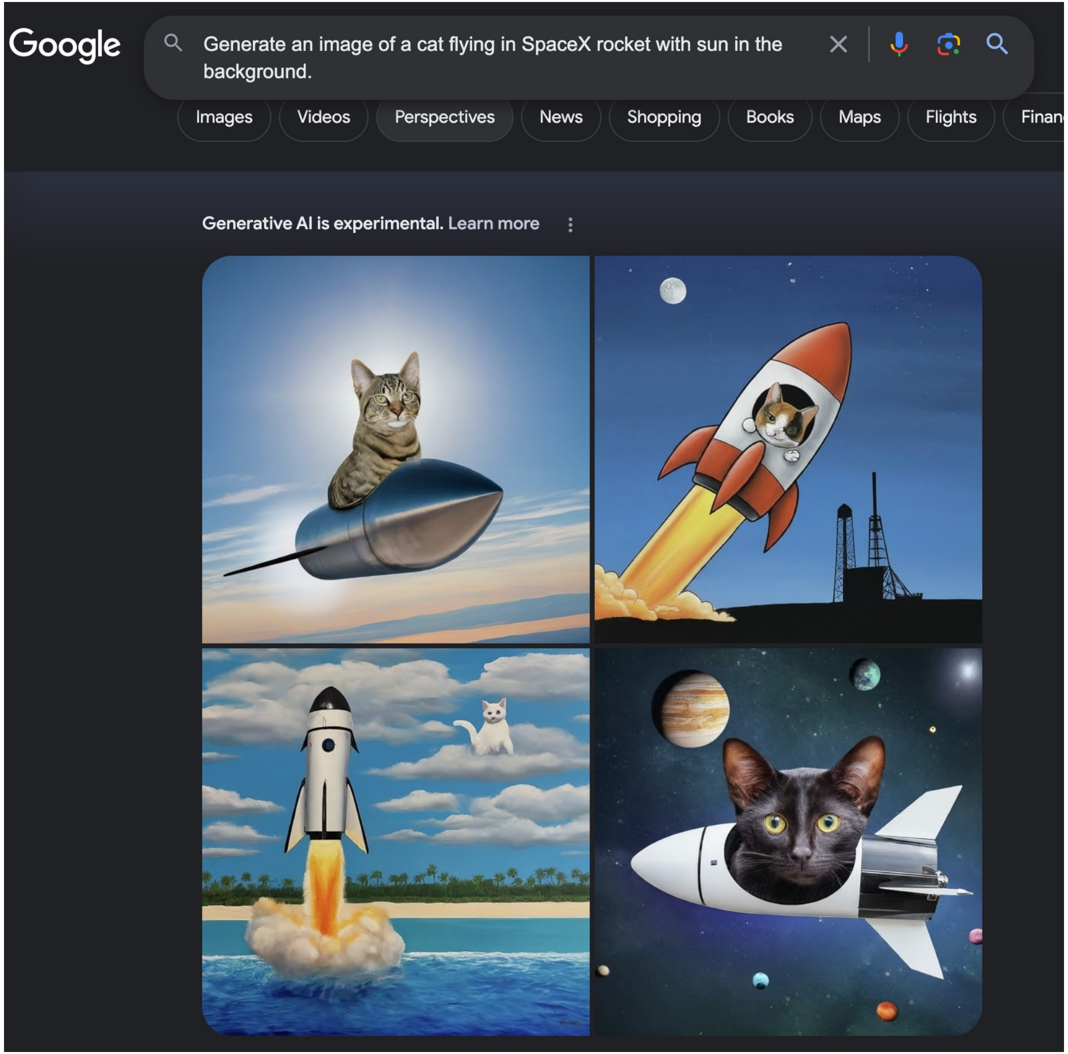
Imagen

Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, 2022



Google
Generate an image of a dancing elephant
Images
Videos
Free
Perspectives
App
News
Shopping
Books
Maps
Generative AI is experimental. Learn more
Generative imagery may display inaccurate, misleading or offensive images that do not represent Google's views


Google
generate an image of a cat reading an upside-down book.
Images
Perspectives
Videos
Shopping
Books
News
Maps
Flights
Finance
Generative AI is experimental. Learn more


Google
Generate an image of a cat flying in SpaceX rocket with sun in the background.
Images
Videos
Perspectives
News
Shopping
Books
Maps
Flights
Generative AI is experimental. Learn more

A ton of work on T → V

- Text to video

Text-to-Video generation: "a panda is playing guitar on times square"

- Text to 3D shapes

"A living room with a lit furnace, couch, and cozy curtains, bright lamps that make the room look well-lit."

- Text to motions/navigation

What about models that generate **text**?

Multi-modal GPT4

- Model: unknown
- Strong zero-shot visual understanding & reasoning capability
- How do we build opensource models that are as good as GPT4?

GPT-4 visual input example, Extreme Ironing:

User: What is unusual about this image?

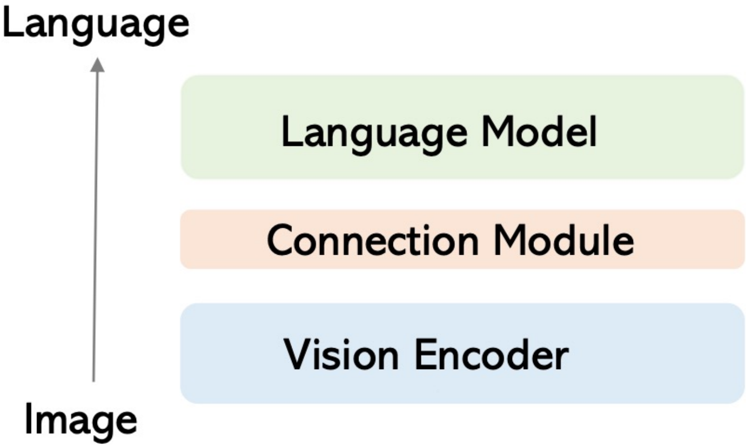
Source: https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg

GPT-4: The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

Image-to-Text Generative Models

- Model architectures:
  - Pre-trained image encoders
  - Pre-trained language models
  - Modules (to be trained) to connect the two modalities

A dog lying on the grass next to a frisbee



Give it a try! https://llava-vl.github.io/

OPEN SOURCE

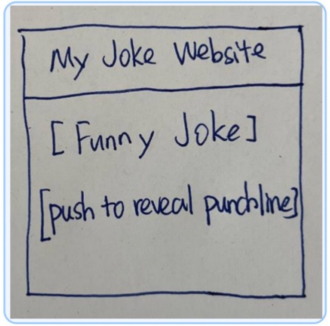
LLaVA Model

User	Can you explain this meme in detail?
LLaVA	The meme in the image is a creative and humorous take on food, with a focus on chicken nuggets, as the center of the universe. The meme begins with the phrase "Sometimes I just look at pictures of the Earth from space and I marvel at how beautiful it all is..." The main image features a tray of cooked chicken nuggets, with the chicken pieces positioned to resemble the continents and islands on a world map. The meme playfully suggests that the chicken nuggets represent the Earth, and the various locations depicted in the photo are actually chicken nugget versions of different places. The punchline of the meme appears in smaller text at the bottom, reading "I mean, it's not the real Earth, but how beautiful it is all is." This lighthearted and entertaining meme combines food and humor, using the image of chicken nuggets to create a funny and imaginative representation of the world.
User	Can you explain this meme?
GPT-4 [36]	This meme is a joke that combines two unrelated things: pictures of the earth from space and chicken nuggets. The text of the meme suggests that the image below is a beautiful picture of the earth from space. However, the image is actually of chicken nuggets arranged to vaguely resemble a map of the world. The humor in this meme comes from the unexpected juxtaposition of the text and the image. The text sets up an expectation of a majestic image of the earth, but the image is actually something mundane and silly.

https://arxiv.org/pdf/2304.08485.pdf

Visual Instruction Tuning, 2023

LLaVA Model



User

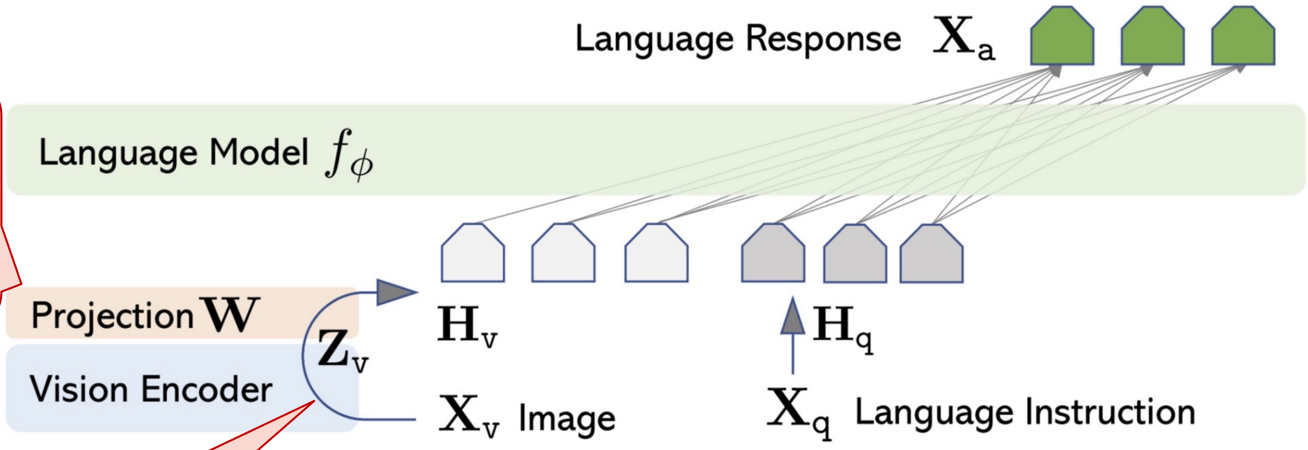
Turn this mock-up into an interactive website using html/js, replace the joke with a real joke.

LLaVA

```
<!DOCTYPE html>
<html>
 <head>
  <title>My Joke Website</title>
  <style>
   body { background-color: white; font-family: Arial, sans-serif; }
   .container { display: flex; flex-direction: column; align-items: center; justify-content: center; height: 100vh; }
   h1 { font-size: 36px; margin-bottom: 20px; }
   p { font-size: 20px; }
   button { background-color: blue; color: white; border: none; padding: 10px 20px; font-size: 18px; cursor: pointer; }
  </style>
 </head>
 <body>
  <div class="container">
   <h1>My Joke Website</h1>
   <p id="punchline">Welcome to my joke website. Push the button to reveal the punchline!</p>
   <button onclick="revealPunchline()">Push me!</button>
  </div>
  <script>
   function revealPunchline() {
    document.getElementById("punchline").innerHTML = "The punchline is: 'Why was the math book sad? Because it had too many 
problems.'";
   }
  </script>
 </body>
</html>
```

LLaVA Architecture

This projection matrix maps the embedded image to the same dimension as the LLM input (word embeddings).



CLIP image encoder

Language instructions

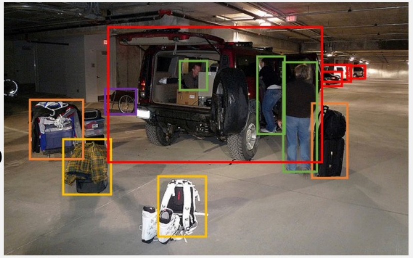
https://arxiv.org/pdf/2304.08485.pdf
Visual Instruction Tuning, 2023

LLaVA: GPT4-assisted data generation

- **The raw data:** image captioning datasets
  - Consists of images, captions and bounding boxes of objects

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.

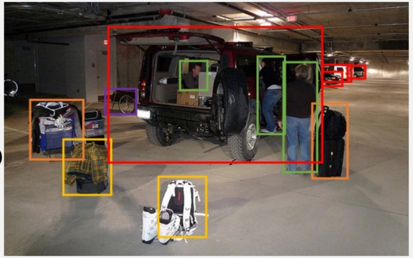


Context type 2: Boxes

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

- **Augmentation:** it uses GPT4 to augment these with detailed captions.

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>

GPT4

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

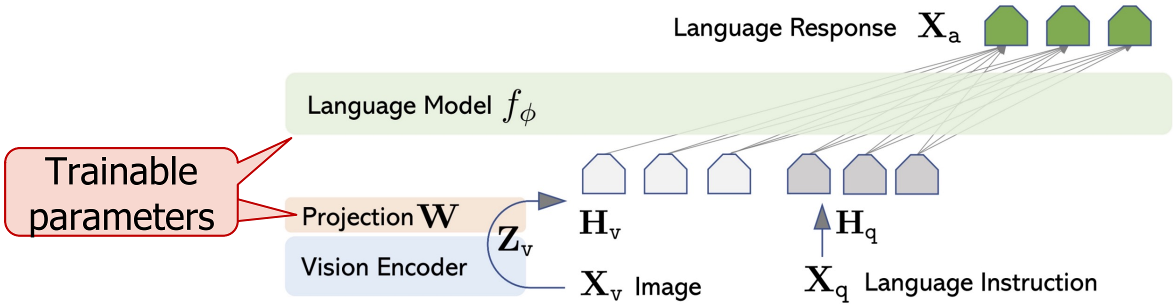
The image is an underground parking area with a black sport utility vehicle (SUV) parked. There are three people in the scene, with one person standing closer to the left side of the vehicle, another person in the middle, and the third person on the right side. They are all working together to pack their 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>

LLaVA Architecture: Training

- **Step 1:** Feature alignment — aligning the representation of Vision Encoder and LLM
  - Both Vision Encoder and LLM are kept frozen.
  - The only training parameter is W (the projection matrix)
- **Step 2:** End-to-end fine-tuning
  - Vision Encoder is kept frozen. The training params are W and LLM.

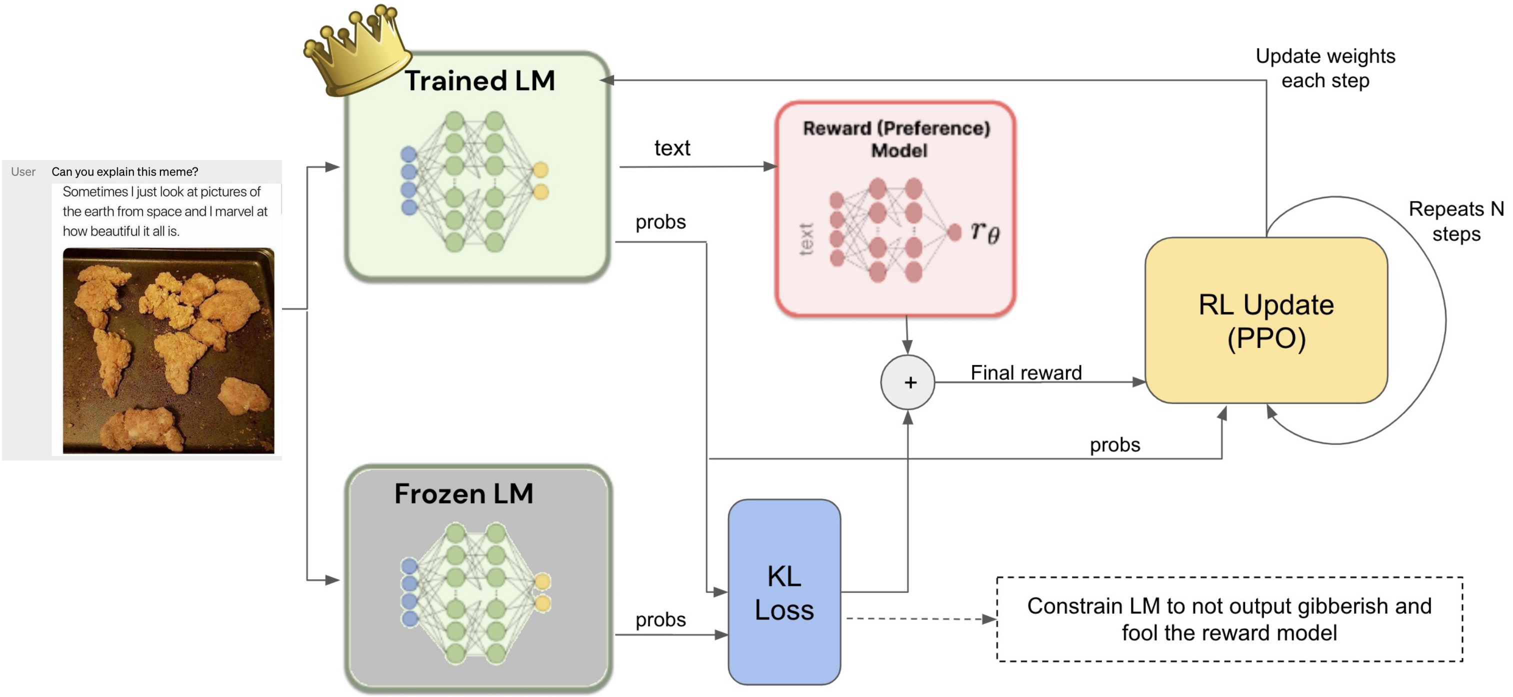


https://arxiv.org/pdf/2304.08485.pdf
Visual Instruction Tuning, 2023

Many open-source models …

- BLIP/BLIP2 [Li et al. 2023]
- Instruct-BLIP [Dai et al. 2023]
- MiniGPT4 [Zhu et al. 2023]
- CoCa [Yu et al. 2022]
- Shikra [Chen et al. 2023]
- LLaVA 1.5 [Liu et al. 2023]
- …

Note: You Can Also do Multimodal RLHF

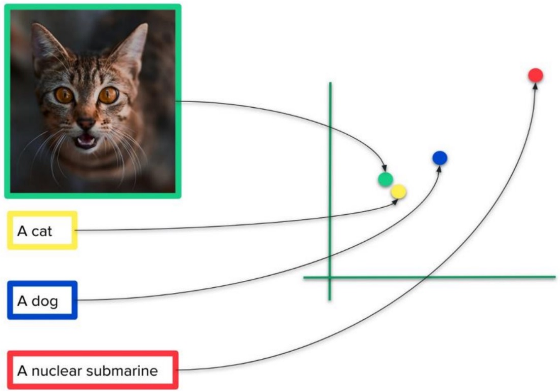


Summary

- There are massive amounts of unimodal data in the world.
  - We can build strong unimodal self-supervised models.

- Multi-modal models: the key challenge is aligning the representations.

- Many recent successes but a lot of challenges remain.
  - Making the models efficient.
  - Further scaling up — data scaling and model scaling
  - Better alignment of the representations — more natural/richer signals for alignment.



Challenges for Future Multi-Modal Models

Different modalities of information: Images, regions, pixels, ….

Different tasks: Classification, captioning, detection, grounding, segmentation, depth, …

Different types of inputs

Temporality: static images, short videos, very long videos, …

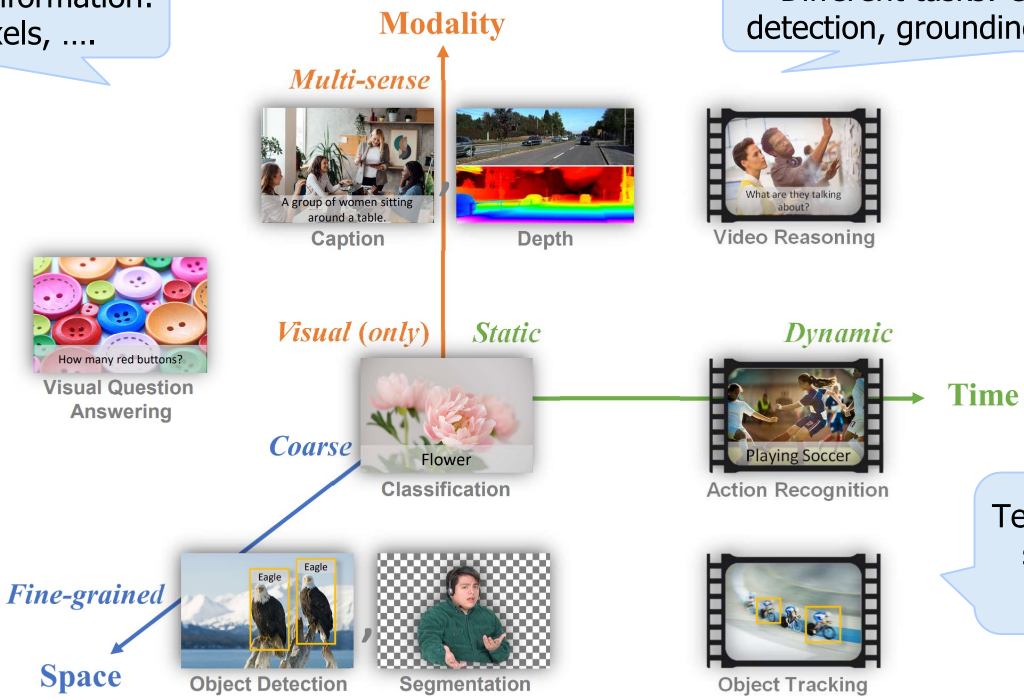
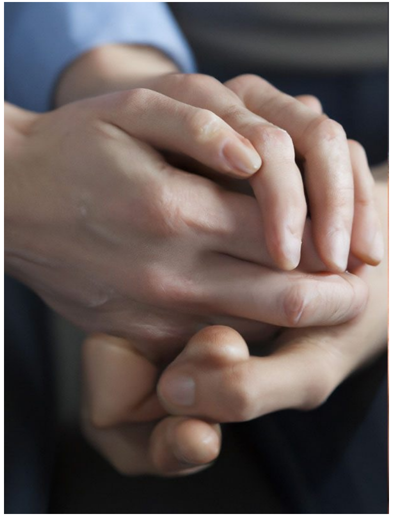
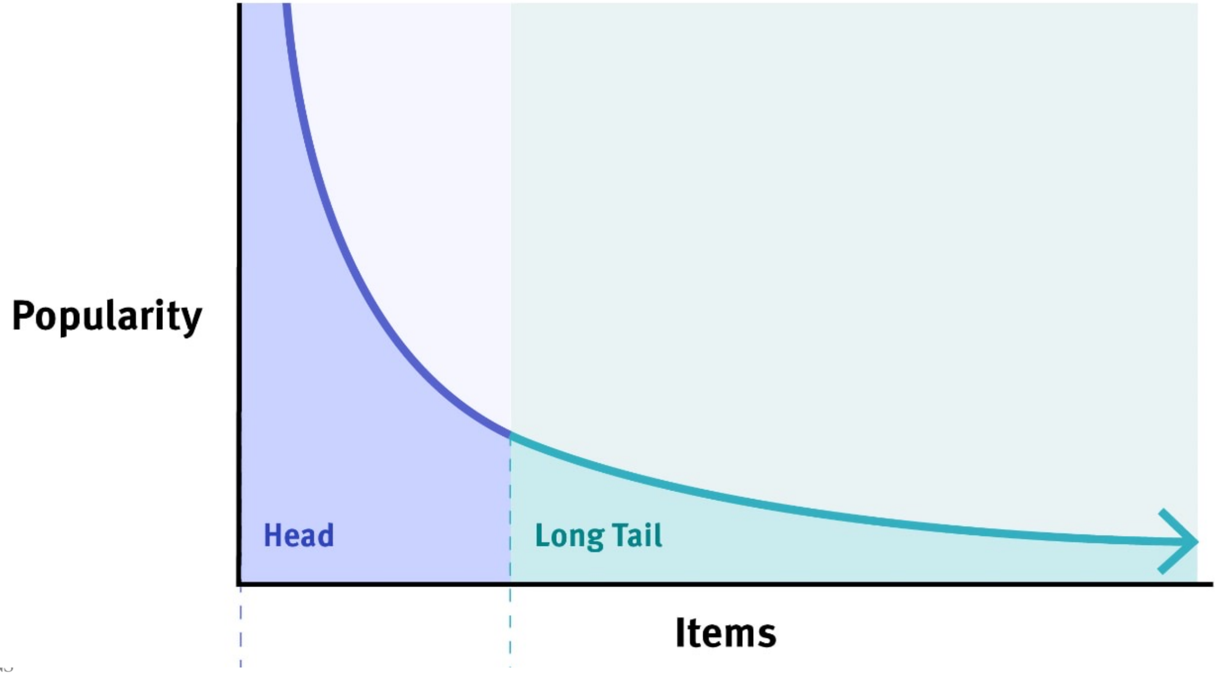


Figure: https://arxiv.org/pdf/2111.11432.pdf

Challenges for Future Multi-Modal Models



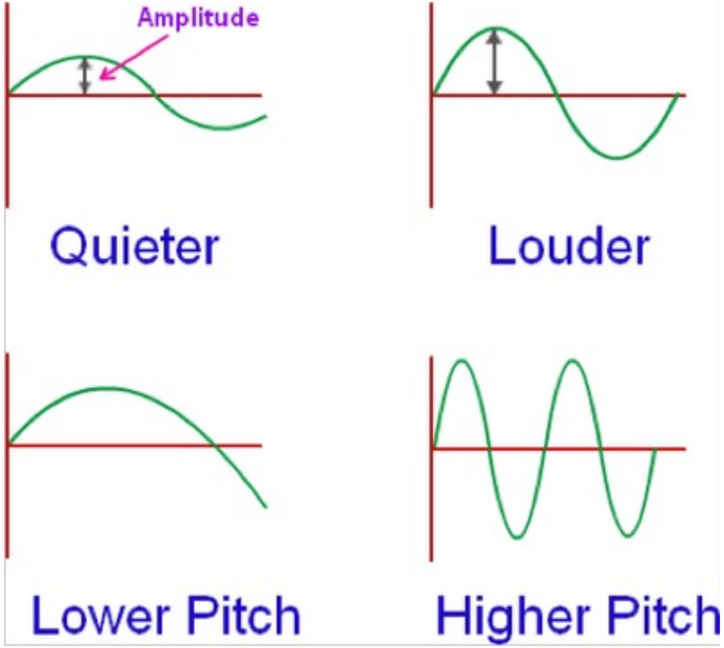
The long tail: *Most things are infrequent*



Transformers for Speech and Audio

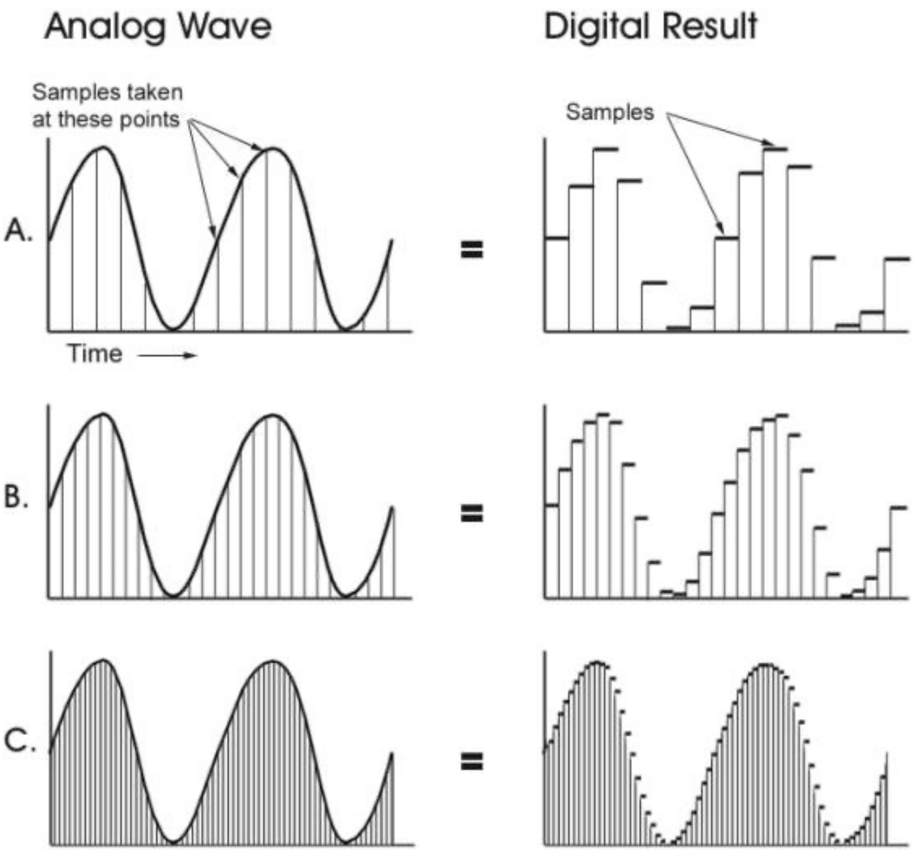
Dealing with Audio Data

- Audio is originally continuous wave.



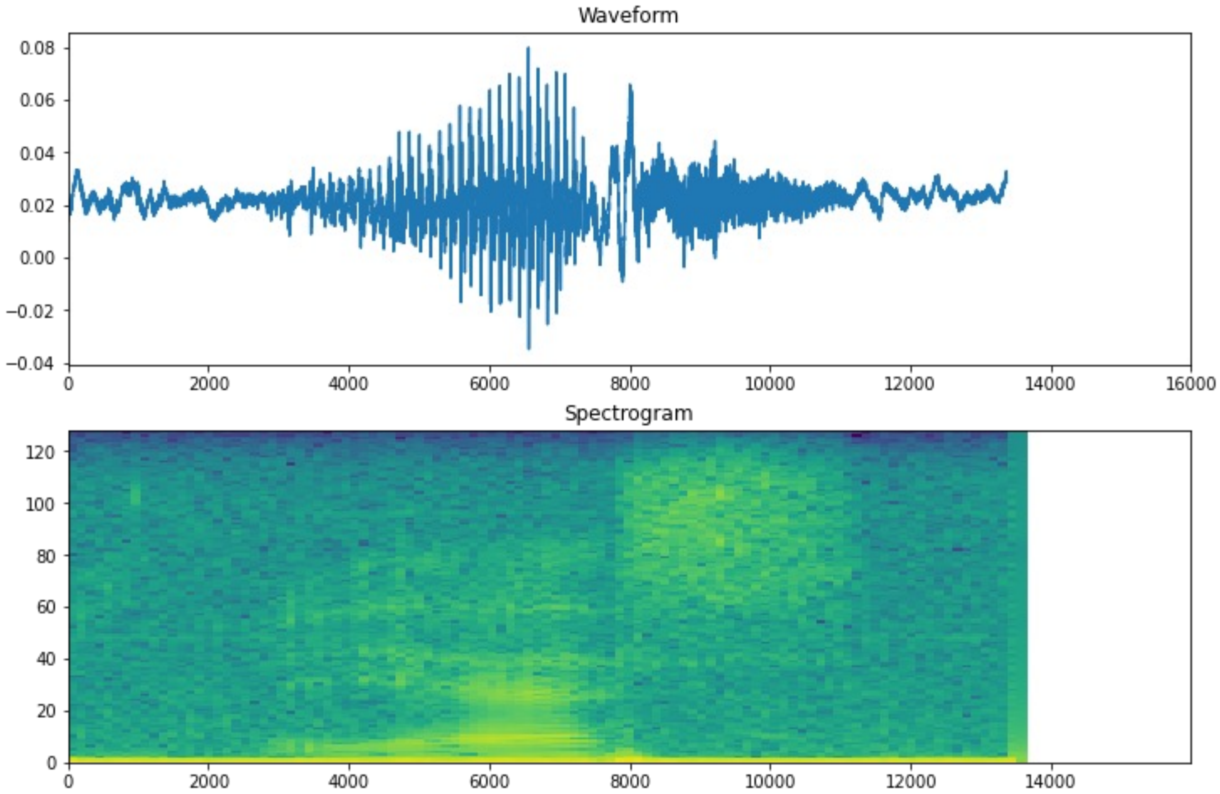
Dealing with Audio Data

- Audio is originally continuous wave.
- When recording it, we sample from it.
- The choice of sampling rate determines the fidelity of the recording.
- If the sampling rate is too low, the digital sound will be muffled.
- Nyquist limit: the minimum rate.



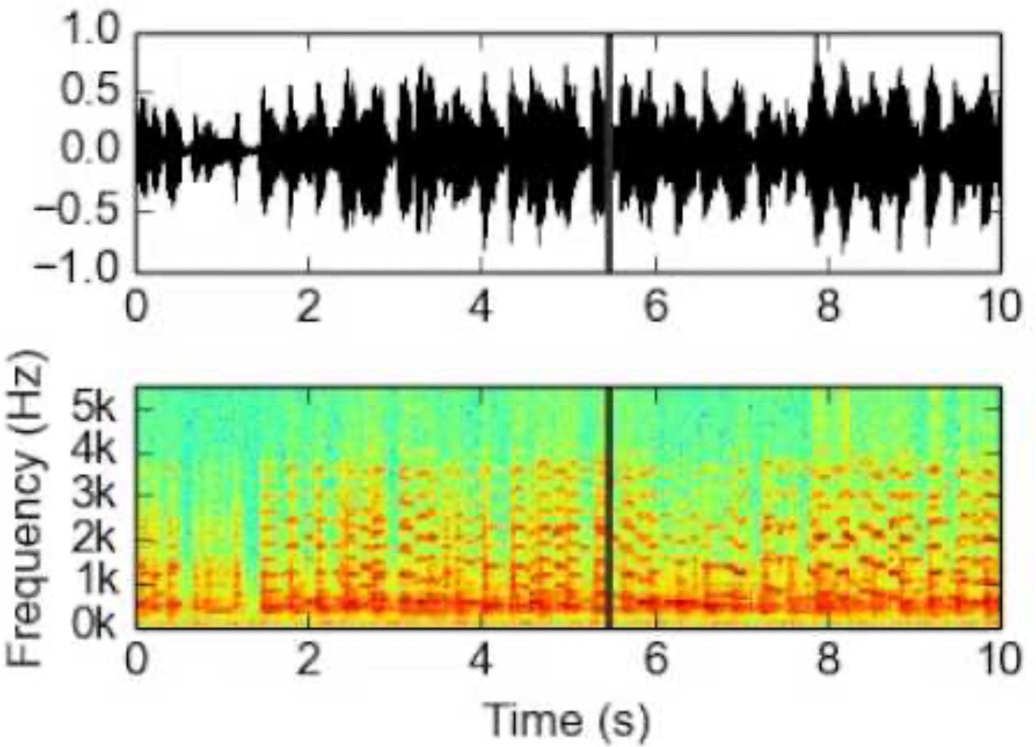
Spectrogram

- A spectrogram shows the **frequency content (y-axis)** of an audio signal as it changes over time.
- In spectrogram, **magnitude** of the sound is shown by **color-coding**.



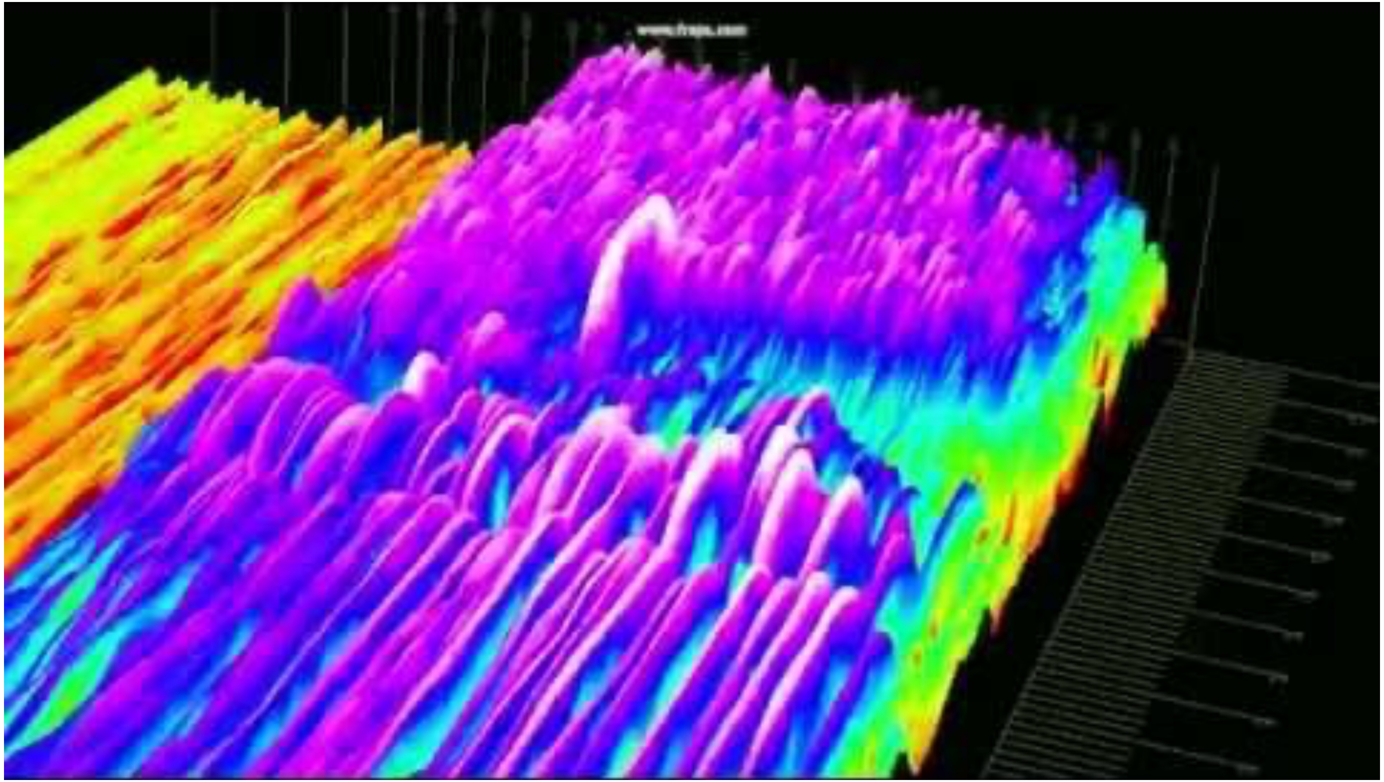
https://huggingface.co/learn/audio-course/en/chapter1/audio_data

Spectrogram: Example



https://huggingface.co/learn/audio-course/en/chapter1/audio_data

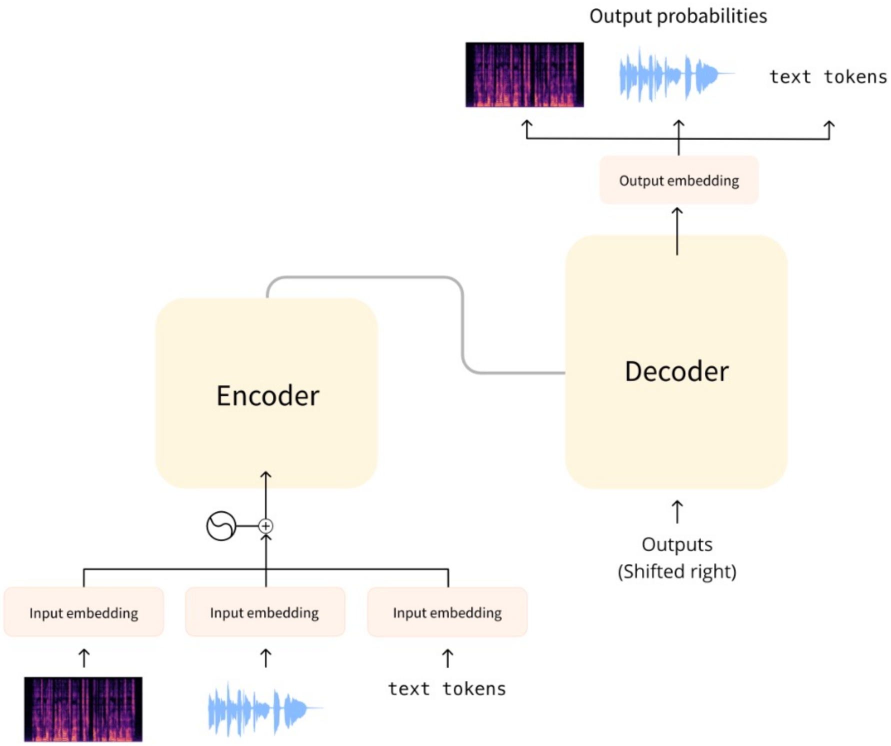
Spectrogram: Example



https://huggingface.co/learn/audio-course/en/chapter1/audio_data

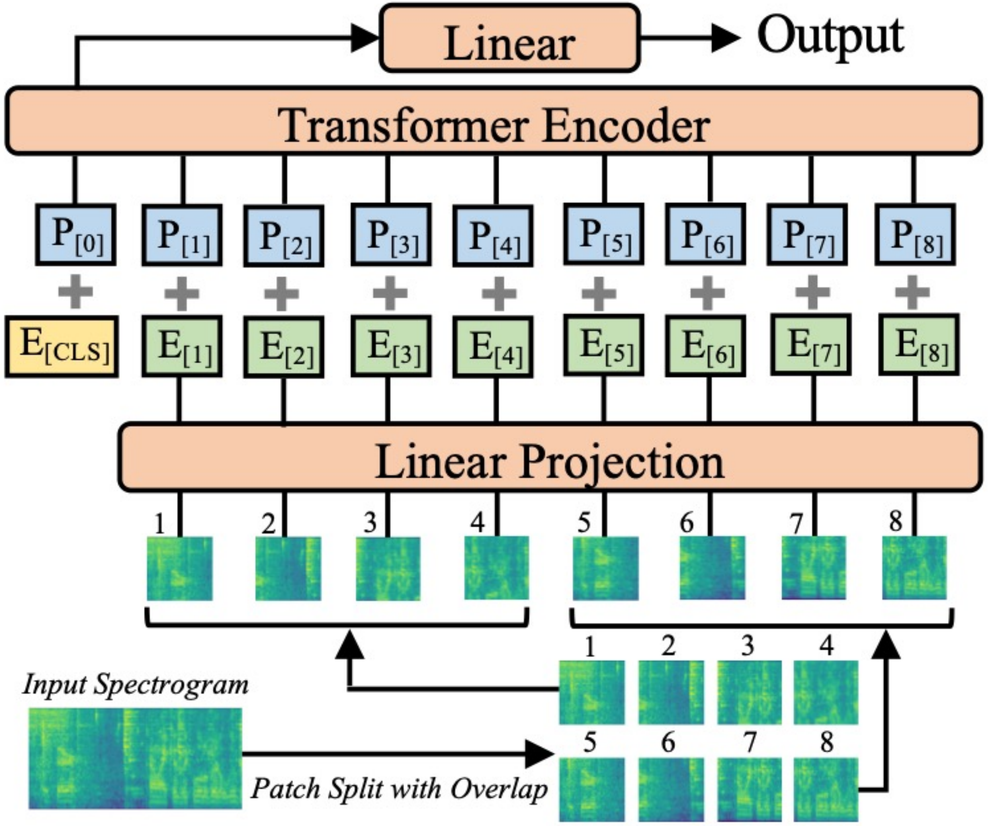
Feeding Speech to Transformer

- Turn your data into a format that can be processed by Transformer.



Audio Spectrogram Transformer (AST)

- Like ViT, AST splits the spectrogram into a sequence of partially overlapping 16×16 images.
- This sequence of patches is then projected into a sequence of embeddings, and these are given to the transformer encoder.
- On top of this is a simple classification layer with sigmoid activation to map the hidden-states to classification probabilities.



AST: Audio Spectrogram Transformer, 2021

Birds of View of Speech Tasks

- The progress in speech/audio is not as mature as areas.
- There are various modality connections:
  - Speech/audio to text, Speech/audio to image, Text to speech/audio, Image to speech/audio, …
- Recording audio is quite tricky:
  - Each microphone is different
  - Each room is different
  - Multiple audio sources (e.g., group of people)
  - …
- There are tons of sounds in the world that are not easily captured.

Summary

- Audio/speech via Transformers?
- Use the same old recipe: pre-process and feed it directly.
- How far are we from solving it? Quite far!!

LMs and Code Generation

How many people have used Github CoPilot?

GitHub
Copilot

https://github.com/features/copilot

Some examples

```python
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

```python
def encode_cyclic(s: str):
    """
    returns encoded string by cycling groups of three characters.
    """
    # split string to groups. Each of length 3.
    groups = [s[(3 * i):min((3 * i + 3), len(s))] for i in range((len(s) + 2) // 3)]
    # cycle elements in each group. Unless group has fewer elements than 3.
    groups = [(group[1:] + group[0]) if len(group) == 3 else group for group in groups]
    return "".join(groups)

def decode_cyclic(s: str):
    """
    takes as input string encoded with encode_cyclic function. Returns decoded string.
    """
    # split string to groups. Each of length 3.
    groups = [s[(3 * i):min((3 * i + 3), len(s))] for i in range((len(s) + 2) // 3)]
    # cycle elements in each group.
    groups = [(group[-1] + group[:-1]) if len(group) == 3 else group for group in groups]
    return "".join(groups)
```

General Idea

The story is the same as what we have seen before! 😮

- 1. Pre-train on web-scale text/code data (what we have seen before)
- 2. More pre-training on a lot of [potentially, noisy] code data (new)
- 3. Fine-tune on smaller but cleaner data (new)

Setup: Pre-train on Code Data

- Start with GPT-3 that is already pre-trained to on text and code.

- **Codex:** Starting from GPT-3, do continued training on code data.
  - Training data: 160GB of Python code (unlabeled!!)

- The goal is to evaluate model for:
  - Generating code from doc-string (implementing an idea)
  - Generating doc-string from code (explaining code)

- Why should this work?
  - Human written code often comes with comments (i.e., free supervision)!! 😍

Fine-tuning on Clean[er] Data

- Competitive Programming (10,000 problems)
  - Problem descriptions as docstrings
  - These have unit test coverage
- Continuous Integration (40,000 functions)
  - "Developers regularly merge code changes into a central repository, after which automated builds and tests are run."
  - These come with free test functions.

- They also do additional filtering.
- Filter out:
  - Low-quality docstring
  - Stateful functions that may be missing variables outside them.
- Approach:
  - Use Codex-12B to generate 100 samples per problem, discard the problem if no generation passes.

Evaluation

- 164 hand-written problems
  - Hand-written to avoid overlap with the training data ("training data leakage")
    - Although in practice it's hard to control for this.
  - Evaluates language comprehension, reasoning, algorithms and simple math

```
• "Check if two words have the same characters."
• "Return median of elements in the list l."
• "sum_to_n is a function that sums numbers from 1 to n."
• "Given a non-empty list of integers lst. add the even elements that are at odd indices."
• "Return true if a given number is prime, and false otherwise."
• "Return n-th Fibonacci number."
```

Evaluation

- Format:
  - function signature
  - docstring with examples
  - unit-tests

```
def vowels_count(s):
    """Write a function vowels_count which takes a
        string representing
    a word as input and returns the number of vowels in
         the string.
    Vowels in this case are 'a', 'e', 'i', 'o', 'u'.
        Here, 'y' is also a
    vowel, but only when it is at the end of the given
        word.

    Example:
    >>> vowels_count("abcde")
    2
    >>> vowels_count("ACEDY")
    3
    """
```

```
def check(candidate):
     # Check some simple cases assert candidate("abcde")
     == 2, "Test 1" assert candidate("Alone") == 3, "Test
     2" assert candidate("key") == 2, "Test 3" assert
     candidate("bye") == 1, "Test 4" assert
     candidate("keY") == 2, "Test 5" assert
     candidate("bYe") == 1, "Test 6" assert
     candidate("ACEDY") == 3, "Test 7"
     # Check some edge cases that are easy to work out by
     hand. assert True, "This prints if this assert fails
     2 (also good for debugging!)"
```

Evaluation

- The traditional NLP metrics are not necessarily good for code generation.

Reference code

```python
def f(a, b):
    c = a - b
    return c
```

Equivalent code

```python
def f(a, b):
    summ = 0
    summ += a
    summ -= b
    return summ
```

BLEU = 66

Non-equivalent code

```python
def f(a, b):
    c = a + b
    return c
```

BLEU = 81

- Instead, they measure functional correctness.
  - Whether the generated code (ignoring its details) passes all unit tests
  - This is the way humans evaluate correctness of the code rather than its content.

Evaluation

- Given a prompt, generate k samples
  - For each, sample tokens until a stop sequence is encountered

- **pass@k**: the probability that **at least one** of the top k-generated code samples for a problem passes the unit tests.
  - How do you compute this?
    - Think about it! ;-)

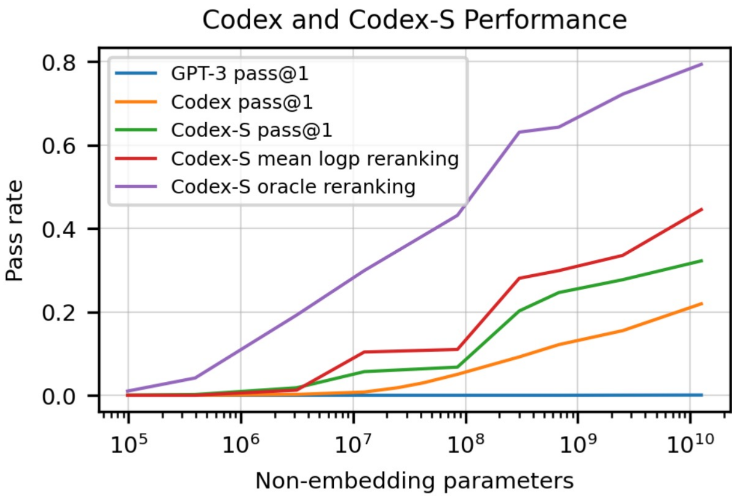
Results

- Scaling improves the results

- Just pre-training code (Codex) gives a major boost.

- Reranking heuristics:

  Generate 100 samples per problem and selecting the sample with the highest mean log-probability or by selecting the sample that passes the unit tests (oracle).



Evaluating Large Language Models Trained on Code, 2021

Results: w/ Open-source Models

- Existing open-source models (GPT-J and GPT-Neo) know a surprising amount about code generation!!

- Reason: The Pile (used for GPT-J) contains a lot of code: 8% GitHub code, along with natural language data

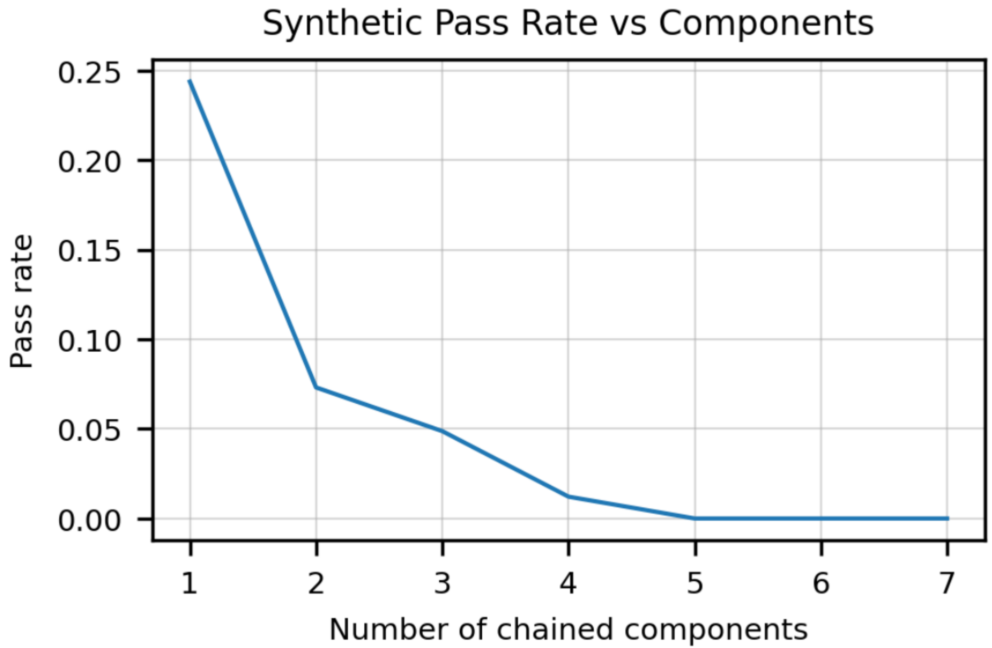
Table 1. Codex, GPT-Neo, & TabNine evaluations for HumanEval. We find that GPT-J pass@1 is between Codex-85M and Codex-300M performance.

	PASS@k		
	$k=1$	$k=10$	$k=100$
GPT-NEO 125M	0.75%	1.88%	2.97%
GPT-NEO 1.3B	4.79%	7.47%	16.30%
GPT-NEO 2.7B	6.41%	11.27%	21.37%
GPT-J 6B	11.62%	15.74%	27.74%
TABNINE	2.58%	4.35%	7.59%
CODEX-12M	2.00%	3.62%	8.58%
CODEX-25M	3.21%	7.1%	12.89%
CODEX-42M	5.06%	8.8%	15.55%
CODEX-85M	8.22%	12.81%	22.4%
CODEX-300M	13.17%	20.37%	36.27%
CODEX-679M	16.22%	25.7%	40.95%
CODEX-2.5B	21.36%	35.42%	59.5%
CODEX-12B	28.81%	46.81%	72.31%

Evaluating Large Language Models Trained on Code, 2021

Result: Degradation with Length

- The longer the inputs instructions are, the lower the model performance is.



Summary

- Our focus: building a bridge between natural language and code.
- Turns out the ideas in prior chapters go a long way!
- The importance of this is more than just increasing the productivity of programmers!
- In the next part, we will use this bridge to enable LMs speak to do various things!

Connecting Language to Actions/Tools

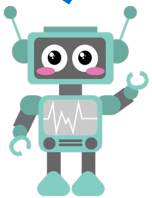
Leverage LLMs for Real-World Tasks

Put this book on the table.

LLM

```
def grab(obj):
    # grab the target object
    ...
def move_ahead(dist):
    # move ahead by dist
    ...
def put(obj, loc):
    # put the object at loc
    ...
```

Available Actions/APIS



```
def put_book(book, shelf):
    # grab the book
    grab(book)
    # move ahead by 1 meter
    move_ahead(1)
    # put the book at the shelf
    put(book, shelf)
```

Policies/Programs

Leverage LLMs for Real-World Tasks

User Interaction:

- Make me a cup of coffee
- Plan a 2-day trip to DC
- Buy my favorite snacks

Planning:

LLM

```
def make_coffe()
...
def plan_route():
...
def buy_snacks():
...
```

Execution:

Tool/Action 1 — Robot Actions

Tool/Action 2 — App Tools (e.g., map, weather)

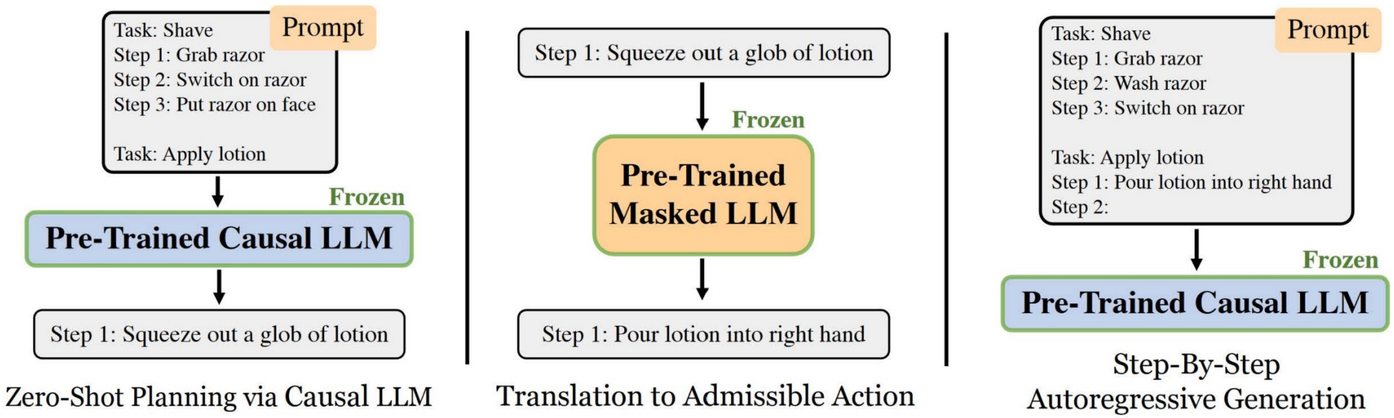
Tool/Action 3 — Web Navigation APIs (e.g., click, type, select)

LLMs as Backbones for Planners

- Rich parametric knowledge about commonsense and procedures
- Impressive capabilities of zero-shot/few-shot code generation

LMs as Planners: Procedural and Commonsense Knowledge

- Key Idea: Large language models learn rich procedural knowledge and such knowledge could be extracted.

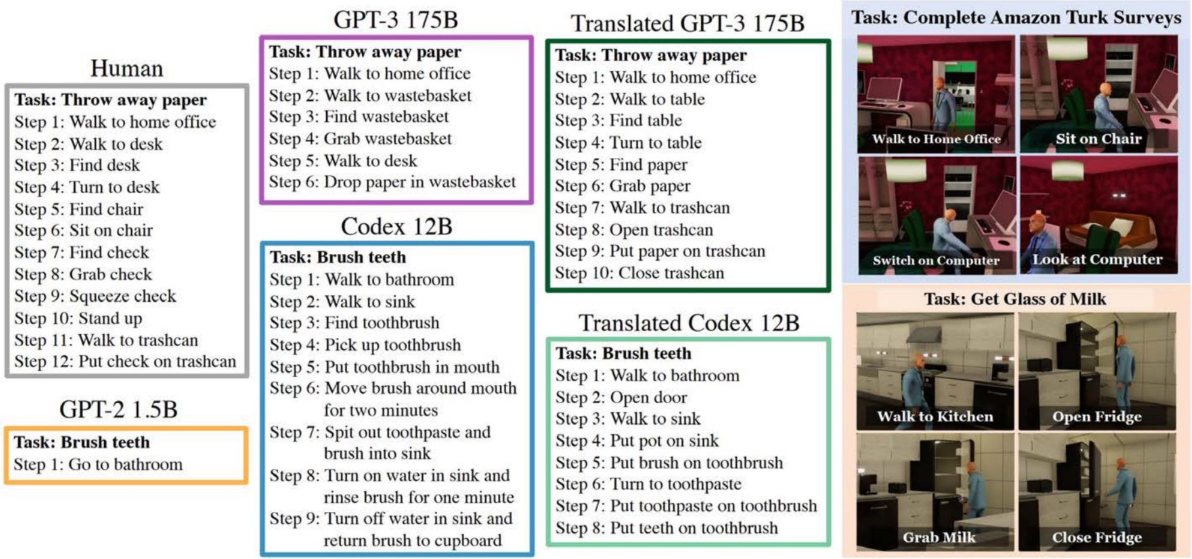


Huang, Wenlong, et al. "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents." International Conference on Machine Learning. PMLR, 2022.

319

LMs as Planners: Procedural and Commonsense Knowledge

- Impressive results.
- Challenge: verification; groundability to real-world videos.

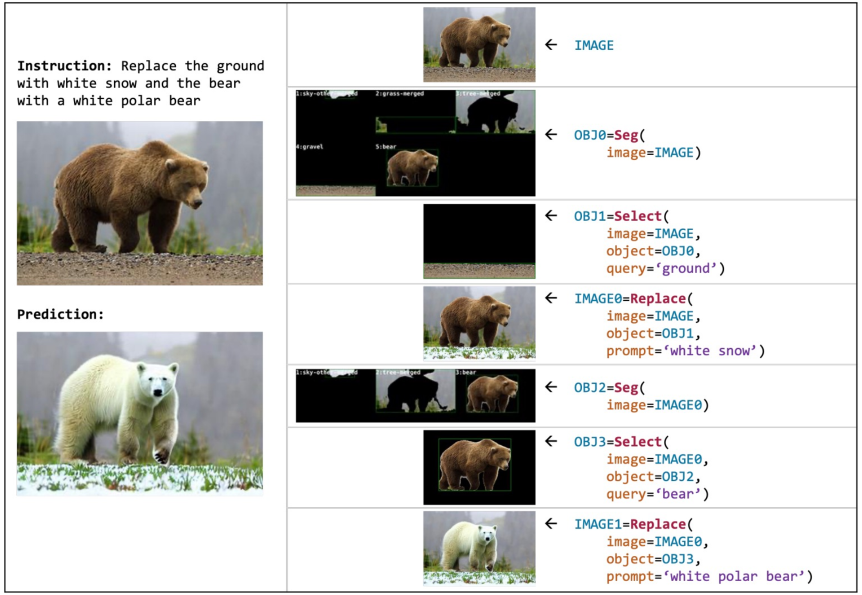


Huang, Wenlong, et al. "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents." International Conference on Machine Learning. PMLR, 2022.

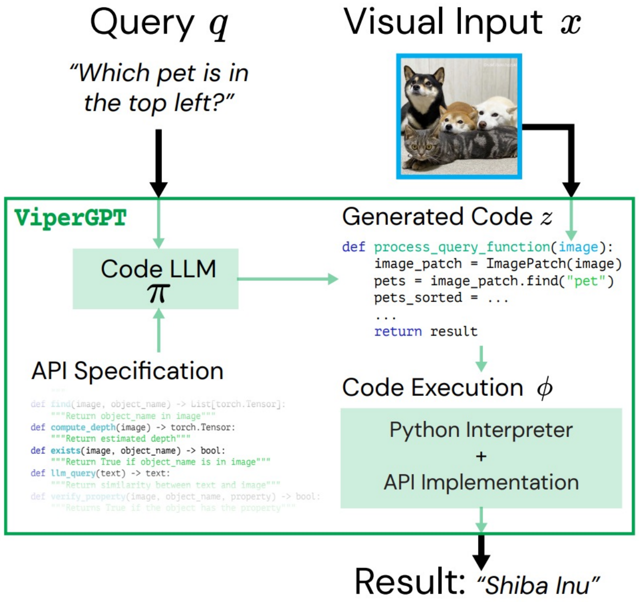
320

LLM as Planners: Composing Tools via Code Generation

Visual Programming: Compositional visual reasoning without training

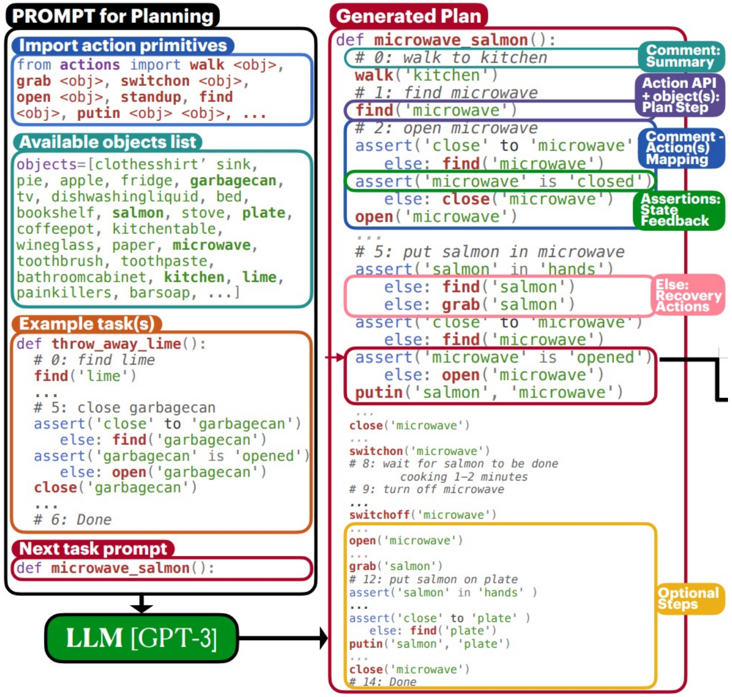


ViperGPT: Visual Inference via Python Execution for Reasoning

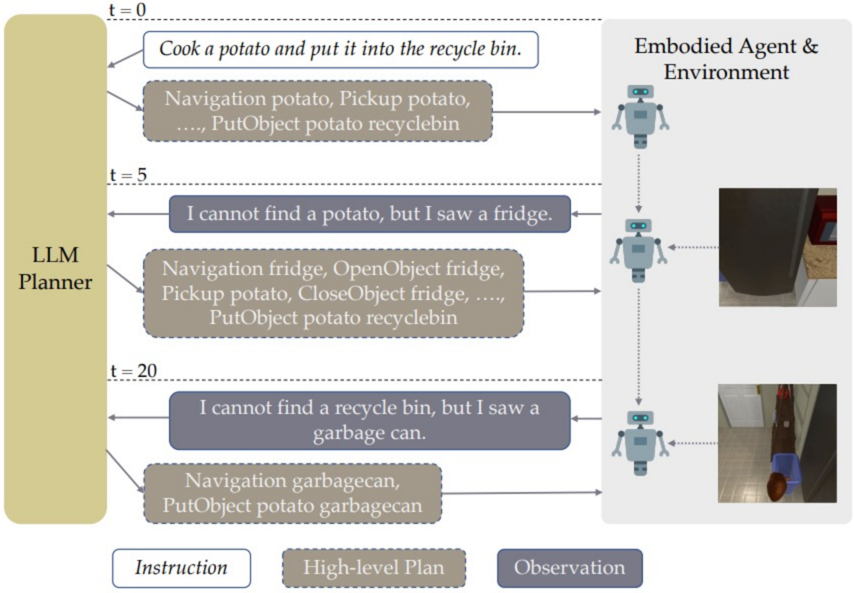


Application: Embodied Planning

Generating Situated Robot Task Plans using LLMs



Few-Shot Grounded Planning for Embodied Agents with LLMs



Application: Vision-Language Agents

MM-ReAct: Connecting LLM to Vision Experts

Visual Math and Text Reasoning

What is “a” in the first equation?
What is y in the last equation?

5 + a = 7	40 + y = 45
10 + y = 15	39 + a = 41
20 + a = 22	24 + y = 29
48 + a = 50	15 + a =17
10 + a = 12	34 + y = 39

The value of “a” in the first equation “5 + a = 7” is 2.

The value of “y” in the last equation “34 + y = 39” is 5.

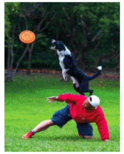
Visual-Conditioned Joke/Meme

What is funny about it?

The humor in the comic may come from the situation, the characters, or the dialogue. The person is feeling unwell and is asked if they are pregnant or have the coronavirus, which is a current and relevant topic. The humor may come from the absurdity of the situation or the unexpectedness of the question.

Spatial/Coordinate Understanding

Where is the frisbee?
What is the relative position of the frisbee and the man?



The frisbee is located at coordinates (63,119,133,177).

The frisbee is located above the man in the image. The man is falling on the ground while the frisbee is in the air. The frisbee is also closer to the top of the image than the man..

Visual Planning and Prediction

What should I do after kneading the dough, before dividing it into pieces? For how long?

After kneading the dough, you should put it in a warm place before dividing it into pieces, according to the recipe in the image.

You should put the dough in a warm place for 1 hour.

Multi-Image Reasoning

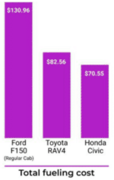
How much in total did I spend on travel?
How much in total did I pay for taxes?

You spent $1299.90 on airfare and $43.83 on an Uber ride for a total of $1343.73 on travel.

You paid $323.23 in total for taxes. This includes $321.90 for taxes and fees for the flight purchase and $1.33 for sales tax for the Windmill Family Restaurant.

Multi-Hop Document Understanding

Excluding the regular cab, what is the average total fueling cost?



The average total fueling cost excluding the Ford F150 (Regular Cab) is $76.55. This is calculated by taking the sum of the total fueling cost of Toyota RAV4 and Honda Civic and dividing it by 2. ($82.56 + $70.55) / 2 = $76.55.

Open-World Concept Understanding

What is the brand of these drink? Which one is unsweetened?

The brand of these drinks is “BETTER THAN MILK”. The unsweetened one is “BETTER THAN MILK organic OAT DRINK”.

Who are the celebrities in this image?

Keira Knightley and Natalie Portman are the celebrities in this image.

Is it safe to eat this mushroom?

Morel mushrooms are generally considered safe to eat when properly cooked.

Video Summarization/Event Localization

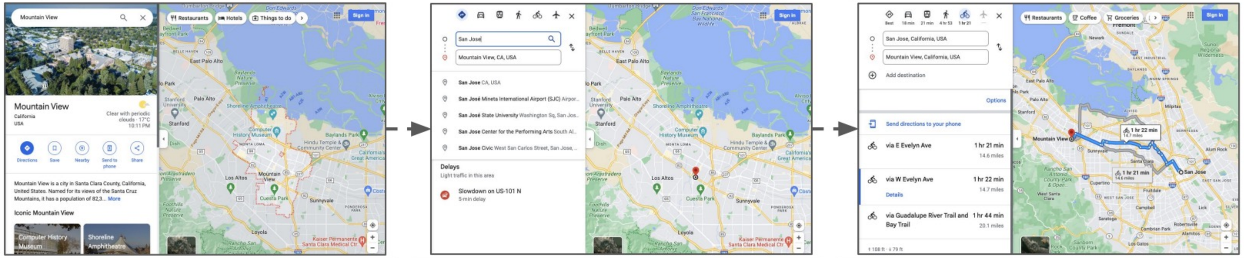
Can you breakdown this video tutorial to step-by-step instructions, with start and end time?

Here's a step-by-step breakdown of the tutorial with start and end time.

··· Add Lettuce Add Tomato Add Bacon ···

Application: Web Agents

map: ***Show me the way from San Jose to Mountain View by 2nd Cycling at map website?***



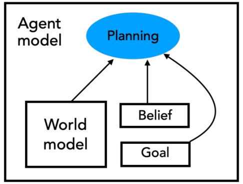
```
# Type Mountain View into search
driver.find_element(By.CSS_SELECTOR,"...").clear()
driver.find_element(
   By.CSS_SELECTOR,"..."
).send_keys("Mountain View")
```

```
# Type San Jose into starting point
driver.find_element(By.CSS_SELECTOR,"...").clear()
driver.find_element(
   By.CSS_SELECTOR,"...").send_keys("San Jose")
```

```
# Click Cycling radio button
driver.find_element(
   By.CSS_SELECTOR,"#Cycling").click()
# Click 2nd trip
driver.find_element(By.CSS_SELECTOR,"#trip1").click()
```

Planning with Environment Feedback

Language models as (autoregressive) planners



Task: Shave
Step 1: Grab razor
Step 2: Wash razor
Step 3: Switch on razor

Task: Apply lotion
Step 1: Pour lotion into right hand
Step 2:

Prompt

LLM

Step 1: Squeeze out a glob of lotion

Step 1: Squeeze out a glob of lotion

All possible actions
in the environment

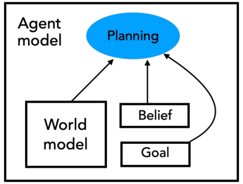
Step 1: Pour lotion into right hand

LLM

...

Huang et al. (2022)

Language models as (autoregressive) planners

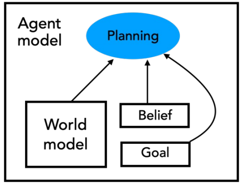


Empty Dishwasher

Organize Closet

Huang et al. (2022)

Language models as (autoregressive) planners



Step 1: Squeeze out a glob of lotion

All possible actions in the environment ?

Step 1: Pour lotion into right hand

LLM

Step 1: Squeeze out a glob of lotion

Task: Shave
Step 1: Grab razor
Step 2: Wash razor
Step 3: Switch on razor

Task: Apply lotion
Step 1: Pour lotion into right hand
Step 2:

Prompt

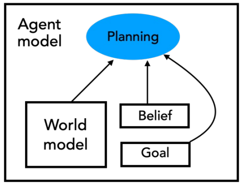
LLM

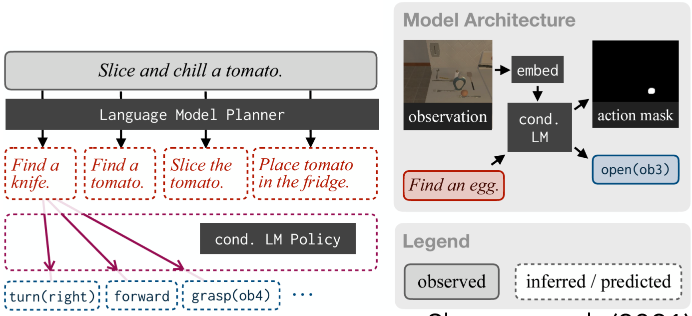
...

? No environment feedback

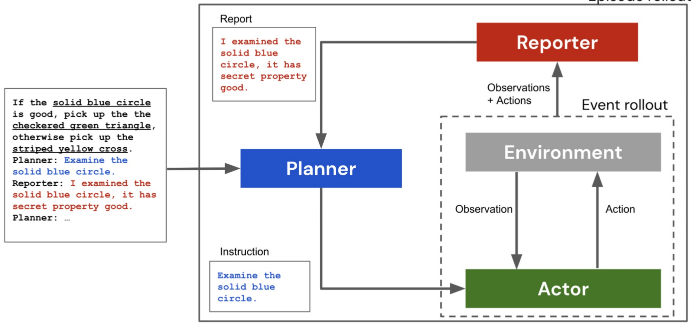
Huang et al. (2022)

Interactive planner

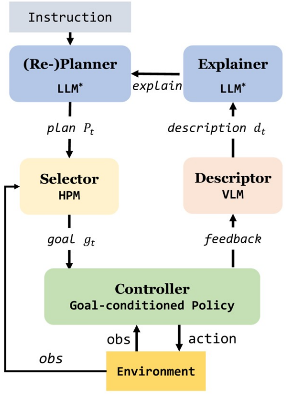




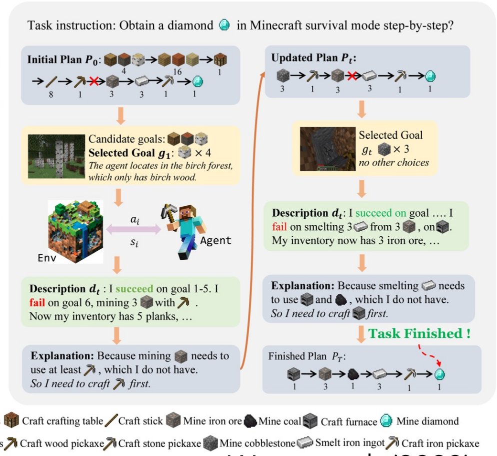
Sharma et al. (2021)



Dasgupta et al. (2022)

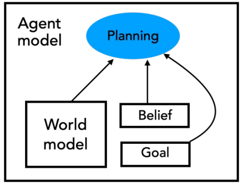


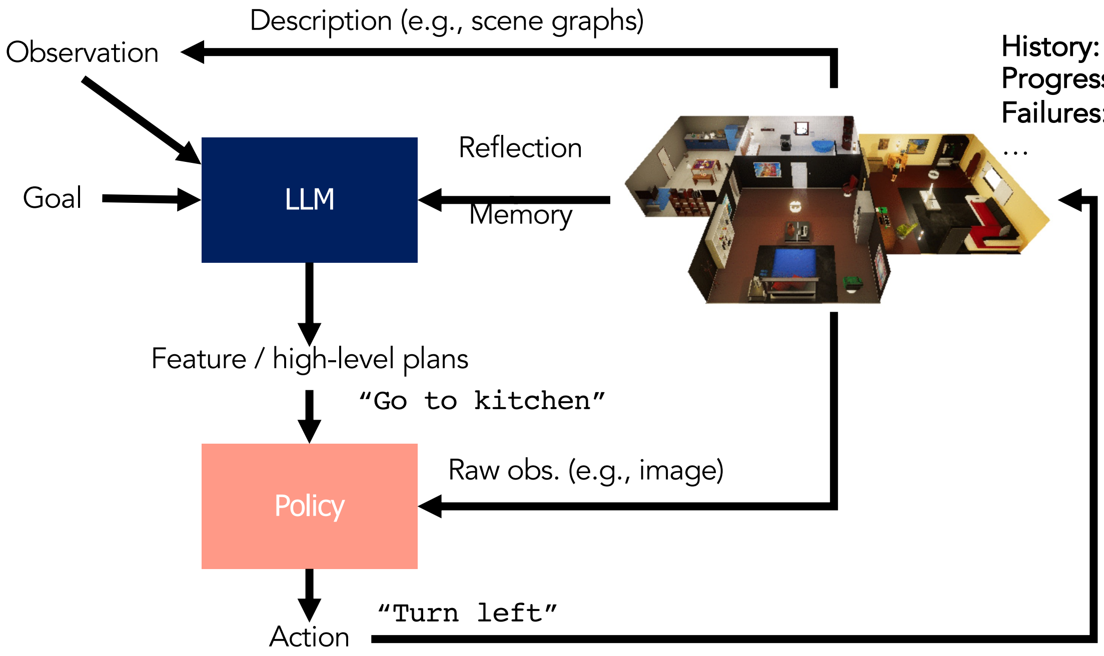
Li et al. (2022)



Wang et al. (2023)

Interactive planner





History: actions taken so far
Progress: I have done xxx
Failures: I couldn’t find xxx
…

"Language Models meet World Models" Zhiting Hu and Tianmin Shu, 2023

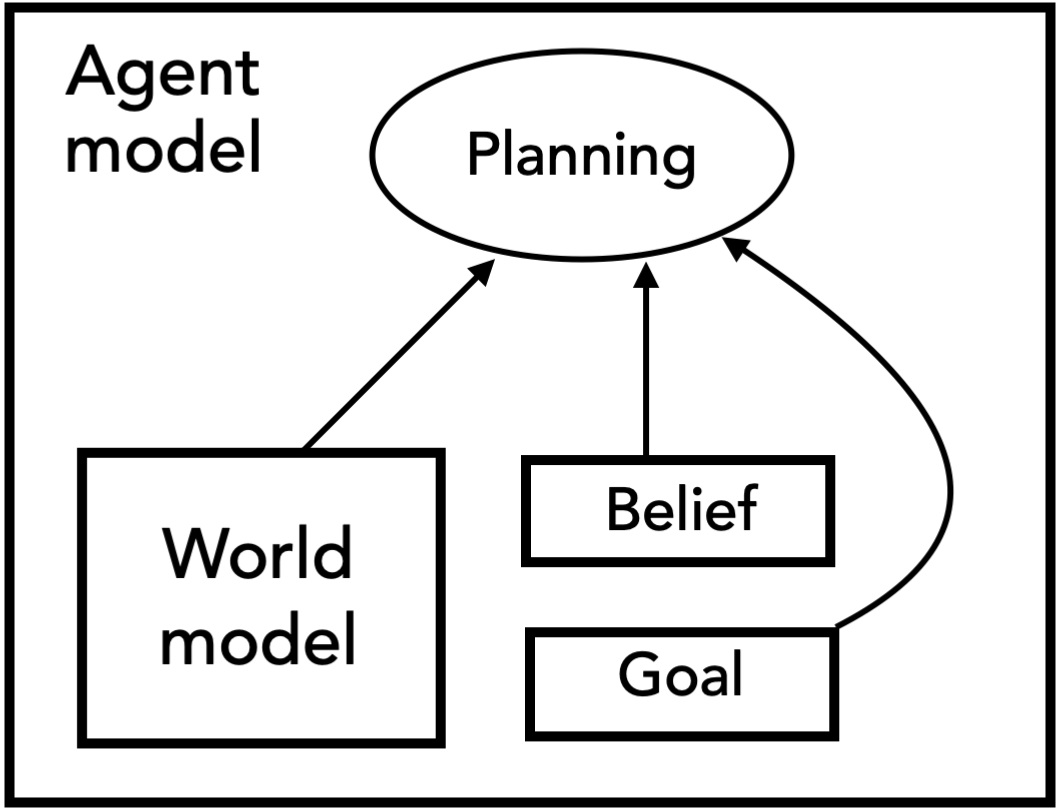
Sustained Embodied Reasoning in Rich Environments

Limits in LLMs

Does this person need help?

"Language Models meet World Models" Zhiting Hu and Tianmin Shu, 2023

World models and agent models

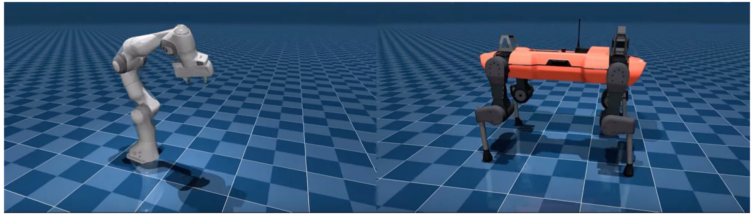


"Language Models meet World Models" Zhiting Hu and Tianmin Shu, 2023

Simulators as world models

- Physics engines / embodied simulators

MuJoCo



Todorov et al. (2012)

AI2-THOR

Kolve et al. (2017)

iGibson 2.0

Li et al. (2021)

Habitat 2.0

Szot et al. (2021)

ThreeDWorld

Gan et al. (2020)

Simulators as world models

- Embodied simulators + synthetic humans

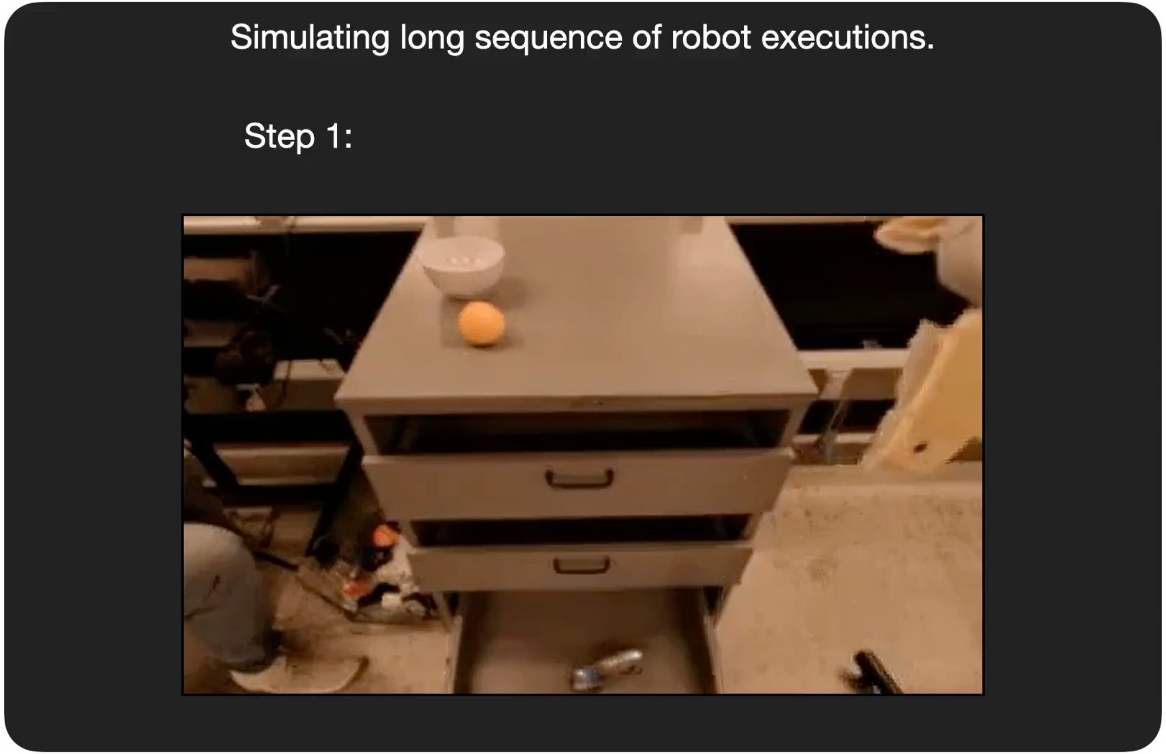
VirtualHome 2.0

Puig et al. (2021)

Habitat 3.0

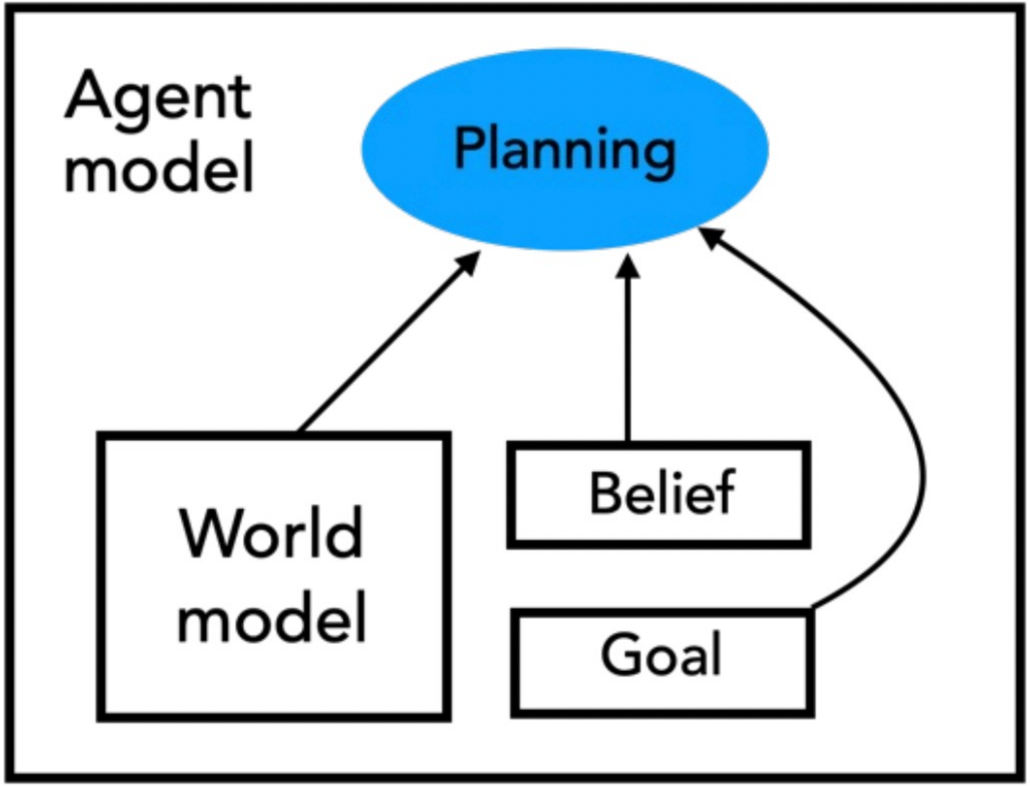
Puig et al. (2023)

Video prediction for robot planning



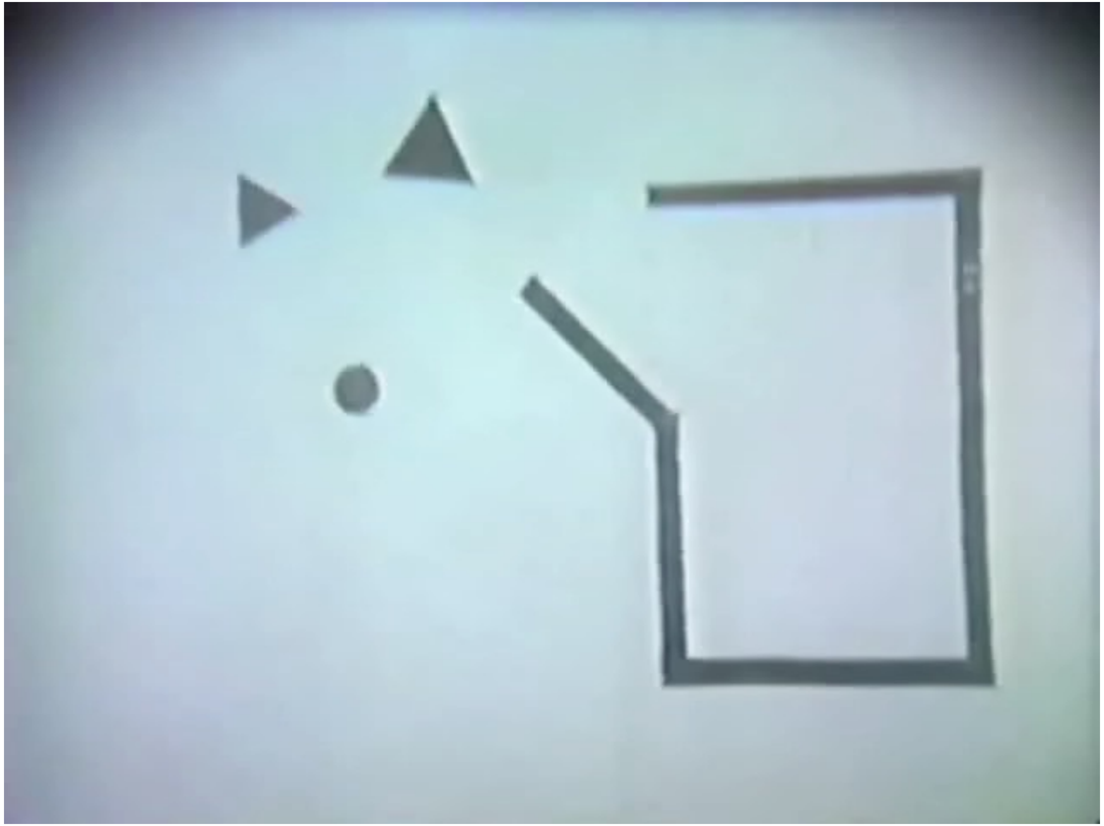
Yang et al. (2023)

Language models as (autoregressive) planners



"Language Models meet World Models" Zhiting Hu and Tianmin Shu, 2023

Humans represent agents differently from objects



Heider & Simmel (1944)

Humans represent agents differently from objects

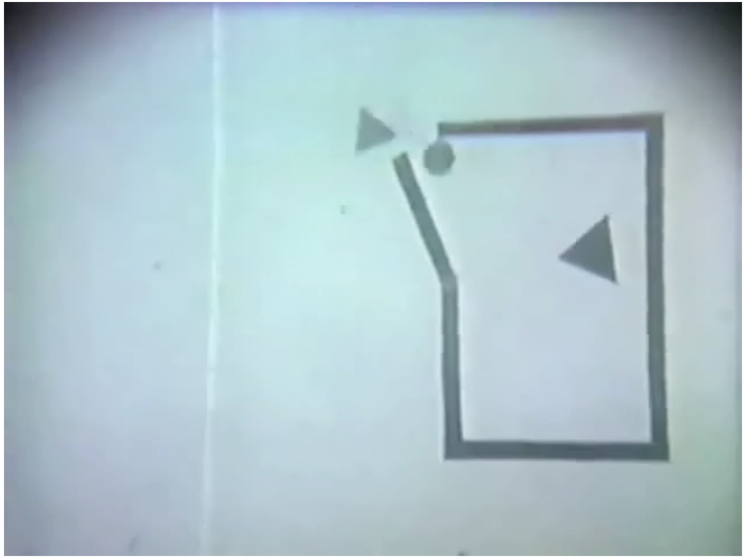
Strengths
strong, weak

Goals
helping, hurting, escaping

Relationships
friends, enemies

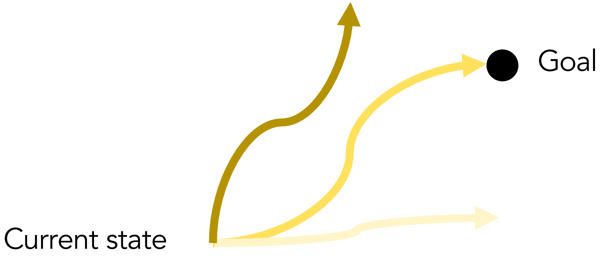
Moral judgment
good guy, bully

←



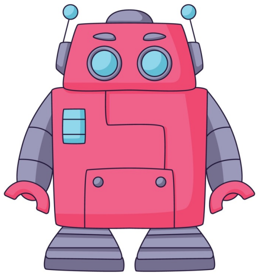
(size / velocity / angle…)
A big triangle moves back and forth, while a small triangle and a small circle rotate 360°…

Level-0 agent models for embodied tasks

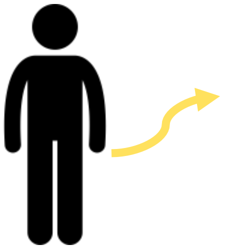


Simulate plans via world model

Choose the best plan

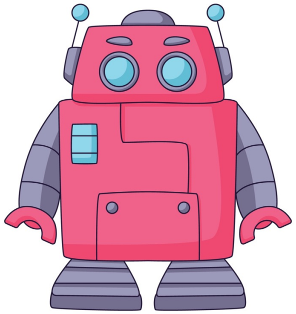


Level-1 agent models for social reasoning

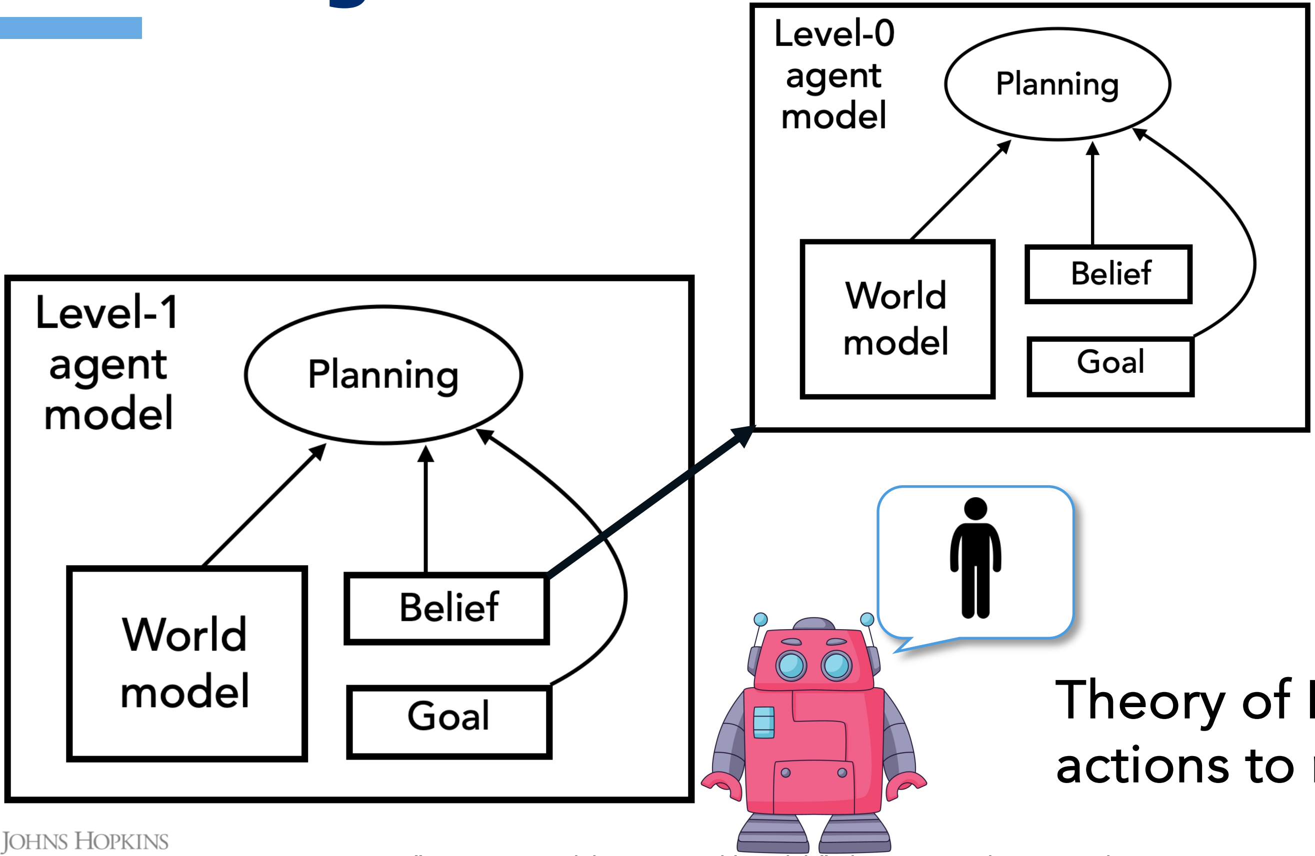


Goal: Office or coffee shop?

An observer



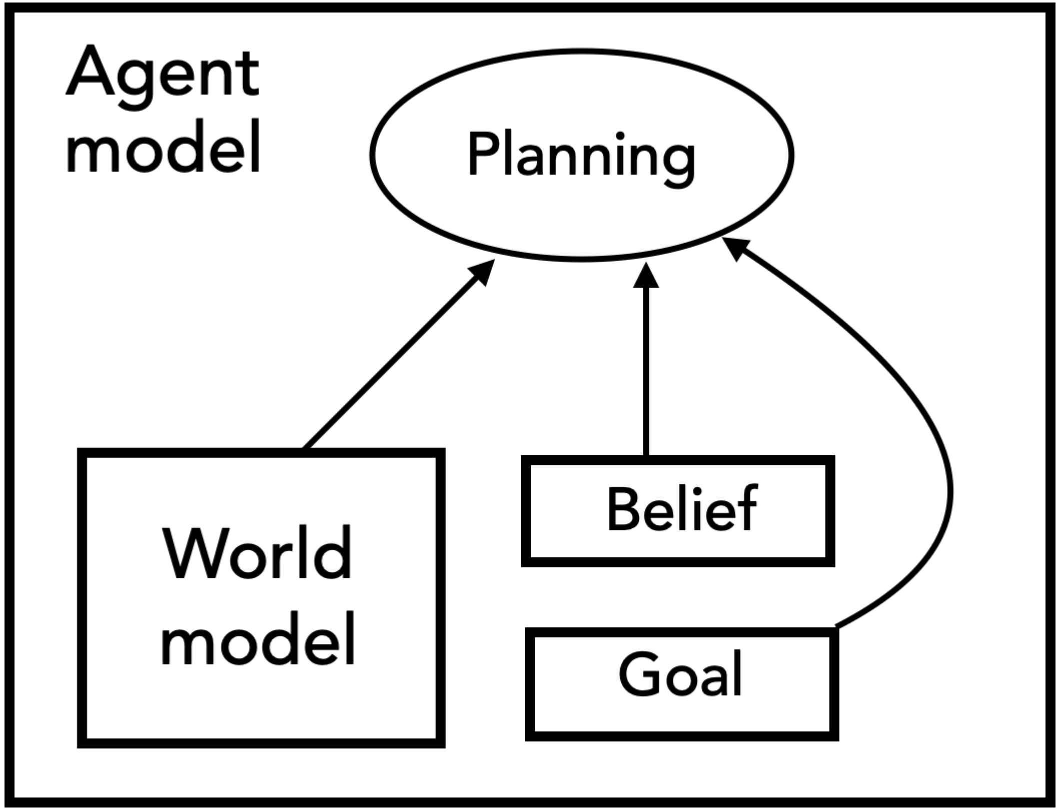
Level-1 agent models for social reasoning



Theory of Mind: mapping actions to mental state

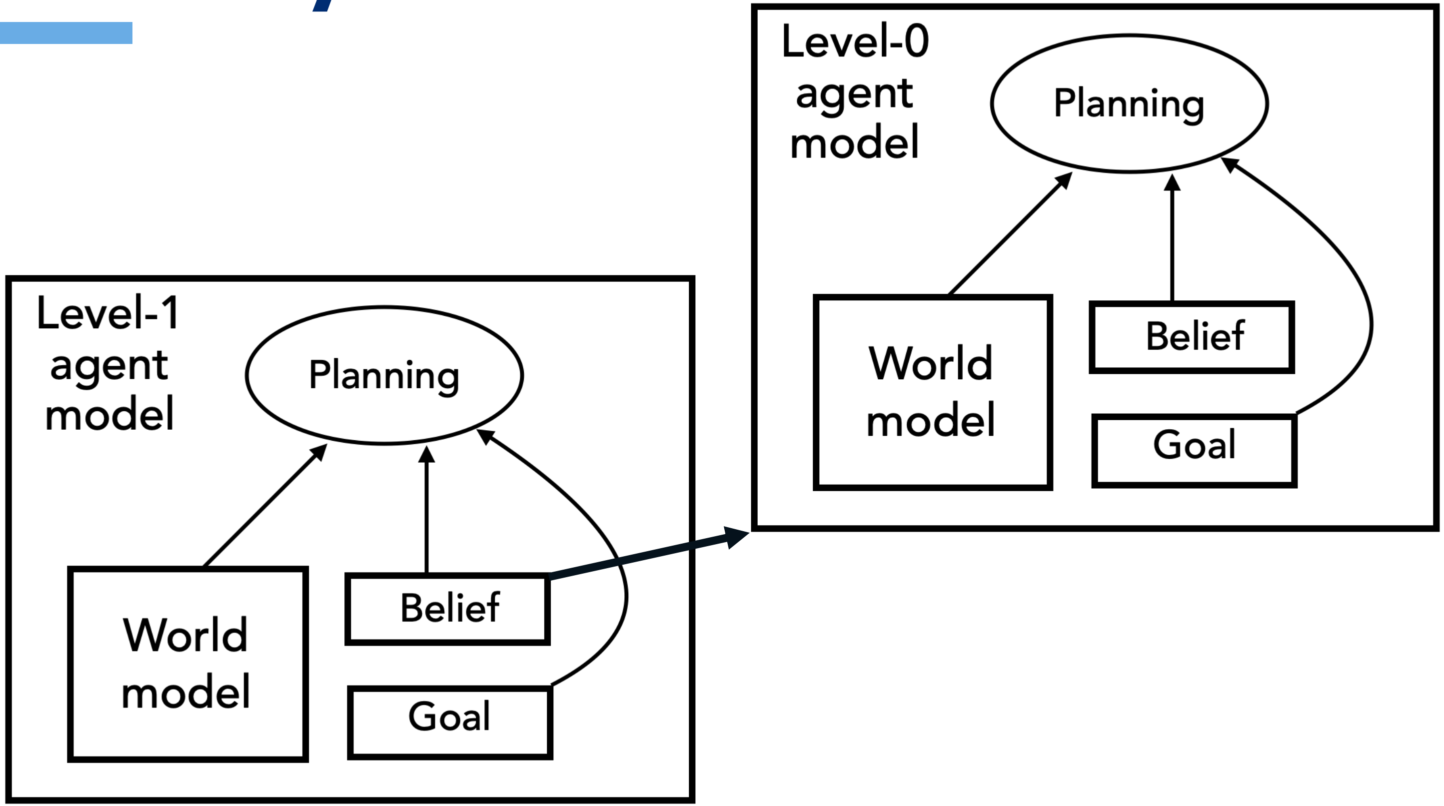
"Language Models meet World Models" Zhiting Hu and Tianmin Shu, 2023

Summary



"Language Models meet World Models" Zhiting Hu and Tianmin Shu, 2023

Summary

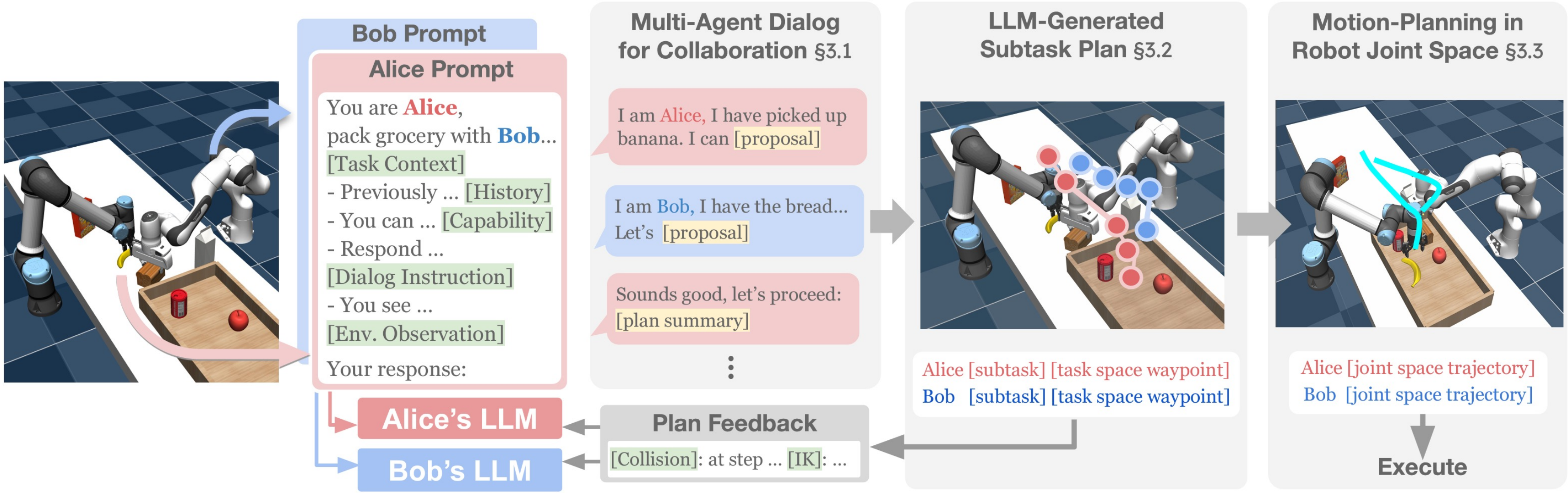


"Language Models meet World Models" Zhiting Hu and Tianmin Shu, 2023

Another issue is interaction with more then one agents

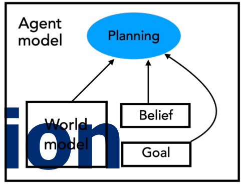
Multi-agent planning and communication

- Multi-agent collaboration



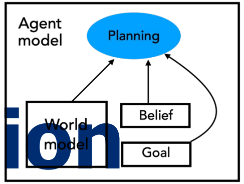
Mandi et al. (2023)

Multi-agent planning and communication



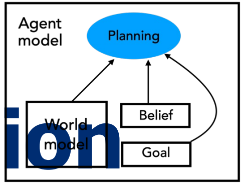
Zhang et al. (2023)

Multi-agent planning and communication



- Social

Park et al. (2023)



Multi-agent planning and communication

- Social interaction between *multiple* simulated agents

- Persona

- Social relationships with other agents

John Lin is a pharmacy shopkeeper at the Willow Market and Pharmacy who loves to help people.

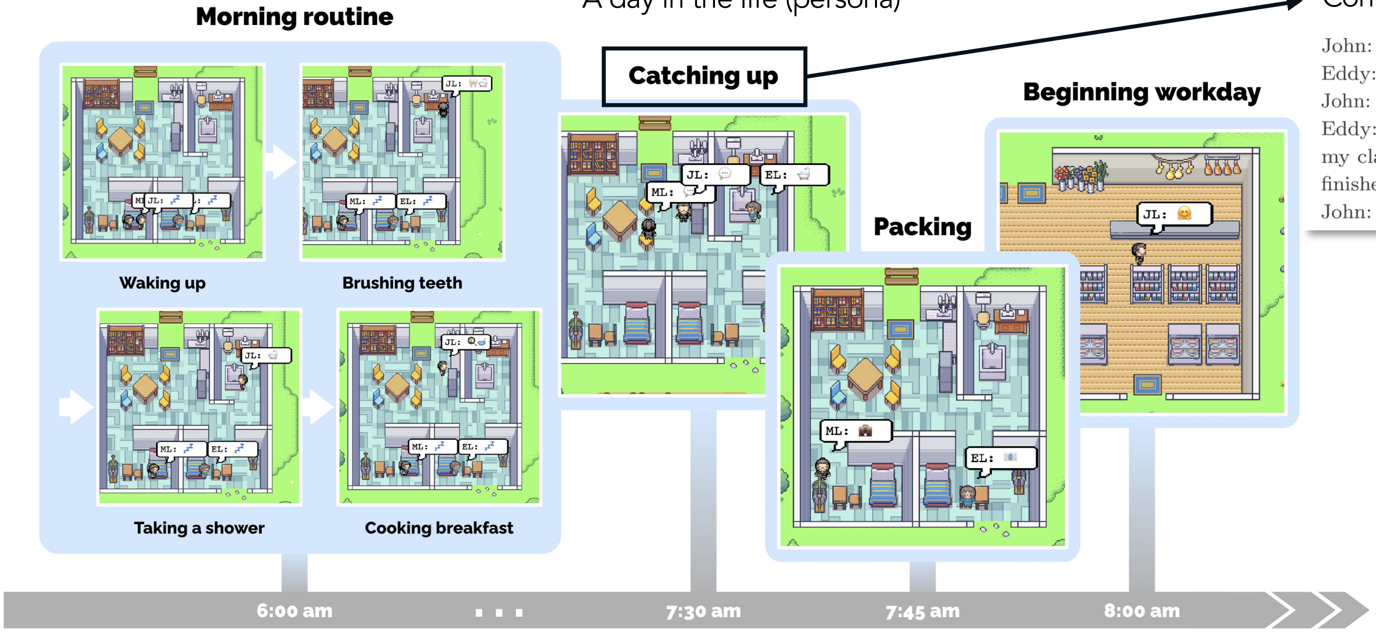
…

John Lin is living with his wife, Mei Lin, who is a college professor, ands on, Eddy Lin, who is a student studying music theory

Park et al. (2023)

Multi-agent planning and communication

A day in the life (persona)

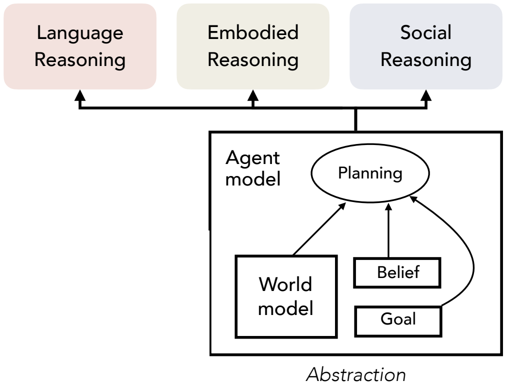


Conversation (social relation)

John: Good morning Eddy. Did you sleep well?
Eddy: Good morning dad. Yeah, I slept great.
John: That's good. What are you working on today?
Eddy: I'm working on a new music composition for my class. It's due this week, so I'm trying to get it finished. But I'm having so much fun with it!
John: That sounds great!

Park et al. (2023)

Summary



- Model-based reasoning
  - Data-efficient
  - Generalizes to new scenarios well

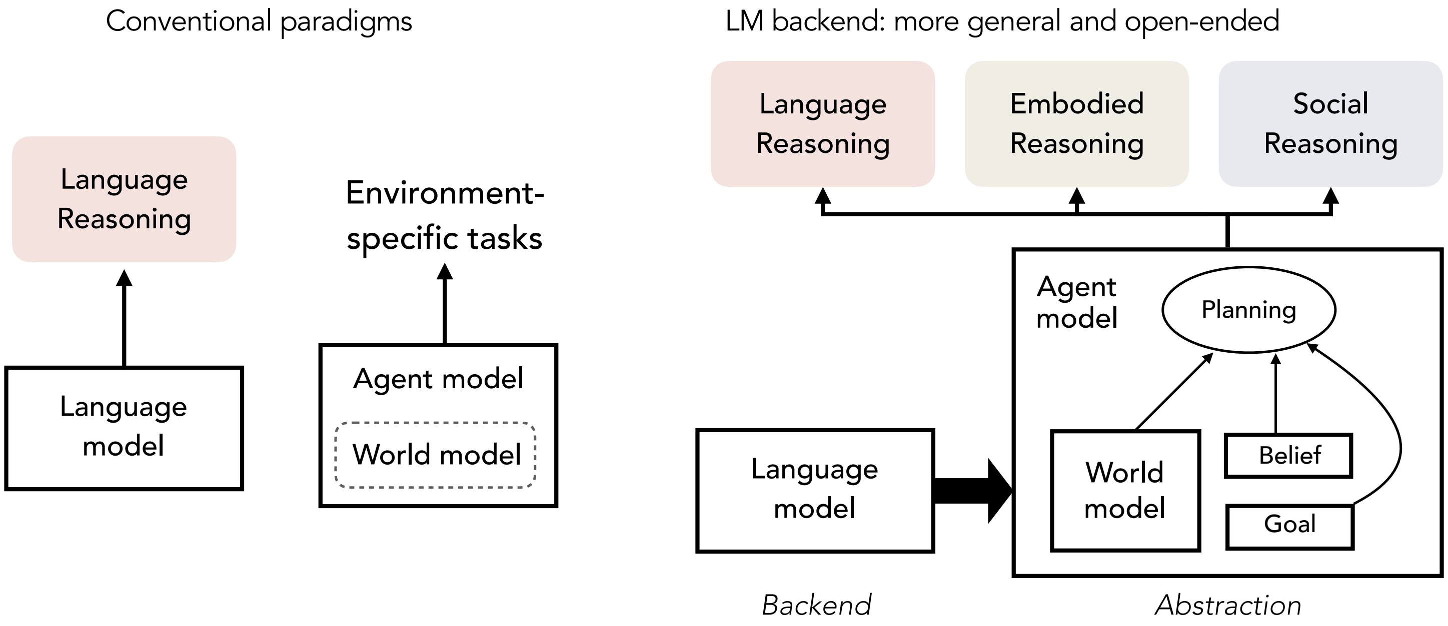
- Traditionally, world models and agent models are all domain specific

- Can we leverage the open-endedness of LMs to construct world and agent models?

- Language models as backend

Summary

- Language models as the backend for world models and agent models, supporting model-based reasoning



Summary

- Language models as the backend for world models and agent models, supporting model-based reasoning
- While language alone is not sufficient for many tasks, language models can be connected with models operating on other modalities to achieve multimodal capacity
- There is still a need for manually crafted components (such as belief / memory modules) or conventional methods (such as classical planning)
- Enhancing the language model backend
  - Multimodality?
  - Single-model or modular design?
  - ...

- Questions?

Envisioning the future

- More modalities — combinations of video (2D, 3D), text, code, etc.
- Large models and more efficient scaling
- More breath — more data and more types of data
- Interaction with physical world — models with hands and actuators
- Better personalization — these agents should serve your 👊 needs
- Better human-machine teaming
  - CoPilot for coding
  - CoPilot for writing
  - ….
  - CoPilot for life!!!

- https://sites.google.com/princeton.edu/cos597f