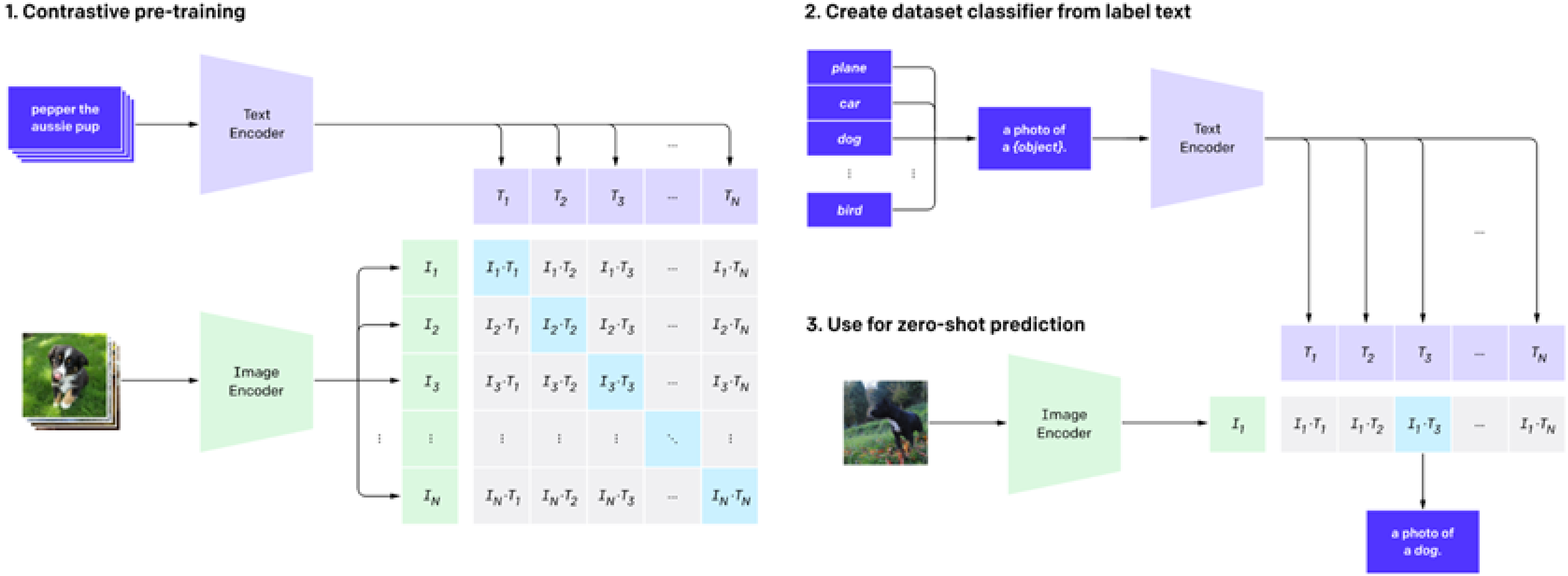
CS 4644 / 7643-A: Lecture 23
Danfei Xu

Large Vision and Language Models

From Self-Supervised Learning Lecture …

Contrastive learning between image and natural language sentences



CLIP (*Contrastive Language–Image Pre-training*) Radford *et al.*, 2021

Vision and Language Models:

Connecting the Pixel and Semantic Worlds at Scale

Why Vision-Language Models?

- Language is the most intuitive interface for an unstructured data space (e.g., natural images)
- Important to ground sensory information to semantic concepts
- Complementary information sources for a given task
- Claim: you cannot learn language without grounding it to the physical world, e.g., through visual sensing.
- Representations are converging (more on this later)

History: the first captioning model (Ordonez, 2011)

Im2Text: Describing Images Using 1 Million Captioned Photographs

Vicente Ordonez **Girish Kulkarni** **Tamara L Berg**

Stony Brook University
Stony Brook, NY 11794
{vordonezroma or tlberg}@cs.stonybrook.edu

Abstract

History: the first captioning model (Ordonez, 2011)

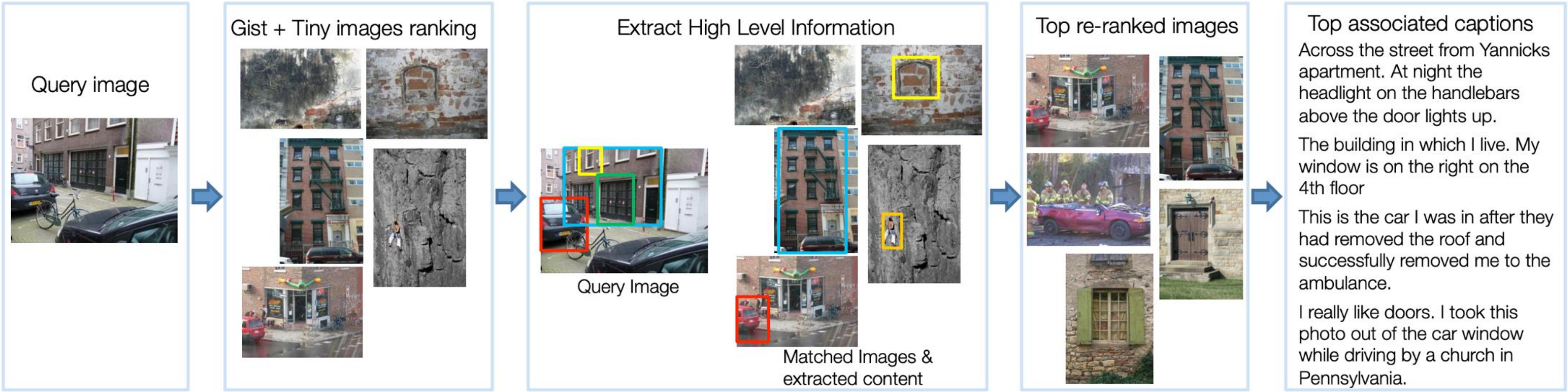


Image -> Image lookup -> match text description -> text stitching

History: the first deep captioning model (Vinyals, 2015)

Show and Tell: A Neural Image Caption Generator

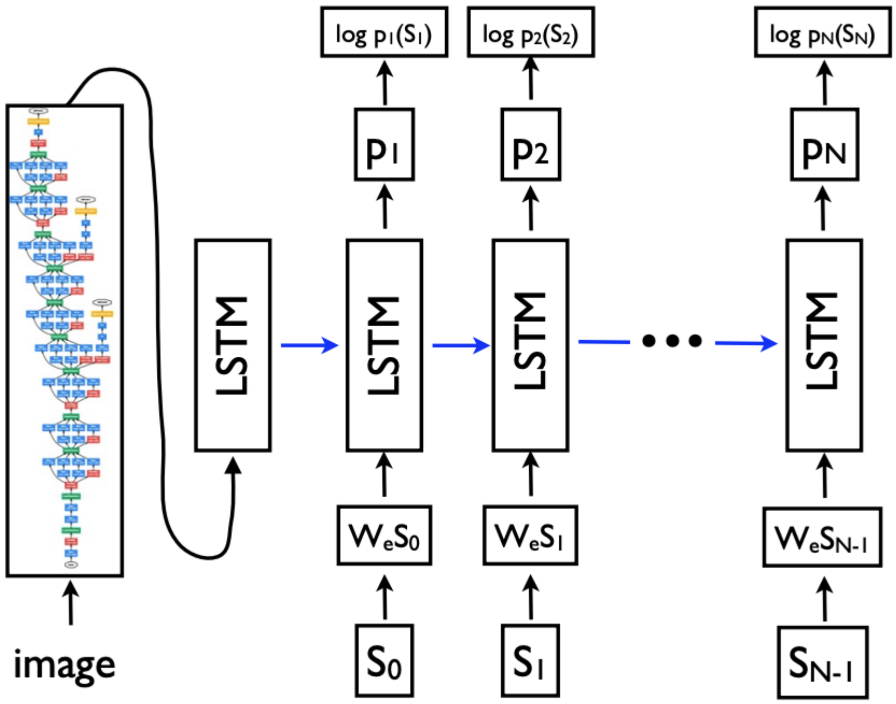
Oriol Vinyals
Google
vinyals@google.com

Alexander Toshev
Google
toshev@google.com

Samy Bengio
Google
bengio@google.com

Dumitru Erhan
Google
dumitru@google.com

History: the first deep captioning model (Vinyals, 2015)



History: the first VQA model (Agrawal, 2015)

VQA: Visual Question Answering

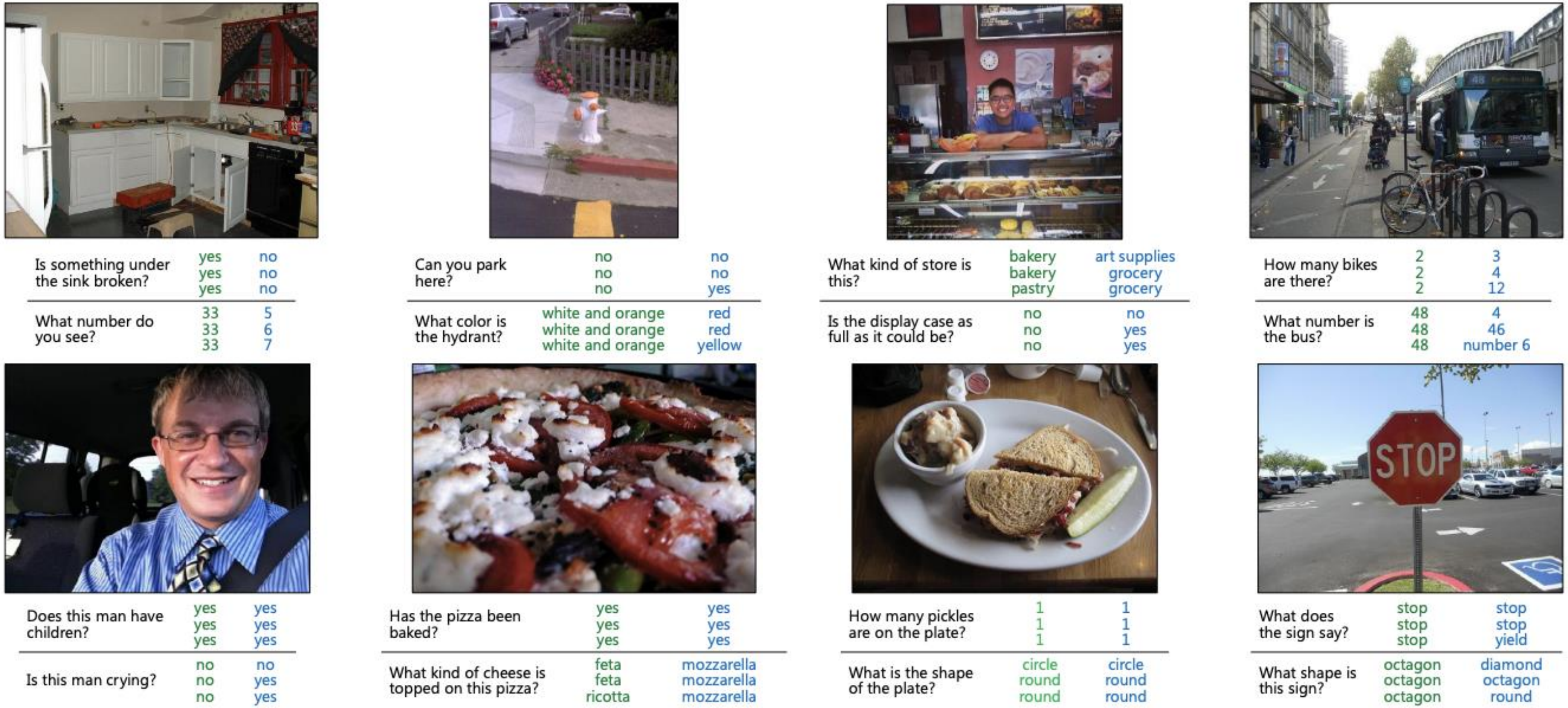
www.visualqa.org

Aishwarya Agrawal*, Jiasen Lu*, Stanislaw Antol*,
Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, Devi Parikh

Abstract—We propose the task of *free-form* and *open-ended* Visual Question Answering (VQA). Given an image and a natural language question about the image, the task is to provide an accurate natural language answer. Mirroring real-world scenarios, such as helping the visually impaired, both the questions and answers are open-ended. Visual questions selectively target different areas of an image, including background details and underlying context. As a result, a system that succeeds at VQA typically needs a more detailed understanding of the image and complex reasoning than a system producing generic image captions. Moreover, VQA is amenable to automatic evaluation, since many open-ended answers contain only a few words or a closed set of answers that can be provided in a multiple-choice format. We provide a dataset containing ∼0.25M images, ∼0.76M questions, and ∼10M answers (www.visualqa.org), and discuss the information it provides. Numerous baselines and methods for VQA are provided and compared with human performance. Our VQA demo is available on CloudCV (http://cloudcv.org/vqa).

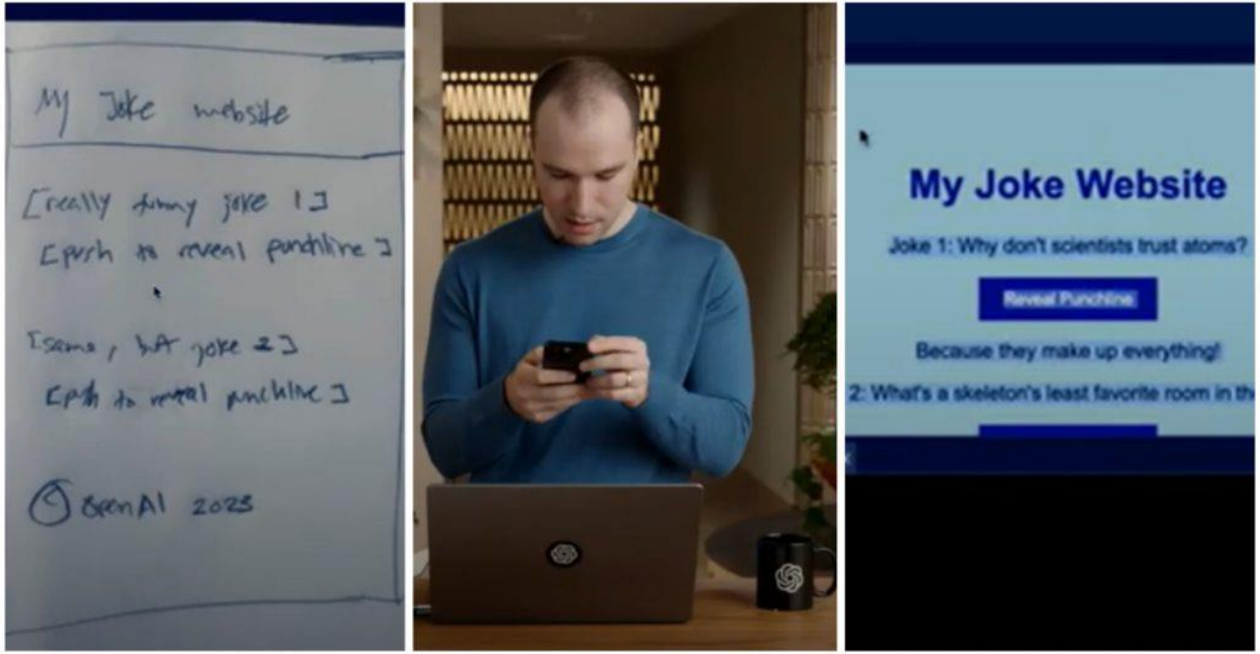
Standard task: Visual Question Answer

History: the first VQA model (Agrawal, 2015)



Free-form Text + Image -> Free-form Text

Foundation VLM (2019-)



Hand-drawn sketch + instruction -> website source code

GPT 4v(ision) (OpenAI, 2023)

Major Areas

- **Representation**: how to convert raw data into meaningful features
- **Translation**: transform one modality to another
- **Alignment**: discover relationships between elements across modalities
- **Fusion**: join features from modalities to support prediction
- **Co-learning**: transferring knowledge from one modality to another for some downstream tasks

Slide credit: Desmont Elliott

Language->Vision: Language-guided Image Gen

TEXT DESCRIPTION

An astronaut   Teddy bears   A bowl of soup

riding a horse   lounging in a tropical resort in space   playing basketball with cats in space

in a photorealistic style   in the style of Andy Warhol   as a pencil drawing

→

DALL·E 2

https://openai.com/dall-e-2/

Vision->Language: Image Captioning

Captions generated using neuraltalk2
All images are CC0 Public domain: cat suitcase, cat tree, dog, bear, surfers, tennis, giraffe, motorcycle

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

Two people walking on the beach with surfboards

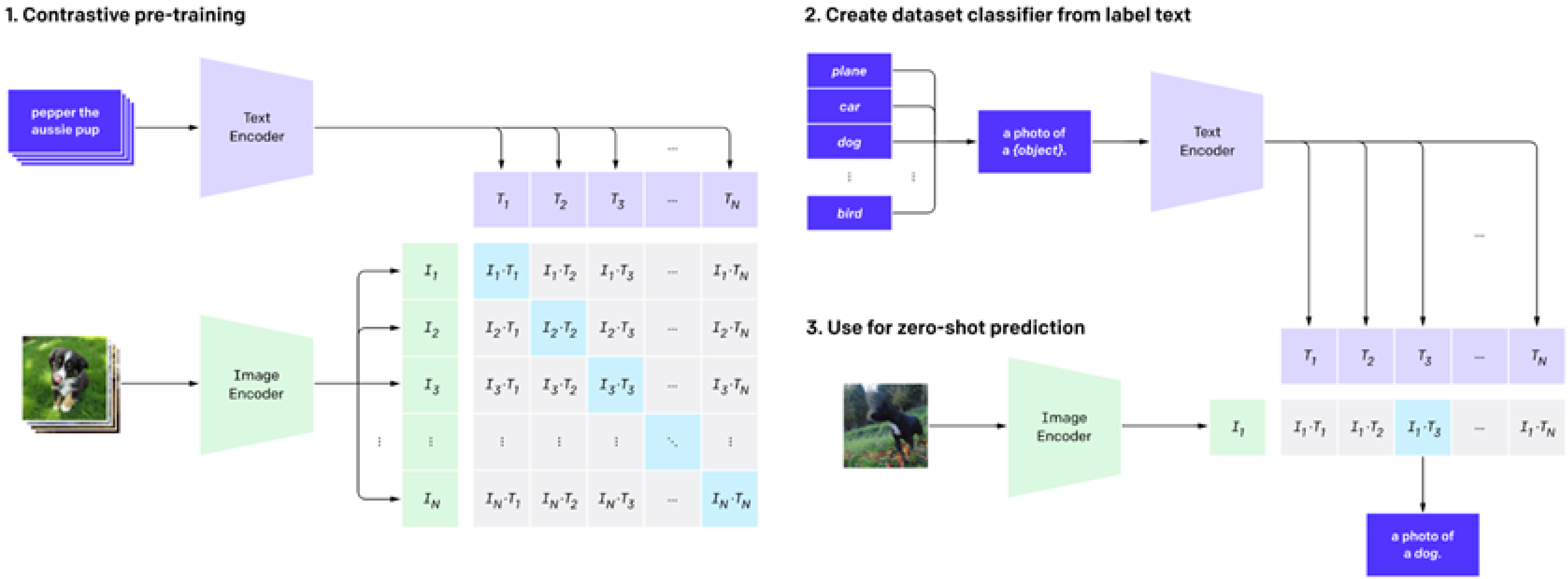
A tennis player in action on the court

Two giraffes standing in a grassy field

A man riding a dirt bike on a dirt track

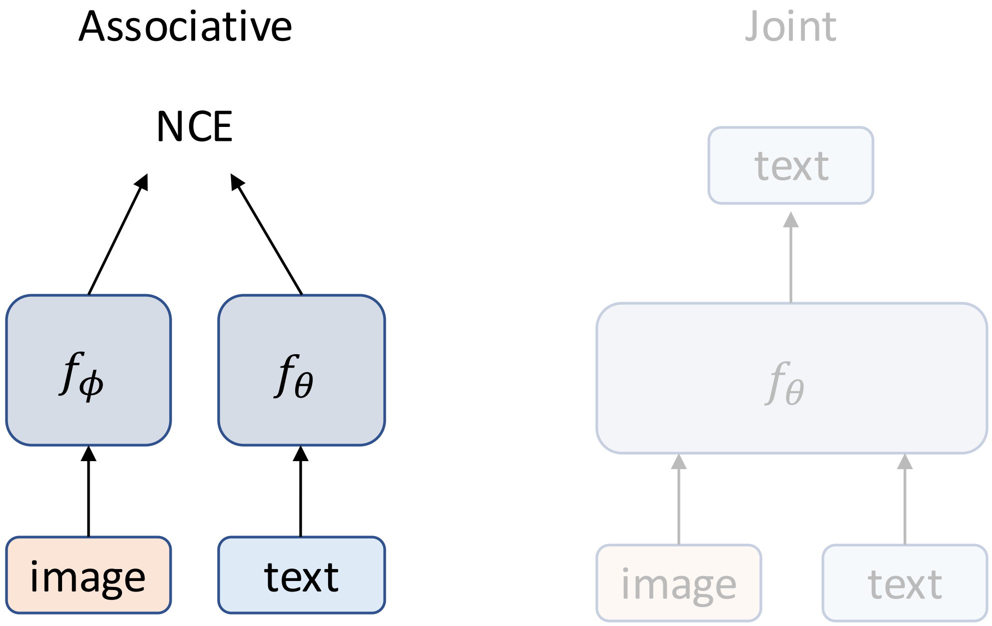
Image – Language Association

Contrastive learning between image and natural language sentences

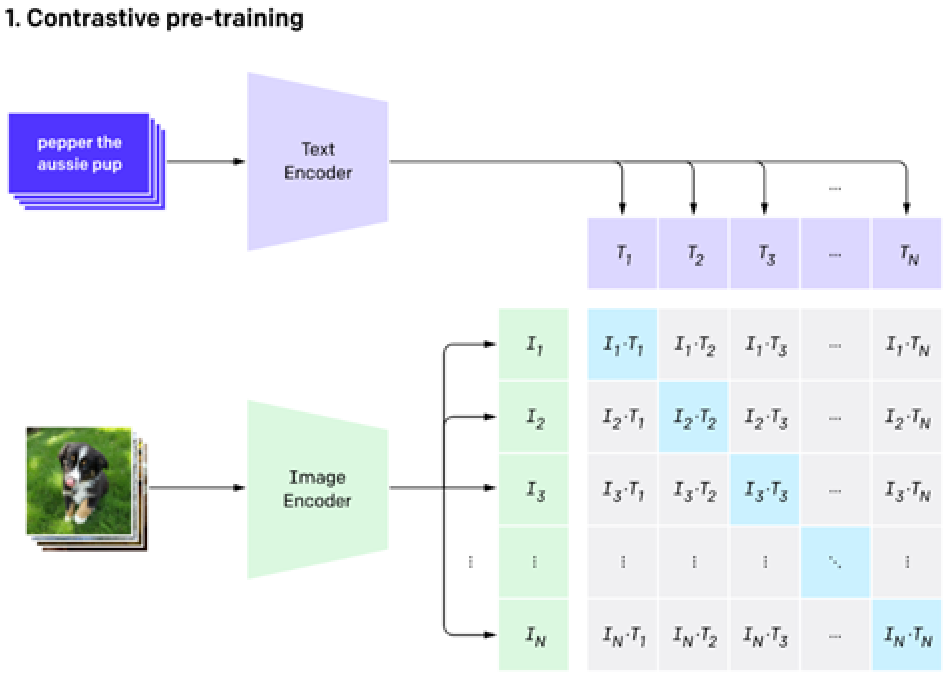


CLIP (*Contrastive Language–Image Pre-training*) Radford *et al.*, 2021

Image – language encoding architectures



CLIP: Associative Encoding



CLIP (*Contrastive Language–Image Pre-training*) Radford *et al.*, 2021

Recall: Noise Contrastive Learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\overline{\exp(s(f(x), f(x^+))}}{\underbrace{\exp(s(f(x), f(x^+))}_{\text{score for the positive pair}} + \underbrace{\sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))}_{\text{score for the N-1 negative pairs}}} \right]$$

Cross entropy loss for a N-way softmax classifier

I.e., learn to find the positive sample from the N samples

CLIP: Training

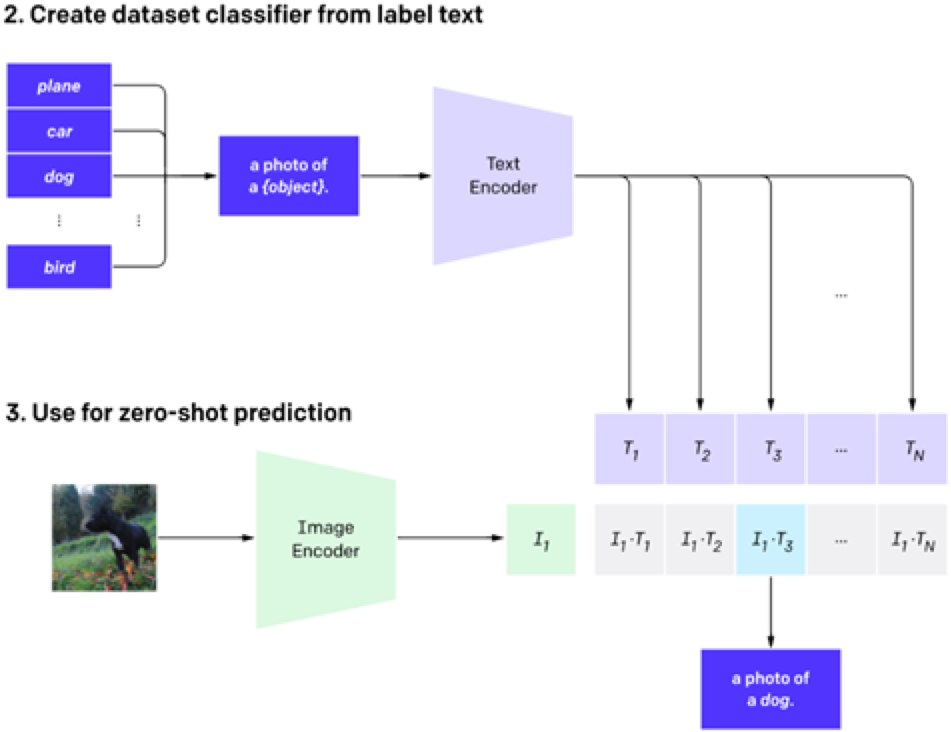
```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned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

Predict image -> text association

Predict text -> image association

CLIP (*Contrastive Language–Image Pre-training*) Radford *et al.*, 2021

CLIP: Zero-shot Classification



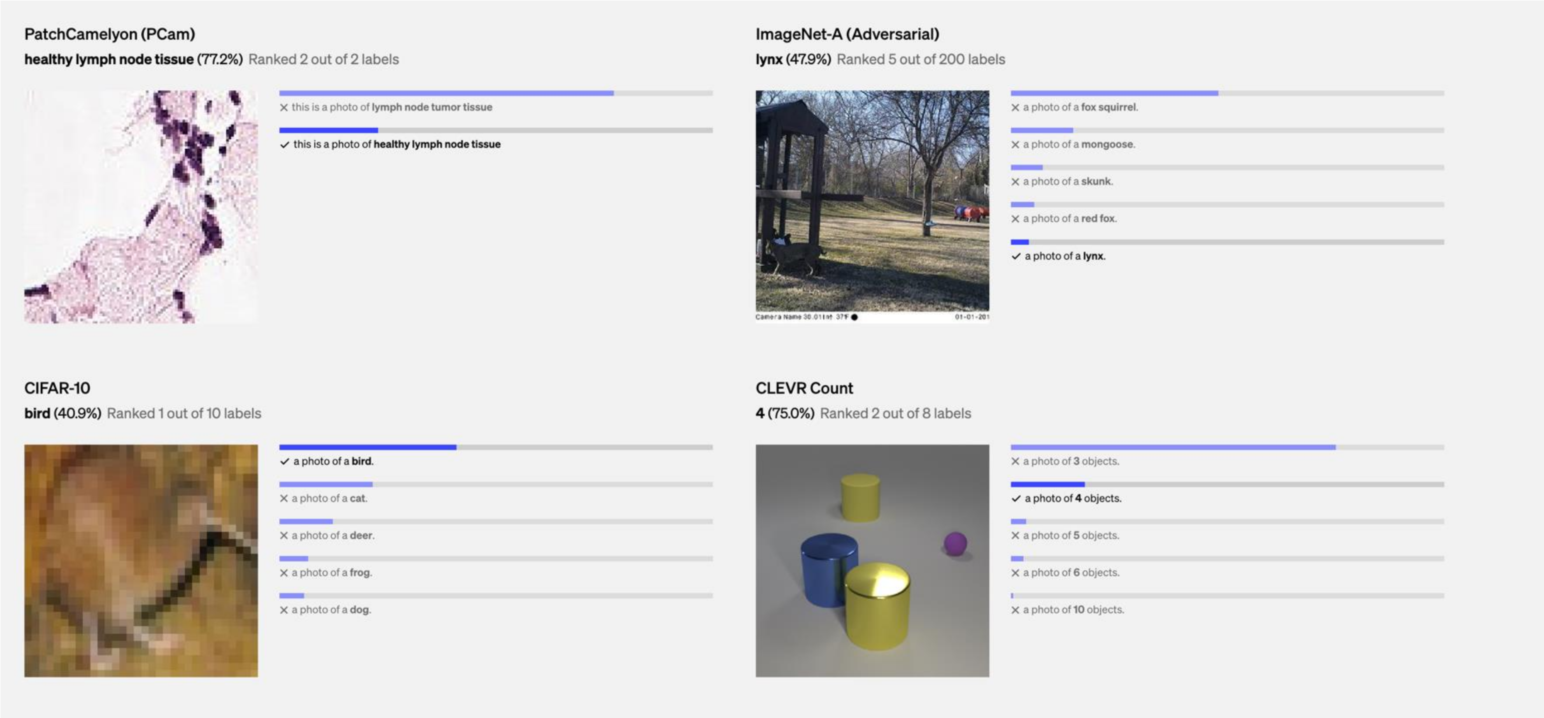
CLIP (*Contrastive Language–Image Pre-training*) Radford *et al.*, 2021

CLIP: Zero-shot Classification

```python
# Load the model
device = "cuda" if torch.cuda.is_available() else "cpu"
model, preprocess = clip.load('ViT-B/32', device)

# Download the dataset
cifar100 = CIFAR100(root=os.path.expanduser("~/.cache"), download=True, train=False)

# Prepare the inputs
image, class_id = cifar100[3637]
image_input = preprocess(image).unsqueeze(0).to(device)
text_inputs = torch.cat([clip.tokenize(f"a photo of a {c}") for c in cifar100.classes]).to(device)

# Calculate features
with torch.no_grad():
    image_features = model.encode_image(image_input)
    text_features = model.encode_text(text_inputs)

# Pick the top 5 most similar labels for the image
image_features /= image_features.norm(dim=-1, keepdim=True)
text_features /= text_features.norm(dim=-1, keepdim=True)
similarity = (100.0 * image_features @ text_features.T).softmax(dim=-1)
values, indices = similarity[0].topk(5)
```

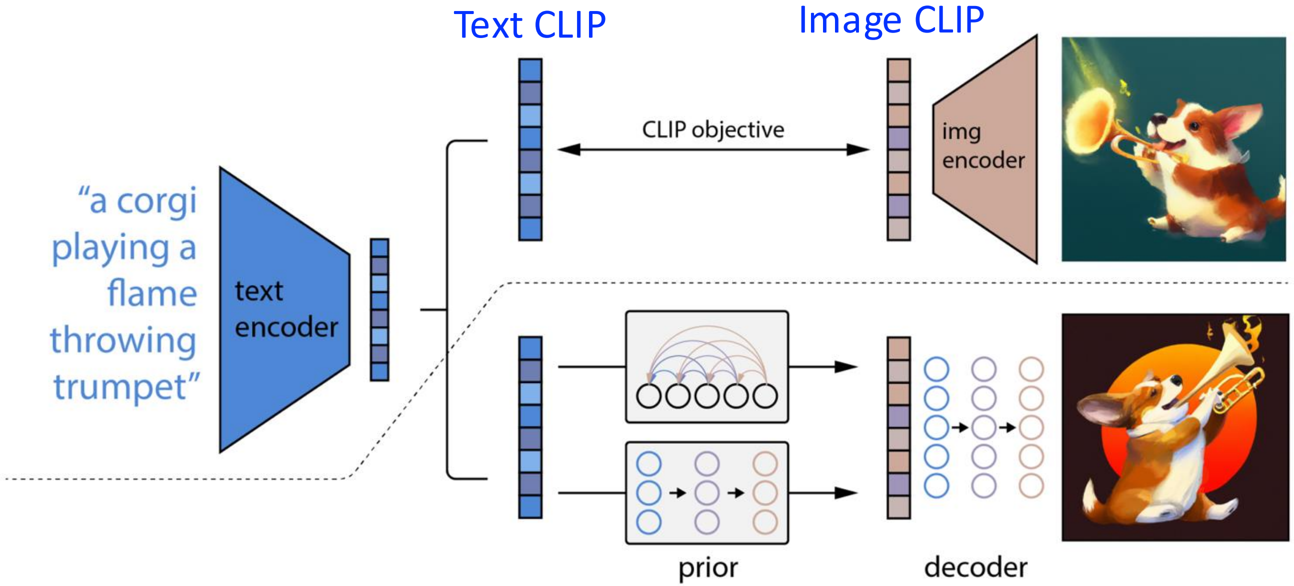
https://github.com/openai/CLIP

CLIP: Zero-shot Classification



CLIP (*Contrastive Language–Image Pre-training*) Radford *et al.*, 2021

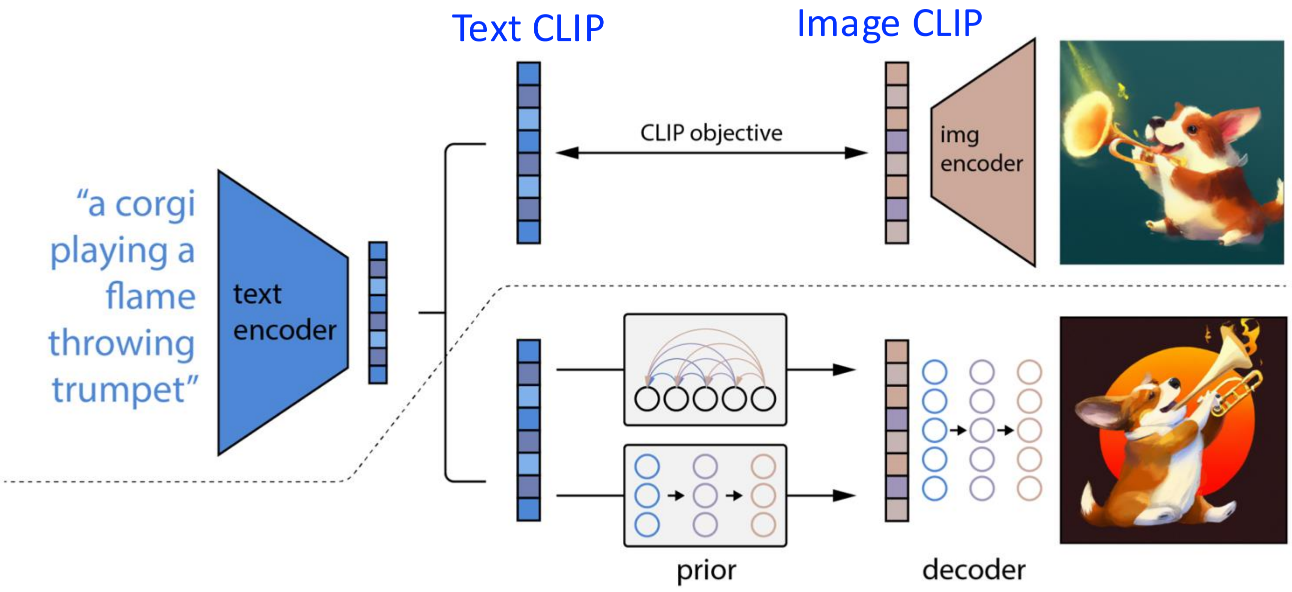
Generating Images from CLIP Latents (DALL-E 2)



- Train image diffusion with classifier-free guidance using CLIP image embedding
- Train another diffusion model to predict CLIP image embedding from the CLIP embedding of the input text.

Hierarchical Text-Conditional Image Generation with CLIP Latents (Ramesh, Dhariwal, Nichol, Chu, Chen, 2022)

Generating Images from CLIP Latents (DALL-E 2)

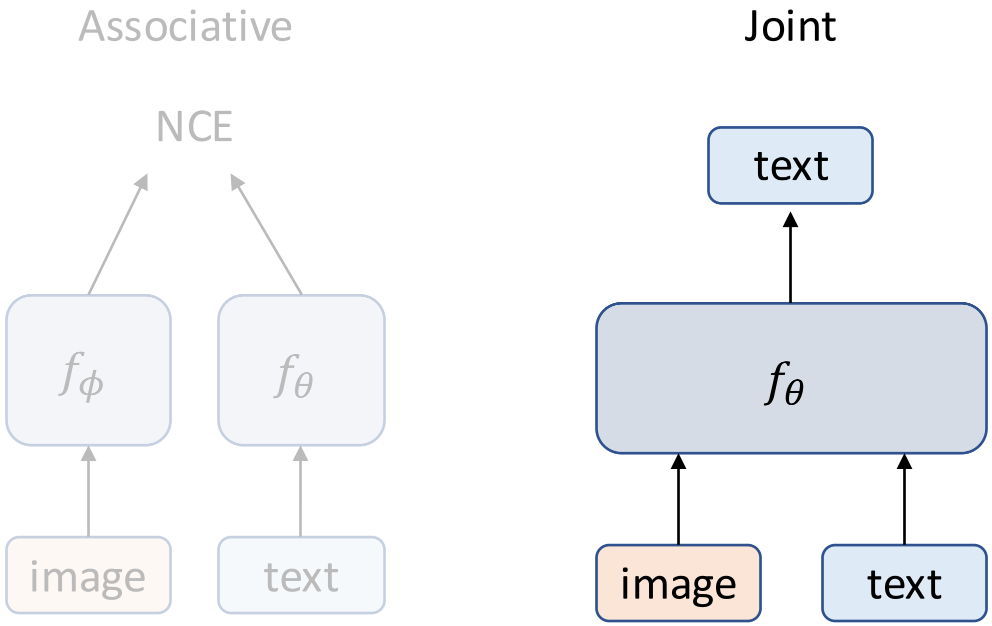


Learning objective for the text to image CLIP embedding diffusion model:

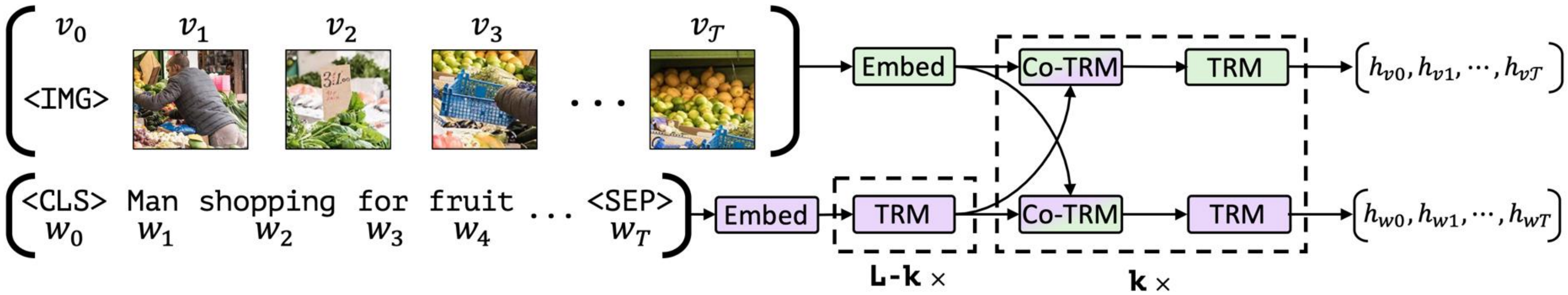
$$L_{\text{prior}} = \mathbb{E}_{t\sim[1,T],z_i^{(t)}\sim q_t}\left[\|f_\theta(z_i^{(t)}, t, y) - z_i\|^2\right]$$

Hierarchical Text-Conditional Image Generation with CLIP Latents (Ramesh, Dhariwal, Nichol, Chu, Chen, 2022)

Image – language encoding architectures



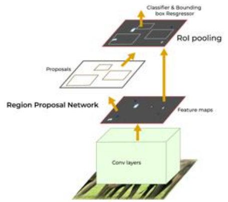
Joint Encodings: ViLBERT (2019)

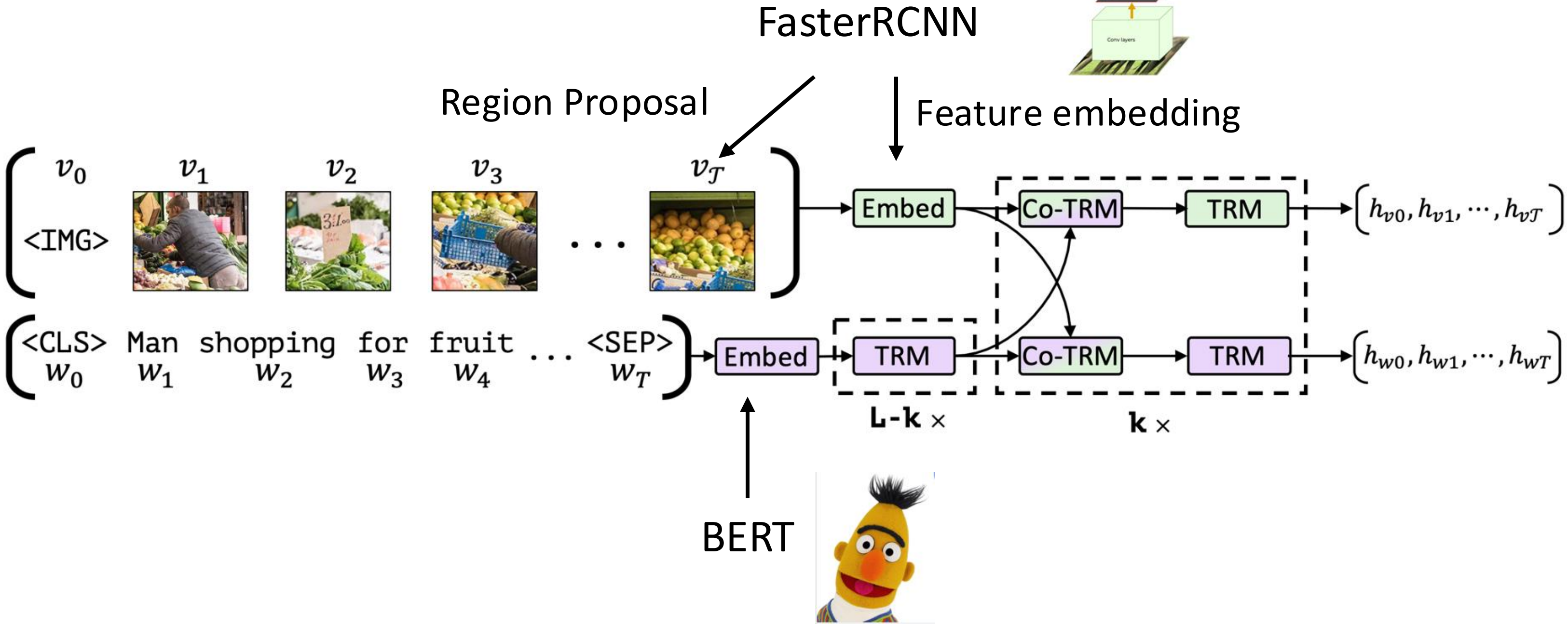


Vision and Language Joint Pretraining

ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks (Lu et al., 2019)

Joint Encodings: ViLBERT (2019)

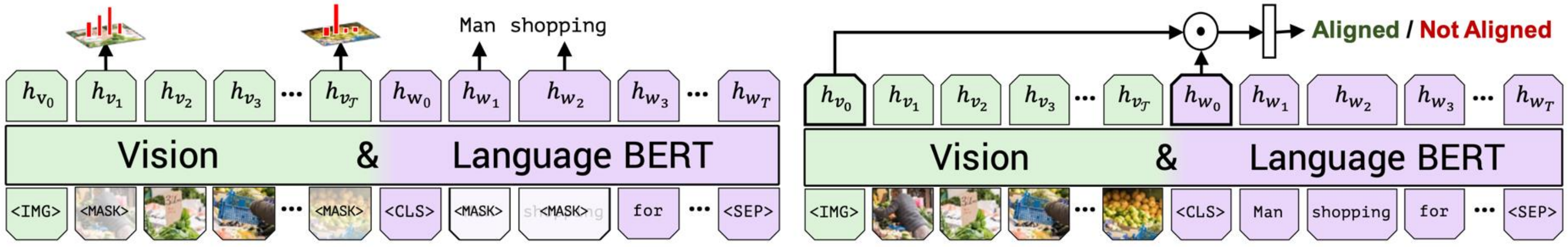




Vision and Language Joint Pretraining

ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks (Lu et al., 2019)

Joint Encodings: ViLBERT (2019)

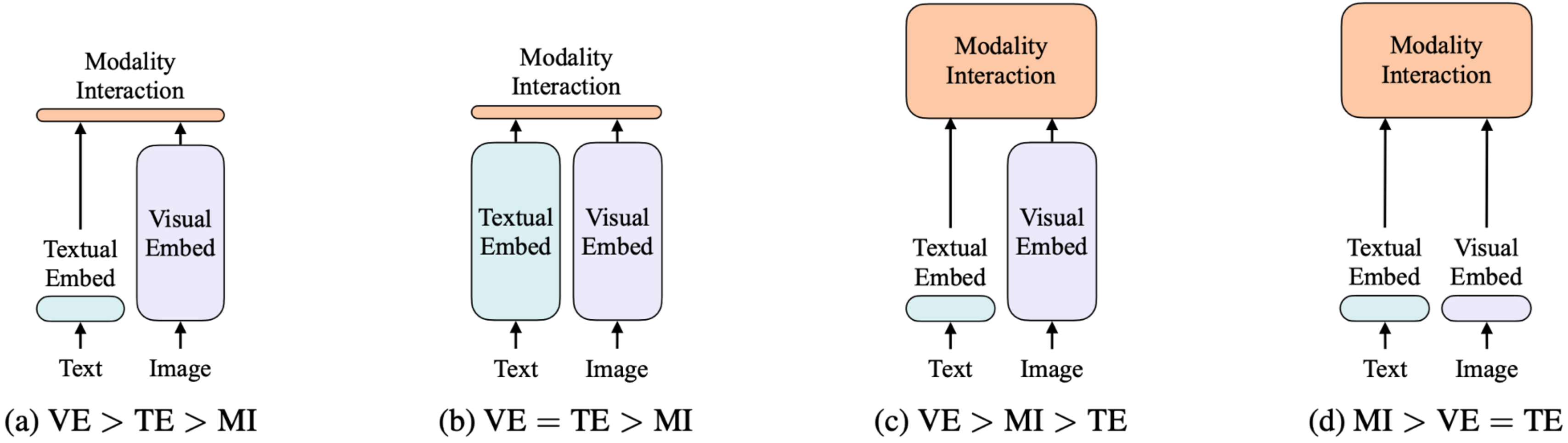


(a) Masked multi-modal learning (b) Multi-modal alignment prediction

Vision and Language Joint Pretraining

ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks (Lu et al., 2019)

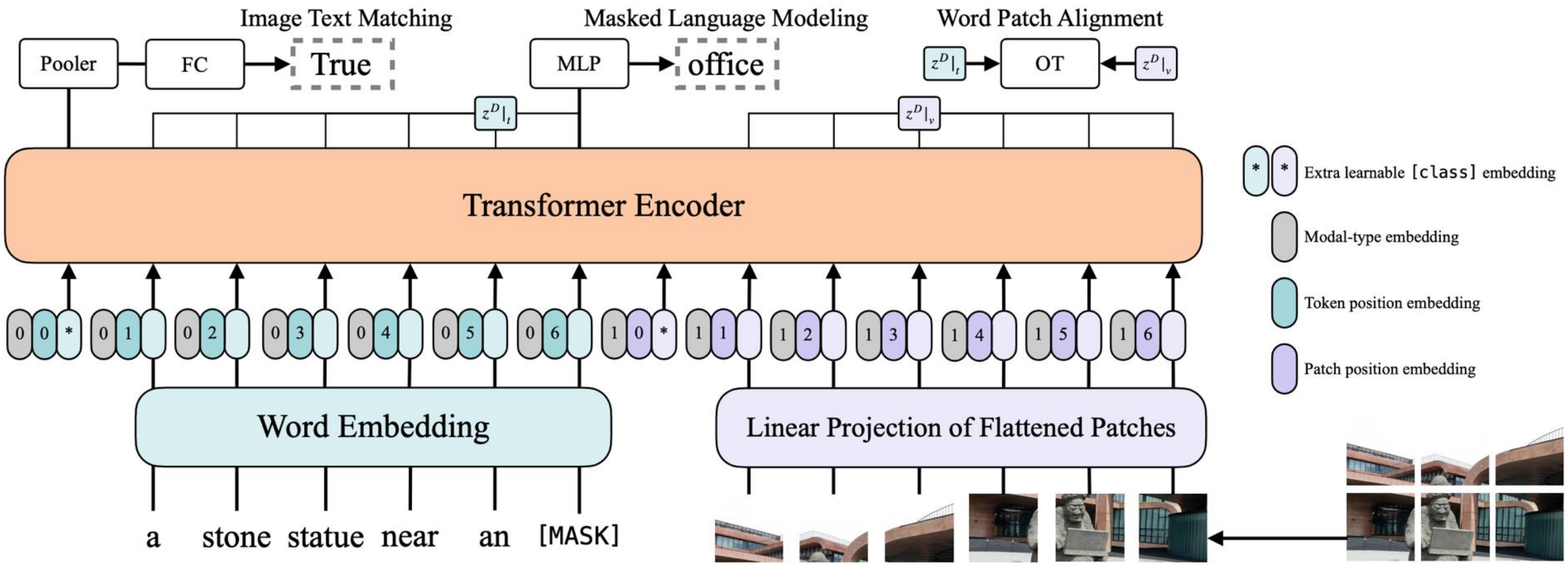
Joint Encodings: ViLT (2021)



Categories of vision-language model in terms of model complexity / capacity

ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision (Kim and Son, 2021)

Joint Encodings: ViLT (2021)



Vision and Language Joint Pretraining

ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision (Kim and Son, 2021)

Data matters

Scaling Up Foundation Vision and Language Models

Pre-foundation model era (2015 – 2020)

Visual Question Answering
(Goyal and Knot, 2017)

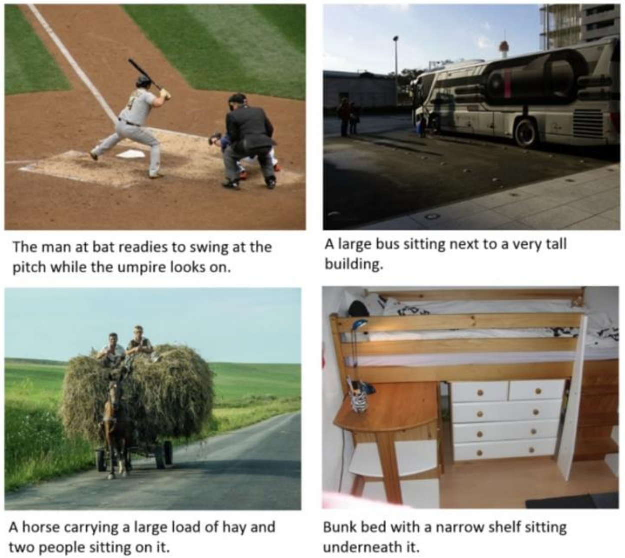
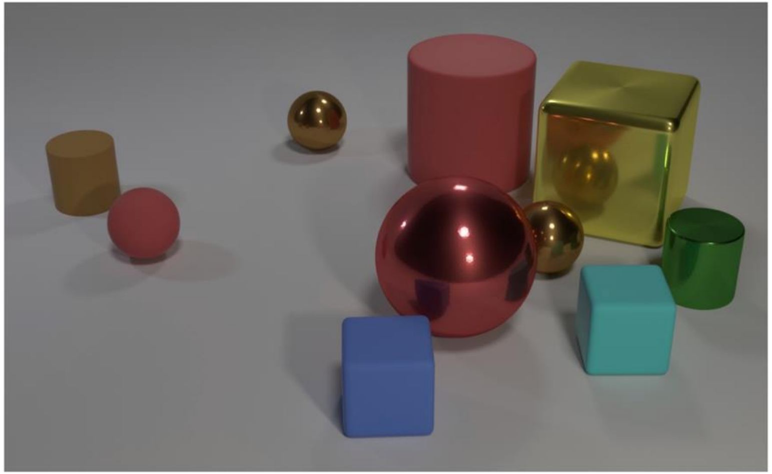


Image Captioning
(MS-COCO)

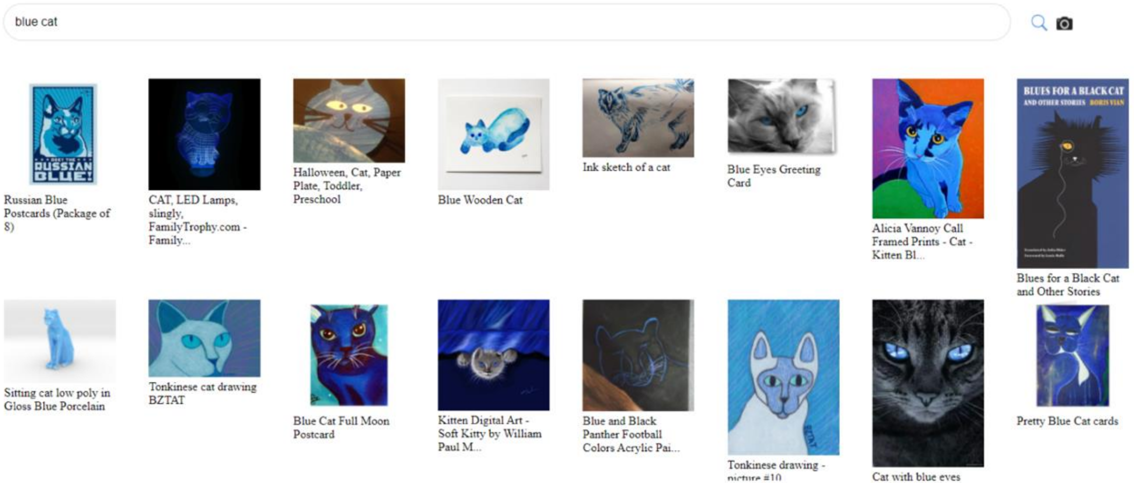
Pre-foundation model era (2015 – 2020)



Q: Are there an equal number of large things and metal spheres?
Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere? **Q:** There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?
Q: How many objects are either small cylinders or metal things?

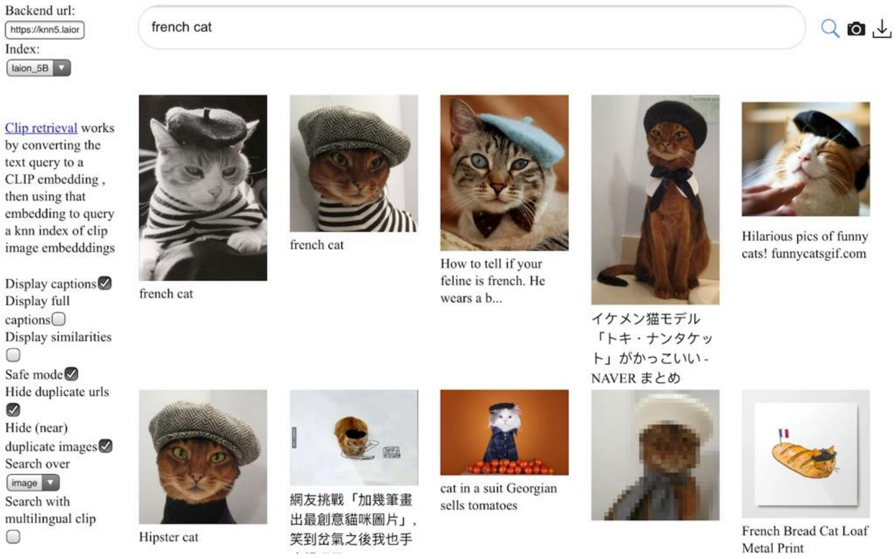
Diagnostic Language and Visual Reasoning
(CLEVR, Johnson et al., 2016)

The “Foundation Model Era” (2020-now)



- **LAION-400M**: 400 million image-text pairs
- Built using Common Crawl datasets,
- Extracting image-text pairs from HTML data.
- Post-processing filters unsuitable pairs using OpenAI's CLIP model.
- A 10TB webdataset with CLIP embeddings and kNN indices.

The “Foundation Model Era” (2020-now)



- **LAION-5B**: Significantly larger than LAION-400M
- Crawled using 50 billion webpages + CLIP filtering
- 2.3 billion pairs in English + 2.2 billions in other languages + 1 billion unassignable languages (e.g., names).

The “Foundation Model Era” (2020-now)

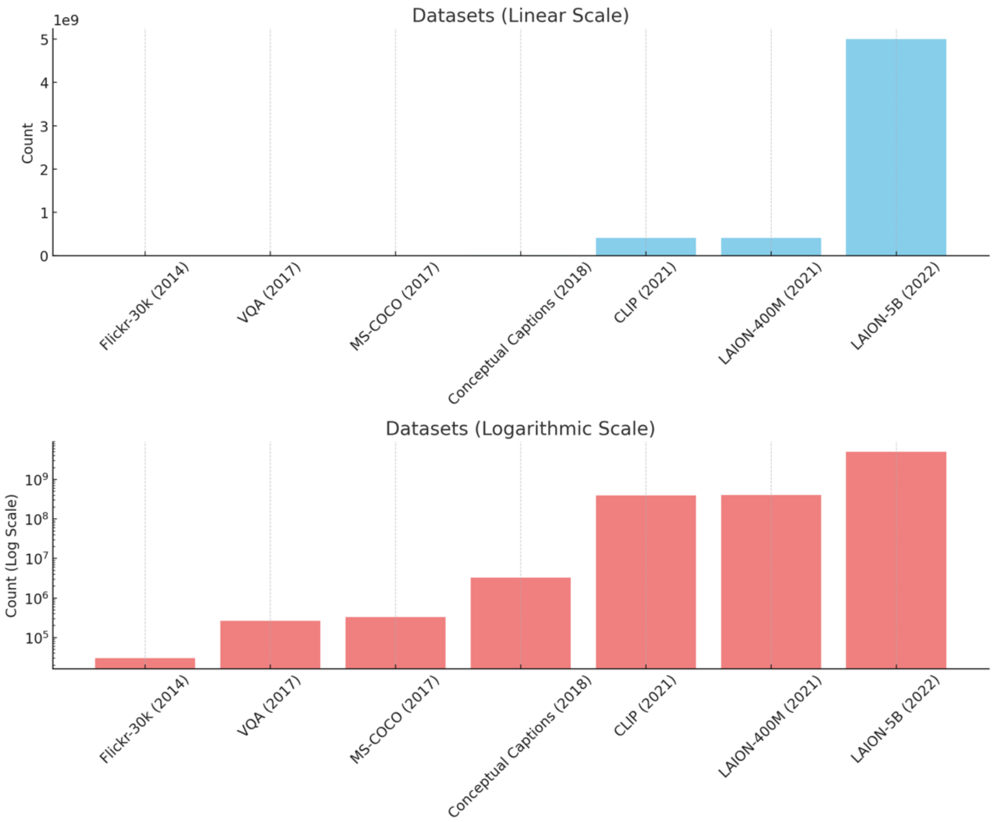
Stable Diffusion

Stable Diffusion was made possible thanks to a collaboration with Stability AI and Runway and builds upon our previous work:

High-Resolution Image Synthesis with Latent Diffusion Models
Robin Rombach*, Andreas Blattmann*, Dominik Lorenz, Patrick Esser, Björn Ommer
CVPR '22 Oral | *GitHub* | *arXiv* | *Project page*

Stable Diffusion is a latent text-to-image diffusion model. Thanks to a generous compute donation from Stability AI and support from LAION, we were able to train a Latent Diffusion Model on 512x512 images from a subset of the LAION-5B database. Similar to Google's Imagen, this model uses a frozen CLIP ViT-L/14 text encoder to condition the model on text prompts. With its 860M UNet and 123M text encoder, the model is relatively lightweight and runs on a GPU with at least 10GB VRAM. See this section below and the model card.

A snapshot of vision-language dataset



Automatic data crawling is great but …

tomclancysthedivision2_gc18images_0001

Enchantments-JUN16-13.jpg

"""""""They Shall Not Grow Old"""". Watching Peter Jackson tinker with WW1 is like watching George Lucas tinker with """"Star Wars"""". Only way more offensive. pic.twitter.com/PkteSrh9tR"""

The International Code Council (ICC) has ratified a change to the 2021 International Building Code (IBC) to allow the use of shipping containers in commercial construction. Photo © www.bigstockphoto.com

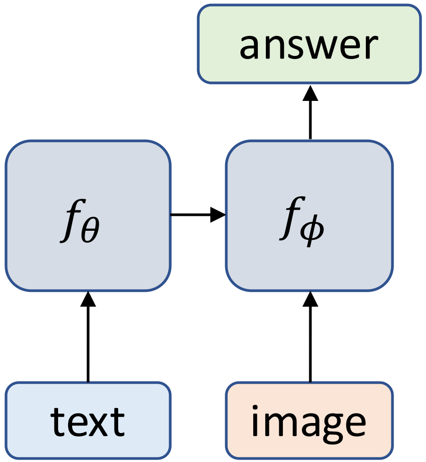
https://laion-aesthetic.datasette.io/laion-aesthetic-6pls/images?_next=300

Composing Vision and Language Models

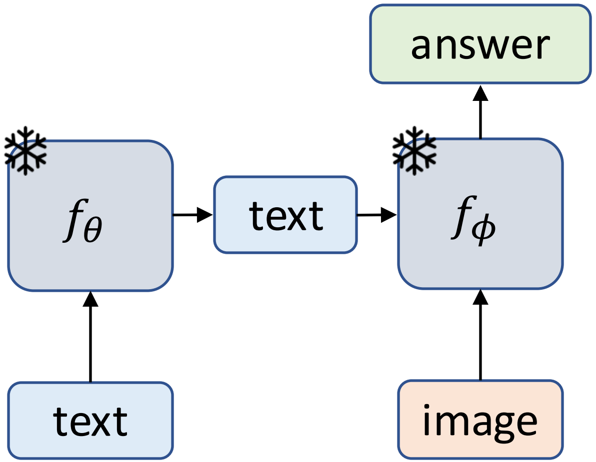
How to compose *trained* L and V models?

How to compose *trained* L and V models?

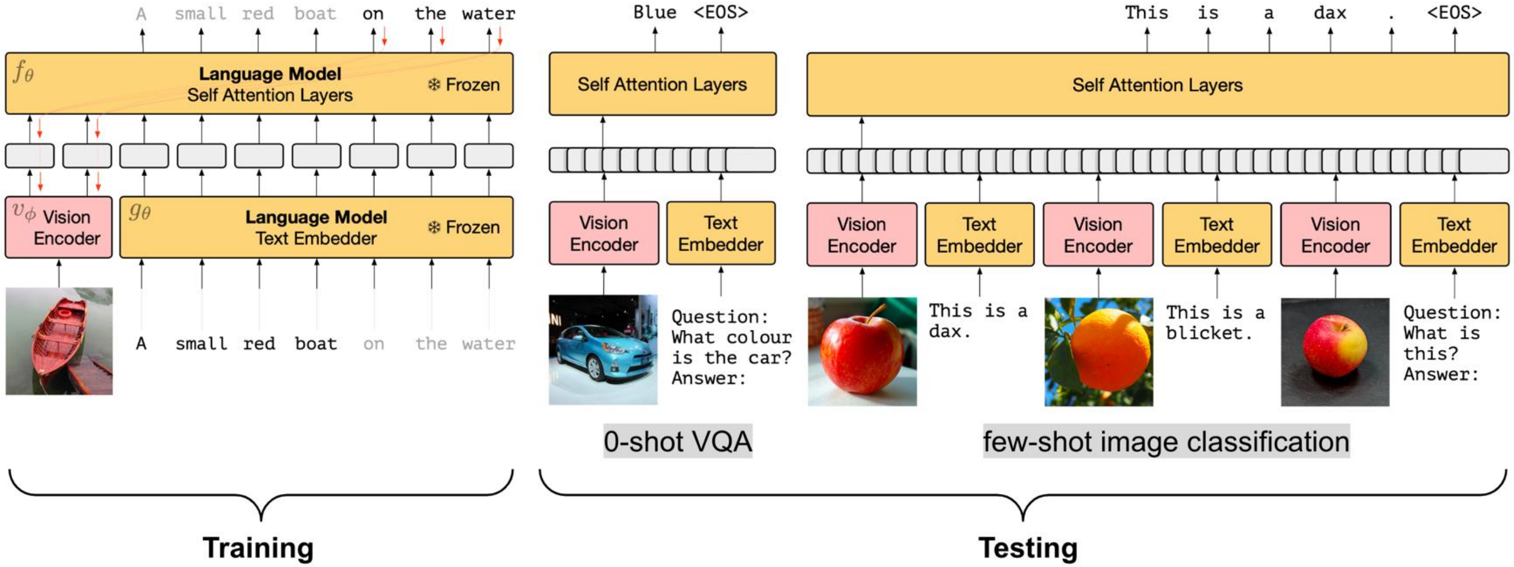
Fast finetuning



Language as interface



Finetuning VLM: Frozen LM, finetune VM



- Train image encoder with frozen language model.
- At test time, can do 0-shot VQA or few-shot classification through in-context learning

Multimodal Few-Shot Learning with Frozen Language Models (Tsimpoukelli et al., 2021)

Finetuning VLM: Frozen LM, finetune VM

- Train image encoder with frozen language model.
- At test time, can do 0-shot VQA or few-shot classification through in-context learning

Multimodal Few-Shot Learning with Frozen Language Models (Tsimpoukelli et al., 2021)

Finetuning VLM: Frozen LM, finetune VM

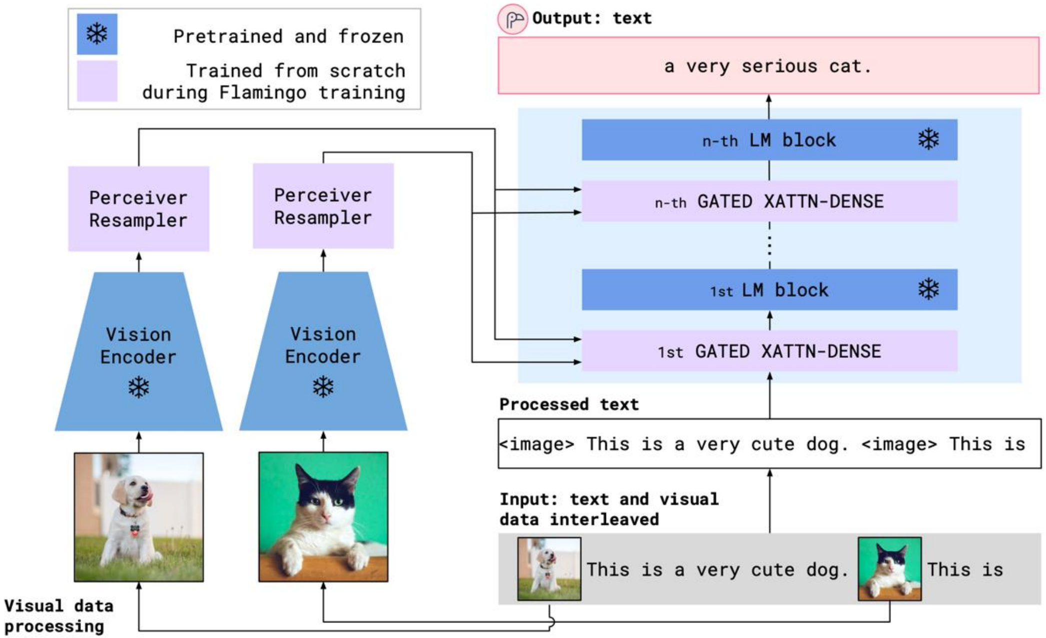
n-shot Acc.	n=0	n=1	n=4	τ
Frozen	29.5	35.7	38.2	✗
Frozen **scratch**	0.0	0.0	0.0	✗
Frozen **finetuned**	24.0	28.2	29.2	✗
Frozen **train-blind**	26.2	33.5	33.3	✗
Frozen **VQA**	48.4	–	–	✓
Frozen **VQA-blind**	39.1	–	–	✓
Oscar [23]	73.8	–	–	✓

n-shot Acc.	n=0	n=1	n=4	τ
Frozen	5.9	9.7	12.6	✗
Frozen **400mLM**	4.0	5.9	6.6	✗
Frozen **finetuned**	4.2	4.1	4.6	✗
Frozen **train-blind**	3.3	7.2	0.0	✗
Frozen **VQA**	19.6	–	–	✗
Frozen **VQA-blind**	12.5	–	–	✗
MAVEx [42]	39.4	–	–	✓

- Training large VLM from scratch does not work at all
- Finetuning LM degrades performance
- “Blind” baselines still works, showing the innate power of LM

Multimodal Few-Shot Learning with Frozen Language Models (Tsimpoukelli et al., 2021)

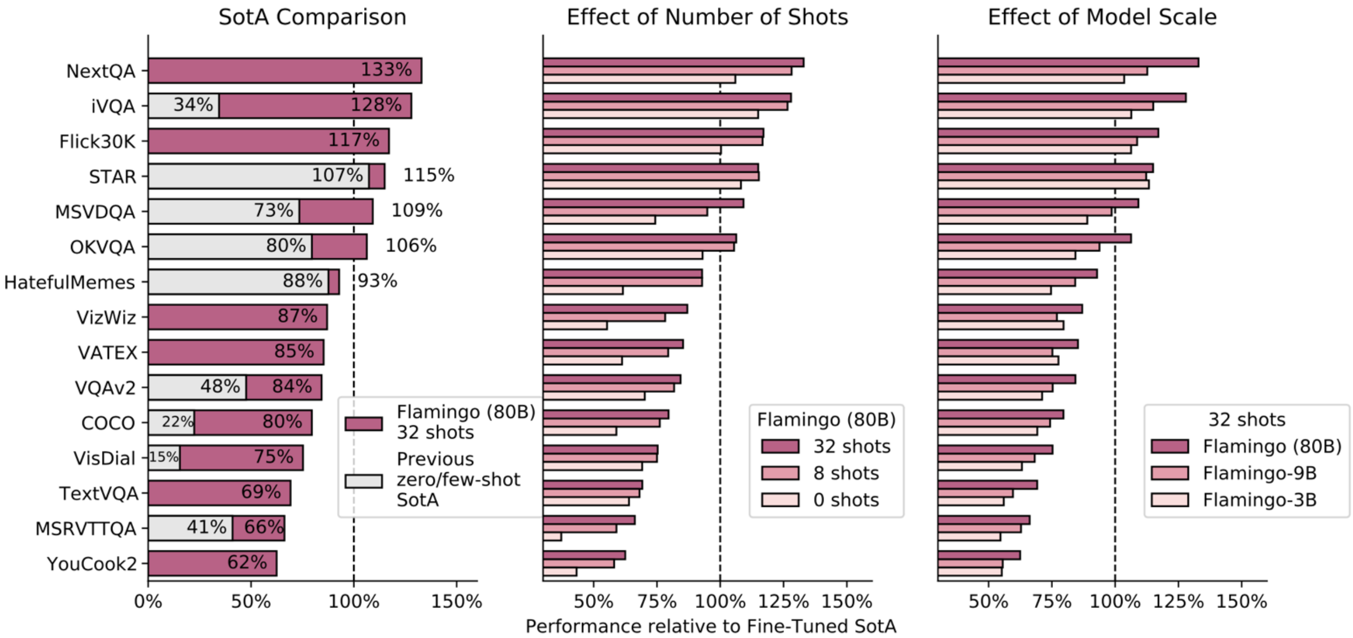
Finetuning VLM: freeze both LM and VM



- Interleaved text-image input
- Only finetune the cross attention (XATTN-DENSE) layers

Flamingo: a Visual Language Model for Few-Shot Learning (Alayrac et al., 2022)

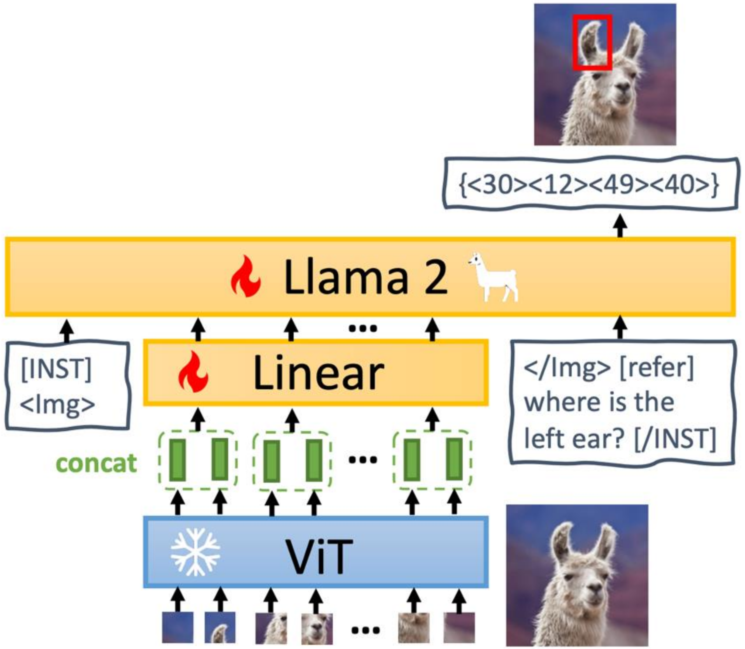
Finetuning VLM: freeze both LM and VM

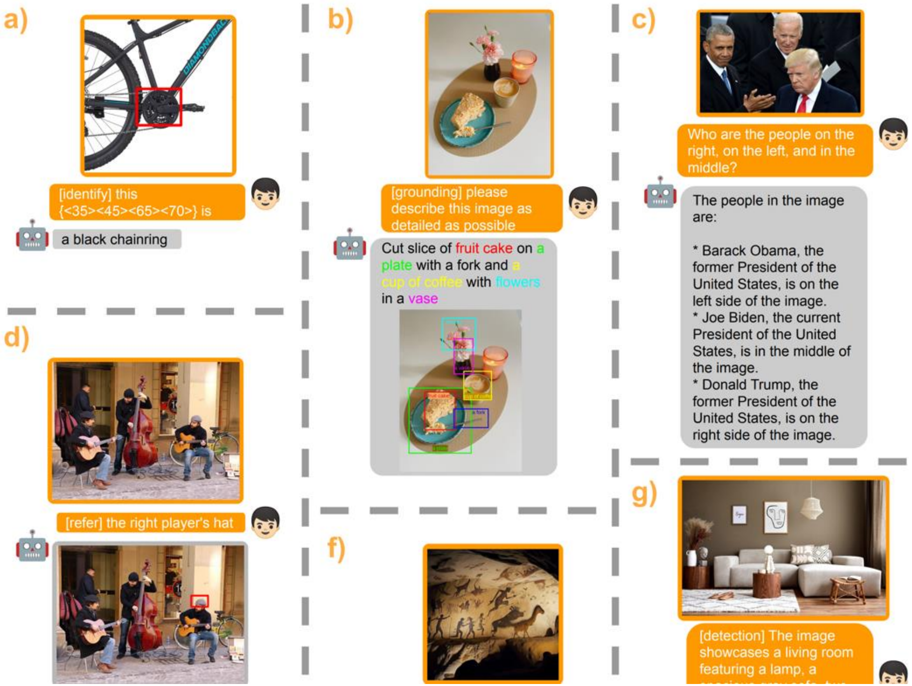


- Largely outperforms previous zero/few shot SotA
- More in-context learning examples do help
- Larger model gives better results

Flamingo: a Visual Language Model for Few-Shot Learning (Alayrac et al., 2022)

Finetuning VLM: freeze both LM and VM





Freeze VM and LM. Train the linear layer and LORA finetune Llama 2

MiniGPT-v2: large language model as a unified interface for vision-language multi-task learning (Chen et al., 2023)

Low-rank finetuning (LORA)

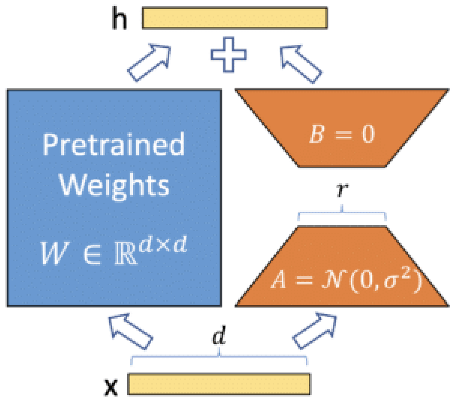
quickly finetune a billion-parameter model

Problem: finetuning still takes a lot of data, especially if the model is huge and/or the domain gap is large.

Fact: finetuning is just adding a W_δ to the existing weight matrix W, i.e., $W^* = W + W_\delta$

Hypothesis: W_δ is *low-rank*, meaning that W_δ can be decomposed into two smaller matrices A and B, i.e., $W_\delta = A^T B$.

So what?: A and B have a lot fewer parameters than the full W. Requires less data and faster to train.



Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models.", 2021

Low-rank finetuning (LORA)

quickly finetune a billion-parameter model

🤗 PEFT

State-of-the-art Parameter-Efficient Fine-Tuning (PEFT) methods

Parameter-Efficient Fine-Tuning (PEFT) methods enable efficient adaptation of pre-trained language models (PLMs) to various downstream applications without fine-tuning all the model's parameters. Fine-tuning large-scale PLMs is often prohibitively costly. In this regard, PEFT methods only fine-tune a small number of (extra) model parameters, thereby greatly decreasing the computational and storage costs. Recent State-of-the-Art PEFT techniques achieve performance comparable to that of full fine-tuning.

Seamlessly integrated with 🤗 Accelerate for large scale models leveraging DeepSpeed and Big Model Inference.

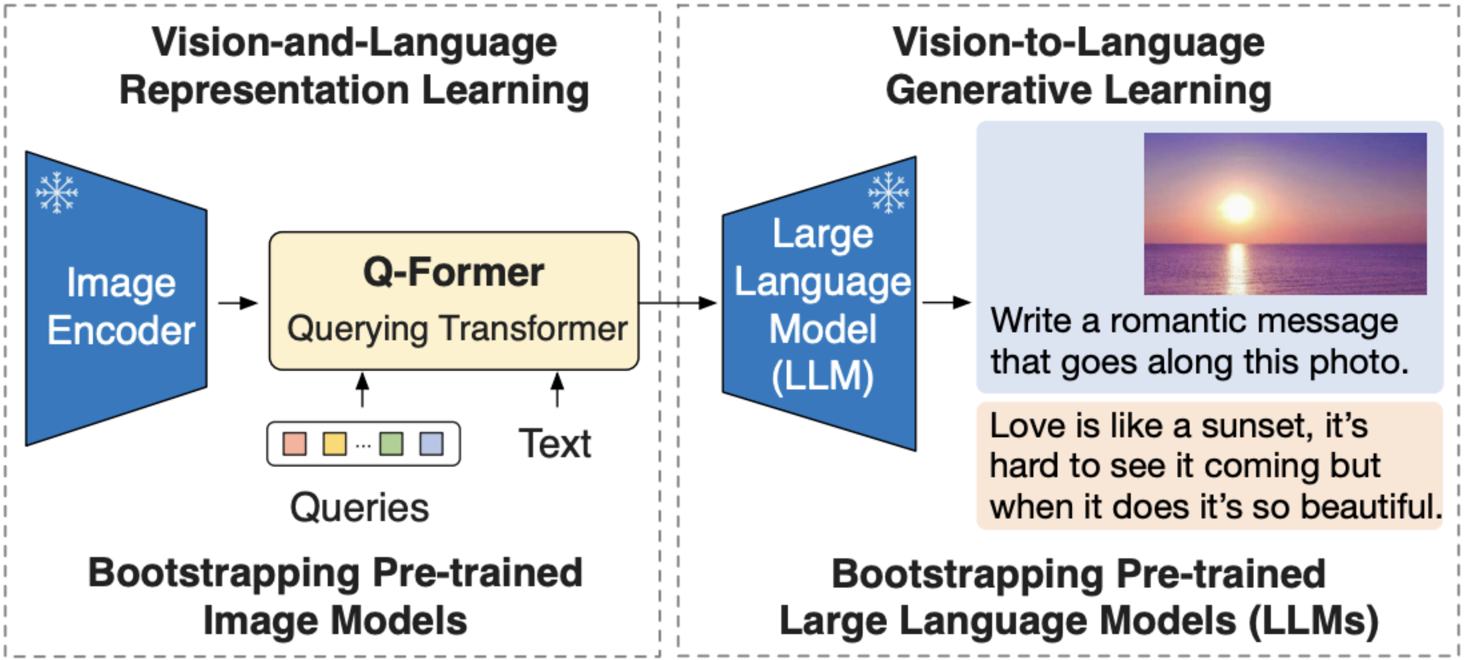
Supported methods:

1. LoRA: LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS
2. Prefix Tuning: Prefix-Tuning: Optimizing Continuous Prompts for Generation, P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks
3. P-Tuning: GPT Understands, Too
4. Prompt Tuning: The Power of Scale for Parameter-Efficient Prompt Tuning
5. AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning
6. $(IA)^3$: Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning
7. MultiTask Prompt Tuning: Multitask Prompt Tuning Enables Parameter-Efficient Transfer Learning
8. LoHa: FedPara: Low-Rank Hadamard Product for Communication-Efficient Federated Learning
9. LoKr: KronA: Parameter Efficient Tuning with Kronecker Adapter based on Navigating Text-To-Image Customization:From LyCORIS Fine-Tuning to Model Evaluation implementation

```python
import torch
from peft import inject_adapter_in_model, LoraConfig

class DummyModel(torch.nn.Module):
    def __init__(self):
        super().__init__()
        self.embedding = torch.nn.Embedding(10, 10)
        self.linear = torch.nn.Linear(10, 10)
        self.lm_head = torch.nn.Linear(10, 10)

    def forward(self, input_ids):
        x = self.embedding(input_ids)
        x = self.linear(x)
        x = self.lm_head(x)
        return x

lora_config = LoraConfig(
    lora_alpha=16,
    lora_dropout=0.1,
    r=64,
    bias="none",
    target_modules=["linear"],
)

model = DummyModel()
model = inject_adapter_in_model(lora_config, model)

dummy_inputs = torch.LongTensor([[0, 1, 2, 3, 4, 5, 6, 7]])
dummy_outputs = model(dummy_inputs)
```

https://github.com/huggingface/peft

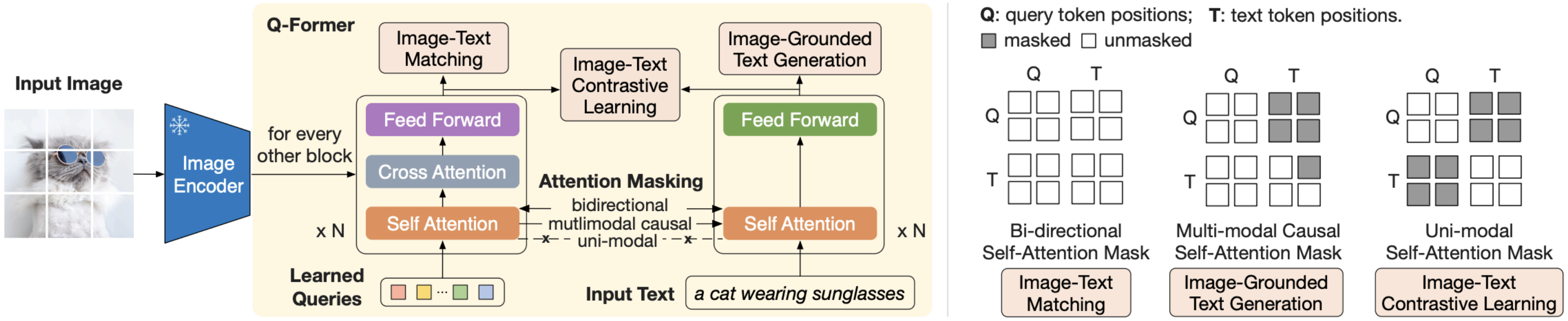
Q-Former: Pretraining to Align Vision to Text



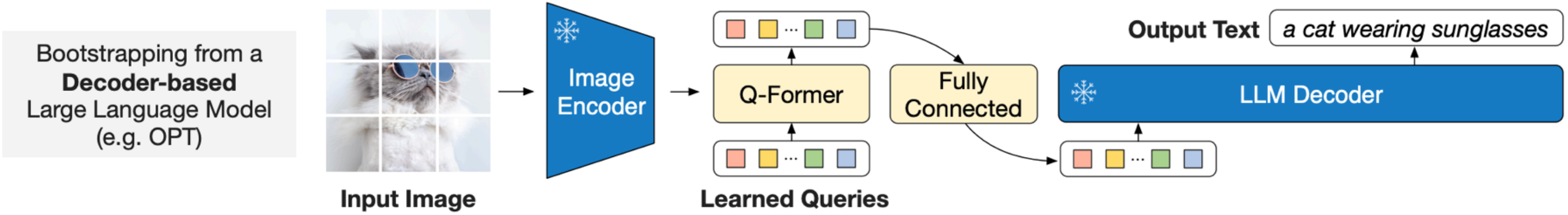
BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models (Li et al., 2023)

Q-Former: Pretraining to Align Vision to Text

1. Extract text-relevant image feature through pretraining:

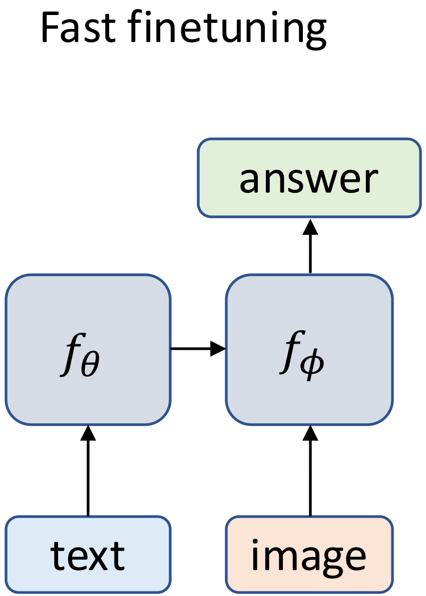


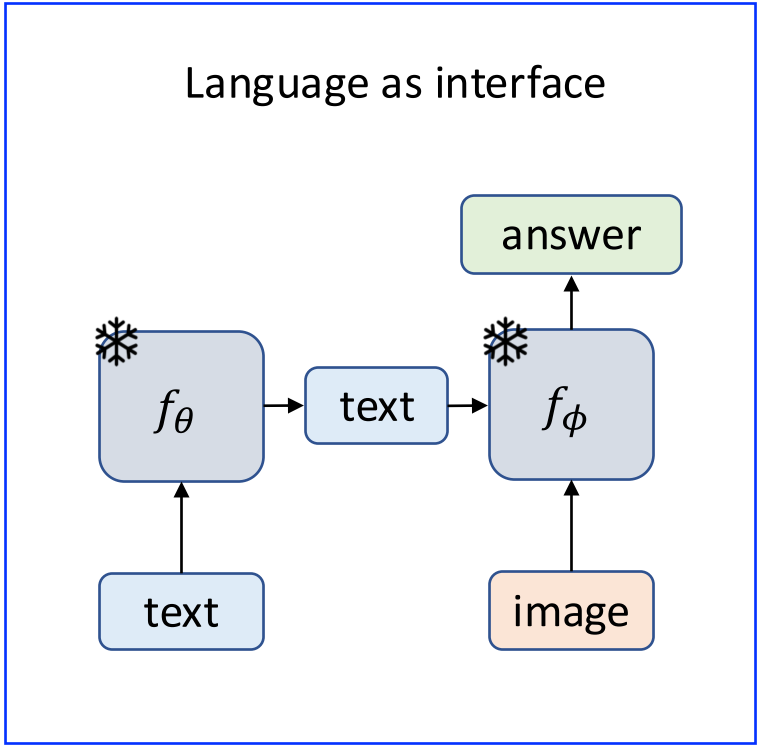
2. Generative finetuning of Q-Former



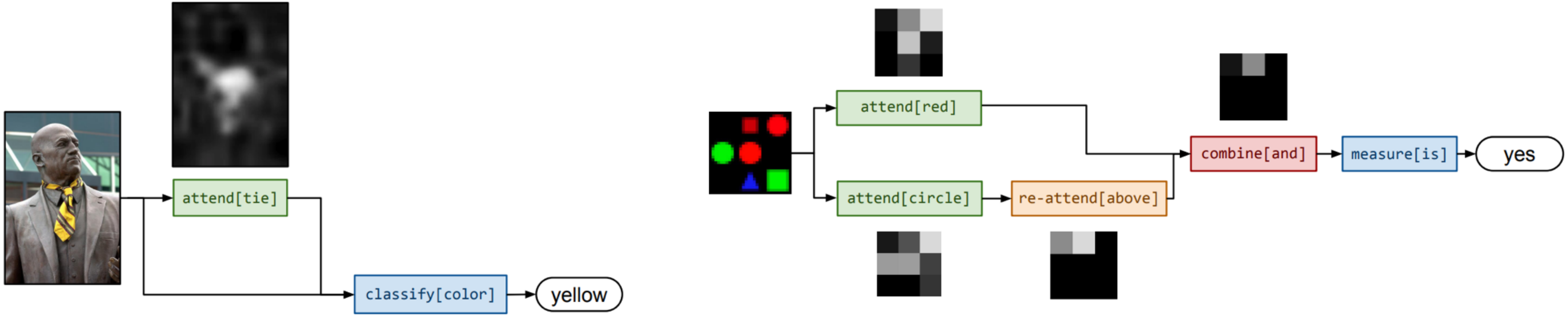
BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models (Li et al., 2023)

How to compose *trained* L and V models?



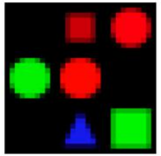


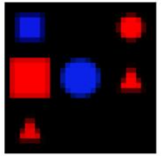
Neural Module Networks (Andreas et al., 2015)



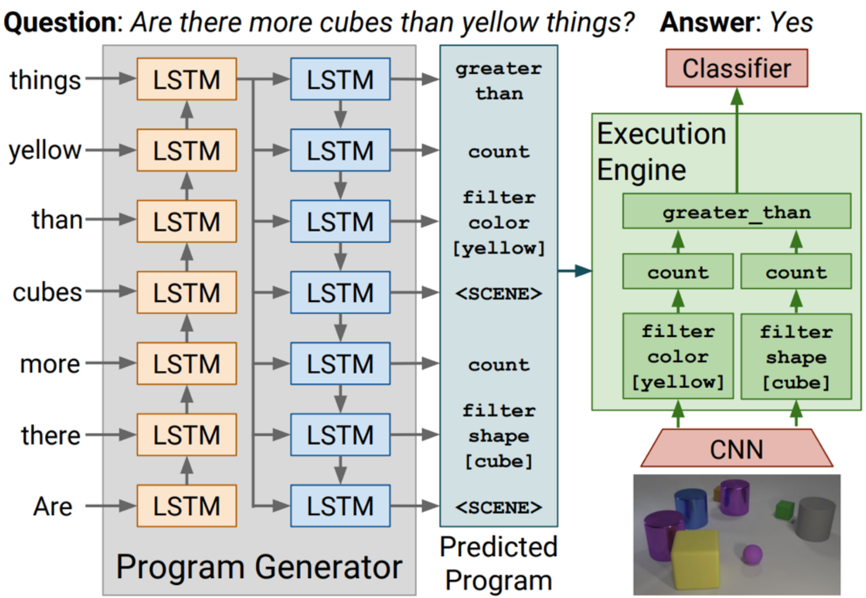
Idea: train modular networks (attend, classify). Use a controller network to decide how to compose the modules together to solve a task

Neural Module Networks (Andreas et al., 2015)

				
how many different lights in various different shapes and sizes?	*what is the color of the horse?*	*what color is the vase?*	*is the bus full of passengers?*	*is there a red shape above a circle?*
`measure[count](attend[light])`	`classify[color](attend[horse])`	`classify[color](attend[vase])`	`measure[is](combine[and](attend[bus], attend[full])`	`measure[is](combine[and](attend[red], re-attend[above](attend[circle])))`
four (four)	brown (brown)	green (green)	yes (yes)	no (no)

				
what is stuffed with toothbrushes wrapped in plastic?	*where does the tabby cat watch a horse eating hay?*	*what material are the boxes made of?*	*is this a clock?*	*is a red shape blue?*
`classify[what](attend[stuff])`	`classify[where](attend[watch])`	`classify[material](attend[box])`	`measure[is](attend[clock])`	`measure[is](combine[and](attend[red], attend[blue]))`
container (cup)	pen (barn)	leather (cardboard)	yes (no)	yes (no)

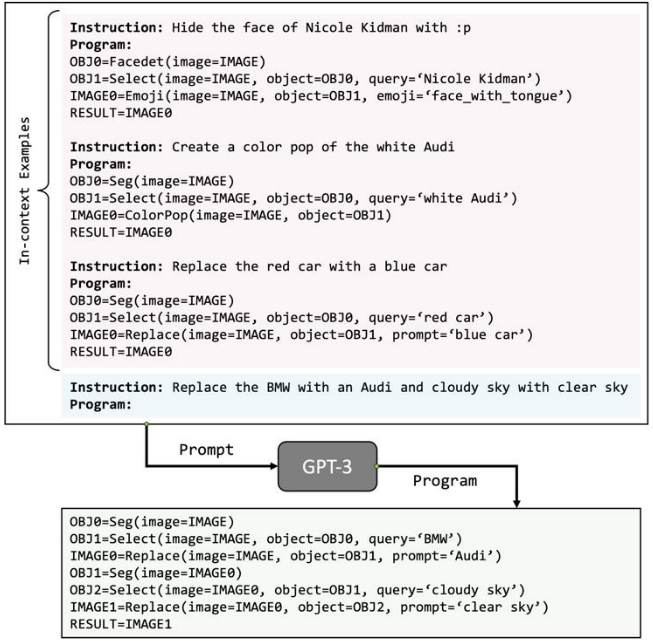
Inferring and Executing Programs for Visual Reasoning (Johnson et al., 2017)

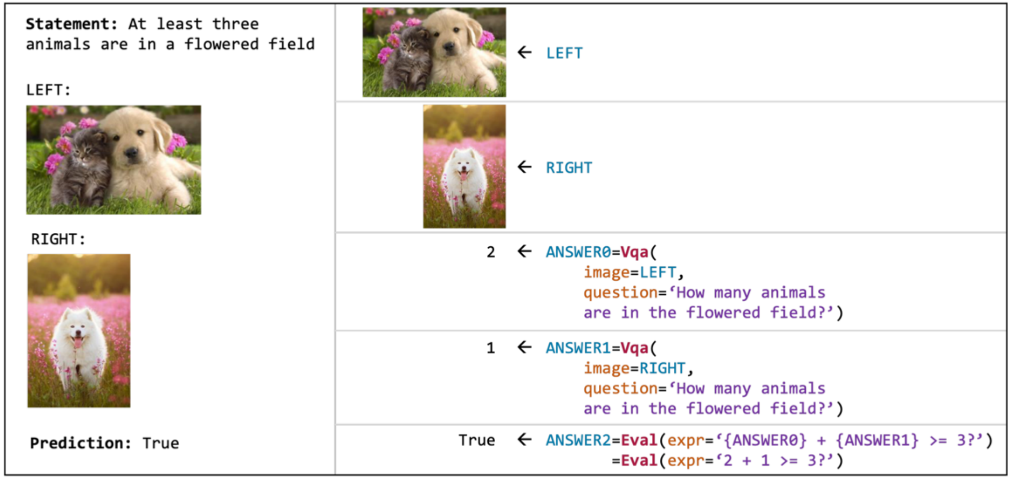


Similar to NMN, but train a *program generator* using REINFORCE

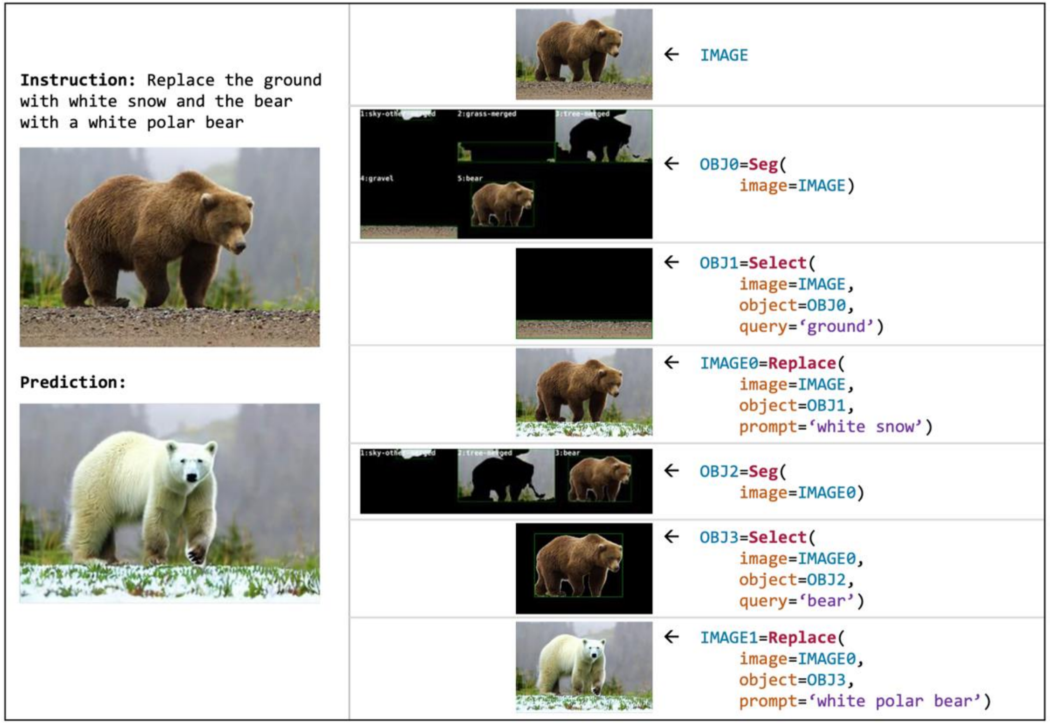
Reward comes from whether the answer is correct

Visual Programming: Compositional visual reasoning without training (Gupta et al., 2023)

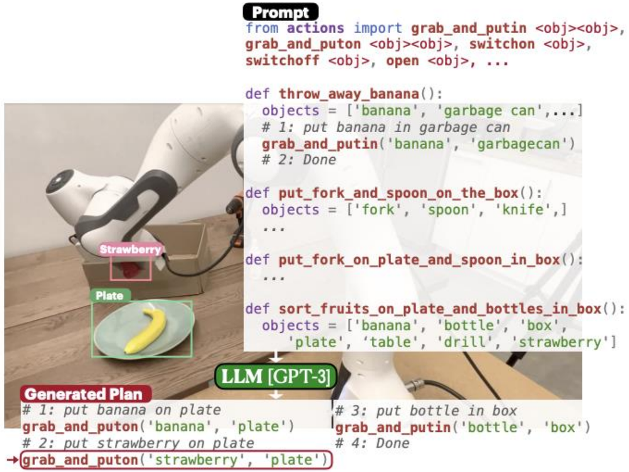


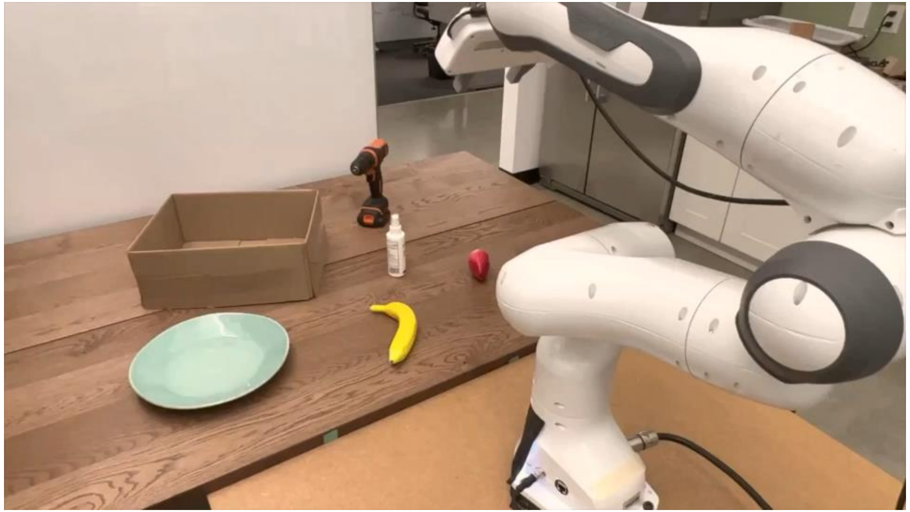


Visual Programming: Compositional visual reasoning without training (Gupta et al., 2023)



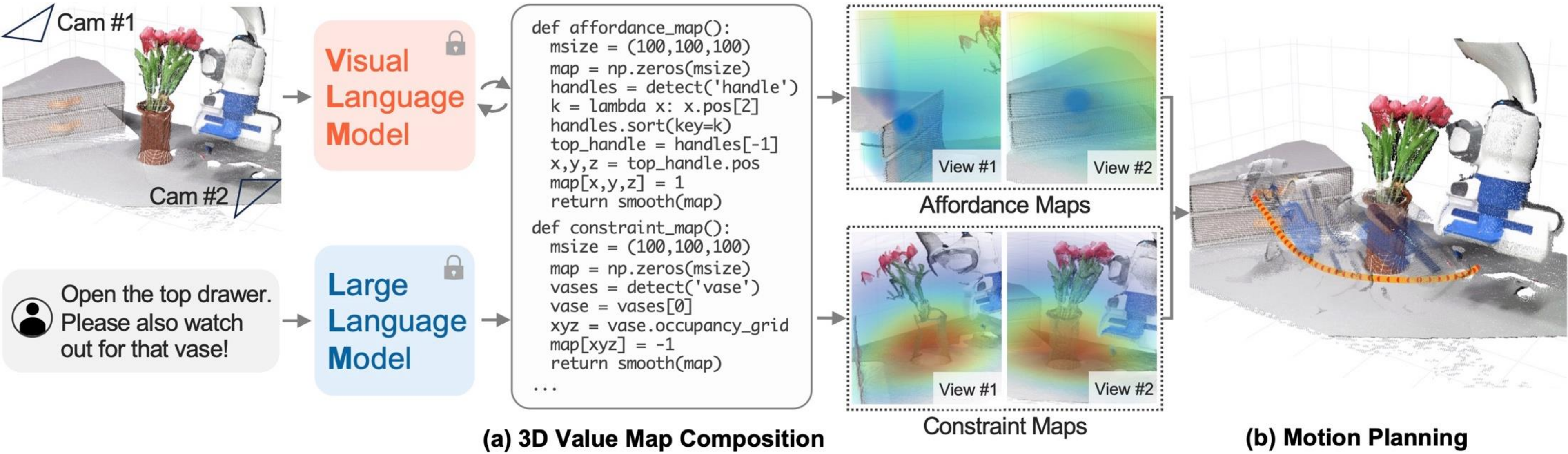
ProgPrompt (Singh et al., 2023): Program to Actions





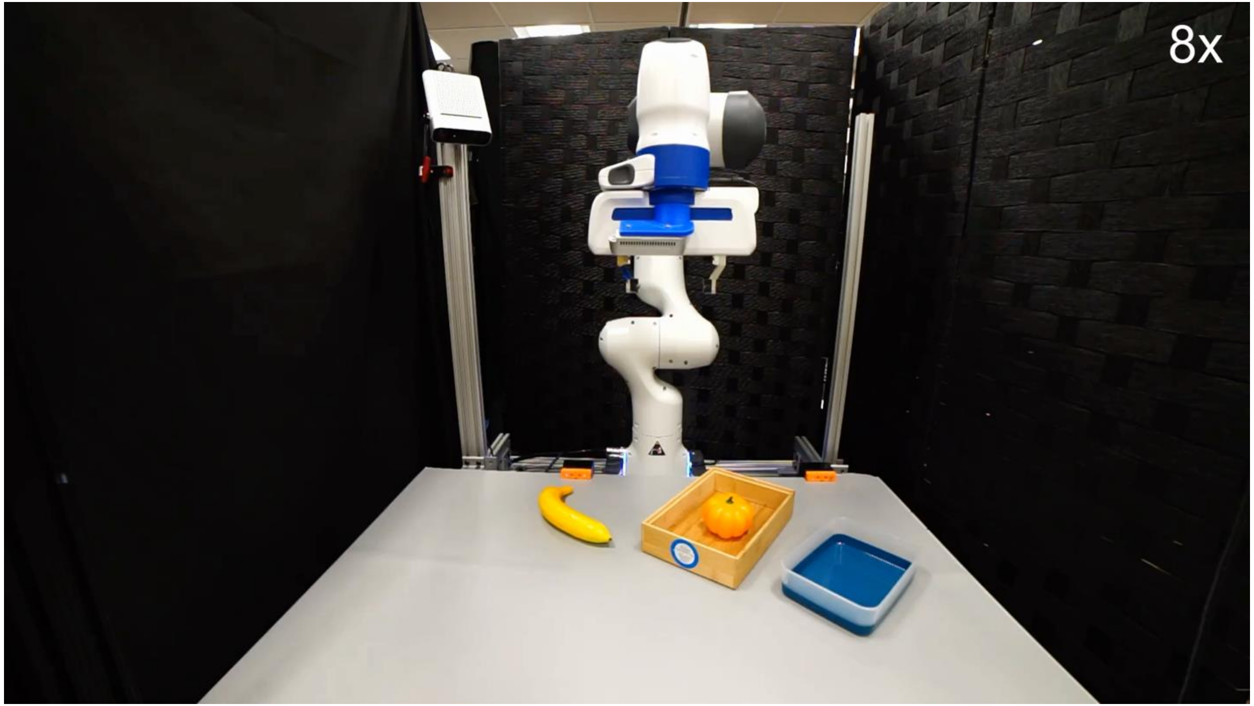
Use large language models (LLMs) to generate program-like semantic plans from natural language command.

VoxPoser (Huang et al., 2023): Program to Grounded Actions



Use LLMs to guide VMs to find where to act next in a 3D scene

VoxPoser (Huang et al., 2023): Program to Grounded Actions

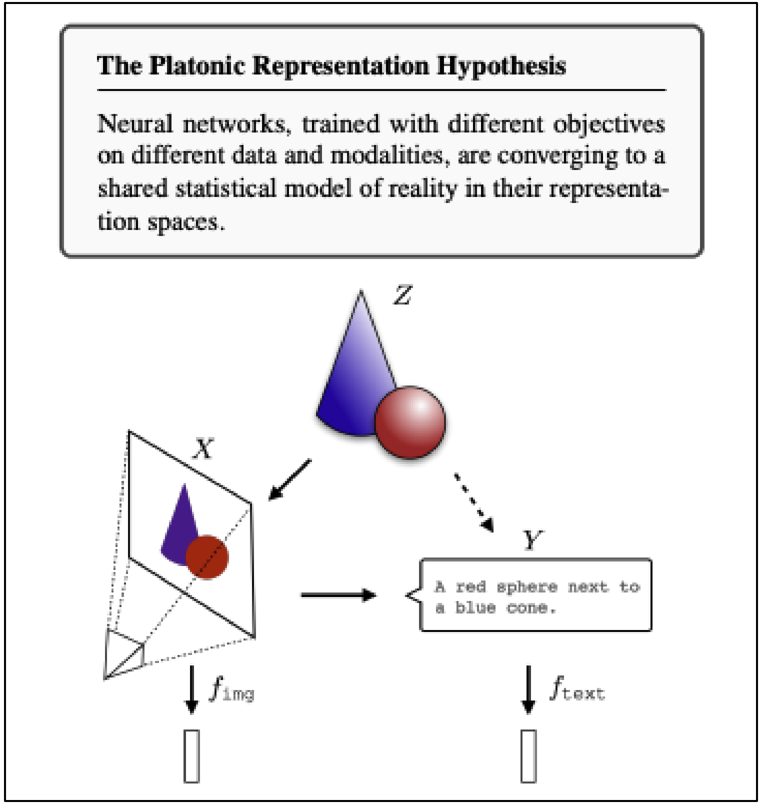


"Sort the paper trash into the blue tray."

Where do we go from here?

The Platonic Representation Hypothesis (2024)

Hypothesis: When trained at large scale, representations learned from different objectives / modalities are converging to the same statistical model that reflects the underlying reality of the world Z



https://arxiv.org/pdf/2405.07987

The Platonic Representation Hypothesis (2024)

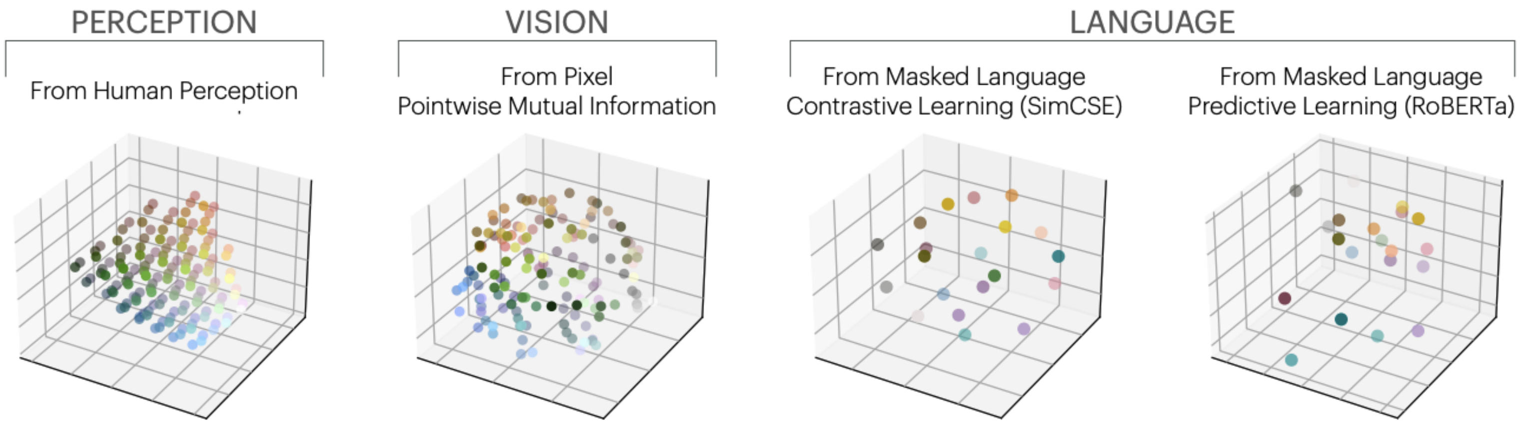
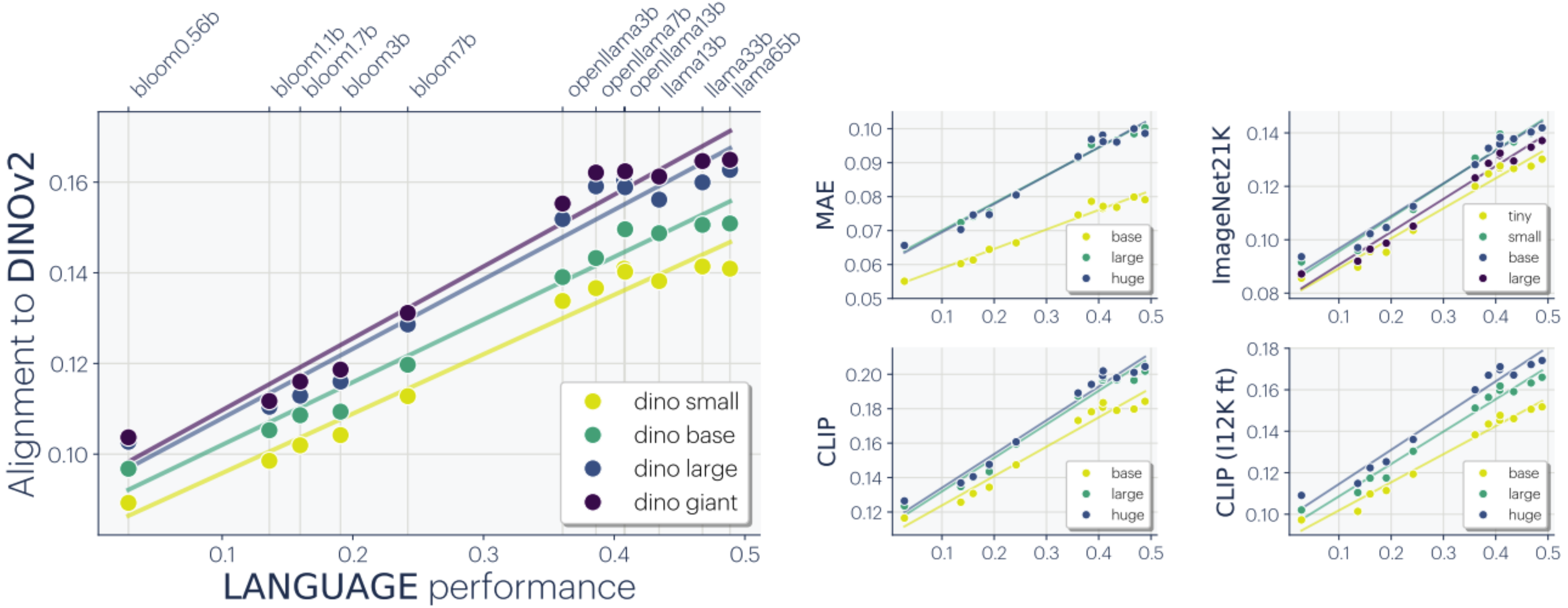


Figure 8. **Color cooccurrence in VISION and LANGUAGE yields perceptual organization:** Similar representations of color are obtained via, **from LEFT to RIGHT**, the perceptual layout from CIELAB color space, cooccurrence in CIFAR-10 images, and language cooccurrence modeling (Gao et al. (2021); Liu et al. (2019); computed roughly following Abdou et al. (2021)). Details in Appendix D.

"… color distances in learned language representations, when trained to predict cooccurrences in text, closely mirror human perception of these distances."

The Platonic Representation Hypothesis (2024)



Stronger LLMs tend to align better with vision model in representation space
(measured in mutual nearest neighbor)

https://arxiv.org/pdf/2405.07987

The Platonic Representation Hypothesis (2024)

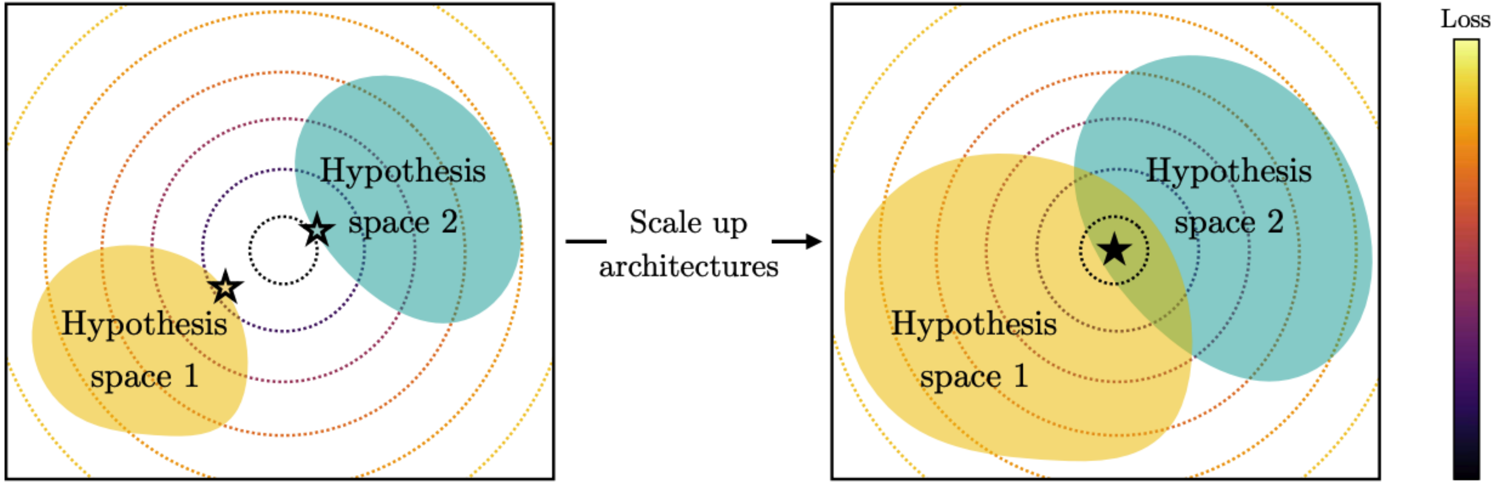


Figure 5. **The Capacity Hypothesis:** If an optimal representation exists in function space, larger hypothesis spaces are more likely to cover it. **LEFT:** Two small models might not cover the optimum and thus find *different* solutions (marked by outlined ☆). **RIGHT:** As the models become larger, they cover the optimum and converge to the same solution (marked by filled ★).

https://arxiv.org/pdf/2405.07987

The Platonic Representation Hypothesis (2024)

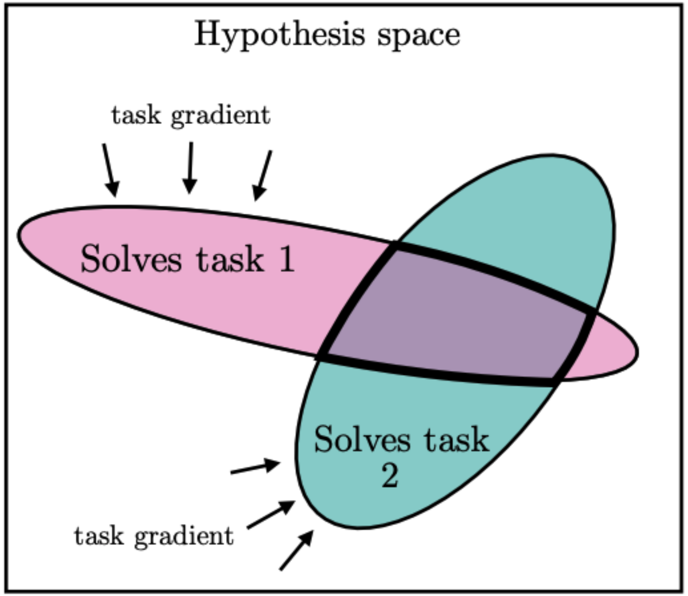


Figure 6. **The Multitask Scaling Hypothesis:** Models trained with an increasing number of tasks are subjected to pressure to learn a representation that can solve all the tasks.

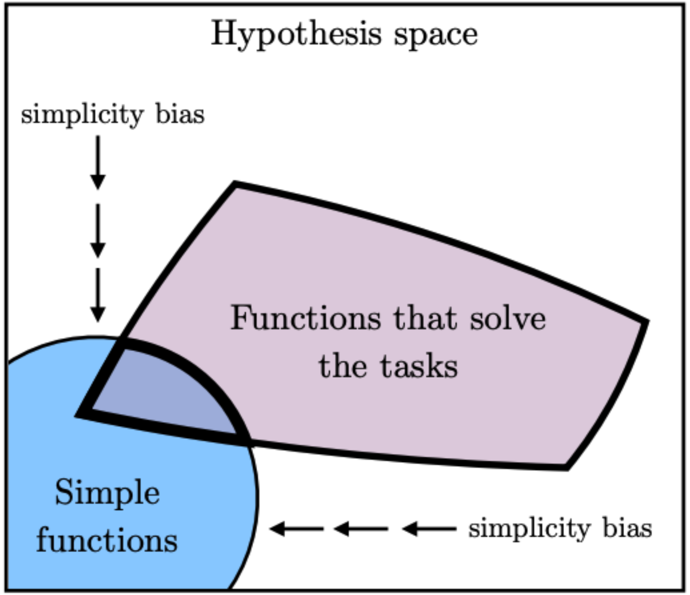


Figure 7. **The Simplicity Bias Hypothesis:** Larger models have larger coverage of all possible ways to fit the same data. However, the implicit simplicity biases of deep networks encourage larger models to find the simplest of these solutions.

https://arxiv.org/pdf/2405.07987

Summary: Large Vision and Language Models

- Very active field of research, with a history as long as modern deep learning (2011 -)
- Foundation vision and language models have revolutionized the research paradigm post 2019.
- Trending towards larger model and dataset.
- Many active research on how to finetune / adapt VLMs with small amount of compute / data.
- The future is going to be multimodal.
- The representations are converging.