

Vision-Language Models (VLMs)

Donald Shenaj

donald.shenaj@di.unipi.it

November 11, 2026

Lecture Outline

1. Aligning Vision and Language
2. Translating Images into Text
3. Text as a Visual Representation and VQA
4. Instruction-Tuned VLMs
5. Other VLM Tasks: Reasoning and Grounding

Recap

- Previous lectures: image classification, detection, and dense prediction using CNNs, ViTs, and attention mechanisms.
- **Self-supervised learning**: SimCLR learns representations through contrastive learning (InfoNCE/NT-Xent), using in-batch negatives and temperature scaling.
- Positive pairs in SimCLR consist of two augmented views of the same image.

From Patch Tokens to Vision-Language Representations

- A vision encoder produces a sequence of visual features:

$$Z_v = f_v(x) \in \mathbb{R}^{N \times D_v}$$

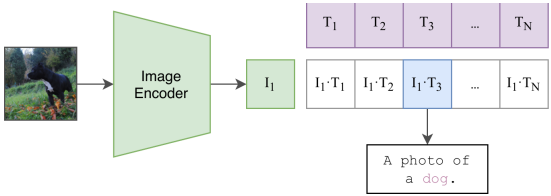
- Vision-language architectures use these representations differently:
  - **Global embedding**: one vector per image, obtained by pooling or selecting a global token and projecting it into a shared space (CLIP).
  - **Visual token sequence**: individual image features are retained for query-based cross-attention (BLIP-2) or projection into the LLM input sequence (LLaVA).

Vision-Language Supervision

- Image-only self-supervised learning uses visual augmentations to define positive pairs.
- Web-scale image-text pairs provide semantic supervision through captions and alt-text.
- Vision-language alignment enables transfer to new text-defined categories, without training a classifier for each target label set.

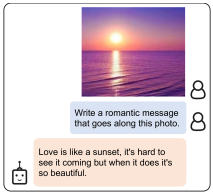
Vision-Language Tasks

Zero-shot classification



no classifier head ever trained for "dog"

Captioning



describe an image in free-form text

Visual question answering



open-ended answers

Instruction-following chat



"What's unusual about this image?"

Aligning Vision and Language

CLIP: alignment, data, evaluation, and limitations

Representing Text for a Neural Network

- **Tokenization**: text is split into discrete tokens, each assigned an integer ID.
- **Byte-Pair Encoding (BPE)**: iteratively merges frequent symbol pairs, balancing vocabulary size and sequence length. Used by CLIP's text encoder.
- **Token embeddings**: each token ID is mapped to a learned d-dimensional vector through an embedding matrix $E \in \mathbb{R}^{V \times d}$, where V is the vocabulary size.

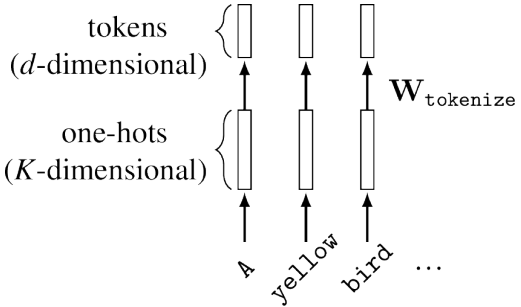


Figure: [*visionbook*]

Precursors to CLIP

- Early methods learned visual representations from image-associated text [*Mori et al., 1999; Ordonez et al., 2011; Joulin et al., 2016; Li et al., 2017*]. Visual n-grams reached 11.5% zero-shot ImageNet accuracy.
- More recent approaches used different objectives: **VirTex** (autoregressive captioning), **ICMLM** (masked language modeling), and **ConVIRT** (contrastive learning).
- **ConVIRT**: bidirectional image-text contrastive learning applied to medical images and reports.

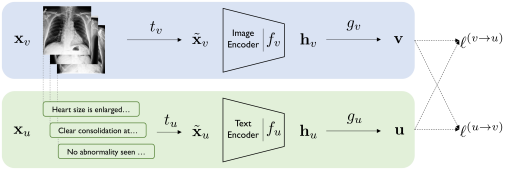


Figure: [*Zhang et al., 2020: ConVIRT*]

$$\mathcal{L}_{\text{ConVIRT}} = \lambda\, \mathcal{L}_{v\to u} + (1-\lambda)\, \mathcal{L}_{u\to v}$$

$$\mathcal{L}_{v\to u} = -\log \frac{\exp(\text{sim}(v,u)/\tau)}{\sum_{u'} \exp(\text{sim}(v,u')/\tau)}$$

CLIP adopts a similar symmetric contrastive objective, trained from scratch on 400M image-text pairs instead of approximately 200k.

CLIP: Contrastive Language-Image Pre-training

- **CLIP** jointly trains image and text encoders using a contrastive objective: matched image-text pairs are aligned, while mismatched pairs are separated.
- **Zero-shot classification**: class names are embedded using text prompts (e.g., "A photo of a {object}."). Images are classified by similarity to these embeddings, without training a task-specific classifier.

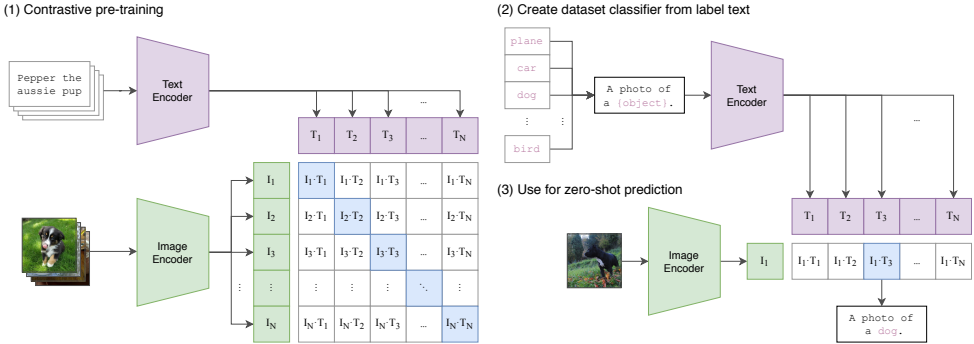


Figure: [*Radford et al., 2021: CLIP*]

CLIP: A Single Image-Text Pair

- **Key idea**: maximize the similarity between true image-text embedding pairs and minimize similarity between mismatched image-text embedding pairs

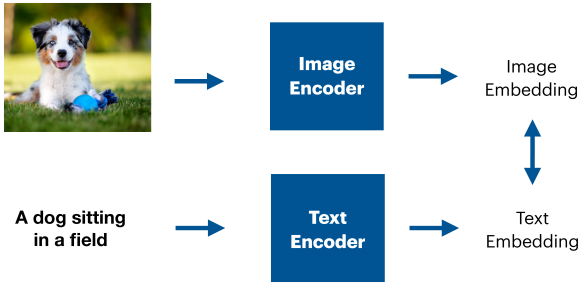


Figure: [*Varma, Stanford CS277/BIODS271*]

CLIP: Choosing an Encoder Architecture

- **Image encoder** f_v: modified ResNet (attention pooling instead of global average pooling) or Vision Transformer (ViT).
- **Text encoder** f_t: decoder-style Transformer with causal self-attention over BPE tokens. The base model has 12 layers, width 512, and 63M parameters. The final `[EOS]` representation is used as the text feature.
- **Projection**: both encoder outputs are mapped linearly into a shared d_z-dimensional space, without the nonlinear projection head used in SimCLR.

CLIP: Normalizing Into a Shared Space

The encoder outputs are linearly projected and **L2-normalized**:

$$v_i = \frac{W_v h_i^v}{\|W_v h_i^v\|_2}, \qquad t_i = \frac{W_t h_i^t}{\|W_t h_i^t\|_2}$$

- $h_i^v \in \mathbb{R}^{D_v}$ and $h_i^t \in \mathbb{R}^{D_t}$ are the encoder features. The projections W_v and W_t map them into a shared d_z-dimensional space.
- Normalization gives $\|v_i\|_2 = \|t_i\|_2 = 1$, so the dot product equals cosine similarity: $v_i^\top t_j = \cos\theta_{ij}$.

CLIP: A Batch of N Pairs

- CLIP trains on batches of N image-caption pairs. Each image is compared with all N captions, and each caption with all N images.

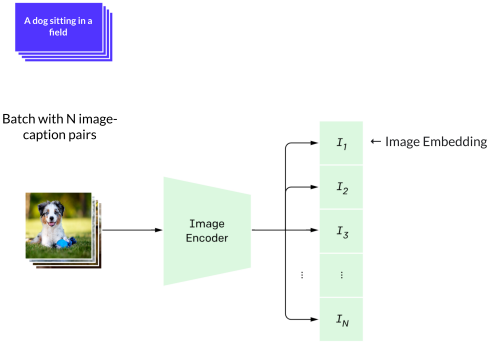


Batch with N image-caption pairs



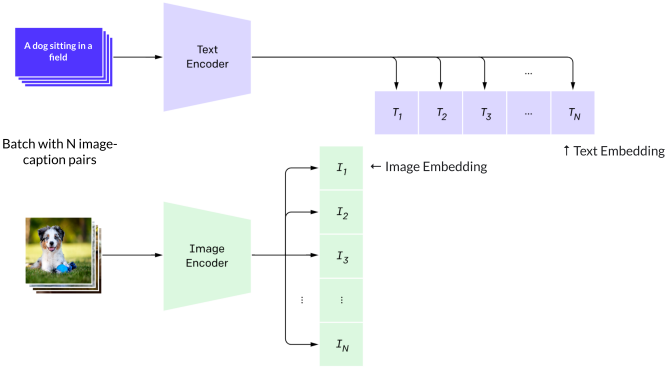
CLIP: Image Embeddings for the Batch

- Each of the N images goes through the *same* image encoder f_ℓ independently, producing $I_1, \ldots, I_N$; batching only changes how many vectors come out, not how any single one is computed



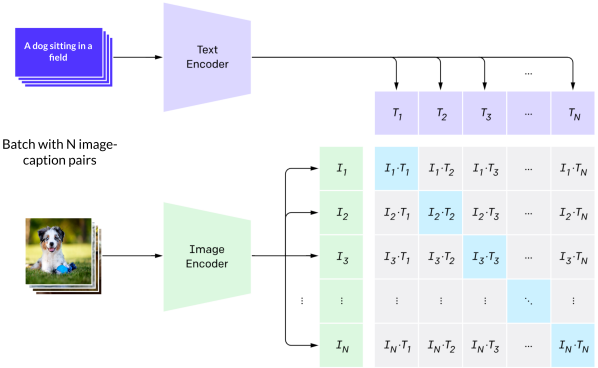
CLIP: Text Embeddings for the Batch

- The N captions are encoded by f_t and projected into the shared embedding space, producing $T_1, \ldots, T_N$.



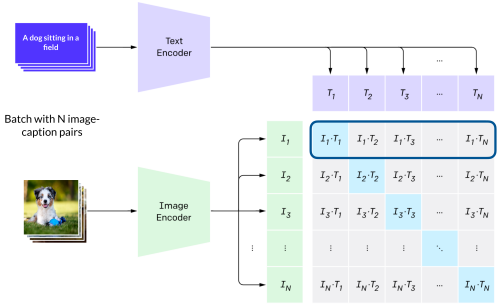
CLIP: All Pairwise Similarities

- Compute all pairwise similarities $S_{ij} = I_i^\top T_j$, forming an $N \times N$ matrix. Matched pairs lie on the diagonal.



CLIP: Image-to-Text Objective

- Apply softmax across row k: image I_k is compared with all N captions.

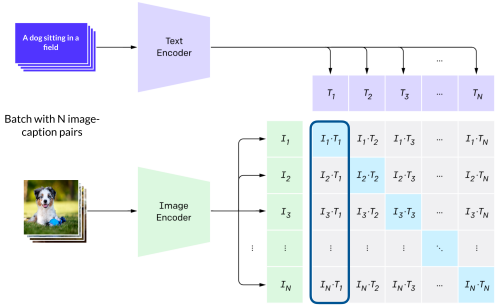


$$\mathcal{L}_{I \to T} = \sum_{k=1}^{N} -\log \frac{\exp(I_k \cdot T_k / \tau)}{\sum_{j=1}^{N} \exp(I_k \cdot T_j / \tau)}$$

- (I_k, T_k) is the positive pair. The other $N-1$ captions are in-batch negatives. The loss is InfoNCE with temperature τ.

CLIP: The Symmetric Objective

- CLIP also applies softmax down each column: caption T_k is compared with all N images.



$$\mathcal{L}_{I \to T} = \sum_{k=1}^{N} -\log \frac{\exp(I_k \cdot T_k / \tau)}{\sum_{j=1}^{N} \exp(I_k \cdot T_j / \tau)}$$

$$\mathcal{L}_{T \to I} = \sum_{k=1}^{N} -\log \frac{\exp(I_k \cdot T_k / \tau)}{\sum_{j=1}^{N} \exp(I_j \cdot T_k / \tau)}$$

$$\mathcal{L} = \mathcal{L}_{I \to T} + \mathcal{L}_{T \to I}$$

- Both directions are summed over the batch. The implementation averages over samples and directions.

CLIP: Training Pseudocode

```
# 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)

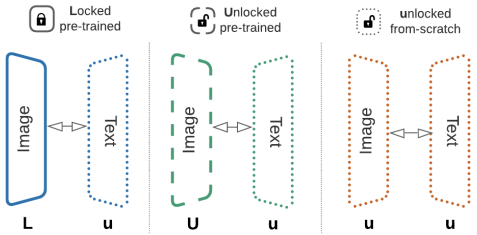
# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# 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
```

Figure: [*Radford et al., 2021: CLIP*]

LiT: Locked-image Text Tuning

- CLIP trains both encoders jointly from scratch. LiT freezes a pretrained image encoder and trains a text encoder using the CLIP contrastive objective.



	ImgNet	IN-v2	C100	Pets
Lu	70.1	61.7	70.9	88.1
Uu	57.2	50.2	62.1	74.8
uu	50.6	43.3	47.9	70.3

Lu = locked pretrained image + unlocked text (LiT)

Figure: [*Zhai et al., 2022: LiT*]

$$v = f_v(x), \qquad t_\theta = f_{t,\theta}(c), \qquad \min_\theta \mathcal{L}_{\text{contrastive}}(v, t_\theta)$$

- Result: Locking the image encoder (Lu) outperforms joint fine-tuning (Uu) and training from scratch (uu) in the reported zero-shot experiments.

CLIP: Representations at Different Depths

- t-SNE visualization of intermediate image and text encoder representations.
- Image features initially cluster by color, then by shape; text features show the opposite progression.
- The contrastive loss acts on the final projected embeddings. Intermediate layers are trained through backpropagation.

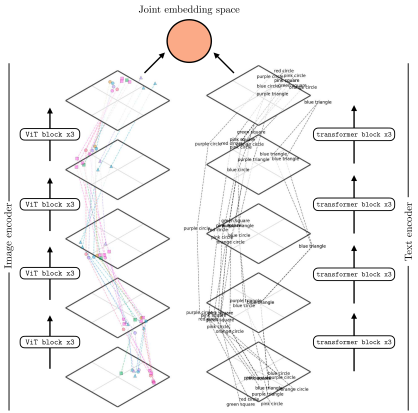


Figure: [*visionbook*]

CLIP: The Joint Embedding Space

- L2-normalized image and text embeddings lie on a unit hypersphere, visualized here using kernel PCA.
- Matched image-text pairs have similar embeddings in the shared space.
- Contrastive learning increases $\cos(\theta_{ii})$ relative to $\cos(\theta_{ij})$, $i \neq j$, without imposing an absolute similarity target.

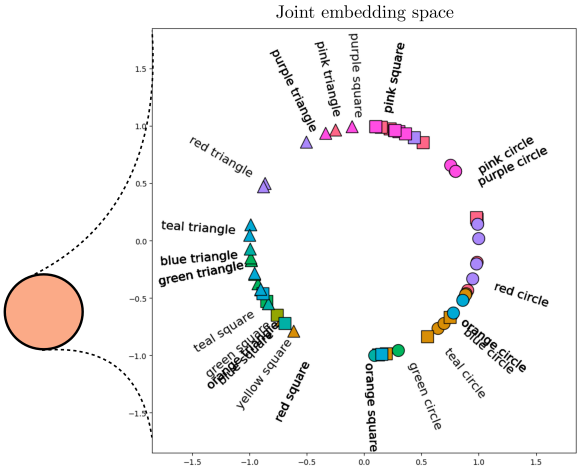


Figure: [*visionbook*]

Training Data for Vision-Language Models

- CLIP was trained on 400M web-collected image-text pairs. Dataset quality, filtering, and diversity are important factors in large-scale vision-language training.

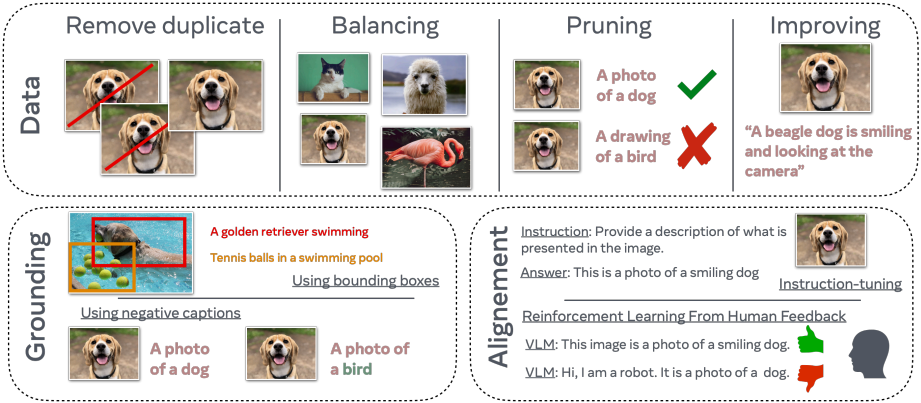


Figure: [*Bordes et al., 2024: An Introduction to Vision-Language Modeling*]

Data at Scale: LAION-5B

- **LAION-5B**: 5.85B image-text pairs from Common Crawl (2.32B English, 2.26B multilingual, 1.27B undetermined language). Public metadata and image URLs support CLIP-style training with **OpenCLIP**.



Figure: [*Schuhmann et al., 2022: LAION-5B*]

- Web alt-text includes product titles, filenames, and incomplete descriptions. CLIP-based filtering improves image-text alignment but does not eliminate dataset noise.

OpenCLIP: Reproducing and Scaling CLIP

- **OpenCLIP**: open-source implementation of CLIP-style dual encoders and contrastive training.
- Uses public datasets (LAION-80M/400M/2B) and releases pretrained models across different architectures and scales.
- Scaling experiments show different performance trends for OpenAI CLIP and OpenCLIP, despite similar architectures and training procedures.

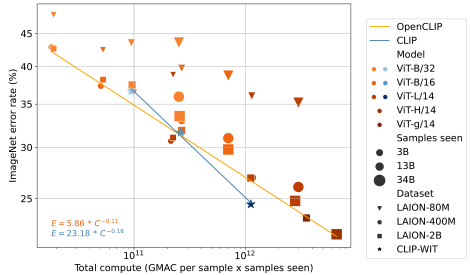


Figure: [*Cherti et al., 2023: Reproducible Scaling Laws for Contrastive Language-Image Learning*]

CLIP: Zero-Shot Classification

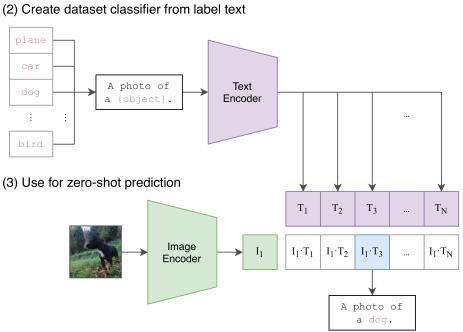


Figure: [*Radford et al., 2021: CLIP*]

- Candidate class names $c_1, \ldots, c_K$ are encoded as text embeddings, defining a classifier without task-specific training.

$$p(y = k \mid x, \mathcal{C}) = \frac{\exp(z_v^x \cdot z_t^{c_k} / \tau)}{\sum_{j=1}^{K} \exp(z_v^x \cdot z_t^{c_j} / \tau)}$$

$$\hat{c} = \arg\max_{c \in \mathcal{C}} z_v^x \cdot z_t^c$$

- The candidate set $\mathcal{C}$ can be changed at inference time.
- Softmax scores depend on $\mathcal{C}$ and are not necessarily calibrated.

Building Custom Zero-Shot Classifiers

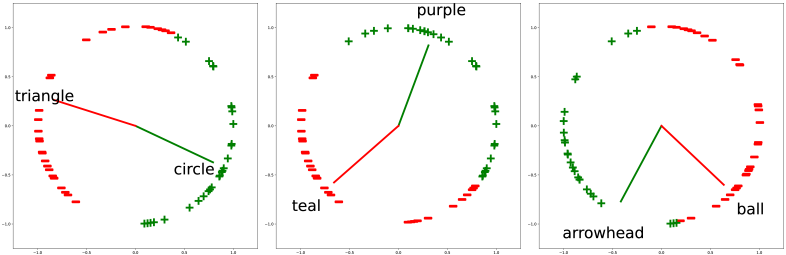


Figure: [*visionbook*]

- The same image embeddings are classified by shape, color, or object type.
- Changing the candidate text embeddings changes the classification boundary, without updating CLIP's parameters.
- With K text embeddings, classification selects the highest cosine similarity.

CLIP: Zero-Shot Results

- Zero-shot CLIP is compared with a supervised linear classifier trained on ResNet-50 features across 27 datasets.
- CLIP performs better on 16 of 27 datasets, including ImageNet.
- Lower performance on specialized domains, such as satellite and medical images, illustrates limitations in transfer.

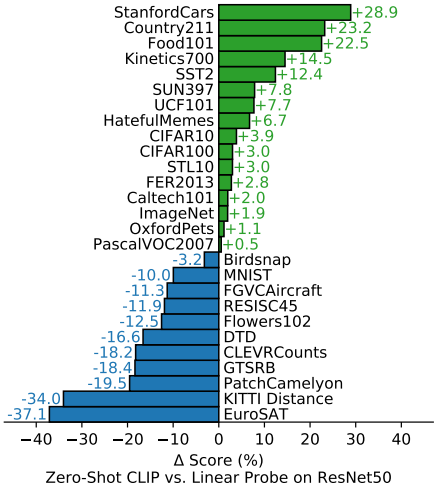


Figure: [*Radford et al., 2021: CLIP*]

Text-to-Image Retrieval

Given a text query q, rank candidate images $\{x_1, \ldots, x_M\}$ by cosine similarity:

$$\hat{\imath} = \arg \max_{i \in \{1,\ldots,M\}} z_t^q \cdot z_v^{x_i}$$

- Image embeddings can be precomputed; retrieval requires a matrix-vector product at inference time.
- Image-to-text retrieval uses the same procedure with the modalities exchanged. **Recall@K** measures the fraction of queries with a relevant result among the top K.

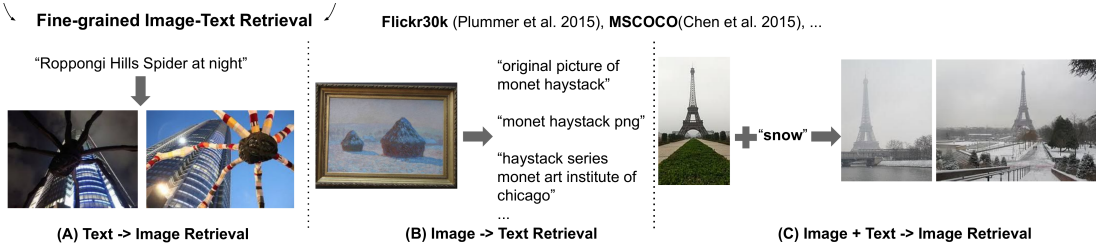


Figure: [*Jia et al., 2021: ALIGN*]

Prompting VLMs: Textual Prompts

- Replacing class names ("dog") with text templates ("a photo of a dog") improves zero-shot ImageNet accuracy by 1.3 percentage points.
- **Prompt ensembling**: average the normalized embeddings of multiple templates for each class, yielding a further 3.5-point improvement.
- Text embeddings can be precomputed, avoiding repeated text encoding for each input image.

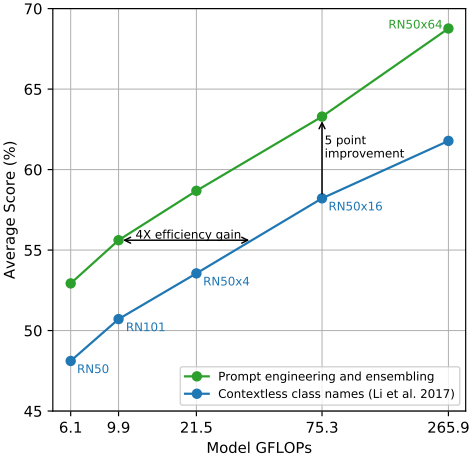


Figure: [*Radford et al., 2021: CLIP*]

Prompting VLMs: Visual Prompts

- Visual prompts modify the input image to indicate a region of interest, without updating model parameters.
- Red circles can direct CLIP toward marked objects while preserving the surrounding image context.
- For visual grounding, candidate regions are marked separately and ranked by image-text similarity.

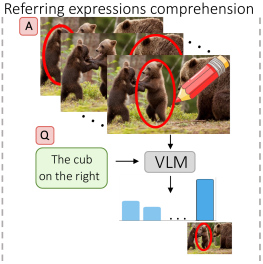


Figure: [*Shtedritski et al., 2023*]



Figure: [*Shtedritski et al., 2023: What Does CLIP Know About a Red Circle?*]

Limitations: Counting

- Standard image-text contrastive learning does not explicitly supervise object counts.
- CLIP often confuses different object counts in the evaluation by Paiss et al. (2023).
- Attention visualizations suggest incomplete coverage of repeated object instances.

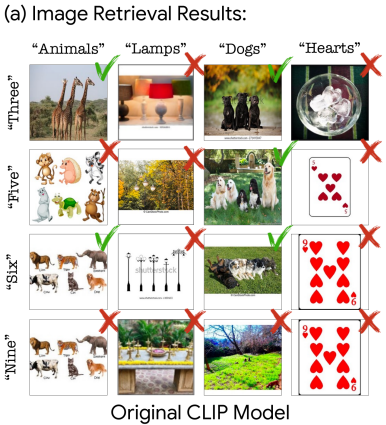


Figure: [*Paiss et al., 2023: Teaching CLIP to Count to Ten*]

Limitations: Relational Understanding

- Image-text alignment can rely heavily on object identity, with limited sensitivity to word order and semantic relations.
- Reversing subject and object changes the sentence meaning while preserving the same words.
- In the example, BLIP assigns similar matching scores to "the horse is eating the grass" and "the grass is eating the horse".

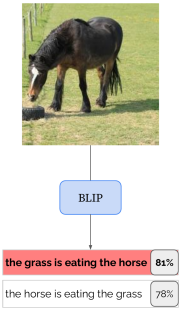


Figure: [*Yuksekgonul et al., 2023: VLMs as Bags-of-Words*]

Translating Images into Text

Image captioning via cross-attention

Encoder-Decoder Translation Between Modalities

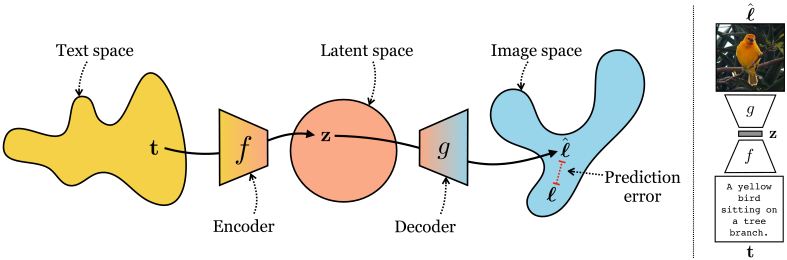


Figure: [*visionbook*]

- An encoder $f : \mathcal{X} \to \mathcal{Z}$ extracts input representations; a decoder $g : \mathcal{Z} \to \mathcal{Y}$ produces outputs in another modality.
- Pretrained encoders and decoders may require a learned connector to align their representation spaces.
- Image-to-text generation produces captions autoregressively while text-to-image generation can instead use a diffusion decoder.

Image Captioning: Autoregressive Generation

A caption $y_1, \ldots, y_T$ is generated autoregressively, conditioned on the image:

$$p_\theta(y \mid x) = \prod_{t=1}^{T} p_\theta(y_t \mid y_{<t}, x), \qquad \mathcal{L}_{\text{caption}} = -\sum_{t=1}^{T} \log p_\theta(y_t \mid y_{<t}, x)$$

- **Training**: teacher forcing uses ground-truth preceding tokens $y_{<t}$.
- **Inference**: tokens are generated sequentially from the model's previous outputs.
- Visual conditioning uses decoder cross-attention or projected visual tokens in the input sequence, depending on the architecture.

Image-to-Text: Cross-Attention

Queries come from decoder hidden states while keys and values come from image features:

$$Q = HW^Q, \quad K = Z_v W^K, \quad V = Z_v W^V$$

$$C = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

- $H \in \mathbb{R}^{T \times D_t}$: decoder states; $Z_v \in \mathbb{R}^{N \times D_v}$: image tokens.
- $QK^\top \in \mathbb{R}^{T \times N}$: each text position attends to N image tokens.
- Causal self-attention and cross-attention are separate decoder sublayers.

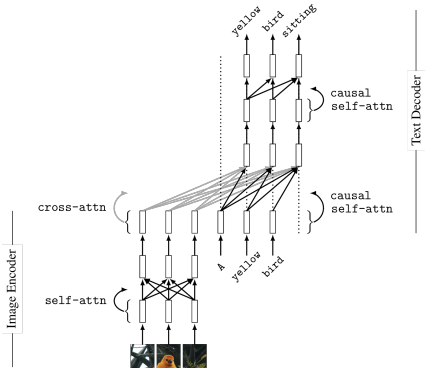


Figure: [*visionbook*]

BLIP-2: Bootstrapping Captioning from Frozen Models

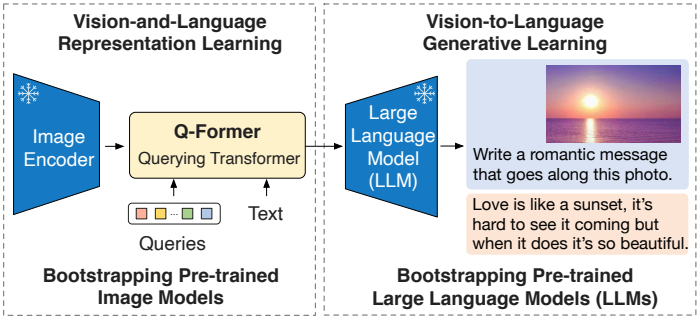


Figure: [*Li et al., 2023: BLIP-2*]

- **Architecture**: a trainable Q-Former connects a pretrained image encoder (CLIP/EVA-CLIP ViT) to a pretrained language model (OPT or FlanT5).
- **Stage 1) Representation learning**: freeze the image encoder and train the Q-Former with image-text contrastive, matching, and generation objectives: $\mathcal{L}_{\text{ITC}} + \mathcal{L}_{\text{ITM}} + \mathcal{L}_{\text{ITG}}$.
- **Stage 2) Generative pretraining**: freeze the image encoder and LLM, train the Q-Former and a linear projection to condition the LLM on visual queries.

BLIP-2: Inside the Q-Former

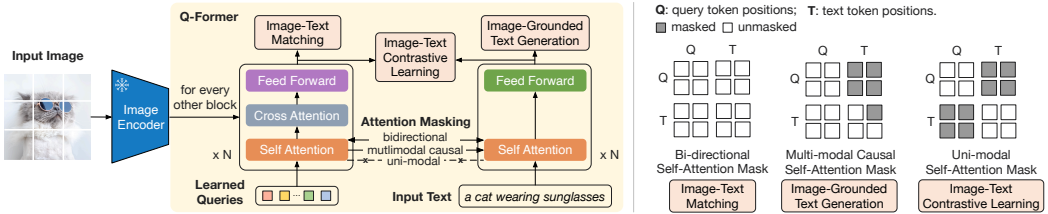


Figure: [*Li et al., 2023: BLIP-2*]

- The Q-Former uses 32 learned query embeddings (768-dimensional) and Transformer layers initialized from BERT-base. Queries cross-attend to frozen image features.
- Three training objectives use different attention masks:
  - **ITC**: image-text contrastive learning (query and text tokens cannot attend to each other).
  - **ITM**: binary image-text matching with bidirectional attention.
  - **ITG**: autoregressive text generation conditioned on image queries.

Text as a Visual Representation and VQA

What else can text represent about an image?

Text as a Visual Representation

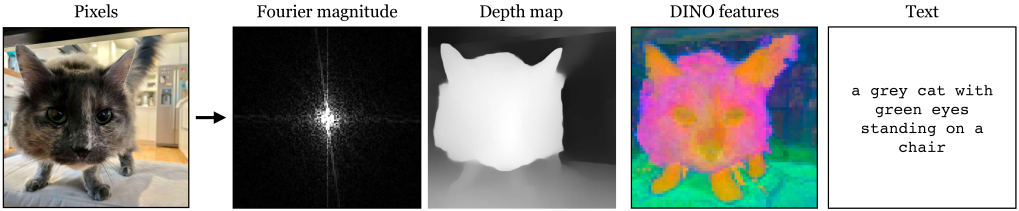


Figure: [*visionbook*]

- Images can be represented through pixels, depth, learned features, or natural language.
- Text provides a semantic representation compatible with language models and NLP methods, but does not preserve all visual information.

Visual Question Answering: Problem Definition



Figure: [*visionbook*]

- **Visual Question Answering (VQA)**: given an image x and a question q, predict an answer a.
- Unlike captioning, VQA requires question-dependent visual information and may involve reasoning.

Visual Question Answering: Commonsense Knowledge

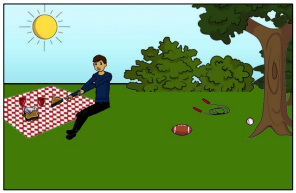
- VQA contains natural-language questions about images, collected through human annotation.
- Questions may require object recognition, spatial relations, and commonsense knowledge.
- Correct answers can depend on both visual evidence and prior world knowledge.



What color are her eyes?
What is the mustache made of?



How many slices of pizza are there?
Is this a vegetarian pizza?



Is this person expecting company?
What is just under the tree?



Does it appear to be rainy?
Does this person have 20/20 vision?

Figure: [*Antol et al., 2015: VQA*]

Visual Question Answering: Two Formulations

Answer classification

$$p_\theta(a \mid x, q), \qquad a \in \{a_1, \ldots, a_K\}$$

- Predict one answer from a fixed vocabulary, typically the most frequent training-set answers.
- Answers outside the vocabulary cannot be predicted.

Generative VQA

$$p_\theta(a \mid x, q) = \prod_{t=1}^{T} p_\theta(a_t \mid a_{<t}, x, q)$$

- Generate answer tokens autoregressively, conditioned on the image and question.
- BLIP-2 uses Q-Former visual queries while LLaVA uses projected visual tokens.

$$\mathcal{L}_{\mathrm{VQA}} = -\sum_{t=1}^{T} \log p_\theta(a_t \mid a_{<t}, x, q)$$

Visual Question Answering: BLIP-2

- **BLIP-2 VQA fine-tuning**: update the image encoder and Q-Former and keep the LLM frozen.
- The question conditions both the Q-Former and the LLM.
- Question tokens guide the Q-Former's cross-attention toward relevant image features. The LLM generates the answer autoregressively.

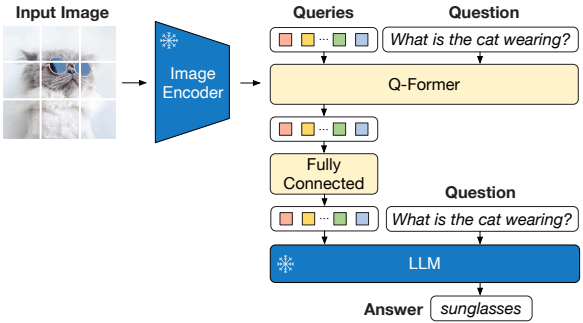


Figure: [*Li et al., 2023: BLIP-2*]

Language as a Visual Representation

- Captions encode selected visual content as natural language.
- VQA extends image captioning by conditioning generation on a question.
- Pretrained language models can combine visual information with linguistic and world knowledge.

Instruction-Tuned VLMs

From frozen encoders to chat-style assistants

LLaVA: Visual Instruction Tuning

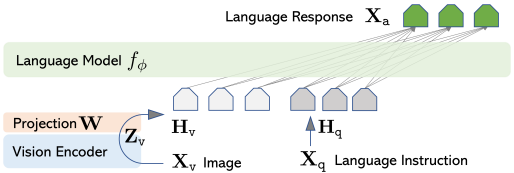


Figure: [*Liu et al., 2023: LLaVA*]

$$Z_v = f_v(x) \in \mathbb{R}^{N \times D_v}, \quad H_v = Z_v W \in \mathbb{R}^{N \times D_{\text{LLM}}},$$

$$\mathcal{L}_{\text{LLaVA}} = -\sum_{t=1}^{T} \log p_\theta(y_t \mid y_{<t}, H_v, X_q)$$

- **Pretrained components**: CLIP ViT-L/14 vision encoder and Vicuna language model.
- A trainable linear projection maps visual tokens into the LLM embedding space, where they are processed with text tokens by causal self-attention.
- The vision encoder remains frozen. LLaVA-1.5 replaces the linear projection with a two-layer MLP.

LLaVA: Training Stages

Component	Stage 1: Feature alignment	Stage 2: Instruction tuning
CLIP vision encoder	Frozen	Frozen
Projection W	Trainable	Trainable
Vicuna LLM	Frozen	Trainable

- **Stage 1) Feature alignment**: train only W on 595K CC3M image-caption pairs, formatted as image-description instructions.
- **Stage 2) Visual instruction tuning**: fine-tune W and Vicuna on 158K GPT-4-generated image-grounded instructions, including conversations, descriptions, and reasoning.

Qwen-VL: A Query-Based Cross-Attention Adapter

- **Pretrained components**: OpenCLIP ViT-bigG/14 vision encoder and Qwen-7B LLM.

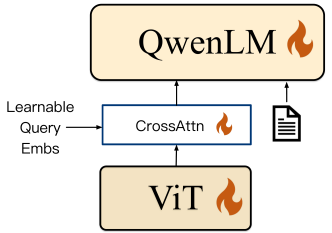


Figure: [*Bai et al., 2023: Qwen-VL*]

- **VL Adapter**: 256 learned queries cross-attend to ViT features, producing a fixed-length visual token sequence.
- A single cross-attention layer uses 2D positional encodings to retain spatial information during token compression.
- Unlike LLaVA's token-wise projector, the adapter reduces the number of visual tokens; unlike BLIP-2, it does not use a multi-layer Q-Former.

Qwen-VL: Training Stages

Component	Stage 1	Stage 2	Stage 3
ViT	Trainable	Trainable	Frozen
VL Adapter	Trainable	Trainable	Trainable
Qwen-LM	Frozen	Trainable	Trainable

- **Stage 1) Pretraining**: 1.4B cleaned image-text pairs at 224^2 resolution. Train the ViT and adapter; freeze the LLM.
- **Stage 2) Multi-task pretraining**: train all components at 448^2 resolution on seven tasks, including captioning, VQA, grounding, OCR, and text generation.
- **Stage 3) Instruction tuning**: train the adapter and LLM on 350K dialogue examples, keeping the ViT frozen. Produces Qwen-VL-Chat.

Comparing VLM Architectures

Model	Visual representation	Connector	Language component
CLIP	Global image embedding	Linear projections	Text encoder
BLIP-2	Visual tokens	Q-Former + projection	Frozen pretrained LLM
LLaVA-1.5	CLIP visual tokens	Two-layer MLP	Fine-tuned pretrained LLM
Qwen-VL	ViT visual features	Query-based cross-attention	Qwen-7B LLM

Other VLM Tasks

Reasoning and Grounding

Visual Reasoning: NLVR2 Benchmark

- **NLVR2**: determine whether a natural-language statement is true for a pair of photographs.

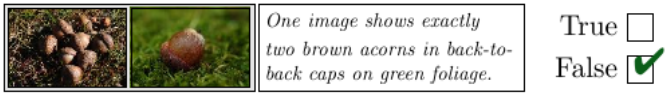


Figure: [*Suhr et al., 2019: NLVR2*]

$$p_\theta(y \mid x_1, x_2, s), \qquad y \in \{\text{True}, \text{False}\}$$
$$\mathcal{L}_{\text{NLVR2}} = -\log p_\theta(y \mid x_1, x_2, s)$$

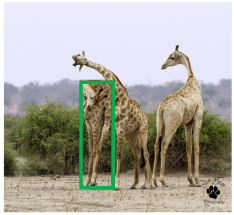
- Binary classification requiring comparison, counting, and relations across two images.

Visual Grounding: Referring Expressions

- Given an image x and a referring expression r, predict the corresponding bounding box.

$$\hat{b} = f_\theta(x, r) \qquad \text{or} \qquad \hat{k} = \arg\max_k s_\theta(b_k, r, x)$$

- Direct box regression or selection among candidate regions $\{b_k\}$.



RefCOCO:
1. giraffe on left
2. first giraffe on left

RefCOCO+:
1. giraffe with lowered head
2. giraffe head down

RefCOCOg:
1. an adult giraffe scratching its back with its horn
2. giraffe hugging another giraffe

Figure: [*Yu et al., 2016: Modeling Context in Referring Expressions*]

- **RefCOCO**, **RefCOCO+**, and **RefCOCOg** evaluate referring-expression comprehension, with different annotation protocols.

Recap: VLM Tasks

Task	Input	Output	Training objective	Example
Zero-shot classification	Image + class names	Class label	Image-text contrastive learning	CLIP
Image captioning	Image	Caption	Autoregressive language modeling	BLIP-2
Visual QA	Image + question	Answer	Answer classification or generation	BLIP-2, LLaVA
Visual reasoning	Image pair + statement	True / False	Binary classification	NLVR2 dataset
Visual grounding	Image + description	Bounding box	Region matching or localization	RefCOCO dataset

Summary

- **Vision-language alignment**: CLIP learns a shared embedding space through contrastive learning, enabling zero-shot classification and image-text retrieval.
- **Multimodal fusion**: BLIP-2 and Qwen-VL use query-based cross-attention, while LLaVA projects visual tokens into the LLM embedding space.
- **Text generation**: generative VLMs predict tokens autoregressively, conditioned on image and text inputs.
- **Training strategies**: BLIP-2 freezes both pretrained backbones during pretraining while LLaVA and Qwen-VL fine-tune different components across training stages.

Next Lectures

- Parameter-efficient fine-tuning (PEFT)
- Lab on VLMs and PEFT

References

- Radford et al. (2021). Learning Transferable Visual Models From Natural Language Supervision (CLIP). ICML 2021.
- van den Oord, Li & Vinyals (2018). Representation Learning with Contrastive Predictive Coding (InfoNCE). arXiv 2018.
- Ordonez, Kulkarni & Berg (2011). Im2Text: Describing Images Using 1 Million Captioned Photographs. NeurIPS 2011.
- Mori, Takahashi & Oka (1999). Image-to-Word Transformation Based on Dividing and Vector Quantizing Images With Words. 1999.
- Joulin, Van Der Maaten, Jabri & Vasilache (2016). Learning Visual Features from Large Weakly Supervised Data. ECCV 2016.
- Li, Jabri, Joulin & van der Maaten (2017). Learning Visual N-Grams from Web Data. ICCV 2017.
- Antol et al. (2015). VQA: Visual Question Answering. ICCV 2015.
- Suhr, Zhou, Zhang, Bai & Artzi (2019). A Corpus for Reasoning About Natural Language Grounded in Photographs (NLVR2). ACL 2019.
- Bordes et al. (2024). An Introduction to Vision-Language Modeling. arXiv 2024.
- Schuhmann et al. (2022). LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models. NeurIPS 2022.
- Jia et al. (2021). Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision (ALIGN). ICML 2021.
- Cherti et al. (2023). Reproducible Scaling Laws for Contrastive Language-Image Learning (OpenCLIP). CVPR 2023.
- Paiss, Ephrat, Tov, Zada, Mosseri, Irani & Dekel (2023). Teaching CLIP to Count to Ten. ICCV 2023.
- Li, Li, Savarese & Hoi (2023). BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Encoders. ICML 2023.
- Liu, Li, Wu & Lee (2023). Visual Instruction Tuning (LLaVA). NeurIPS 2023.
- Yuksekgonul, Bianchi, Kalluri, Jurafsky & Zou (2023). When and Why VLMs Behave Like Bags-of-Words. ICLR 2023.
- Shtedritski, Rupprecht & Vedaldi (2023). What Does CLIP Know About a Red Circle? Visual Prompt Engineering for VLMs. ICCV 2023.
- Zhai, Wang, Mustafa, Steiner, Keysers, Kolesnikov & Beyer (2022). LiT: Zero-Shot Transfer with Locked-image Text Tuning. CVPR 2022.
- Liu, Li, Li & Lee (2024). Improved Baselines with Visual Instruction Tuning (LLaVA-1.5). CVPR 2024.
- Bai et al. (2023). Qwen-VL: A Versatile Vision-Language Model. arXiv 2023.
- Desai & Johnson (2020). VirTex: Learning Visual Representations from Textual Annotations. arXiv 2020.
- Bulent Sariyildiz, Perez & Larlus (2020). Learning Visual Representations with Caption Annotations (ICMLM). ECCV 2020.
- Zhang, Jiang, Miura, Manning & Langlotz (2020). Contrastive Learning of Medical Visual Representations (ConVIRT). MLHC 2022.
- Torralba, Isola & Freeman. Foundations of Computer Vision (MIT visionbook), Ch. 51: Vision and Language.
- Varma. Introduction to Vision-Language Models. Stanford CS277/BIODS271 lecture notes.
- Marr (1982). Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. W.H. Freeman.


Vision-Language Models (VLMs)
Donald Shenaj
Università degli Studi di Padova
November 11, 2026
1


Lecture Outline
2


Recap
3


From Fixed Tokens to Dense Language Representations
4

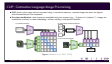

Vision-Language Foundation
5


Vision-Language Tasks
6


Aligning Vision and Language
7


Representing Text for a Vision Network
8


Projection to LLM
9


LLaVA - Connecting Language Image Processing
10


LLaVA: A Single Linear Text Pro
11


LLaVA: Choosing an Encoder Architecture
12

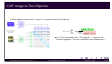

LLaVA: Connecting Into a Pooled Space
13


LLaVA: A Bunch of It Text
14


LLaVA: Image Embeddings Are the Bunch
15


LLaVA: Text Embeddings Are the Bunch
16


LLaVA: LM Process Architecture
17


LLaVA: Image as Text Objectives
18


LLaVA: Text Synthesis Objective
19


LLaVA: Training Procedure
20


LLaVA: Learned Image Text Tuning
21


LLaVA: Representations on Different Depths
22


LLaVA: The Learn Embedding Space
23


Training Data: A Visual-Language Mixture
24


Data and Scale Learning
25

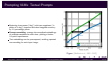

LLaVA-NP: Objectives and Training LLaVA
26


LLaVA: Zero-Shot Classification
27


Scaling: Custom Zero-Shot Classifiers
28


LLaVA: Zero-Shot Results
29


Data to Image Retrieval
30


Prompting VLMs: Textual Prompts
31


Prompting VLMs: Visual Prompts
32

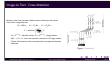

Underscore: Grounding
33


Underscore: Behavioral Understanding
34


Translating Images into Text
Image captioning as a core architecture
35


Encoder-Decoder: Transformer Between Modalities
36


Image Captioning: Autoregressive Generation
37


Images as Text: Cross-Attention
38


BLIP-2: Bootstrapping Capturing Text From Frozen Models
39


BLIP-2: Inside the Q-Former
40

Text as a Visual Representation and VQA
What else can we capture about an image?
41

Text as a Visual Representation
42

Visual Question Answering: Problem Definition
43

Visual Question Answering: Common Failure Modes
44

Visual Question Answering: Two Directions
45

Visual Question Answering: BLIP-2
46

Language as a Visual Representation
47

Instruction-Tuned VLMs
From frozen encoders to instruction-aware systems
48

LLaVA: Visual Instruction Tuning
49

LLaVA: Training Stages
50

Qwen-VL: A Query-Based Cross-Attention Adapter
51

Qwen-VL: Training Stages
52

Comparing VLM Architectures
53

Demo: VLM Tools
Reasoning and Grounding
54

Visual Reasoning: Multi-Benchmark
55

Visual Grounding: Bounding Expressions
56

Survey: VLM Tasks
57

Summary
58

Next Lectures
59

References
60