








Tutorial: Multimodal Foundation Models

Multimodal Foundation Models: : From Specialists to General-Purpose Assistants

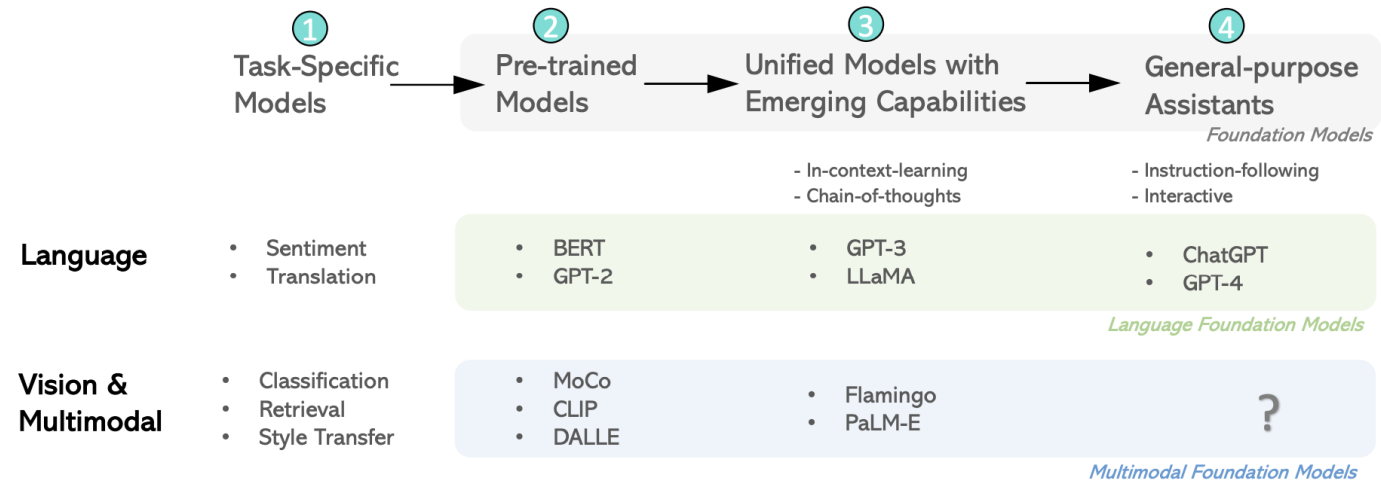
Authors:  [Chunyuan Li](#),  [Zhe Gan](#),  [Zhengyuan Yang](#),  [Jianwei Yang](#),  [Linjie Li](#),  [Lijuan Wang](#),  [Jianfeng Gao](#) | [Authors Info & Claims](#)

[Foundations and Trends® in Computer Graphics and Vision, Volume 16, Issue 1-2](#) • Pages 1 - 214
<https://doi.org/10.1561/06000000110>

Published: 06 May 2024 [Publication History](#)

From Task-Specific Models to General-Purpose Assistants

- AI has evolved from **single-task specialists** to **general-purpose multimodal agents**.
- **Language domain:**
 - *BERT / GPT-2 → GPT-3 / LLaMA → ChatGPT / GPT-4*
- **Vision & Multimodal domain:**
 - *MoCo / CLIP / DALL·E → Flamingo / PaLM-E → (Next: GPT-4V / Gemini)*
- Both domains share a similar trajectory:
Pre-training → Unification → General-purpose assistants
- Key challenge today:
Discovering the “recipe” for a truly multimodal GPT-4.



Three Core Research Directions

•Q1 Visual Understanding — “How to learn visual representations?”

- Self-supervised, language-supervised, and contrastive learning (CLIP, MoCo).

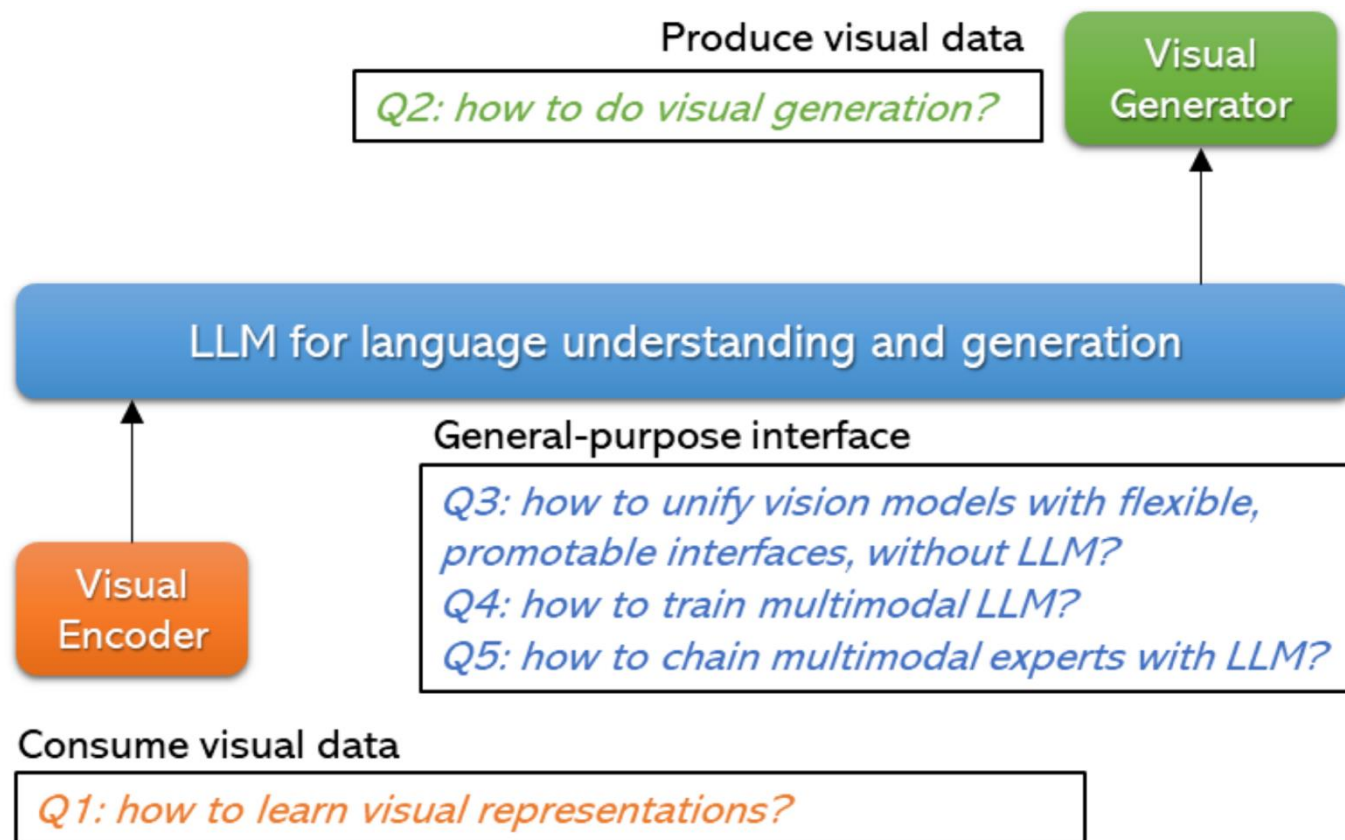
•Q2 Visual Generation — “How to generate visual data?”

- Text-to-image/video models (DALL·E 2, Stable Diffusion, Imagen).

•Q3–Q5 General-Purpose Interface — “How to make models interactive like ChatGPT?”

- Unified vision models (CLIP → OpenSeg);
- End-to-end multimodal LLMs (Flamingo, GPT-4V);
- Chaining LLM with vision tools (Visual ChatGPT, MM-REACT).

Key Idea: *Understanding = seeing, Generation = drawing, Interface = thinking & collaborating.*



From Specialists to Visual Assistants

- **Specialist Models:** CLIP, SimCLR, BEiT, SAM, Stable Diffusion — strong but task-specific.
- **General-Purpose Assistants:**
 - Understand human intentions and follow instructions across tasks.
 - Built through three pathways:

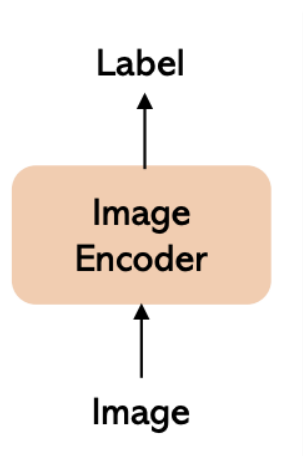
Unified vision modeling (without LLM)

End-to-end LLM training for vision inputs

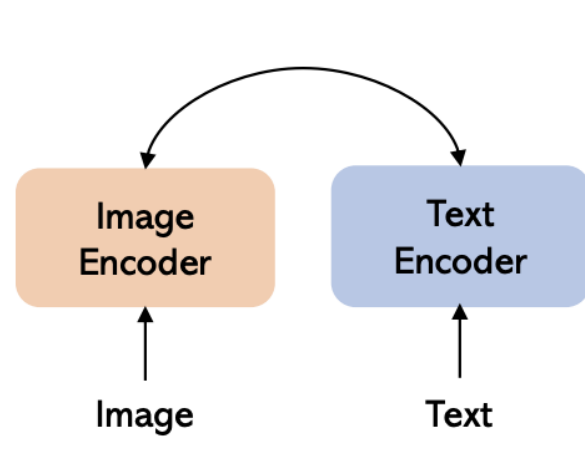
Training-free LLM tool chaining

Image Representation Learning Paradigms

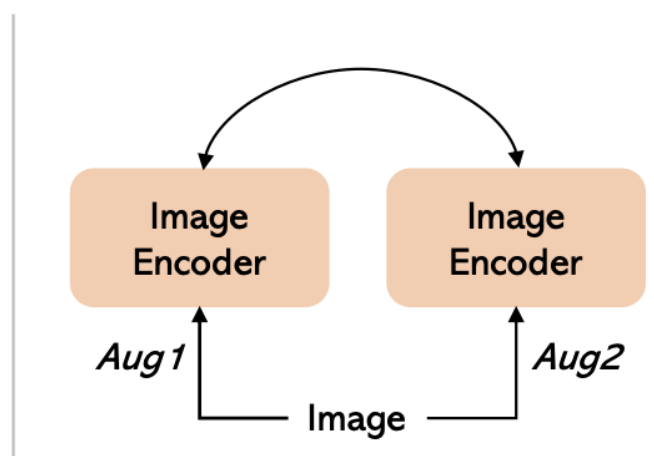
- Different approaches aim to learn **generalizable image encoders**.
- Four key paradigms:
 - Supervised learning** – learns from labeled data.
 - Contrastive language–image pre-training (CLIP)** – aligns image and text spaces.
 - Image-only self-supervision** – contrastive / non-contrastive (SimCLR, BYOL, DINO).
 - Masked image modeling (MIM)** – reconstructs masked visual content (BEiT, MAE).



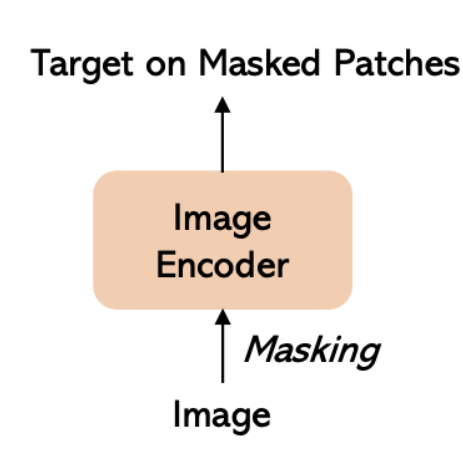
(a) Supervised Learning



(b) CLIP



(c) Image-only (non-)contrastive learning



(d) Masked image modeling

Contrastive Language–Image Pre-training

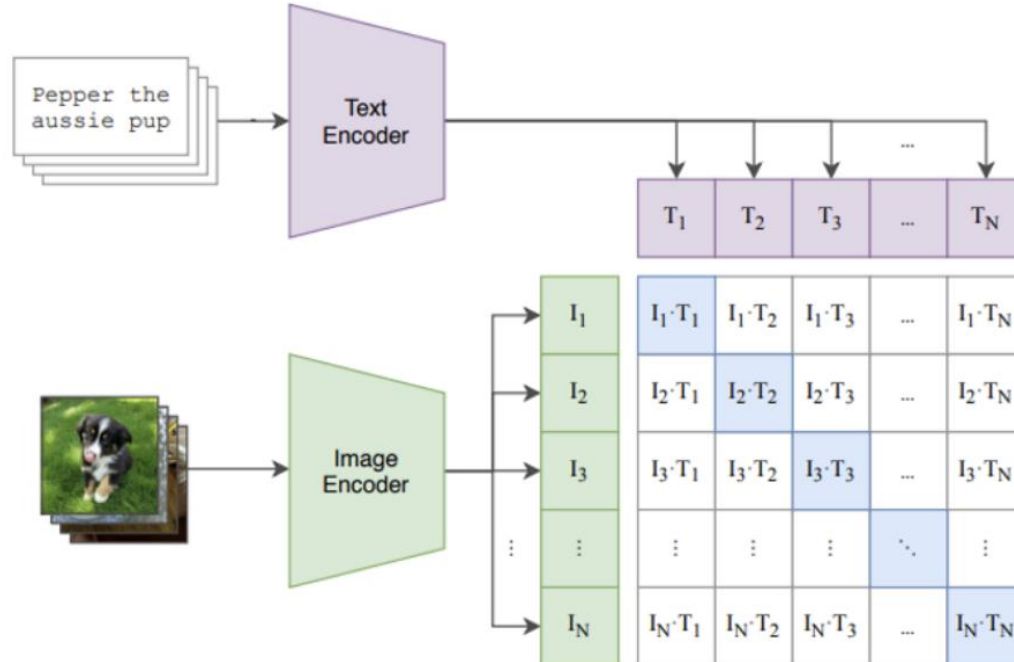
- Training:**

- Each image–caption pair forms a **positive example**.
- Contrastive loss pushes matching pairs closer and mismatched pairs apart.
- Large batch sizes (16 K – 32 K) and billions of image–text pairs enable scalability.

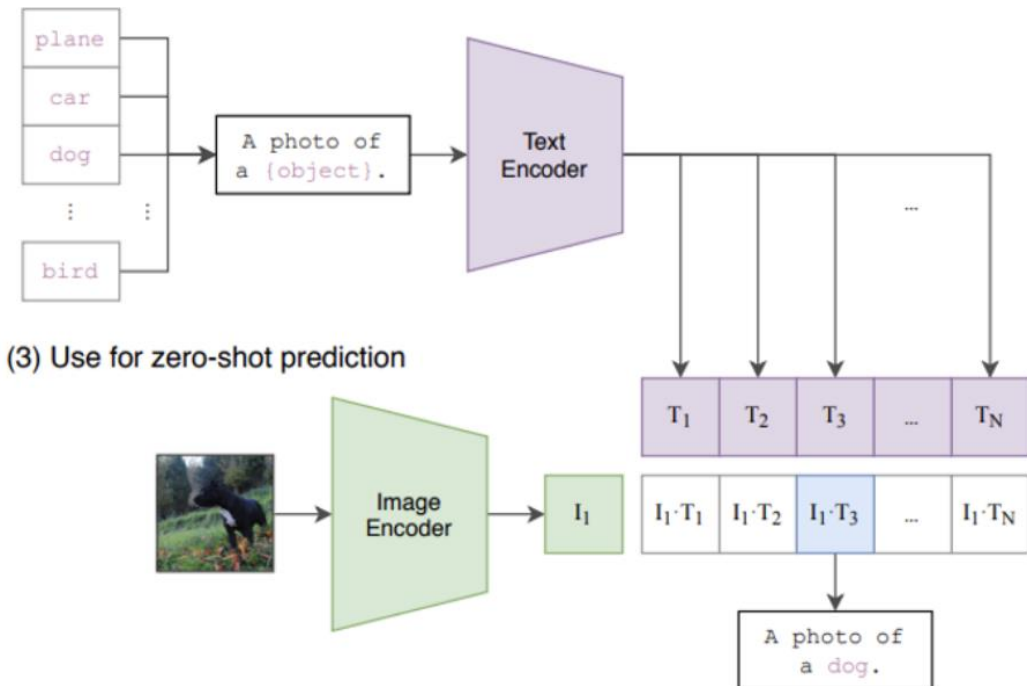
- Zero-shot use:**

- Reformulate classification as **text–image retrieval**.
- Class names converted to prompts (“a photo of a dog”).
- Model predicts by comparing image embeddings to text embeddings.

(1) Contrastive pre-training



(2) Create dataset classifier from label text



(3) Use for zero-shot prediction

ImageBind: Unified Multimodal Representation Learning

- Goal:** Extend contrastive pre-training beyond image–text pairs to *six modalities*:

Images Videos Text Audio Depth Thermal IMU

- Key idea:**

- Align *naturally paired* data (e.g., image–text, video–audio) in a **shared embedding space**.

- Leverage *emergent alignment* to connect unpaired modalities (e.g., depth↔audio) through image as a bridge.

- Outcome:** Enables zero-shot cross-modal retrieval and reasoning — e.g.,
“Find the sound corresponding to this video” or “Retrieve a thermal view for this image.”

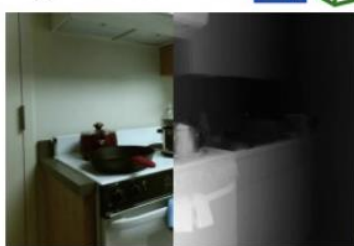


Web Image-Text



Sheep basking in the sun

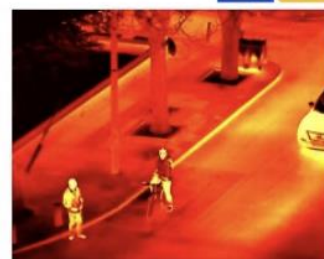
Depth Sensor Data



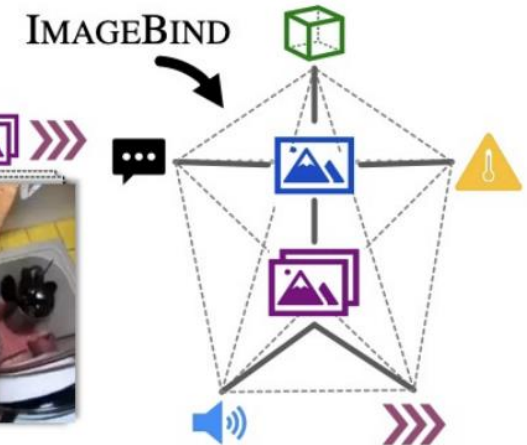
Web Videos



Thermal Data



Egocentric Videos



From CLIP to CoCa: Expanding Vision–Language Learning

- CLIP (a):**

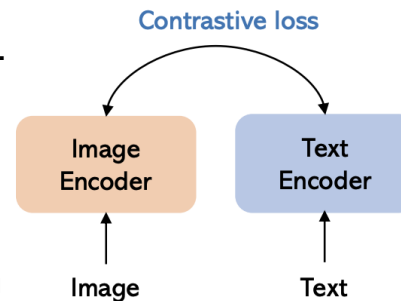
- Uses *contrastive loss* between image and text encoders.
- Learns shared embeddings for image–text retrieval and zero-shot classification.

- VirTex / SimVLM (b):**

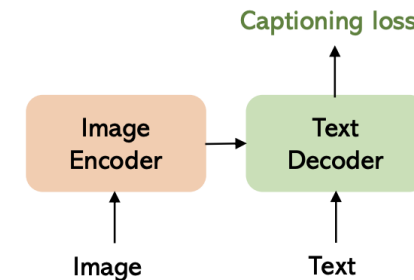
- Uses *captioning loss* only.
- Treats image–text pre-training as a language generation problem.
- The text decoder learns to generate captions from image features.

- CoCa (c):**

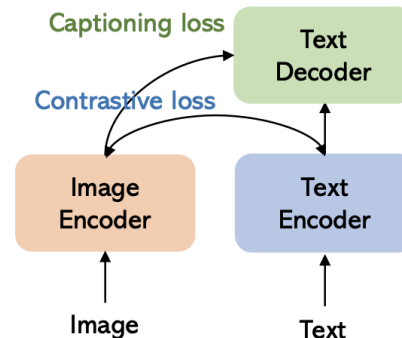
- Combines both **contrastive** and **captioning** losses.
- Jointly trains image encoder, text encoder, and text decoder.
- Achieves strong performance across retrieval and captioning tasks.



(a) CLIP



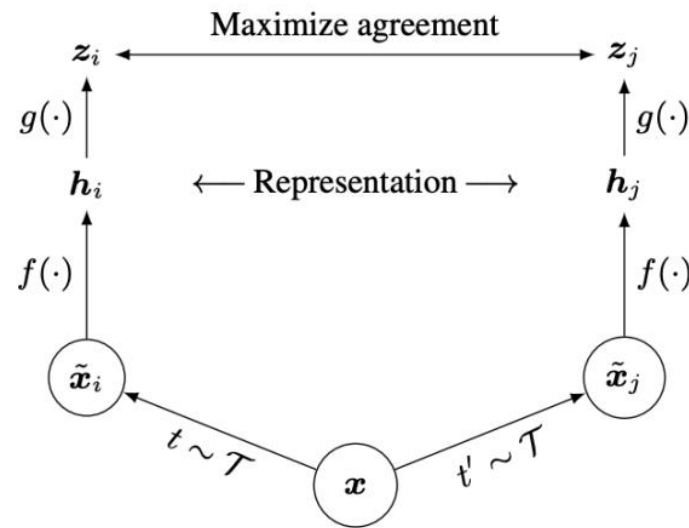
(b) VirTex/SimVLM



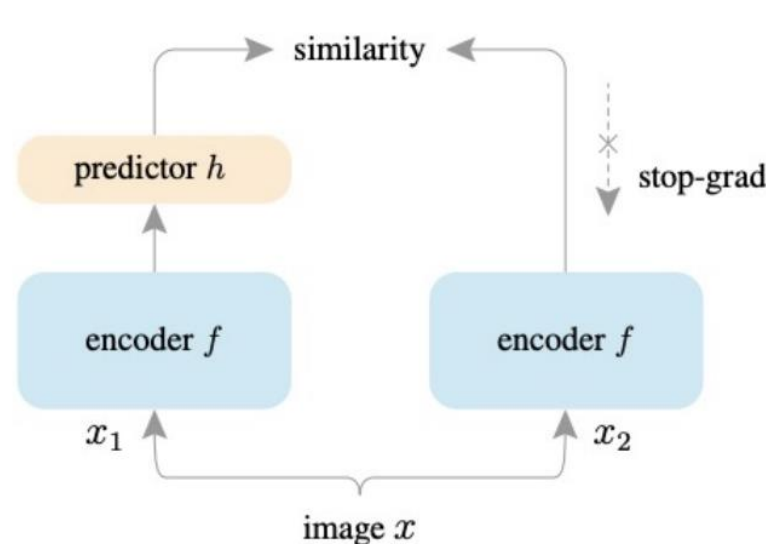
(c) CoCa

Image-Only (Non-Contrastive Learning: From SimCLR to DINO

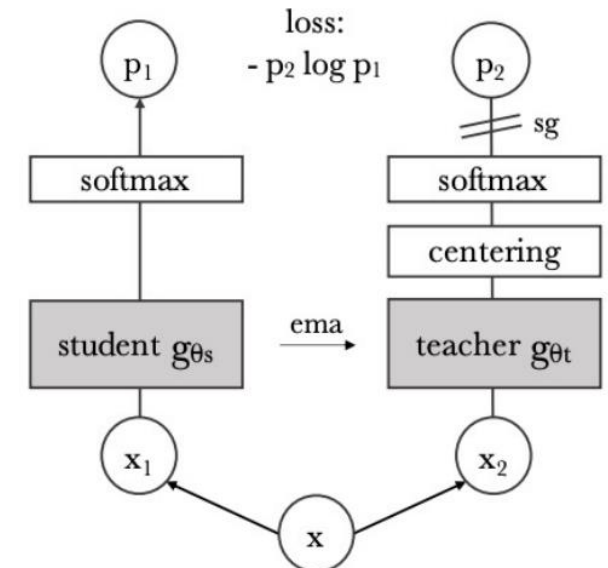
- These methods learn *visual representations* without using labels or text.
- **(a) SimCLR (Chen et al., 2020):**
 - Learns by maximizing agreement between augmented image pairs.
 - Requires *large batch sizes* and *negative samples*.
- **(b) SimSiam (Chen & He, 2021):**
 - Removes negatives via a *stop-gradient trick* to prevent collapse.
 - A predictor head aligns features between two augmented views.
- **(c) DINO (Caron et al., 2021):**
 - Uses *teacher-student* networks.
 - The teacher provides stable targets through EMA updates.
 - Learns cluster-like features, enabling attention maps and segmentation.



(a) SimCLR



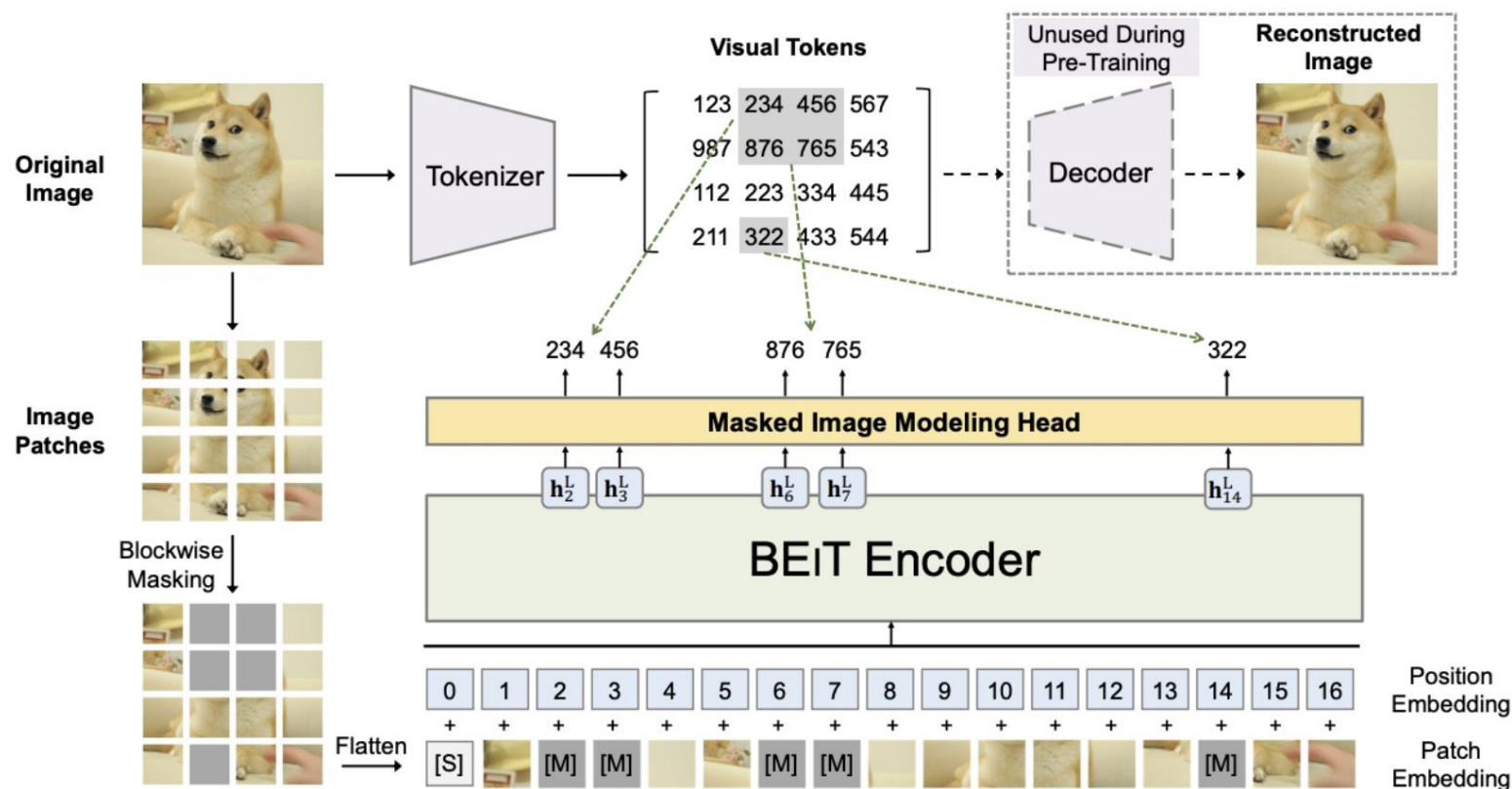
(b) SimSiam



(c) DINO

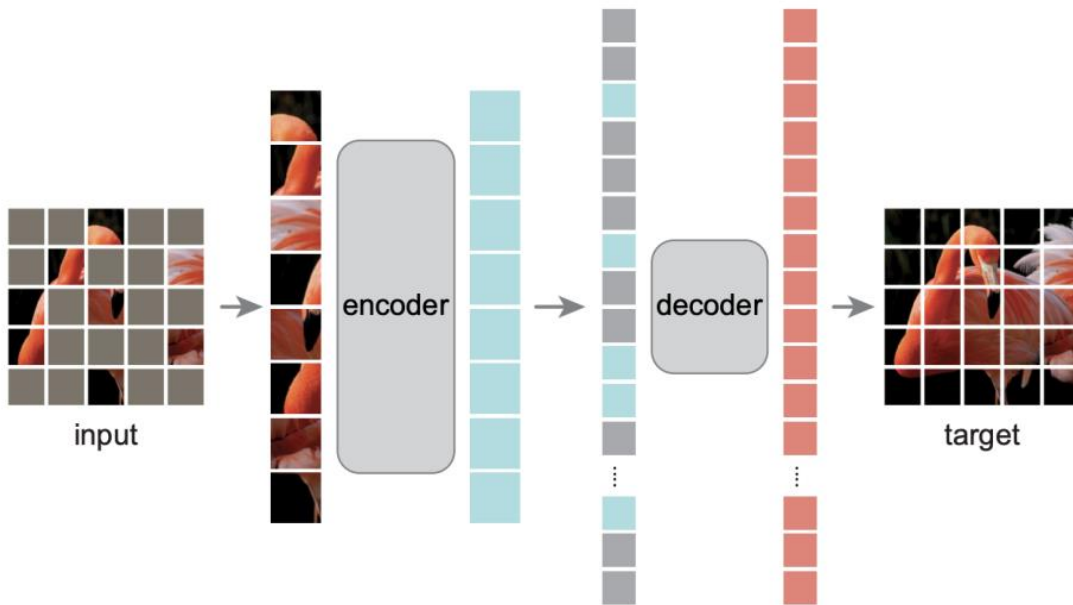
Masked Image Modeling (MIM): The BEiT Framework

- **Goal:** Learn visual representations by reconstructing masked image patches — analogous to BERT in NLP.
- **Process overview:**
 1. Split the image into patches and apply **blockwise masking**.
 2. **Tokenizer** converts visible patches into **visual tokens** (discrete IDs).
 3. The **BEiT encoder** predicts the missing tokens through a **masked image modeling head**.
 4. Decoder (optional) reconstructs the full image.
- **Key idea:**
 - Predict discrete visual tokens rather than raw pixels → more semantic learning.
 - Enables **self-supervised pre-training** without labels.
- **Extensions:**
 - Inspired follow-ups: **MAE**, **SimMIM**, **MaskFeat**, **BEiT v2/v3**, integrating vision–language and generative objectives.

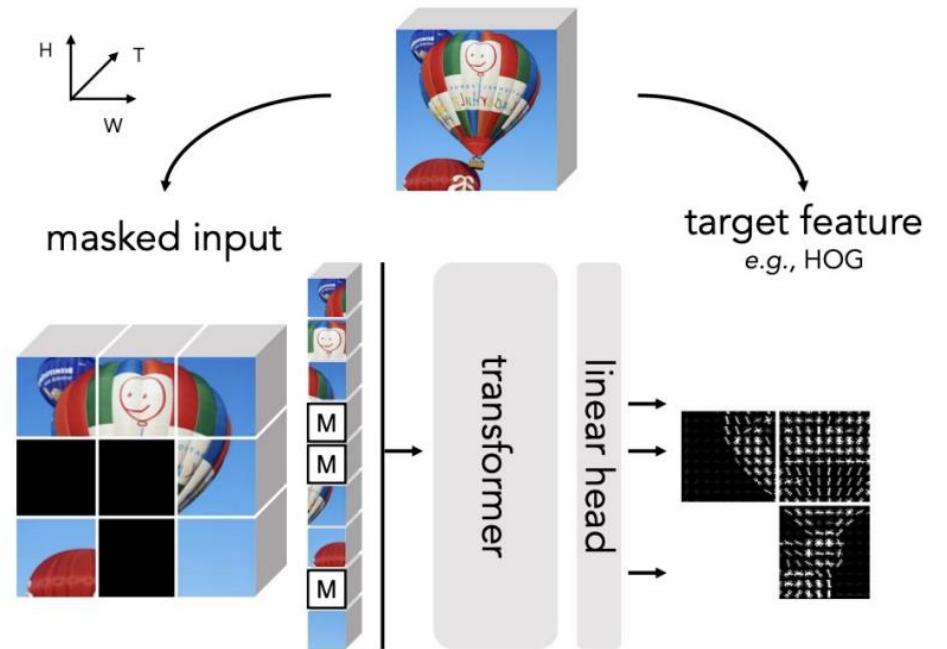


Advancements in Masked Image Modeling: MAE and MaskFeat

- (a) **MAE (Masked Autoencoder)**
 - Uses **high masking ratio** ($\approx 75\%$) on image patches.
 - Encoder** processes only visible patches \rightarrow reduces computation.
 - Lightweight decoder** reconstructs the full image from encoded + masked patches.
 - Learns strong visual priors for fine-tuning downstream tasks.
- (b) **MaskFeat**
 - Extends MIM beyond pixel reconstruction.
 - Predicts **handcrafted target features** (e.g., HOG, gradient maps).
 - Encourages learning **semantically meaningful features** over low-level pixels.
 - Works well even with shallow decoders, improving transfer efficiency.



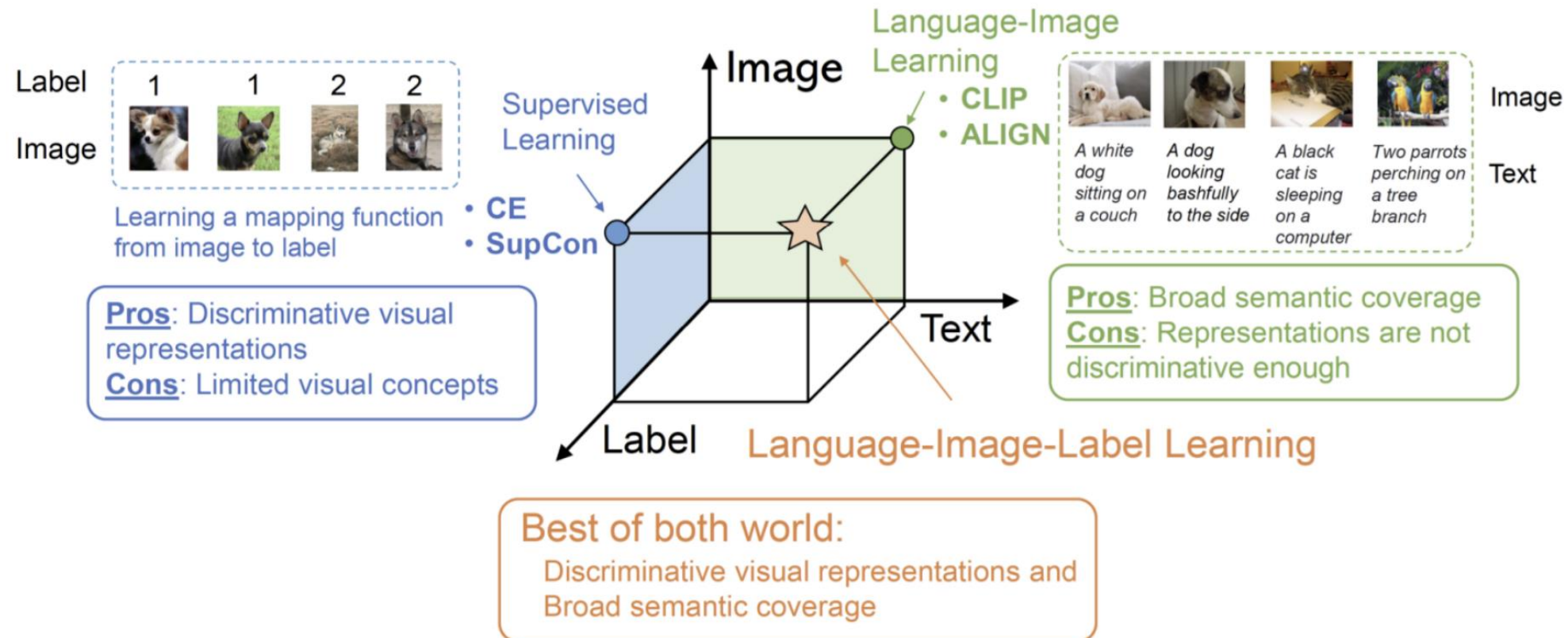
(a) MAE



(b) MaskFeat

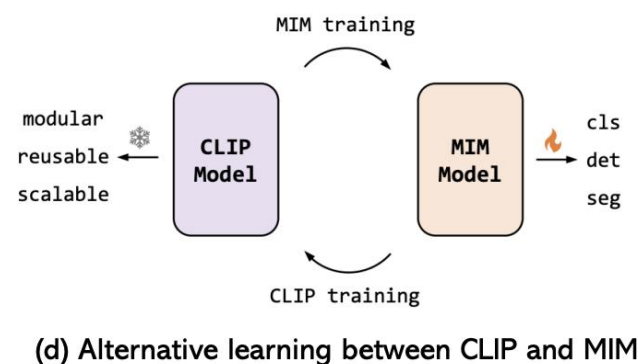
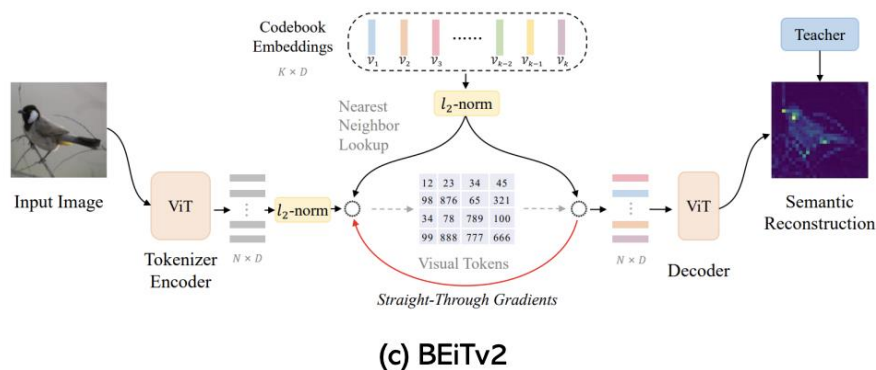
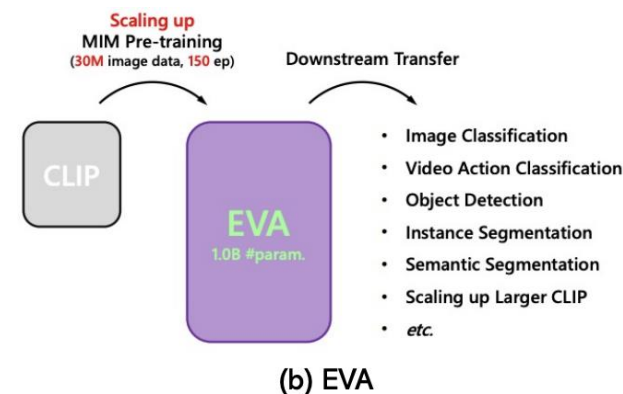
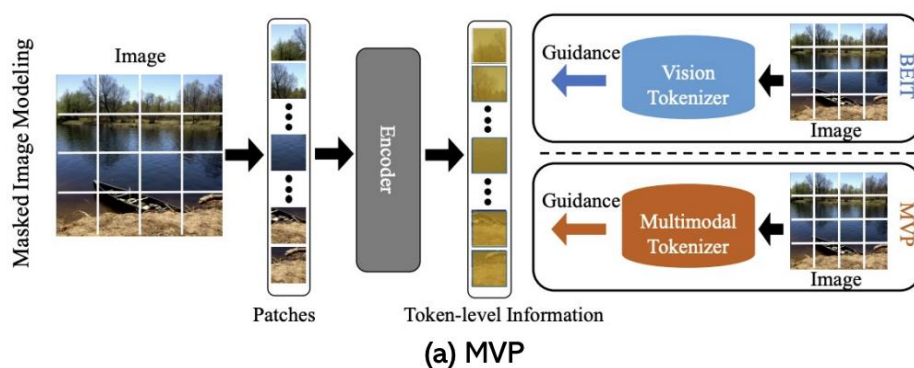
Bridging Supervised and Language-Image Learning

- **Motivation:**
 - Supervised learning (image–label) provides *discriminative visual representations* but limited concept diversity.
 - CLIP-style image–text contrastive learning offers *broad semantics* but weaker discriminability.
- **UniCL (Yang et al., 2022):**
 - Unifies image–label and image–text data into a **joint contrastive learning space**.
 - Learns from both categorical labels and descriptive texts.
 - Framework extended to large-scale system **Florence (Yuan et al., 2021)**.
- **Outcome:**
 - Combines advantages of both paradigms —
→ **Strong discrimination + Rich semantic coverage**.



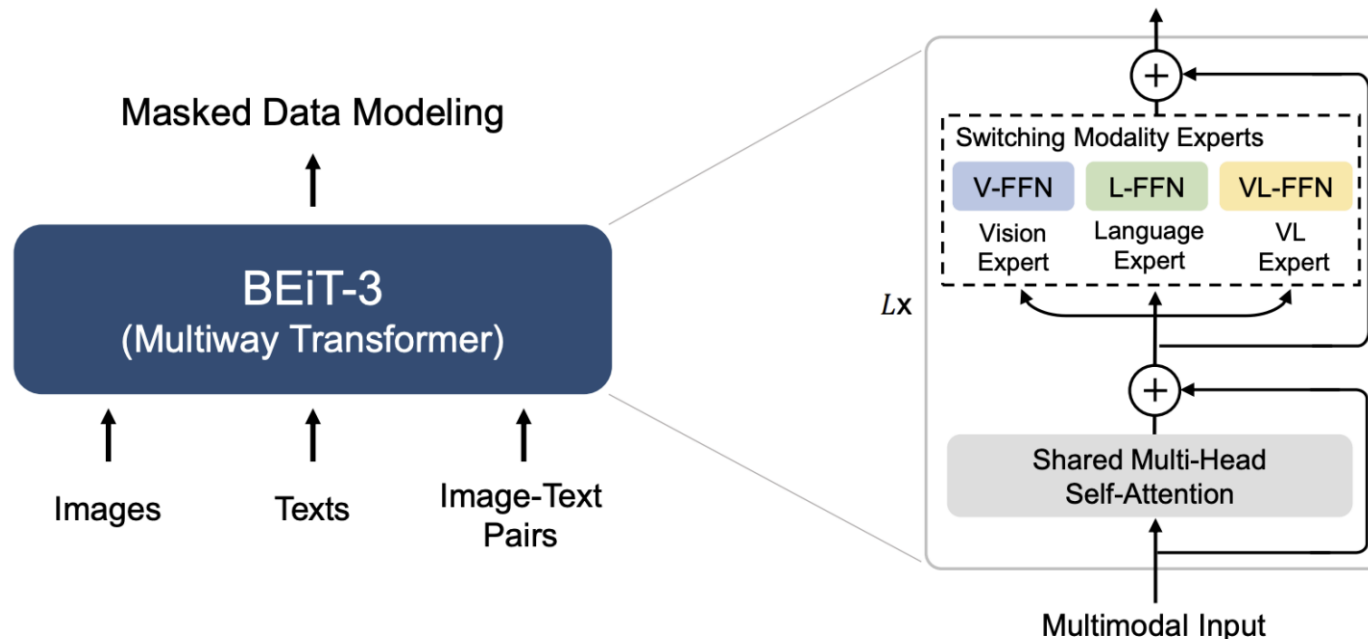
Bridging CLIP and Masked Image Modeling (MIM)

- **Goal:** Combine the strengths of **contrastive pre-training (CLIP)** and **masked reconstruction (MIM)**.
- **(a) MVP (Wei et al., 2022)**
 - Uses CLIP features as **guidance targets** for MIM pre-training.
 - Multimodal tokenizer integrates visual + text semantics.
- **(b) EVA (Fang et al., 2023)**
 - Scales up MIM pre-training (30 M images, 1 B params).
 - Enables large-scale downstream transfer (Cls, Seg, Det).
- **(c) BEiTv2 (Peng et al., 2022)**
 - Compresses CLIP features into discrete visual tokens for BEiT-style training.
- **(d) Alternative Learning**
 - MIM \leftrightarrow CLIP cycle: use CLIP features as MIM targets & MIM encoder to warm-start CLIP.



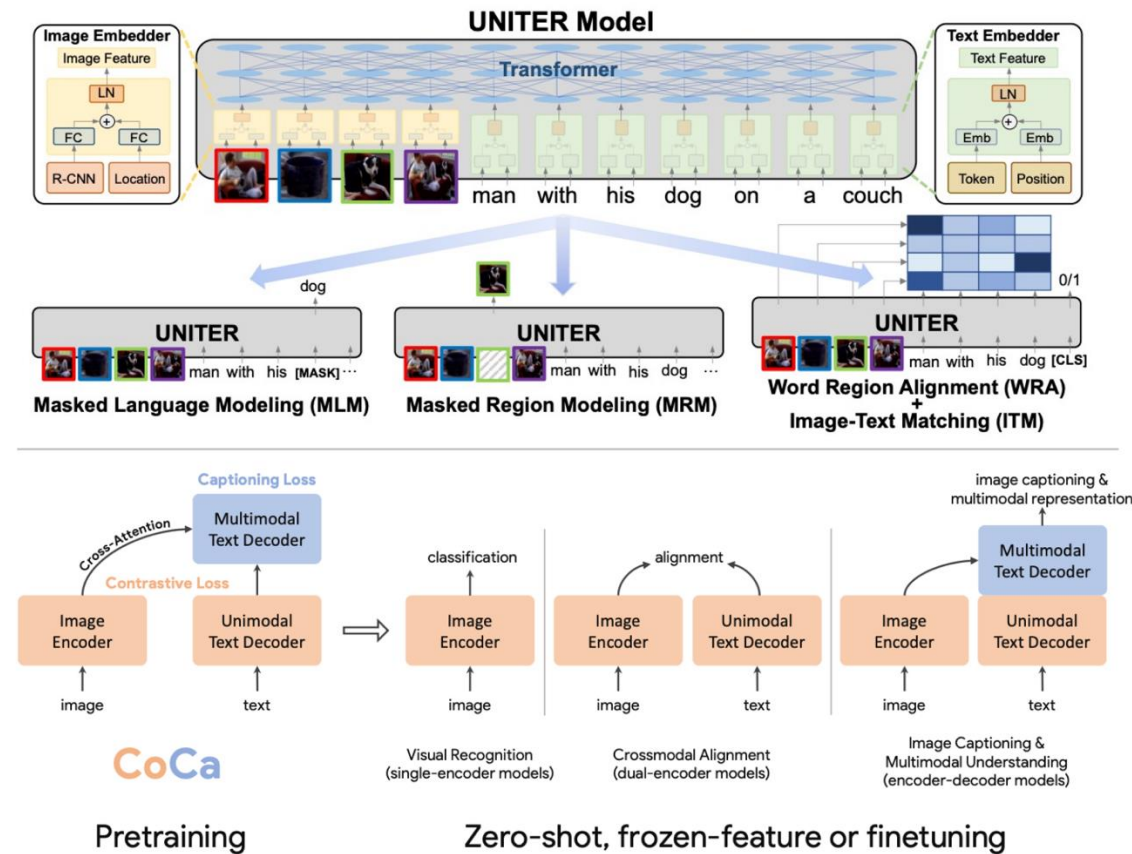
BEiT-3: Unified Multimodal Masked Modeling

- **Goal:** Extend MIM to a unified framework for both vision and language.
- **Input:**
 - Images, Texts, and Image-Text Pairs.
- **Architecture:**
- **Shared multi-head self-attention** for cross-modal alignment.
- **Modality-specific experts:**
 - V-FFN (Vision Expert)
 - L-FFN (Language Expert)
 - VL-FFN (Vision-Language Expert)
- The system dynamically switches experts according to modality.
- **Learning objective:**
- **Masked Data Modeling** across all modalities (like BERT + BEiT).
- **Result:**
 - Achieves **state-of-the-art performance** on vision and vision-language tasks (e.g., classification, captioning, VQA).



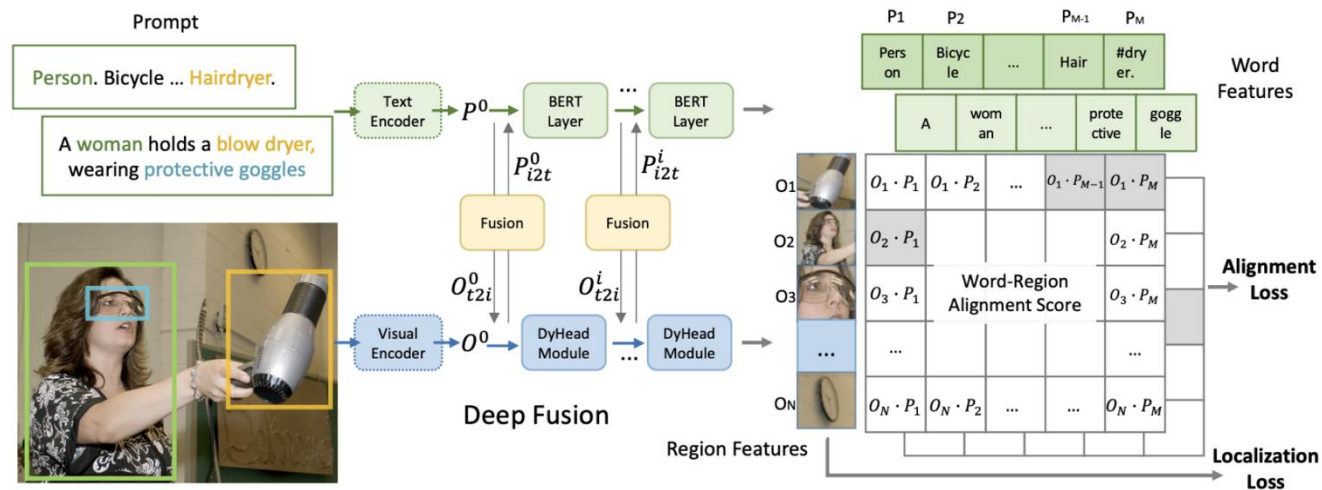
From UNITER to CoCa — Evolution of Multimodal Fusion

- **Motivation:** CLIP handles image–text alignment but lacks *deep fusion*.
- **UNITER (Chen et al., 2020)**
 - Uses **object-detector features** (e.g., R-CNN + LN + FC).
 - Models cross-modal interactions via a **joint transformer**.
 - Pre-trained with:
 - **MLM:** Masked Language Modeling
 - **MRM:** Masked Region Modeling
 - **ITM:** Image-Text Matching
 - **WRA:** Word-Region Alignment
- **CoCa (Yu et al., 2022)**
 - Fully end-to-end; trains all components from scratch.
 - Combines **Contrastive Loss** (CLIP-style) and **Captioning Loss** (text decoder).
 - Enables both **image recognition** and **image captioning** tasks.



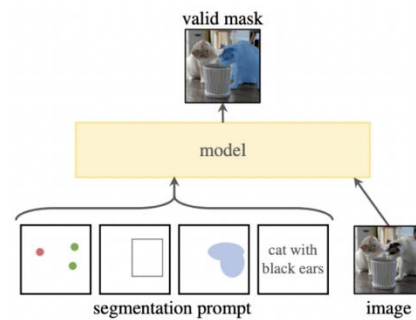
GLIP — Grounded Language-Image Pre-training for Detection

- **Goal:** Extend vision-language pre-training to **region-level understanding**, enabling open-vocabulary object detection.
- **Input:**
 - Natural-language **prompts** (“person, hairdryer, goggles...”)
 - **Images** with region proposals.
- **Model Design:**
 - **Text Encoder (BERT-based):** encodes phrase- or word-level semantics.
 - **Visual Encoder (Region Extractor):** extracts region-level visual features.
 - **DyHead Fusion Modules:** progressively align text and vision features.
- **Training Objective:**
 - **Alignment Loss:** Align words \leftrightarrow regions (word-region similarity matrix).
 - **Localization Loss:** Detect and localize object instances.
- **Outcome:**
 - Enables **zero-shot detection** for unseen categories by grounding text to regions.

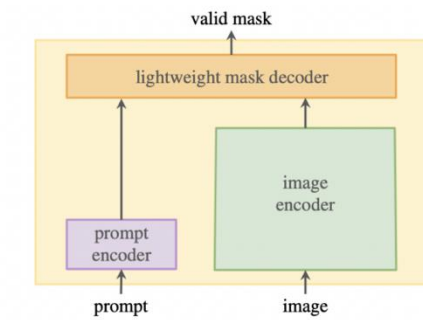


Segment Anything Model (SAM): Pixel-Level Foundation Model

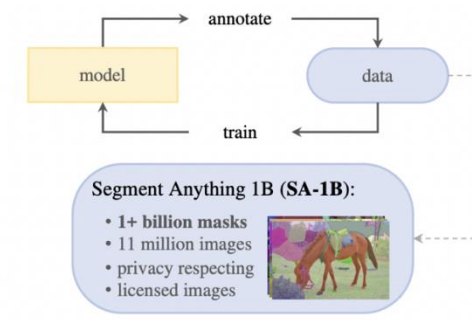
- **Goal:** Build a **general-purpose vision foundation model** for segmentation, capable of adapting to any segmentation prompt.
- **Promptable Segmentation (Task):**
 - Input prompts = points, boxes, masks, or text (e.g., “cat with black ears”).
 - Output = corresponding **segmentation mask**.
- **Model Architecture:**
 - **Image Encoder:** ViT pre-trained via MAE (He et al., 2022).
 - **Prompt Encoder:**
 - Sparse input → CLIP text encoder.
 - Dense input → CNN encoder.
 - **Lightweight Mask Decoder:** Transformer-based head produces pixel-level masks.
- **Data Engine (SA-1B):**
 - Model-in-the-loop annotation creates > **1 billion masks** from **11 million images**.
 - Enables iterative train–annotate–retrain cycle for scalable segmentation.



(a) **Task:** promptable segmentation



(b) **Model:** Segment Anything Model (SAM)

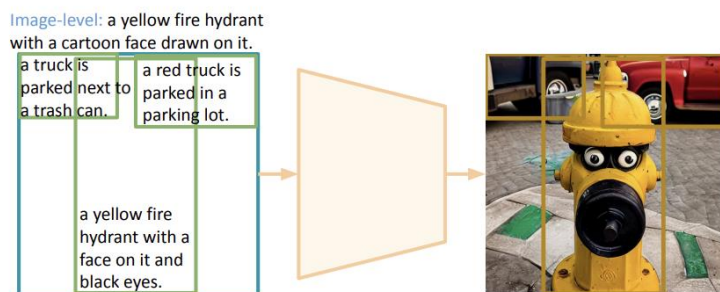


(c) **Data:** data engine (top) & dataset (bottom)

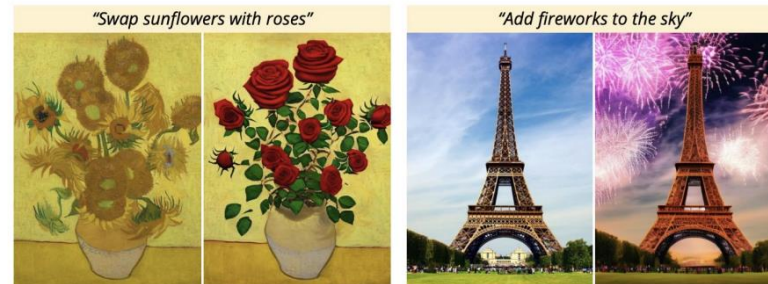
Text-to-Image Foundation Models: Aligning Visual Generation with Human Intent

- Four key themes:
 1. Spatial controllable T2I generation
 2. Text-based editing
 3. Text-prompt following
 4. Concept customization
- End with unified human-alignment tuning

(a) Spatial Controllable T2I Generation



(b) Text-based Editing



(c) Text Prompts Following

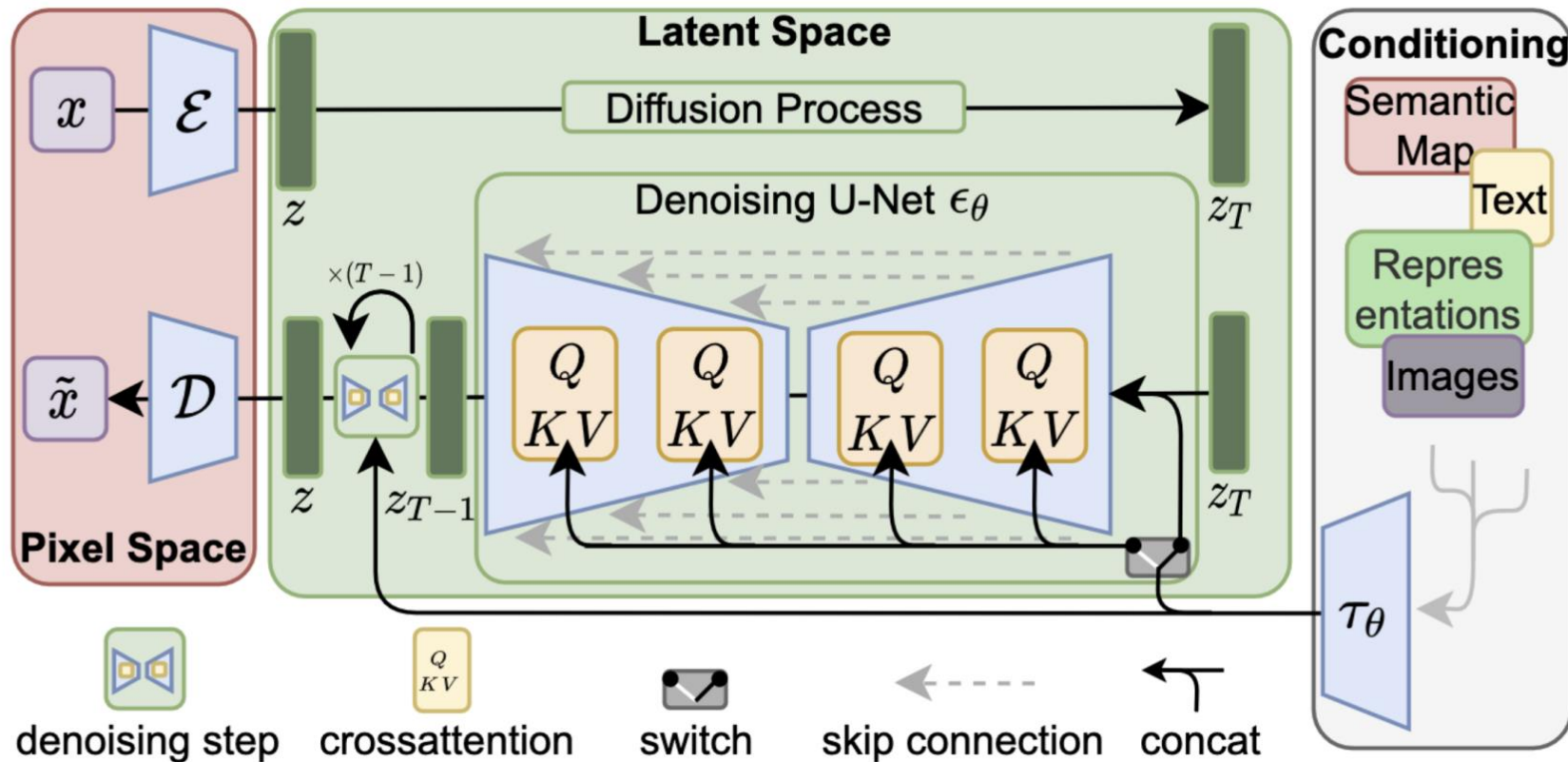


(d) Concept Customization



Latent Diffusion Framework

- Pixel \rightarrow latent \rightarrow denoised \rightarrow reconstructed image pipeline
- Denoising U-Net uses **cross-attention** to link text embeddings and visual latents
- Conditioning signals: text, semantic maps, dense features



Region-Controlled T2I (ReCo)

- Adds *positional box tokens* <204>, <687> to text for localized control
- Each region description encoded separately; diffusion model learns layout consistency
- Enables generation with multi-object spatial relationships

Region-Controlled Input Sequence



#1 Region Description: baseball player is swinging a bat and wearing a blue and white jersey.

#2 Region Description: a catcher in a gray and black uniform is crouching and ready to catch the ball.

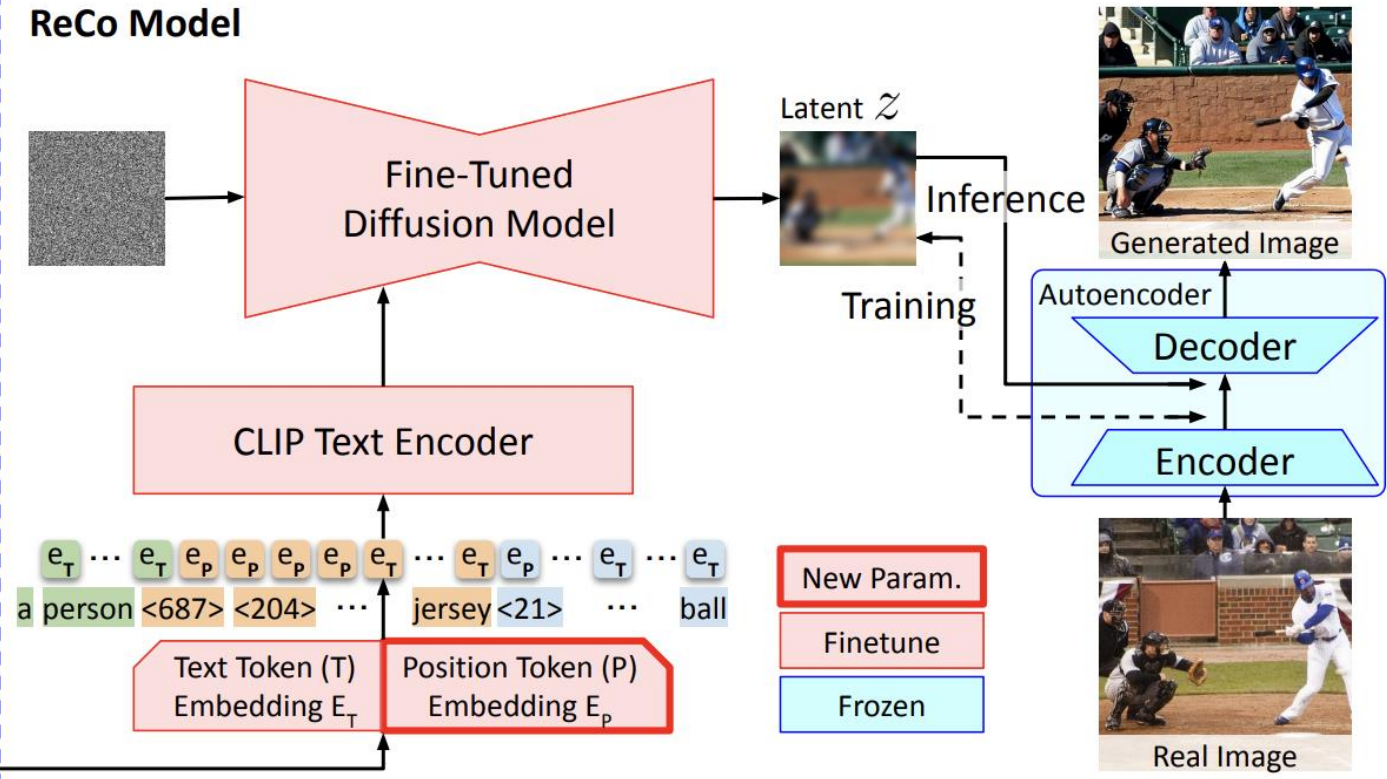
#3 Region Description: a baseball player with the number 19 on his jersey.

Image Description: a person standing at the plate in mid swing of a bat.

Input Query = Image Description + [Region-Controlled Text] * #Regions:

a person standing at the plate in mid swing of a bat
 <687> <204> <999> <833> baseball player ... jersey.
 <21> <447> <433> <840> a catcher in gray ... ball.
 <0> <323> <123> <827> a baseball player ... jersey.

ReCo Model



Dense Conditional Controls (ControlNet)

- Beyond box tokens → dense visual maps (depth, edges, poses)
- ControlNet injects these maps into diffusion layers for pixel-level control
- Supports sketch-to-image, pose-transfer, and semantic layout generation

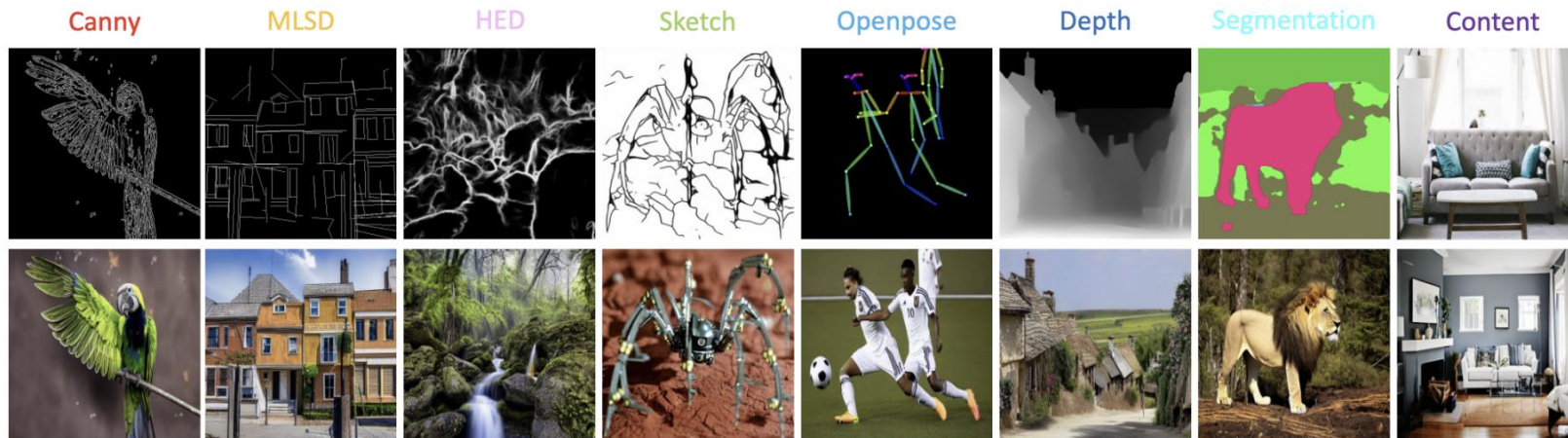


Figure 3.5: Examples of the dense controls and the corresponding generated images. Image credit: [Zhao et al. \(2023b\)](#).



(a) A headshot of a woman with a dog in winter.

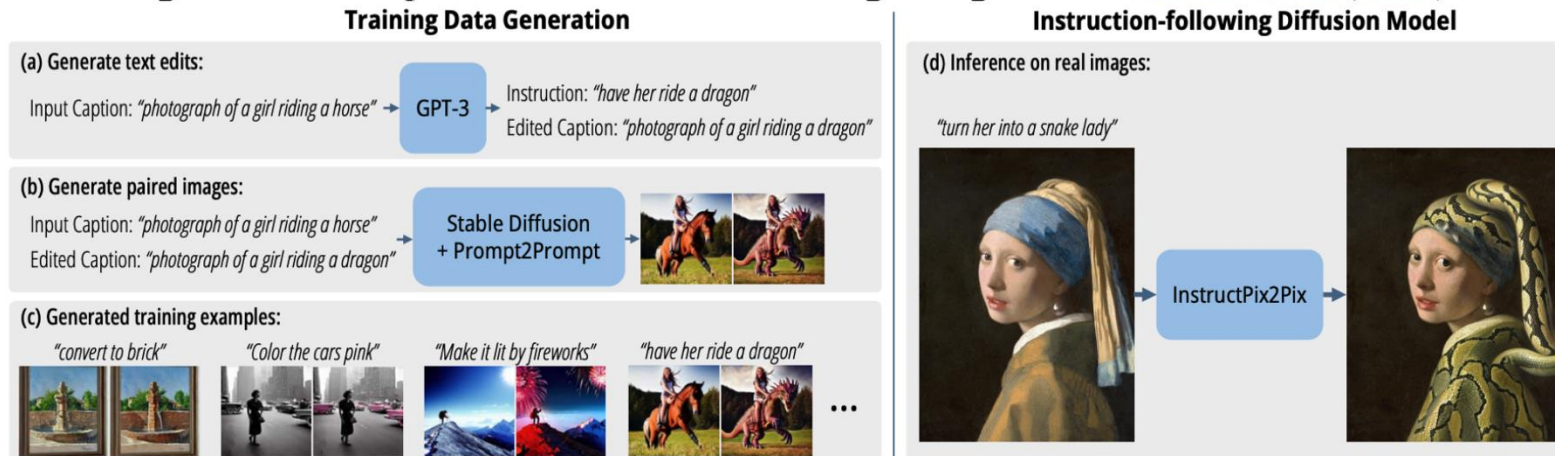
(b) A headshot of a woman with a dog on beach.

Text Instruction Editing (InstructPix2Pix)

- Model trained on synthetic instruction–edit pairs
- Editing performed without re-training
- Enables open-ended editing like “swap sunflowers with roses”, “add fireworks”, “replace the fruits with cake”



Figure 3.8: Examples of text instruction editing. Image credit: [Brooks et al. \(2023\)](#).



Improving Text–Image Alignment during Inference

- Vanilla T2I may mis-assign attributes (e.g., red car ↔ white sheep)
- **StructureDiffusion**: parses prompt into noun phrases to balance attention
- **Attend-and-Excite**: adjusts latent z_t with loss that boosts under-attended tokens

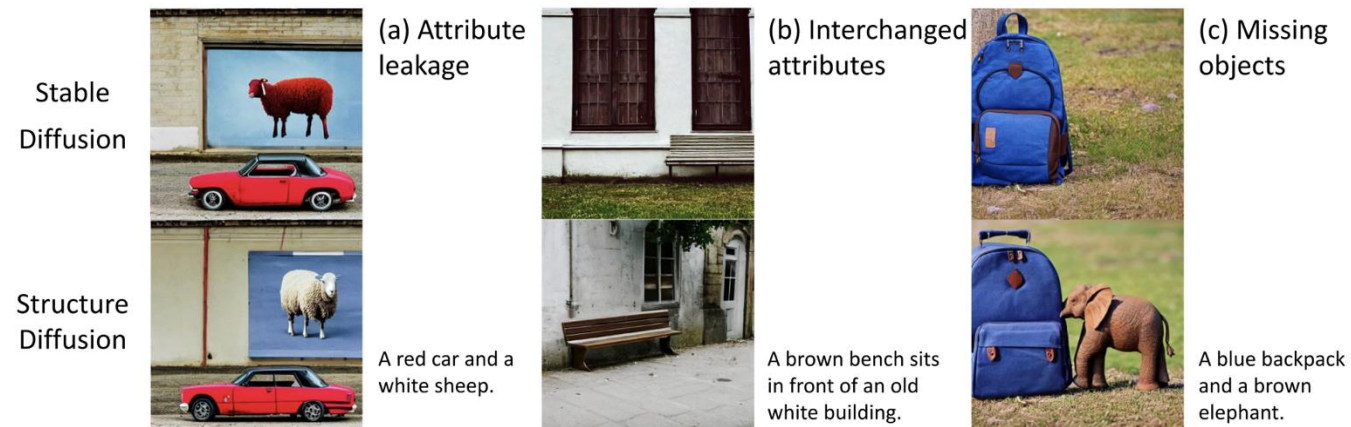
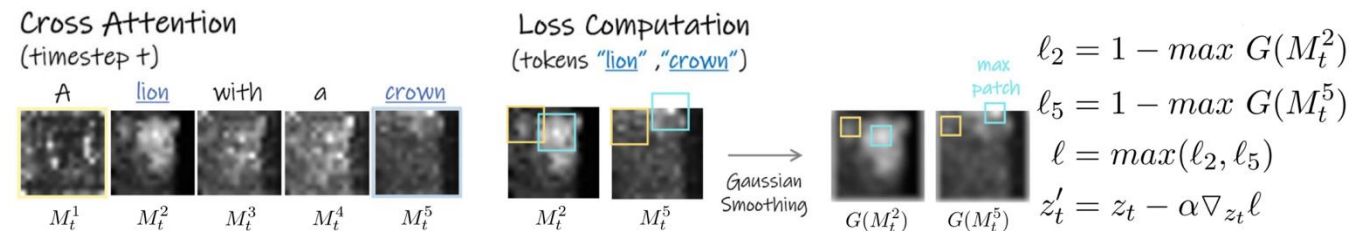
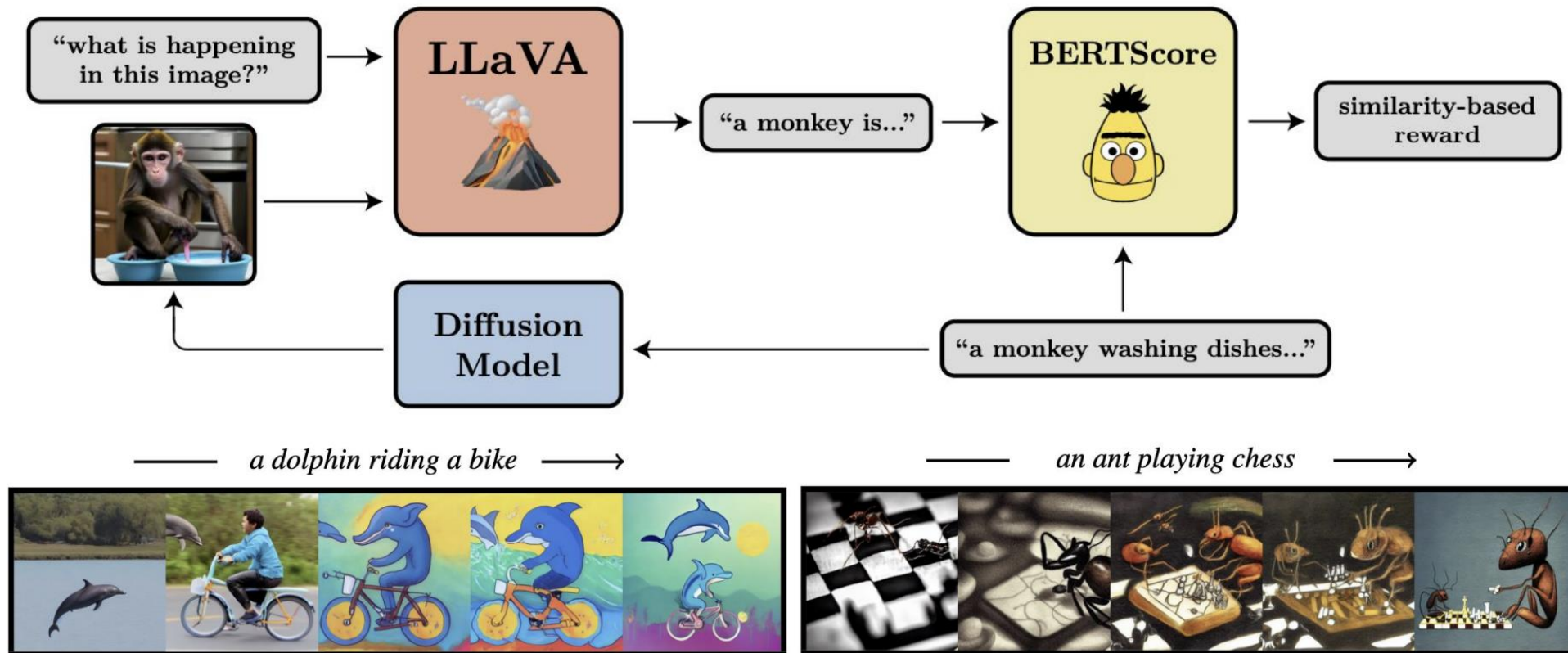


Figure 3.10: Failure cases of vanilla T2I model in text prompt following. Image credit: [Feng et al. \(2022b\)](#).



Model Tuning with Reward-Based Alignment

- DDPO: uses vision-language model (LLaVA) + BERTScore for reward signal
- Reward = text-image similarity; model updated for better adherence
- Mirrors RLHF in LLMs → improves semantic alignment and quality

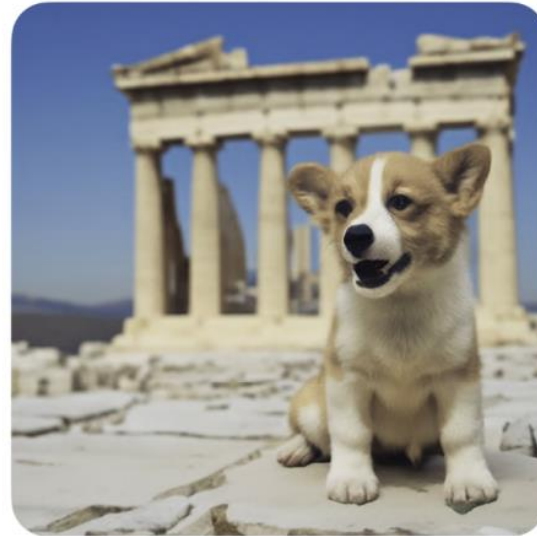


Personalizing T2I Models with New Visual Concepts

- Learn new token [V] for a concept (e.g., user's dog)
- DreamBooth → fine-tune with prior preservation to retain class diversity
- Enables generating unique subjects in arbitrary scenes



Input images



in the Acropolis



swimming



sleeping



in a doghouse



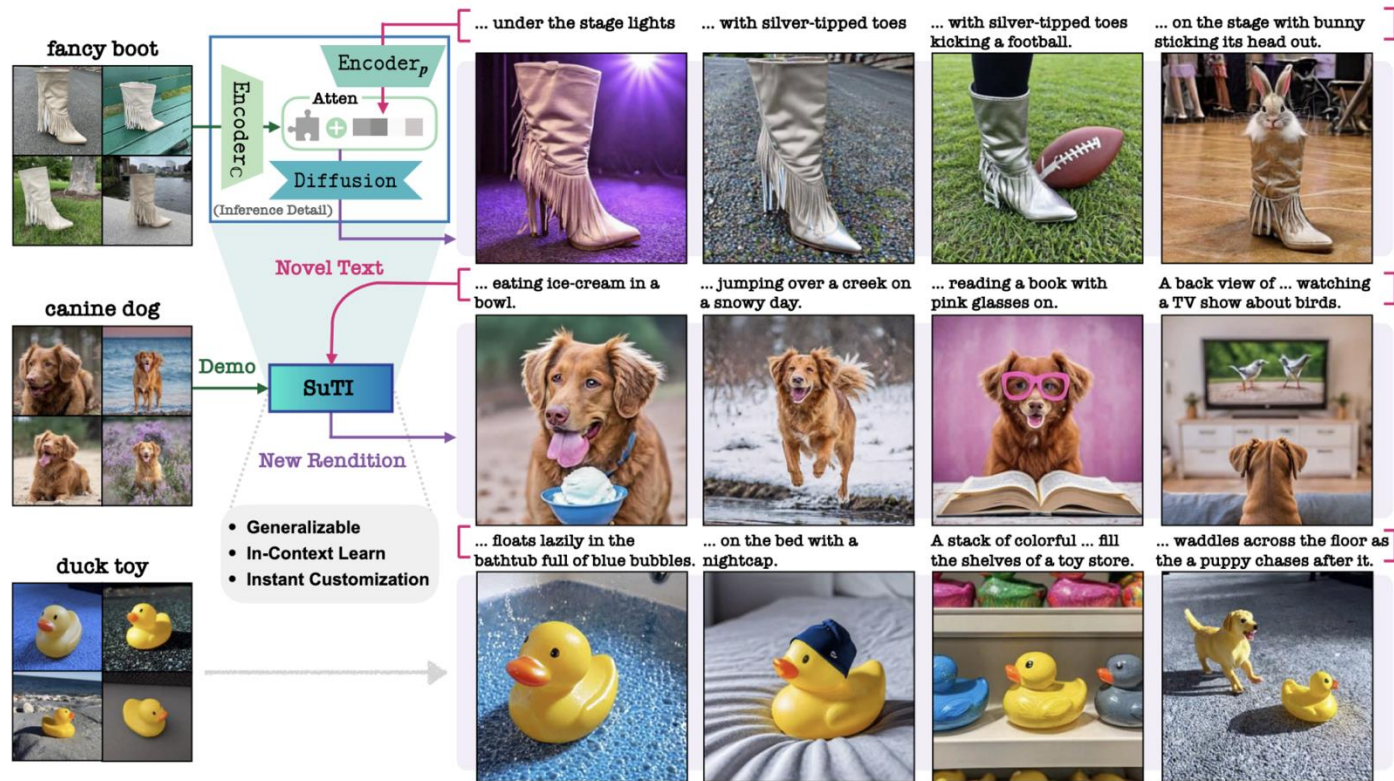
in a bucket



getting a haircut




In-Context Customization without Fine-Tuning (SuTI)

- SuTI learns from subject image + text in one unified diffusion model
- No per-concept fine-tuning; supports new subjects dynamically
- Output guided by text and demonstration images



Toward Unified Multimodal Alignment Tuning

- Unified interface combines text, image, and instruction inputs
- Alignment tuning adds aesthetic / safety / instruction / similarity rewards
- Future → closed-loop understanding + generation models (image–text in → image–text out)

Related topics	Instruction text input	Content text input	Image input
T2I models (Sec. 3.1.2; SD)	None	Image description	None
Region-controlled T2I (Sec. 3.2; ReCo)	None	Image description + Box Tokens	None
T2I with dense conditions (Sec. 3.2; ControlNet)	None	Image description	Dense conditions (segmentation, edgemap, depth, keypoints, etc.)
Text instruction editing (Sec. 3.3; InstructPix2Pix)	Editing instruction ("change the dog's color to blue")	Contents for editing instruction	Image 
Concept customization (Sec. 3.5; SuTI)	Customization instruction ("generate a dog looks like this one")	Image description	Image 
Alignment tuned T2I models (Sec. 3.6)	Arbitrary instruction ("generate a dog looks like the left one but in blue")	Image description + Box Tokens ("in <687>, <204>, <999>, <833>")	Image or dense conditions 

Key Design Principles of Unified Vision Models

•Three Core Principles of Unified Vision Models:

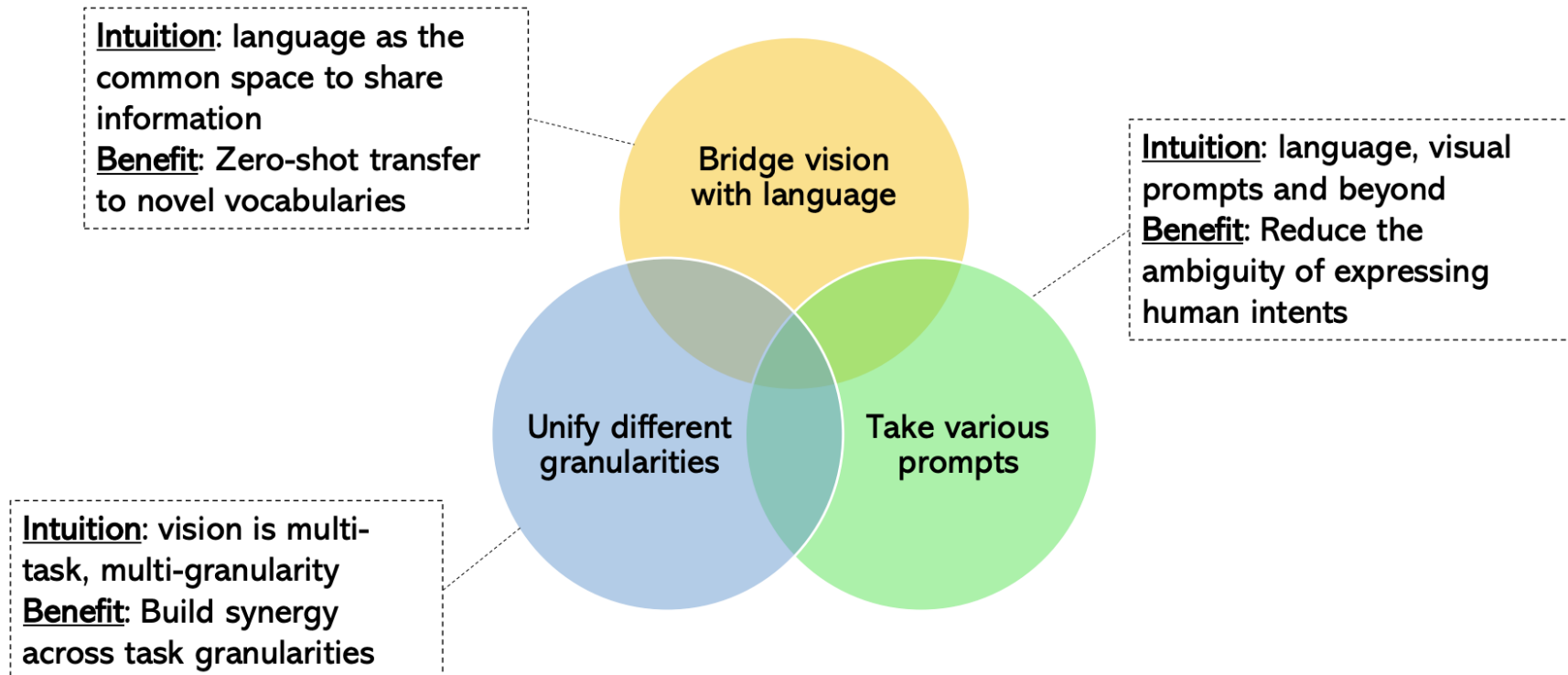
Bridge vision with language – Use language as the shared semantic space → Zero-shot transfer to new concepts.

Unify different granularities – Integrate image-level, region-level and pixel-level tasks → Cross-task synergy.

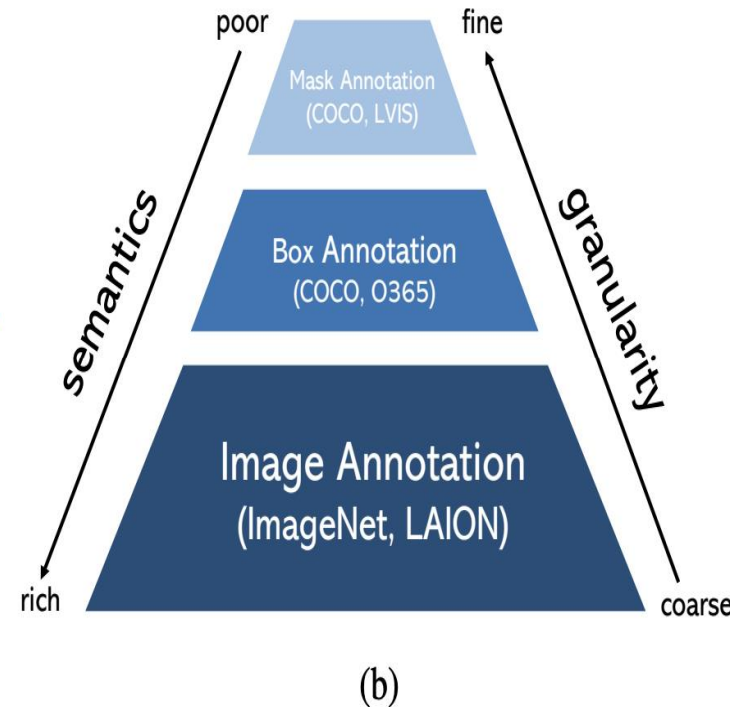
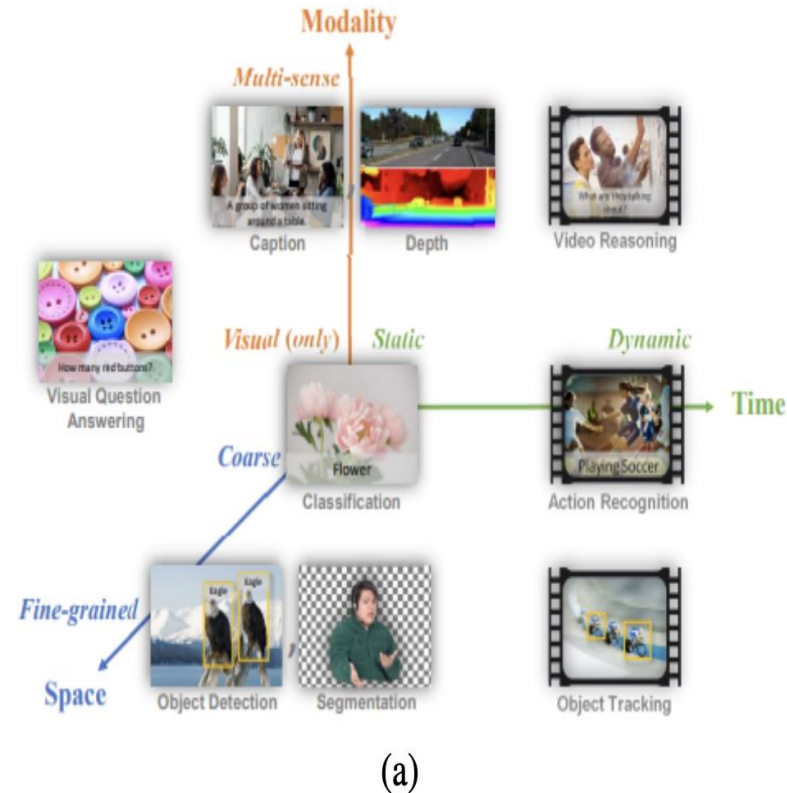
Take various prompts – Support text, visual and mixed prompts → Reduce ambiguity in human intent expression.

•Outcome:

These three axes jointly enable open-vocabulary, multi-granularity, and interactive vision models—the foundation for general-purpose



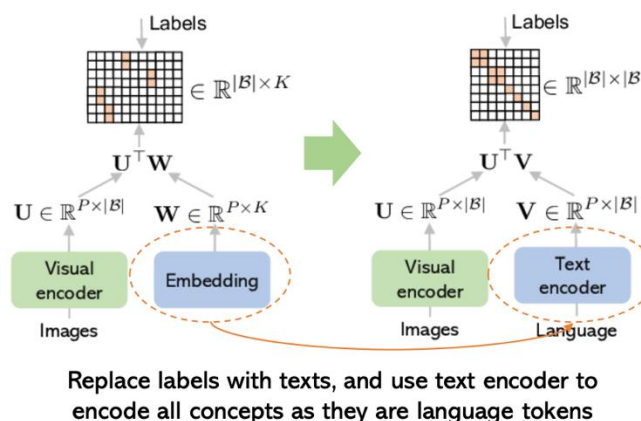
Challenges in Unifying Vision Tasks



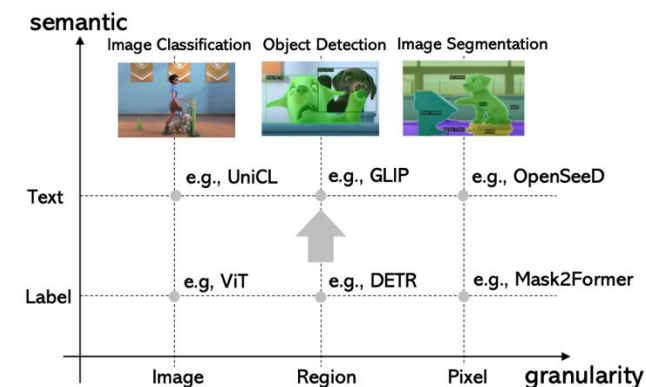
- **Why Unified Vision Models Are Difficult to Build**
- Computer vision tasks span multiple fundamental axes, making it extremely challenging to unify them under a single model:
- **1. Modality Dimension**
- Vision tasks may involve multiple modalities: images, videos, depth maps, text, etc.
- Tasks like video reasoning or depth estimation require additional signals beyond static RGB.
- **2. Spatial Granularity Differences**
- **Image-level:** classification, retrieval (coarse)
- **Region-level:** object detection (medium)
- **Pixel-level:** segmentation (fine-grained)
Each granularity requires different types of supervision (image labels, bounding boxes, masks).
- **3. Temporal Axis**
- **Static tasks:** classification, detection, segmentation
- **Dynamic tasks:** action recognition, video reasoning, object tracking
Temporal modeling introduces extra complexity compared to static images.
- **4. Supervision Varies in Semantics and Granularity**
- ImageNet / LAION: rich semantics but coarse-grained
- COCO bounding boxes: medium granularity
- COCO panoptic / LVIS masks: fine-grained but limited category coverage

Replacing Labels with Text for Unified Semantics

- **Key Idea**
- Traditional vision models learn from **label embeddings**.
- Unified models replace labels with **text descriptions**, letting a **text encoder** represent all visual concepts.
- **Why This Helps**
- Text naturally expresses rich semantics.
- Enables **zero-shot** generalization to new categories.
- Provides a **shared representation space** across image, region, and pixel tasks.
- **Semantic vs Granularity Landscape**
- Image → Region → Pixel tasks span increasing granularity.
- Models using **text supervision** (e.g., UniCL, GLIP, OpenSeeD) achieve richer semantics across all levels.



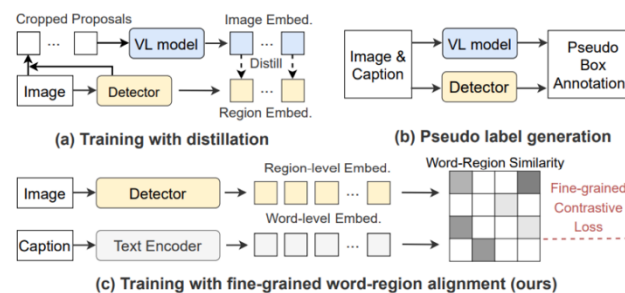
(a)



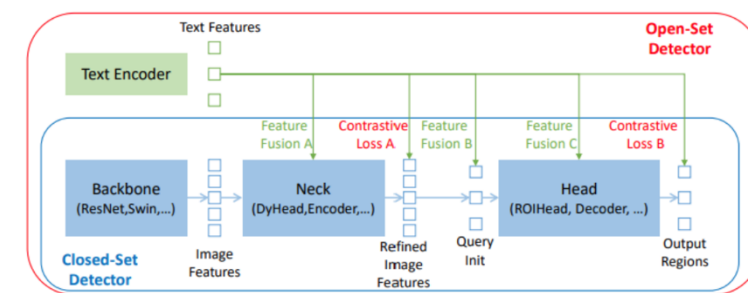
(b)

DetCLIP-v2 and Grounding-DINO

- **DetCLIPv2:** Learns fine-grained word-region alignment by combining object detection data with large-scale image-text pairs. Uses text embeddings to supervise region-level features through contrastive word-region matching.
- **Grounding-DINO:** Injects text features into multiple stages of the detector (backbone, neck, head). Contrastive losses at different levels significantly improve open-set grounding performance.



(a) DetCLIPv2



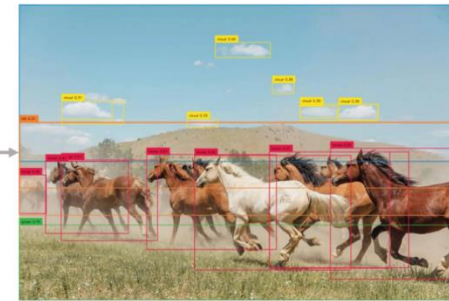
(b) Grounding-DINO

Grounding-SAM: Detect and Segment Everything

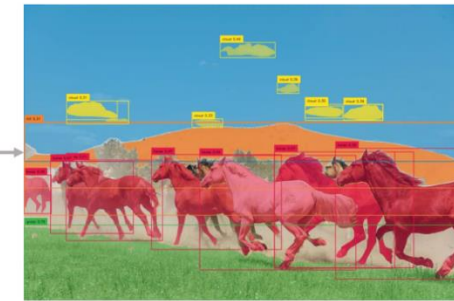
- Grounding-SAM combines Grounding-DINO and SAM to achieve open-world detection and segmentation. Given a text prompt (e.g., “Horse, Clouds, Grasses, Sky, Hill”), Grounding-DINO localizes all relevant objects, and SAM produces precise masks for each detected region. This enables fully open-vocabulary, prompt-driven detection and segmentation in complex scenes.



Text Prompt:
“Horse, Clouds, Grasses, Sky, Hill.”



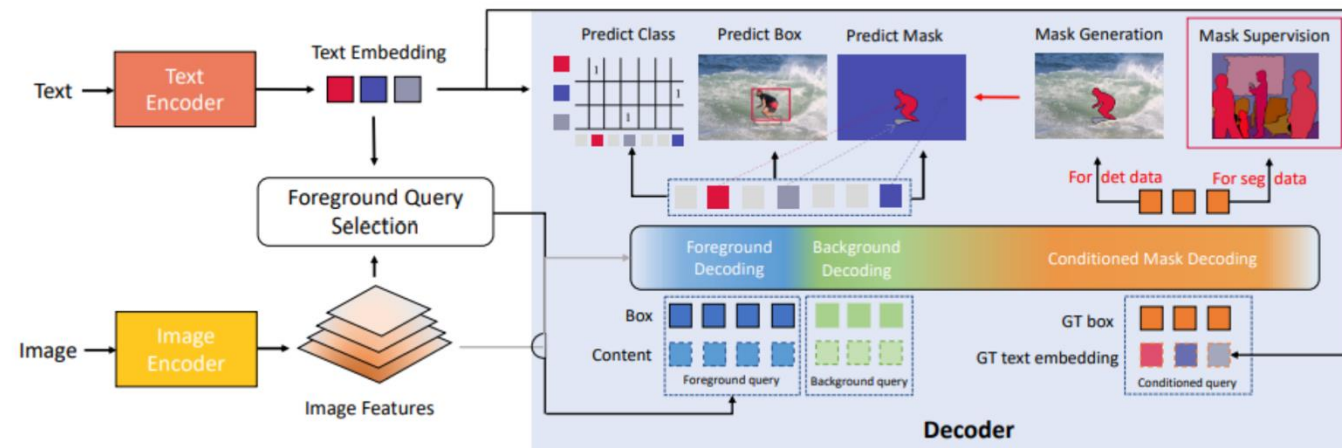
Grounding DINO:
Detect Everything



Grounded-SAM:
Detect and Segment Everything

OpenSeeD: Unified Detection and Segmentation with Text Queries

- OpenSeeD unifies open-vocabulary detection and segmentation by conditioning queries on text embeddings. The text encoder provides semantic embeddings, which guide the selection of foreground queries. The image encoder produces visual features, and the decoder predicts classes, boxes, and masks. A conditioned mask decoder allows the model to train jointly on detection data and segmentation data, enabling open-vocabulary, cross-granularity understanding.



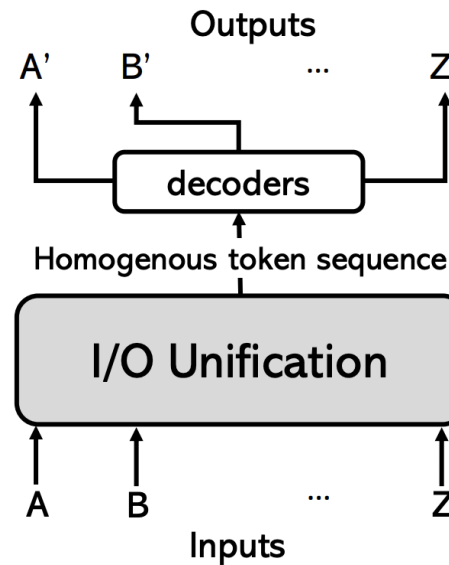
I/O Unification vs Functional Unification

•I/O Unification

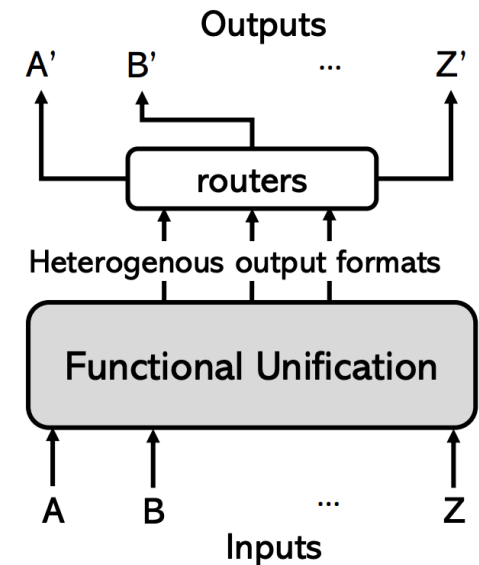
- Converts all tasks into a **homogeneous token sequence**.
- A **shared decoder** takes this unified sequence and generates outputs for different tasks.
- Core idea: unify the *format* of inputs/outputs so the model sees everything in one common space.
- Examples: Pix2Seq, Unified-IO, UniTAB.

•Functional Unification

- Maintains **heterogeneous output formats** (boxes, masks, texts, etc.).
- A **router** selects task-specific heads while sharing a unified backbone and feature space.
- Core idea: unify the *functions* the model performs while keeping specialized output structures.
- Examples: X-Decoder, GPV, GLIP-v2.

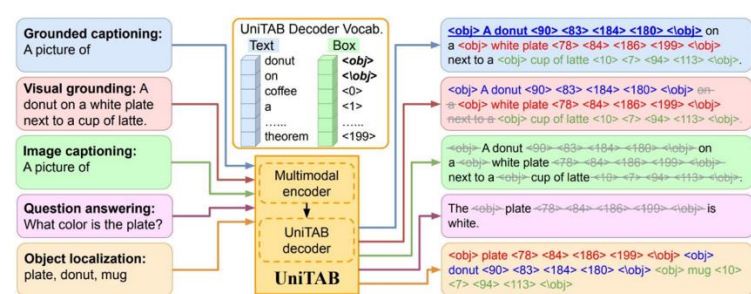


Tasks: [A, B,..., Z]

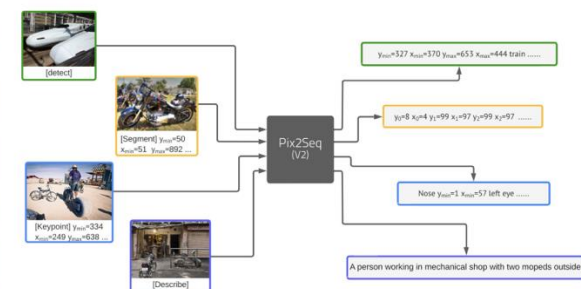


UniTab & Pix2Seq-v2: I/O Unification in Vision

- **UniTab**
- Unifies multiple tasks—grounded captioning, visual grounding, image captioning, VQA, and object localization.
- Converts all outputs (text, boxes, objects) into a **shared token vocabulary**.
- Uses a multimodal encoder + unified decoder to handle diverse instructions as sequences.
- Key idea: treat vision tasks as **token generation problems** under a unified text/box vocabulary.
- **Pix2Seq-v2**
- Extends sequence modeling to unify detection, segmentation, keypoint detection, and captioning.
- Encodes all predictions (boxes, masks, keypoints, descriptions) as **ordered sequences of tokens**.
- Enables multi-task training within one model without changing architecture.
- Key idea: represent structured outputs as **learnable sequences**, making vision tasks language-like.



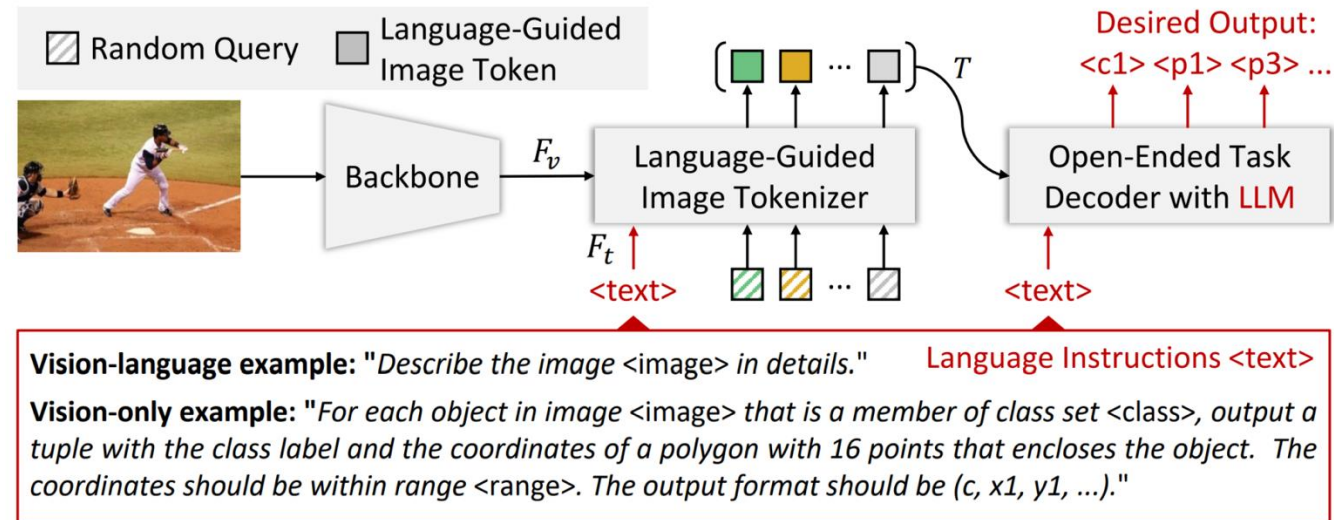
(a) UniTab



(b) Pix2Seqv2

Language-Guided Image Tokenization for Open-Ended Vision Tasks (VisionLLM)

- Uses a **backbone** to extract image features, then a **language-guided image tokenizer** converts them into semantic tokens informed by text.
- Text instructions provide **task specifications**, guiding which image regions and concepts to focus on.
- The tokenizer produces a sequence of **language-aligned visual tokens**, suitable for open-ended reasoning.
- An **LLM-based decoder** generates flexible outputs (classes, positions, polygons, captions, etc.) depending on the instruction.
- Supports both **vision-language tasks** (e.g., “Describe the image”) and **vision-only structured tasks** (e.g., object tuples with coordinates).



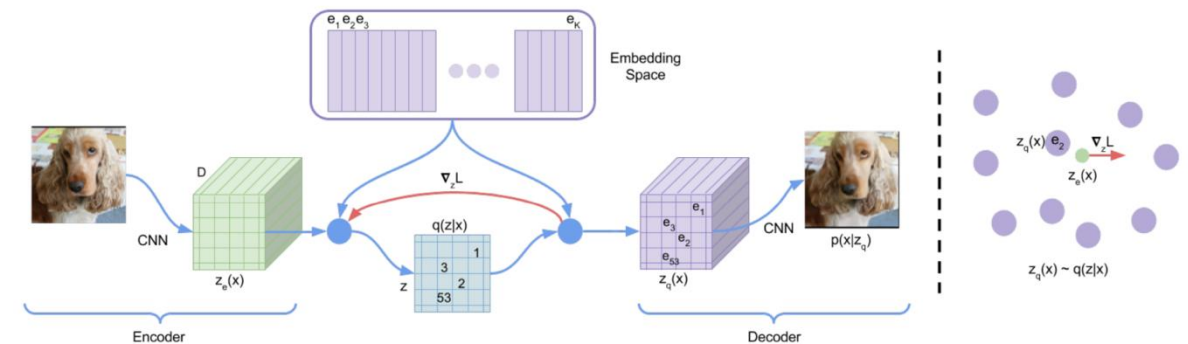
VQ-VAE and VQ-GAN: Discrete Visual Token Representation

• VQ-VAE

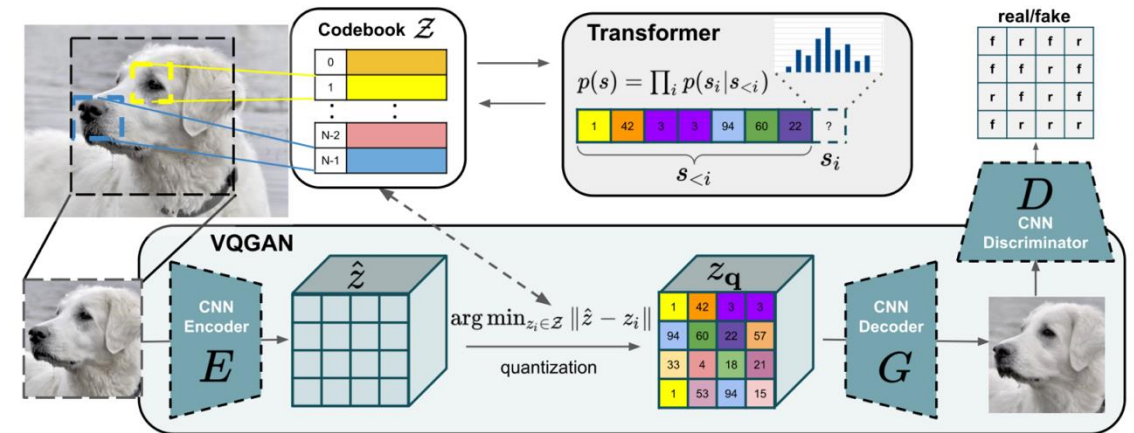
- Encodes an image into continuous features, then **quantizes** them using a discrete **codebook**.
- Each feature vector is replaced with the nearest codebook entry → produces **discrete latent tokens**.
- Decoder reconstructs the image from these tokens.
- Enables image modeling using sequence models (e.g., Transformers) over discrete tokens.

• VQ-GAN

- Extends VQ-VAE with an additional **GAN discriminator** to improve visual fidelity.
- Preserves discrete tokenization while producing **sharper, more realistic reconstructions**.
- Combines quantization + adversarial loss + perceptual loss to maintain both semantic accuracy and image detail.
- Forms the foundation of latent diffusion and modern text-to-image models.



(a) VQ-VAE



(b) VQ-GAN

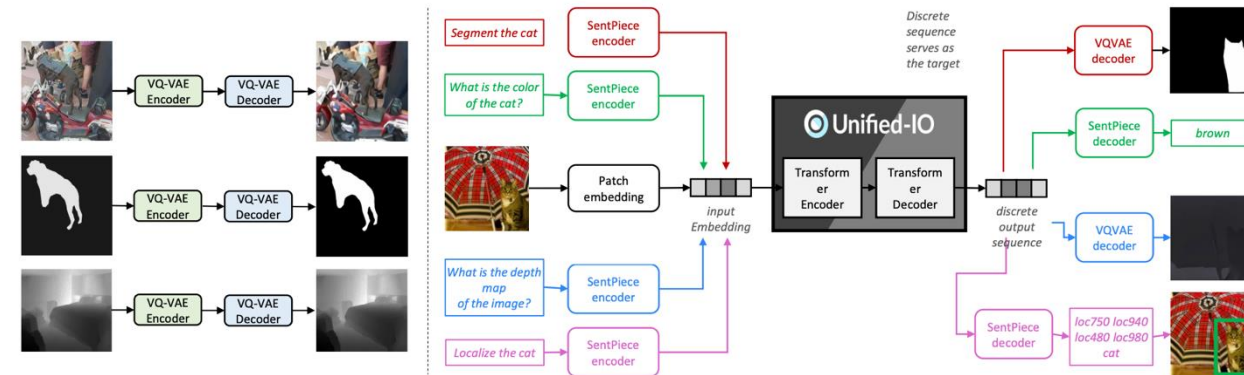
Unified-IO: A Single Model for All Vision Tasks

• Key Ideas

- Uses **VQ-VAE** to convert each task's output (segmentation mask, depth map, bounding box, caption, etc.) into a **discrete token sequence**.
- Images are encoded into patches, while **text instructions** are encoded by a SentencePiece encoder.
- All inputs—images + instructions—are combined into a **shared input embedding** for a unified Transformer encoder-decoder.
- The Transformer outputs a **discrete output sequence**, which is then decoded using either VQ-VAE decoders (for masks/depth) or text decoders (for captions/answers).

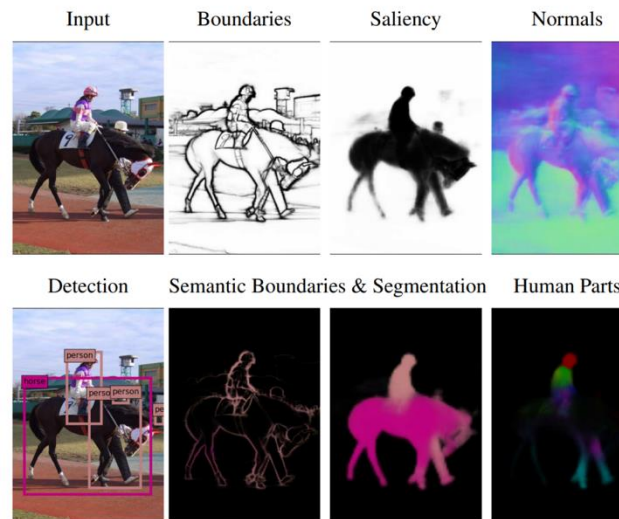
• What This Achieves

- Represents **all tasks**—detection, segmentation, depth, captioning, VQA—using **one unified token space**.
- Enables an encoder-decoder model to solve diverse tasks **without changing architecture**.
- Forms one of the earliest fully **I/O-unified vision systems**.

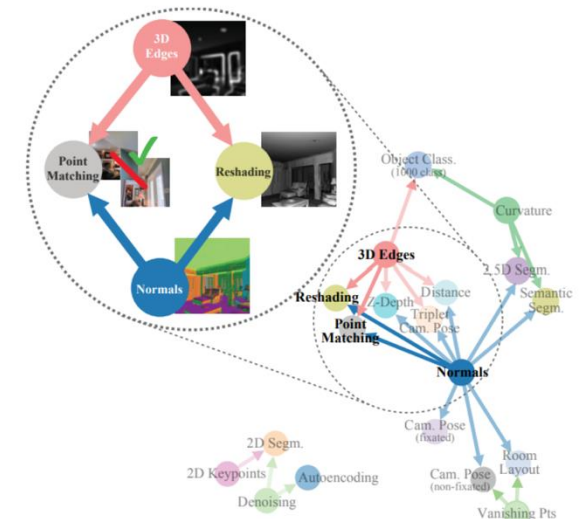


UberNet & Taskonomy: Early Multi-Task Vision Unification

- **UberNet (2017)**
- A single **budget-controllable CNN** that handles **7 diverse vision tasks** (detection, boundaries, saliency, normals, segmentation, human parts, etc.).
- Shares a **unified backbone** while attaching lightweight task-specific heads.
- Demonstrates early evidence that many vision tasks can be solved by a **shared feature extractor**.
- Focuses on **efficiency**: dynamic compute allocation depending on task budget.
- **Taskonomy (2018)**
- Studies **relationships between vision tasks** via large-scale multi-task transfer learning.
- Builds a **task affinity graph**, showing which tasks can best transfer to others.
- Provides a structured **taxonomy of tasks** indicating fundamental tasks (e.g., normals) vs. downstream tasks (e.g., semantic segmentation).
- Influential for understanding **which tasks should share features** in unified/multitask models.



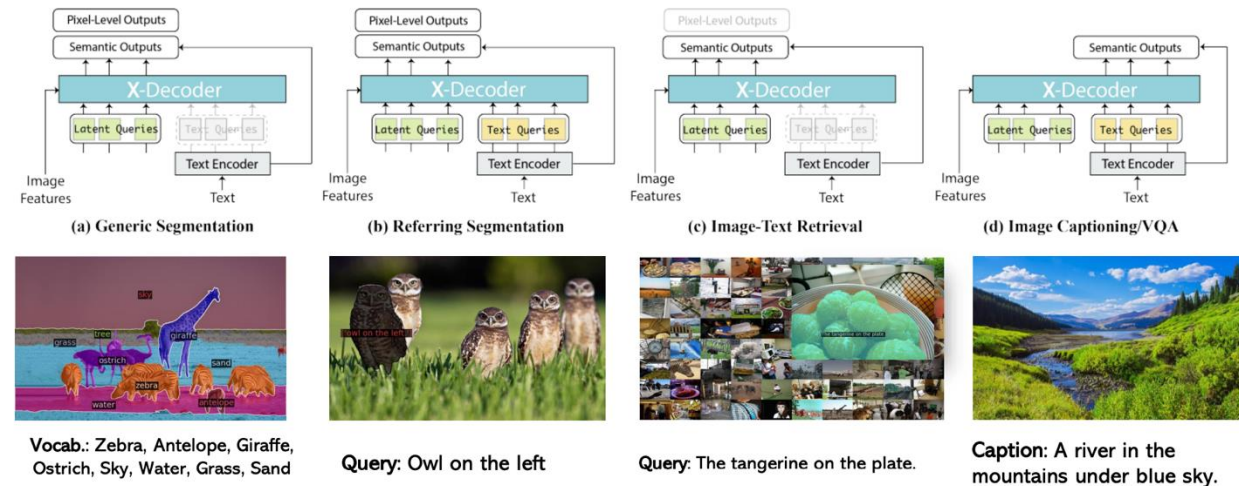
(a) UberNet



(b) Taskonomy

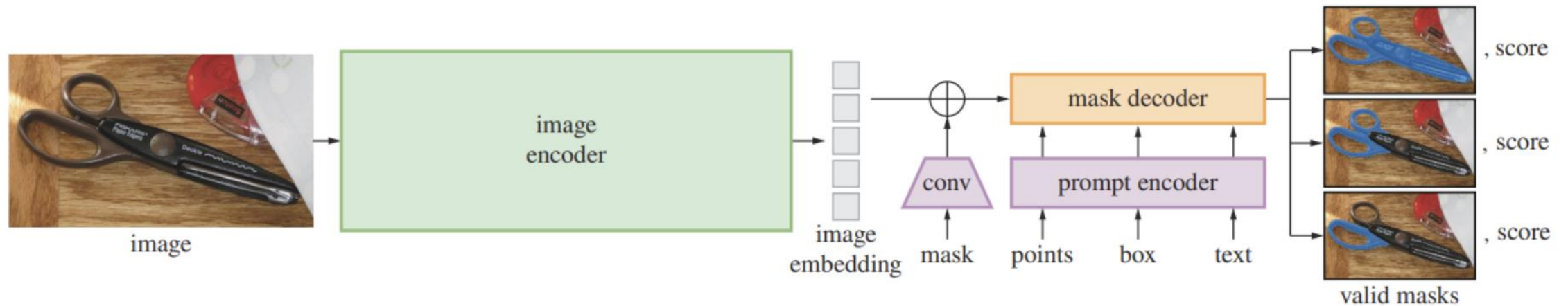
X-Decoder — A Unified Vision Decoder

- **Core Idea**
- A single decoder architecture handles a wide range of vision tasks.
- Text encoder provides task-specific guidance via text queries.
- Latent queries + text queries jointly drive pixel-level and semantic outputs.
- **How It Works**
- **Image features** extracted by backbone → fed into X-Decoder.
- **Latent Queries**: general visual tokens for universal decoding.
- **Text Queries**: optional task prompts (e.g., referring expressions, captions).
- **Shared Decoder**: produces segmentation masks, retrieval embeddings, or text outputs depending on query type.



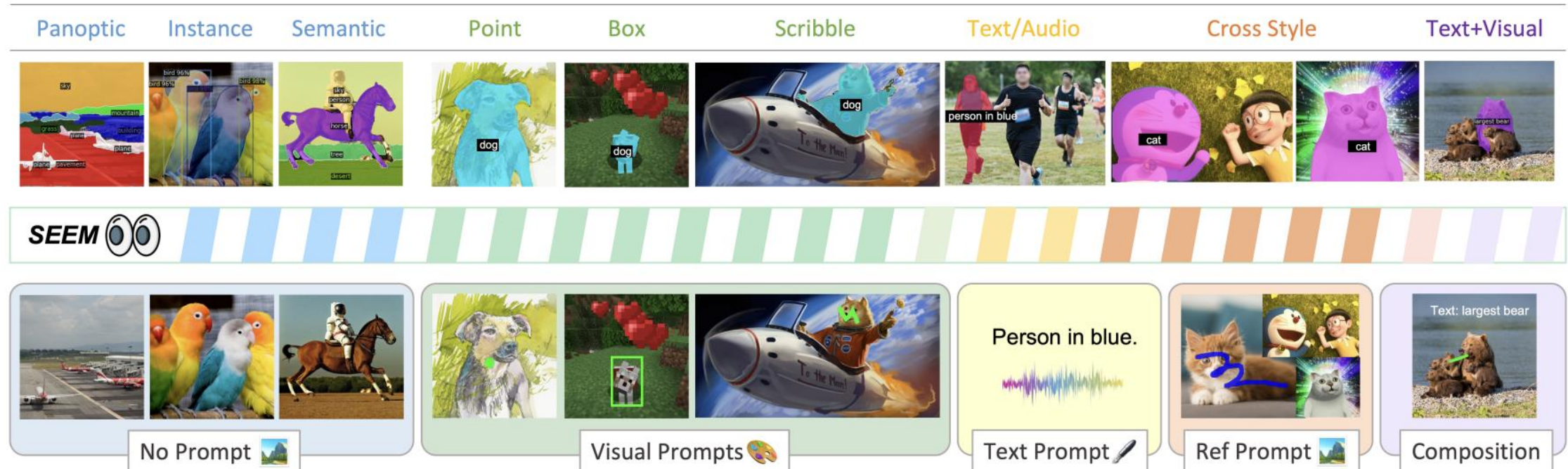
Segment Anything Model (SAM)-- Promptable

- **Promptable segmentation** model
- Supports **points, boxes, masks, text** as prompts
- **Image Encoder** extracts dense features
- **Prompt Encoder** embeds spatial/text cues
- **Mask Decoder** produces multiple candidate masks
- **Scoring module** selects the best mask
- Strong generalization, **zero-shot segmentation**
- Includes a **large-scale auto-labeling engine** for massive mask data



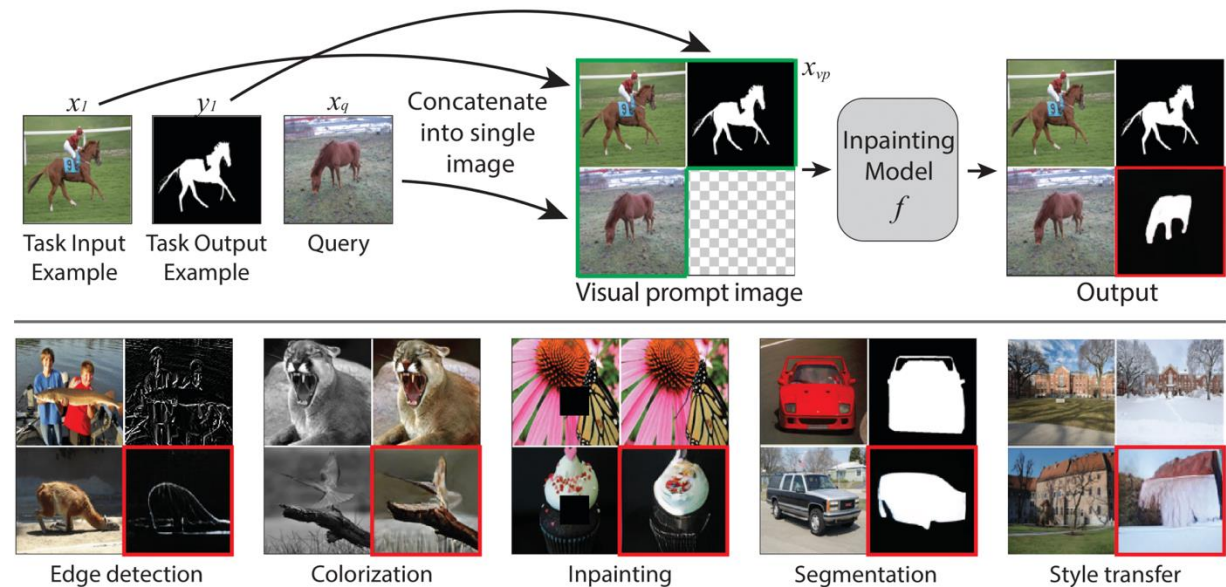
SEEM -- Universal Promptable Segmentation

- Handles **many segmentation types**: panoptic, instance, semantic, etc.
- Supports **multiple prompt modalities**:
 - **Visual**: points, boxes, scribbles, reference images
 - **Language**: text, audio
- **Cross-style + Text+Visual** prompts
- Unified model that generalizes across **free-form prompts**
- Works on both **real images** and **stylized / synthetic** content
- Enables composition of prompts for flexible segmentation tasks



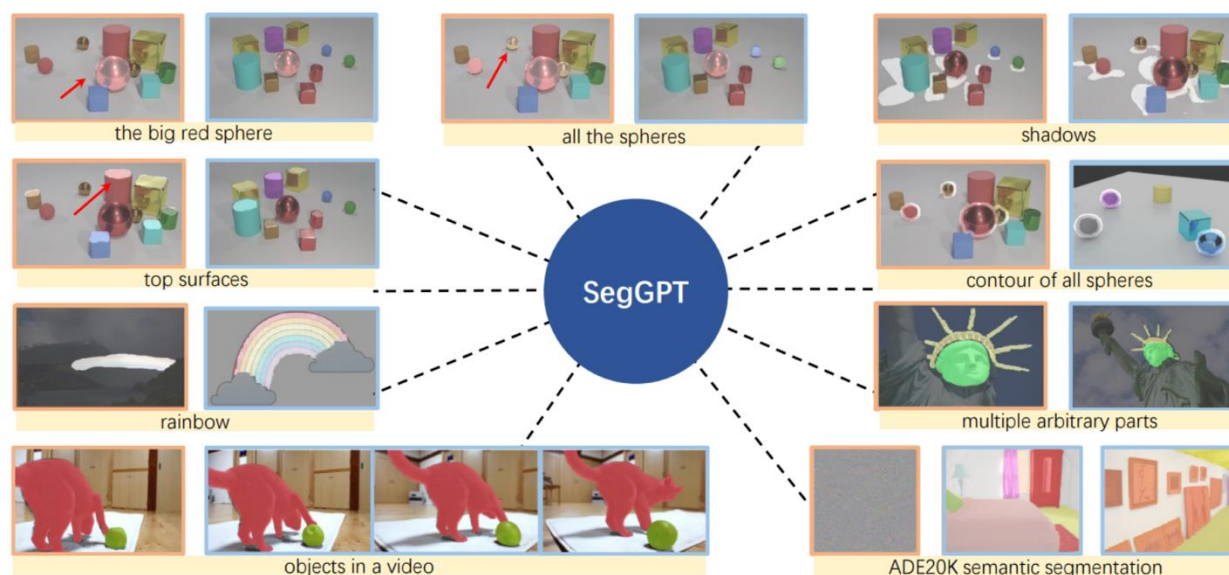
Visual Prompting via Inpainting

- Uses **visual examples** instead of text to specify a task.
 - Build a **visual prompt image** by concatenating:
 - Task input example
 - Task output example
 - Query image
 - Empty slot to be predicted
 - Feed the visual prompt into an **inpainting model** to infer the missing output.
 - A *single* model can perform many tasks by changing the visual prompt:
- **Edge detection**
 - **Colorization**
 - **Inpainting**
 - **Segmentation**
 - **Style transfer**



SegGPT — In-Context Segmentation

- SegGPT performs **in-context learning** for segmentation.
- The model takes **example image-mask pairs** and generalizes to new queries.
- Supports a wide range of segmentation tasks without task-specific training:
- Segment a **specific object** (e.g., the big red sphere)
- Segment **all objects of a category** (all spheres)
- Extract **shadows**, **top surfaces**, **contours**, etc.
- Segment **arbitrary parts** (e.g., Statue of Liberty face)
- Segment in **videos** (object tracking + segmentation)
- Perform **semantic segmentation** (e.g., ADE20K)
- Handle **unusual or creative prompts** (rainbow, abstract parts)
- Key idea:
Give segmentation examples → model infers the intended rule → applies it to new images.



Hummingbird — In-Context Scene Understanding

- A retrieval-based in-context segmentation model
- Core idea: **use nearest-neighbor examples to guide prediction**
- Pipeline:
 - **Input prompt images + labels** provide segmentation examples
 - For each query image, retrieve **nearest prompt examples**
 - **Aggregate retrieved labels** based on visual similarity
 - Produce segmentation for the query (no model fine-tuning needed)
- Supports open-set and varied object categories through retrieval
- Emphasizes **non-parametric generalization** via example-based reasoning
-



Large Multimodal Models (Training with LLM)

- Goal: enable vision–language models to act like **GPT-4-style assistants**
- Combine *visual perception* + *language reasoning* + *instruction alignment*
- Pipeline → image encoder + LLM decoder + cross-modal attention

A dog lying on the grass next to a frisbee



Language

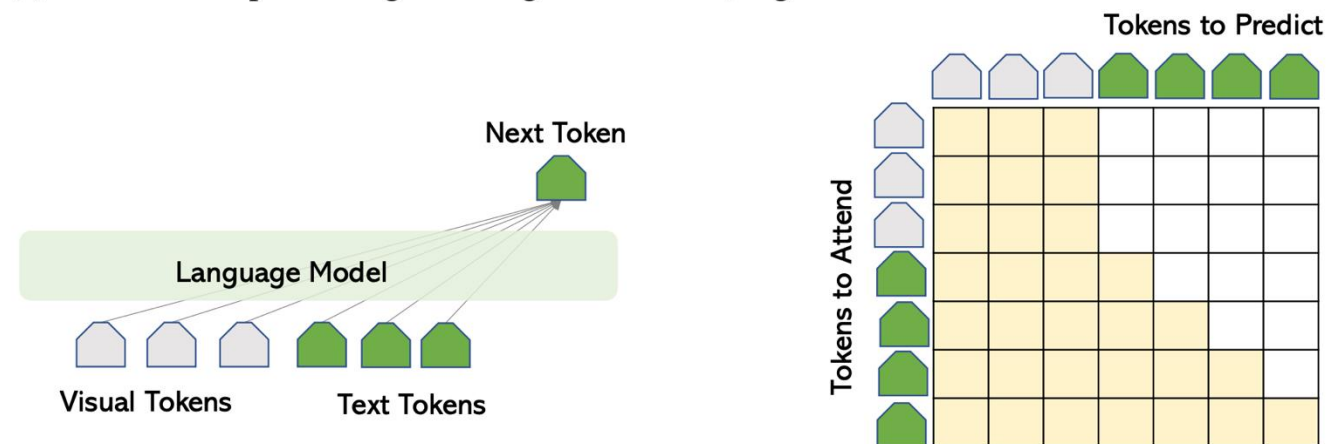
Image

Language Model

Connection Module

Vision Encoder

(a) Left: An example of image-to-text generation task; Right: Model architecture.

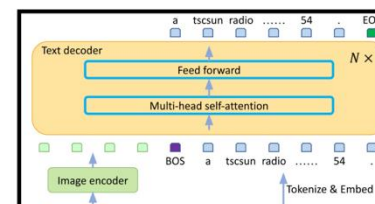


(b) Training objective and attention mask. For each row, the yellow elements indicate that the prediction token attends the tokens on the left.

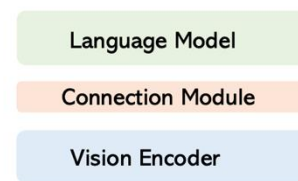
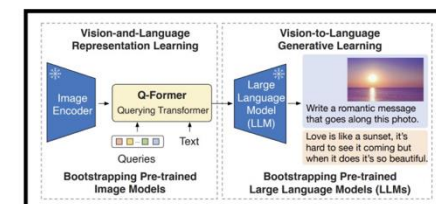
Core Architecture & Key Models

- **Flamingo (2022)**: Frozen LLM + visual adapters for image–text reasoning
- **BLIP-2 (2023)**: Q-Former bridges CLIP encoder and LLM
- **Kosmos-1 / GIT-2**: Fully end-to-end multimodal transformers
- Unified goal: learn image-to-text generation and visual dialogue in one framework

• GIT



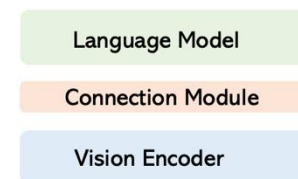
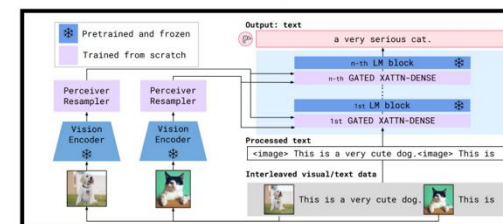
• BLIP2



From Scratch	Pre-trained: FLAN-T5/OPT
	Q-Former: Lightweight Querying Transformer
Contrastive pre-trained: Florence/CLIP	Contrastive pre-trained: EVA/CLIP

(a) Example 1: LMM trained with image-text pairs.

• Flamingo

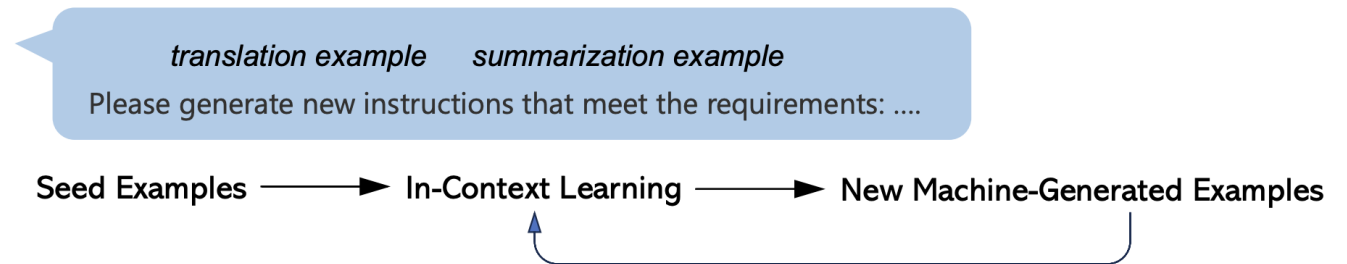


Pre-trained: 70B Chinchilla
Perceiver Resampler Gated Cross-attention + Dense
Pre-trained: Nonnormalizer-Free ResNet (NFNet)

(b) Example 2: LMM trained with image-text pairs and interleaved image-text data.

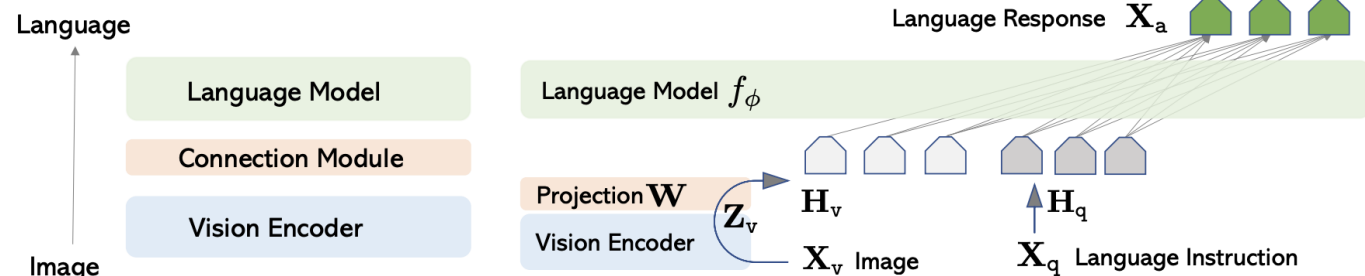
Instruction Tuning for Multimodal Alignment

- Instruction tuning → train models to follow natural language commands
- **InstructGPT / ChatGPT / Vicuna / Alpaca** → **Self-Instruct** generate training pairs
- Extended to multimodality via image–question–answer triples
- Result: LLM learns to “see before answering,” enabling visual instruction



Instruction-Tuned Large Multimodal Models

- **LLaVA**: CLIP encoder + Vicuna LLM
→ visual conversation
- **MiniGPT-4 / mPLUG-Owl / Otter**:
lightweight adapters for efficient training
- **Applications**: VQA, captioning, OCR, medical AI, reasoning dialogue
- **Training strategy**: freeze LLM, align vision features via small projection layers



Challenges & Future Directions

- Still behind **GPT-4** in reasoning and data scale
- **Main gaps:**
 - Limited high-quality multimodal instruction data
 - High compute for end-to-end optimization
 - Inconsistent multi-turn visual dialogue
- **Future:**
 - Open multimodal datasets & efficient training
 - Stronger visual–language fusion modules
 - Better alignment and safety tuning

