



INTERACTION WITH LARGE SCALE MODELS

LIACD/1

University of Beira Interior,
Department of Informatics

Hugo Pedro Proença,
hugomcp@di.ubi.pt, 2024/2025



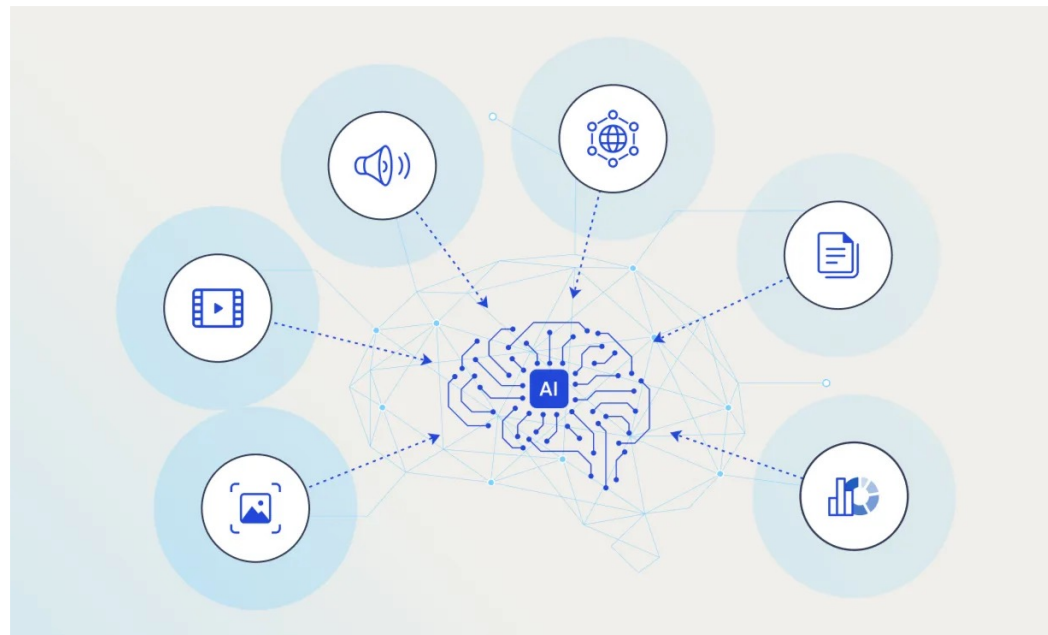
INTERACTION WITH LARGE SCALE MODELS

[07]

- Introduction to multimodal AI (text-to-image, text-to-video, etc.).
- - Designing prompts for text-to-image generation
- - Challenges and limitations of multimodal prompting.

Multimodal AI

- **Multimodal AI** refers to artificial intelligence systems capable of processing and integrating information from multiple modalities—such as text, images, audio, video, and structured data—to perform tasks like classification, generation, reasoning, and interaction.
- In essence, multimodal models learn shared representations across different data types, enabling richer understanding and more human-like capabilities.



Source: <https://www.appen.com/multi-modal-ai/>

Multimodal AI

- Multimodal AI utilizes deep learning architectures to process different types of data simultaneously.
- Unlike traditional AI which operates in only one modality, such as a text-based chatbot, multimodal models merge diverse inputs such as audio, visual, and text data to generate unique outputs.
 - For example, to take meeting minutes, an AI model must process video and audio input then generate a text summary, effectively operating across three different modalities

Text to Image

Training large language models to generate images based on text descriptions.

Image to Text

Leverage AI computer vision technology to produce textual descriptions from images.

Video to Text

Automatically summarizing video content by analyzing both visual and auditory data.

Text to Audio

Produce natural speech, real-world sounds, and music from text inputs.

Text to Video

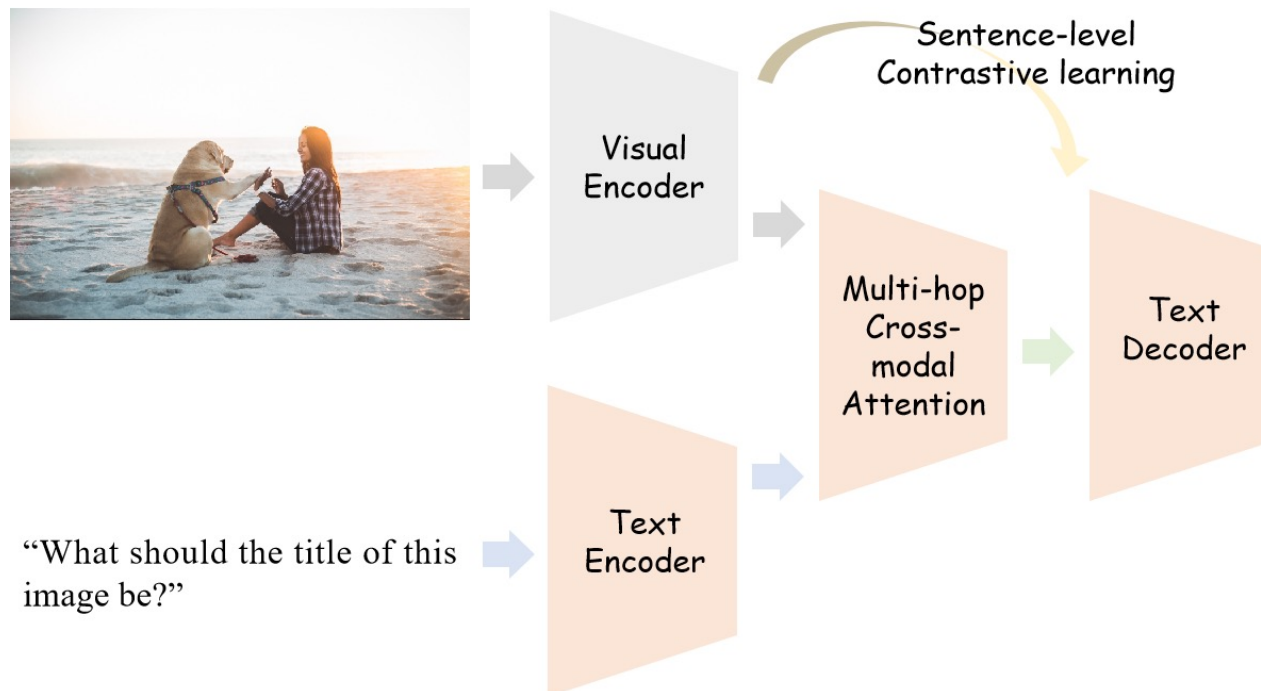
Generating video content from text inputs or prompts, a key feature in modern generative AI applications.

Multimodal Training Data (Ground Truth)

- Text descriptions of multimodal data
 - Used for the development of multimodal AI by helping models to process real-world images, videos, and more. Audio data, for example, can be transcribed to capture the exact speech content.
 - Taking annotation a step further, by providing the model with text descriptions that summarize the speech content, trains the model on narrative structure. Text descriptions encourage relational understanding and train models to recognize increasingly complex patterns in multimodal data.
- Prompts for given multimodal inputs
 - With LLMs growing in popularity, people are interacting with visual data in increasingly complex ways. Queries about an image range from simple questions like 'What species of bird is this?', to more complex knowledge-based queries that require additional logic such as 'What meals can I cook with these ingredients?' Creating prompts, or prompt-response pairs, linked with input data can improve the model's natural language processing and alignment to human end-users.

Multimodal Training Data (Ground Truth)

- COCO Multi-Modal Reasoning dataset (**COCO-MMR**) is a dataset introduced in a paper titled "*Enhancing Human-like Multi-Modal Reasoning: A New Challenging Dataset and Comprehensive Framework*".
- It aims to facilitate cutting-edge research in the field of multi-modal reasoning, particularly focusing on open-ended questions.
 - Encompasses a wide array of open-ended questions, rationales, and answers derived from the COCO.



Source: <https://github.com/weijingxuan/COCO-MMR?tab=readme-ov-file>

What is COCO?



COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- ✓ Object segmentation
- ✓ Recognition in context
- ✓ Superpixel stuff segmentation
- ✓ 330K images (>200K labeled)
- ✓ 1.5 million object instances
- ✓ 80 object categories
- ✓ 91 stuff categories
- ✓ 5 captions per image
- ✓ 250,000 people with keypoints

Source: <https://cocodataset.org/#home>

Multimodal Representation Learning

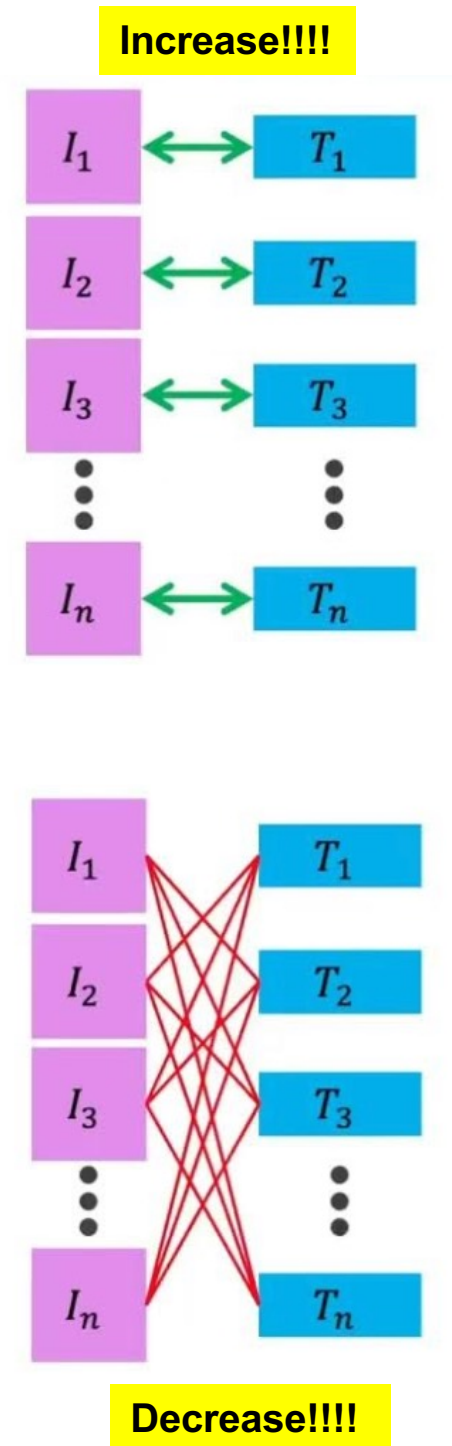
- **Goal:** Find a **common embedding space** or **cross-modal alignment** function.
 - Joint representation: Map all modalities to a shared latent space (e.g., CLIP).
 - Coordinated representation: Maintain modality-specific features but align them via shared objectives (e.g., cross-modal contrastive loss).
 - Modality fusion: Combine features from different modalities into a single representation (e.g., concatenation, attention).
- **Challenges:**
 - Modality alignment: Temporal or semantic synchronization between modalities.
 - Modality heterogeneity: Differences in structure, noise, and sampling rates.
 - Missing modalities: Incomplete input scenarios (e.g., text only at test time).
- **Architectures:**
 - Early Fusion: Merge raw features early (e.g., concatenate text and image features).
 - Late Fusion: Merge decisions or high-level representations.
 - Hybrid Fusion: Combine early and late fusion (e.g., transformers with modality-specific encoders and a shared cross-modal decoder).

Multimodal Representation Learning

Approach	Description	Examples
Contrastive Learning	Learn embeddings such that matching (image, text) pairs are close, non-matching are distant.	CLIP, ALIGN
Transformer-based Fusion	Use multi-head self-attention and cross-attention to fuse modalities.	Flamingo, Gato
Diffusion Models	Generative models for image synthesis conditioned on text.	Stable Diffusion, Imagen
Cross-modal Retrieval	Given a query in one modality (e.g., text), retrieve relevant items in another (e.g., images).	CLIP, BLIP
Masked Modeling	Predict masked parts of one modality given others.	BEiT-3, Florence

CLIP

- Developed by OpenAI, stands for Contrastive Language – Image Pretraining.
- It learns a shared embedding space for text and images using a contrastive learning objective.
- Once trained, it enables zero-shot classification, retrieval, and acts as a general image-text matching model.
- It is based in a Dual Encoder Architecture:
 - **Image Encoder:** Vision Transformer (ViT) or ResNet variant
 - **Text Encoder:** Transformer-based (similar to GPT-2)
- Both encode their input into fixed-size vectors:
 - Image $\rightarrow f(I)$
 - Text $\rightarrow g(T)$



CLIP

CLIP is trained on **N** image–text pairs in a batch.

- For each pair, the goal is to **maximize the cosine similarity** between the matching image and text embeddings, and **minimize it for non-matching pairs**.
- Trained on 400 million (image, text) pairs collected from the internet.
- Learns general semantic alignment between text and image—no task-specific tuning needed.

Image → Text

Text → Image

$$\mathcal{L} = \frac{1}{2N} \sum_{i=1}^N \text{CE}(\text{sim}(f(I_i), g(T_j))) + \text{CE}(\text{sim}(g(T_i), f(I_j)))$$

Each term of the loss **adds positive weight** to the total loss if:

- The similarity of the correct pair is **not higher** than the maximum of the incorrect ones → pushes the model to fix that.

- Performs zero-shot classification at or near the level of fine-tuned models.
 - Provide class names as prompts: e.g., “a photo of a {class}”
- Facilitates prompt engineering for vision: you can steer CLIP by choosing the right textual description.
 - Rank text captions for a given image or vice versa

DALL-E

- DALL-E is a family of generative models by OpenAI that can synthesize images from text prompts, trained via generative pretraining techniques.
- The first version used variational auto-encoders to convert images into sequences of discrete tokens.
- Next text + image tokens were concatenated and the task was to maximize the probability of predicting correctly the next token, according to an autoregressive loss.
- The, the second version, split the problem into two stages:
- Stage 1: Text → Image Embedding (CLIP Prior)
 - Text is embedded using a CLIP text encoder.
 - A prior model (autoregressive or diffusion) generates a CLIP image embedding conditioned on the text embedding.
 - This generates an image idea, not the image itself.
- Stage 2: Image Embedding → Image (Decoder)
 - A diffusion decoder generates an image from the image embedding.
 - Trained to map CLIP image embeddings to high-resolution images.
- CLIP already aligns images and text semantically.
- Diffusion decoders can generate higher fidelity images than discrete token generators.
- Modular: each component can be pretrained separately.

Multimodal Prompts

- **Prompt design changes** from text-only to text-to-image, text-to-audio, text-to-video, or image+text→text tasks.

Feature / Aspect	Text-Only Prompting	Multimodal Prompting
Modalities involved	Text input → Text output	Text + image/audio/video → Image/Text/Video
Prompt function	Specify task, style, or behavior	Also acts as semantic control signal for generation
Token structure	Natural language tokens	Natural language + latent features/visual tokens
Context resolution	Typically linear (text sequence)	Requires cross-modal fusion & alignment
Prompt ambiguity	Easier to resolve with language cues	Ambiguity increases without grounding (e.g., spatial concepts)
Length constraints	Truncated to a fixed token count	Text usually much shorter; can heavily affect visual output
Evaluation	BLEU, ROUGE, etc.	FID, CLIP-Similarity, Human Preference

Multimodal Prompts

1. Semantic Precision

- Unlike text-only tasks, where the model “completes” language,
- In multimodal generation, prompts must be **semantically tight**:
 - *"A highly detailed oil painting of a fox sitting under a cherry blossom tree at sunset"* is much richer than *"a fox in a forest"*

2. Visual Grammar Matters

- Keywords like “photorealistic,” “cinematic lighting,” “isometric view,” or “8k ultra detail” act like **modifiers** for model behavior.
- These tokens aren’t syntactic but **learned style cues** from training.

3. Order of Words is Critical

- *"A cat wearing a red hat"* ≠ *"A red hat wearing a cat"*
- For text-to-image models like **Stable Diffusion**, the **tokenization order** can significantly affect interpretation.

Multimodal Prompts

4. Underspecified Prompts Can Collapse Output

- Prompt: *“A man and a woman in a room.”* → Model may not know:
 - Where are they?
 - What are they doing?
 - What style?
- Fix: *“A digital painting of a man sitting at a wooden desk, a woman standing behind him, both in 19th-century attire.”*

5. Bias Amplification

- Ambiguous prompts (e.g., “a doctor” or “a CEO”) may default to **biased stereotypes**.
- Prompt design must often **proactively steer away from bias**:
 - e.g., *“a female Black doctor in a modern hospital”*

Multimodal Prompts

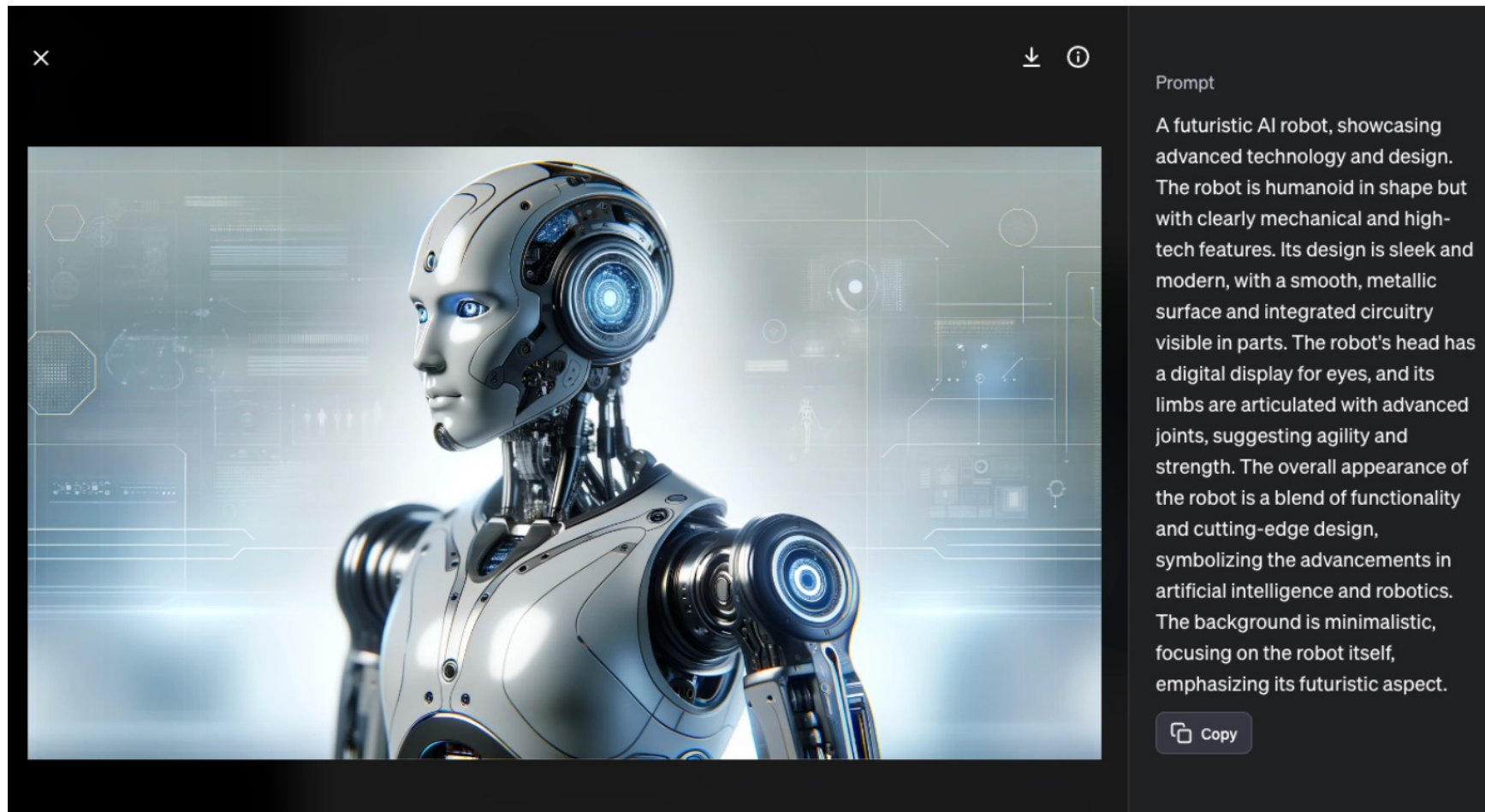
- There are, however, some strategies that can be used as general directions for multimodal prompting.

Strategy	Description	Example Prompt
Explicit Visual Framing	Set scene layout, composition, or position of objects	"A fox in the foreground, mountains in the background"
Stylistic Control	Add modifiers like "realistic", "watercolor", "isometric", "low-poly"	"A cyberpunk cityscape in pixel art style"
Perspective Anchoring	Define the camera angle or zoom level	"Close-up portrait of a smiling child, shot with a 50mm lens"
Lighting and Mood	Lighting affects model output strongly	"Candle-lit room with soft shadows"
Time & Era Anchoring	Mention time of day, era, or artistic period	"A Renaissance-style oil painting of a knight in battle"
Object Count Control	Be explicit if you want multiple objects	"Three identical robots standing side by side"

Multimodal Prompts

- **Text-to-Image (e.g., DALL-E, SDXL)**

“A [style] [subject] [action/pose] in [setting], [lighting], [camera type], [resolution]”



Multimodal Prompts


- **Text-to-Video (e.g., Sora, Runway)**
- Similar to text-to-image, but must imply **temporal evolution**.
- Use verbs and transitions:
 - *“Close-up shot zooming in on the hands and tools, capturing wood shavings and focused lighting”*



Source: <https://blog.segmind.com/how-to-write-ai-video-prompts-with-examples/>

Multimodal Prompts

- **Image+Text → Text (e.g., BLIP-2, Flamingo)**
- Strategy: Combine **image context** with a **specific task cue**.
- Example Prompt:
 - Image: [uploaded image of a Pizza]
 - Text: prompt describing the task that I am interested in.



What are the ingredients I need to make this?

Pepperoni, mozzarella cheese, pizza sauce, olive oil, salt, pepper, basil.

What is the first step?

Place the pizza dough on a baking sheet, brush with olive oil, sprinkle with salt, pepper, and basil.

Source: <https://github.com/salesforce/LAVIS/issues/97>