.UBAfiuba

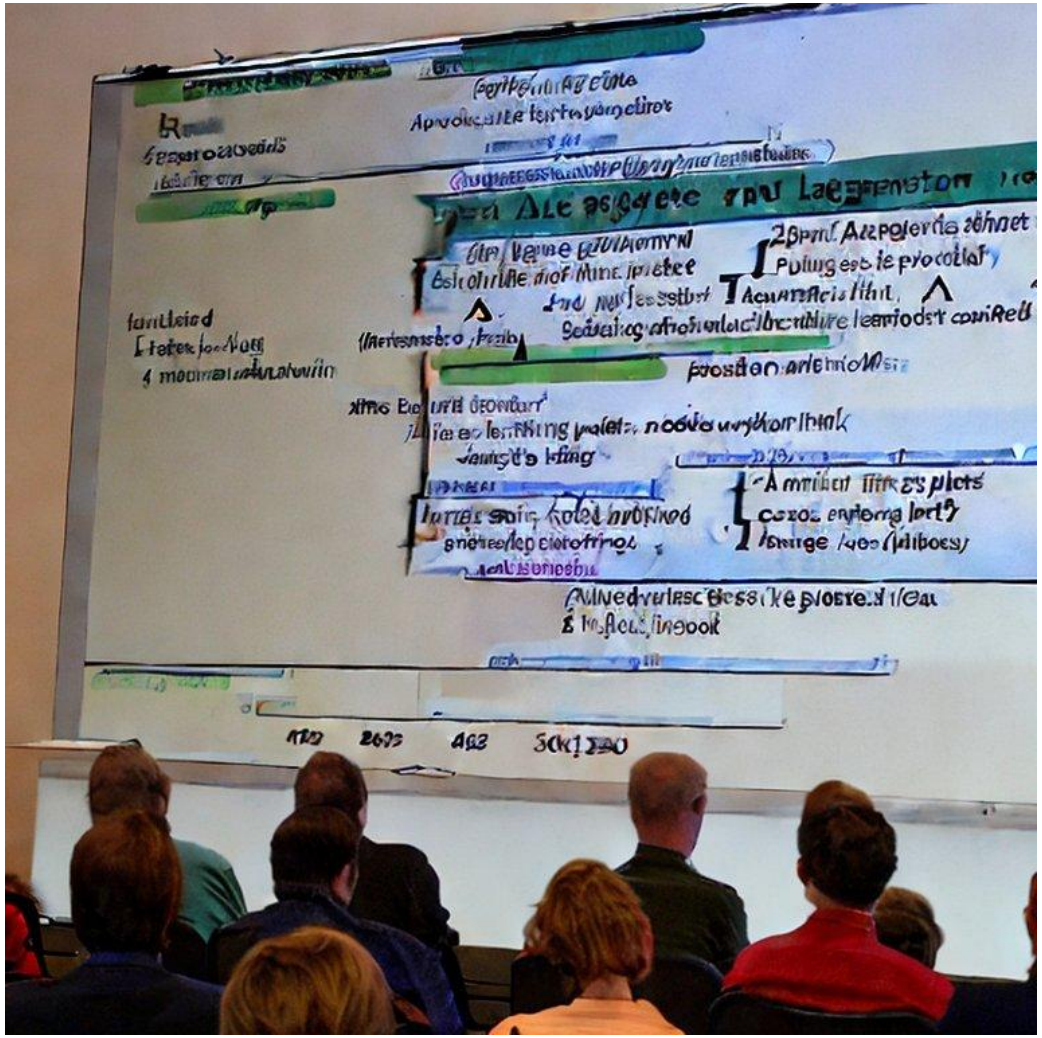
FACULTAD DE INGENIERÍA

LLMs e IA Generativa

Clase 8

Optimización de LLMs
LLMs de Razonamiento
GenIA multimodal

Mg. Ing. Ezequiel Guinsburg
ezequiel.guinsburg@gmail.com



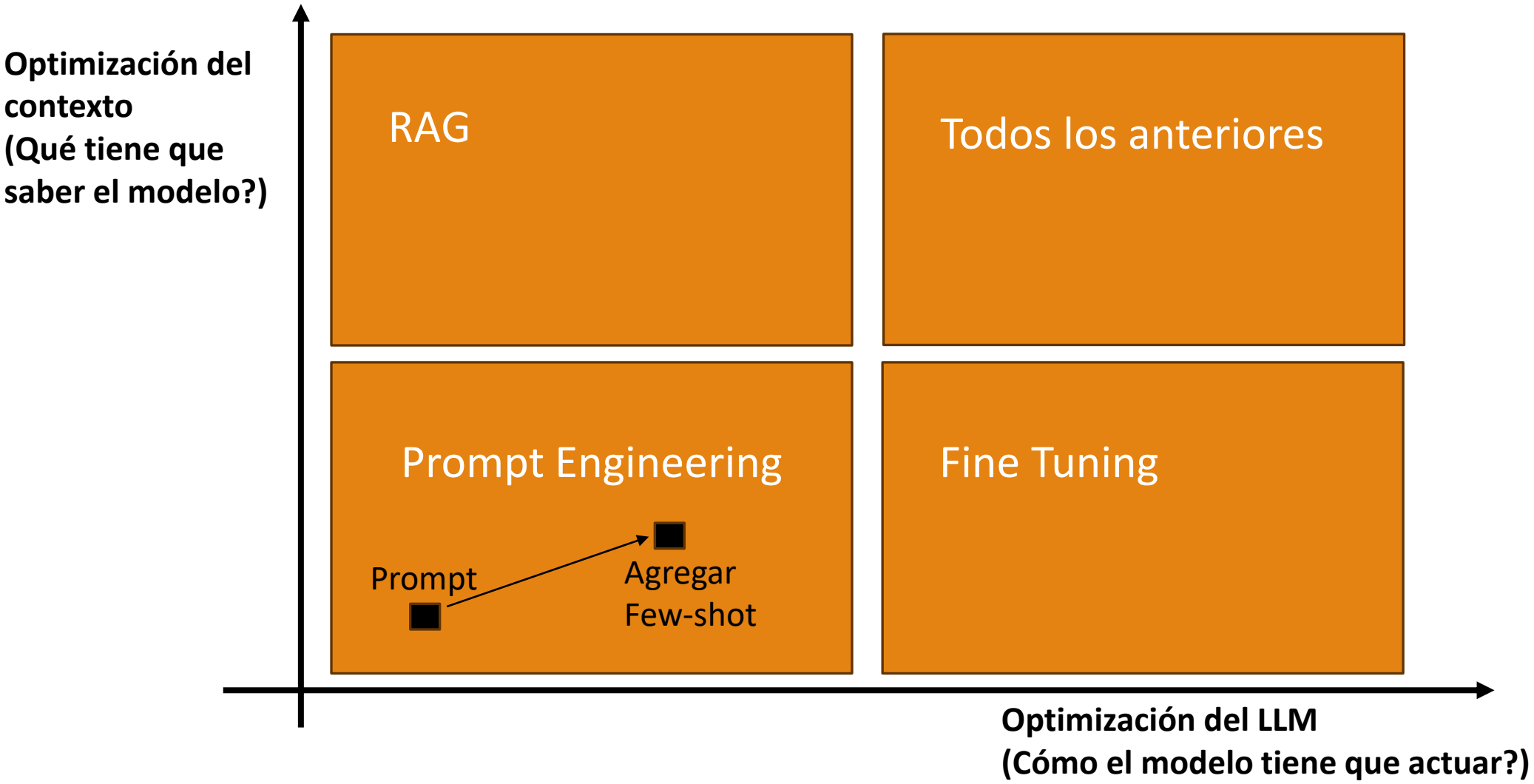
Referencias:

- Paper Reasoning With LLMs, a Survey
- Paper Visual Instruction Tuning - Haotian Liu
- Paper Hicherical Text-Conditional Image Generation with CLIP Latents - https://arxiv.org/pdf/2204.06125
- Denoising Diffusion Probabilistic Models - https://arxiv.org/pdf/2006.11239

Temas:

- El camino de la optimización en el uso de los LLMs.
- LLMs de razonamiento.
- I.A. generativa multimodal.
  - Ejemplo teórico de modelo de texto-imagen.
  - Ejemplo práctico.
- Ejercicio 3.

Prompting Vs. RAG Vs. Fine tuning



Prompting Vs. RAG Vs. Fine tuning

Prompt Engineering

Bueno para:

- Testear y aprender rápidamente.
- Cuando se utiliza y se evalúa nos da una idea de cómo optimizar

Malo para:

- Introducir nueva información.
- Replicar consistentemente un estilo o método (i.e. aprender un nuevo lenguaje de programación).
- Minimizar el uso de tokens.

Prompting Vs. RAG Vs. Fine tuning

Prompt Engineering

Claves:

- Instrucciones claras.
- Dar tiempo para pensar (step-by-step).
- Dividir un problema complejo en instrucciones simples.
- Incluir ejemplos y evidencia -> Few-shot examples.

Prompting Vs. RAG Vs. Fine tuning

RAG

Bueno para:

- Introducir nueva información para actualizar la del modelo.
- Reducir alucinaciones controlando el contenido

Malo para:

- Entendimiento de temas muy abarcativos y complejos.
- Enseñar al modelo a aprender nuevos lenguajes, formatos o estilos.
- Reducir el uso de tokens.

Prompting Vs. RAG Vs. Fine tuning

Fine tuning

Objetivo: Continuar el proceso de entrenamiento en un dataset de dominio menor para optimizar el modelo en una tarea específica.

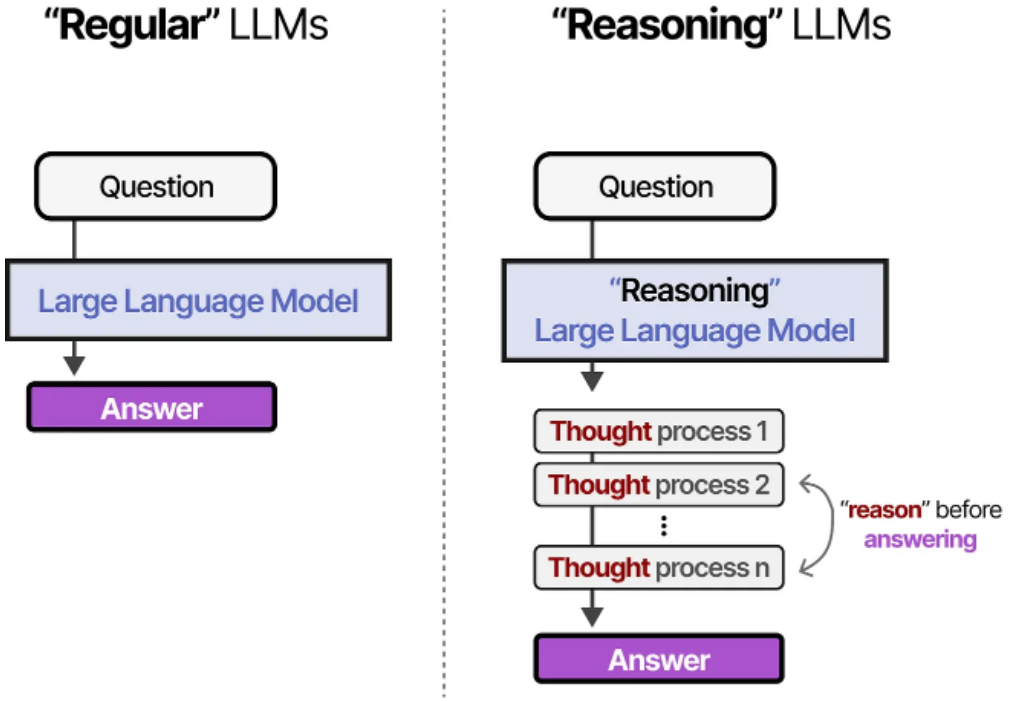
Bueno para:

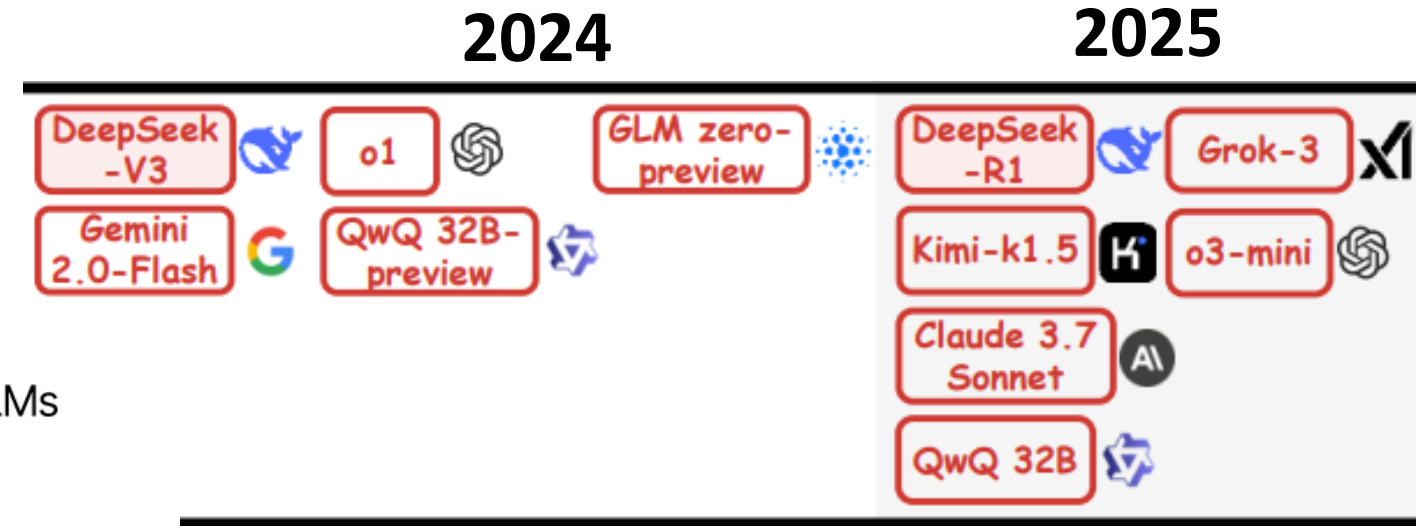
- Mejorar la performance del modelo en un dominio específico.
- Enfatizar el conocimiento que ya existe en un modelo.
- Mejorar la eficiencia (reducción de la cantidad de tokens).

Malo para:

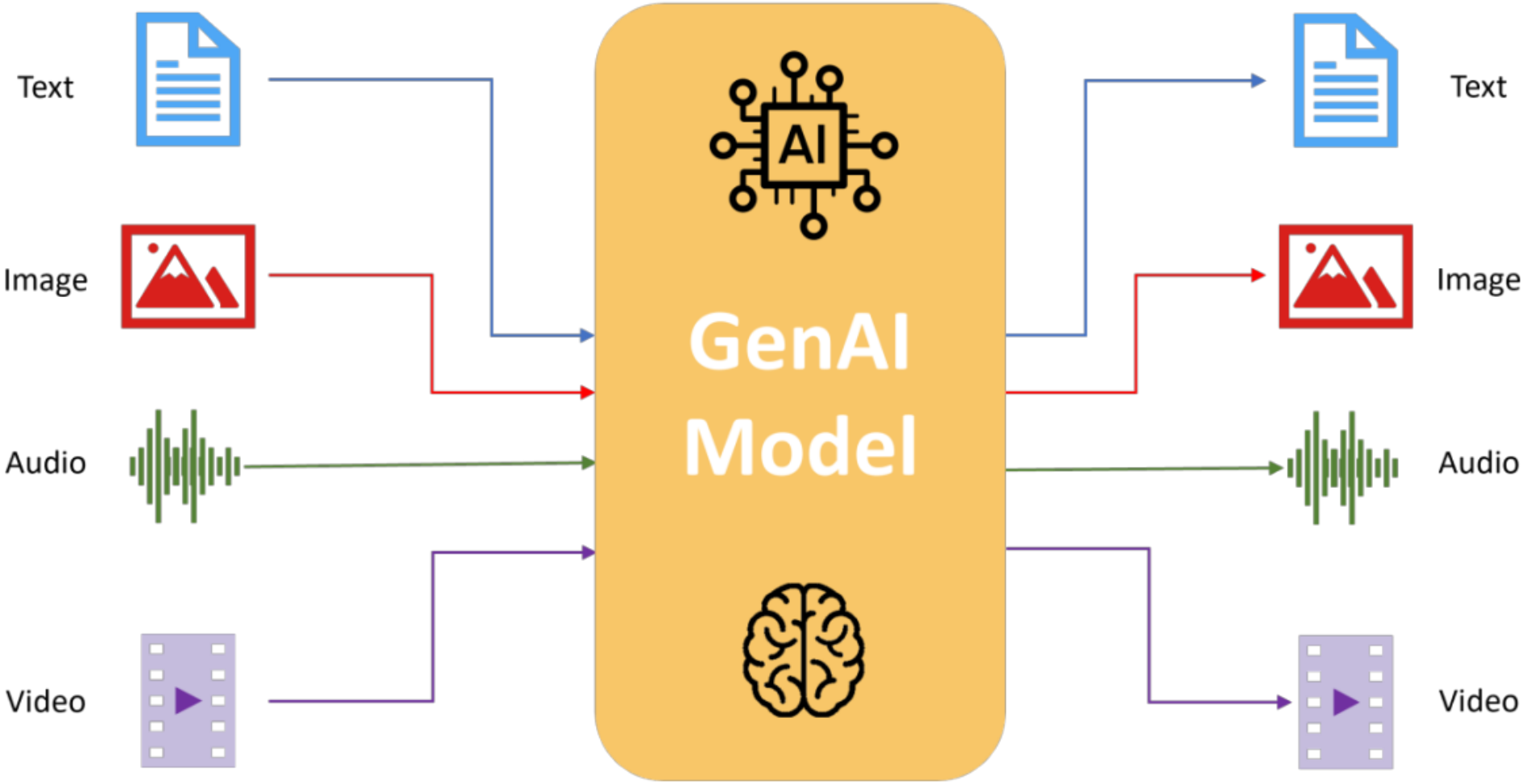
- Agregar nueva información al modelo.
- Iteración rápida en una optimización.

LLMs de Razonamiento

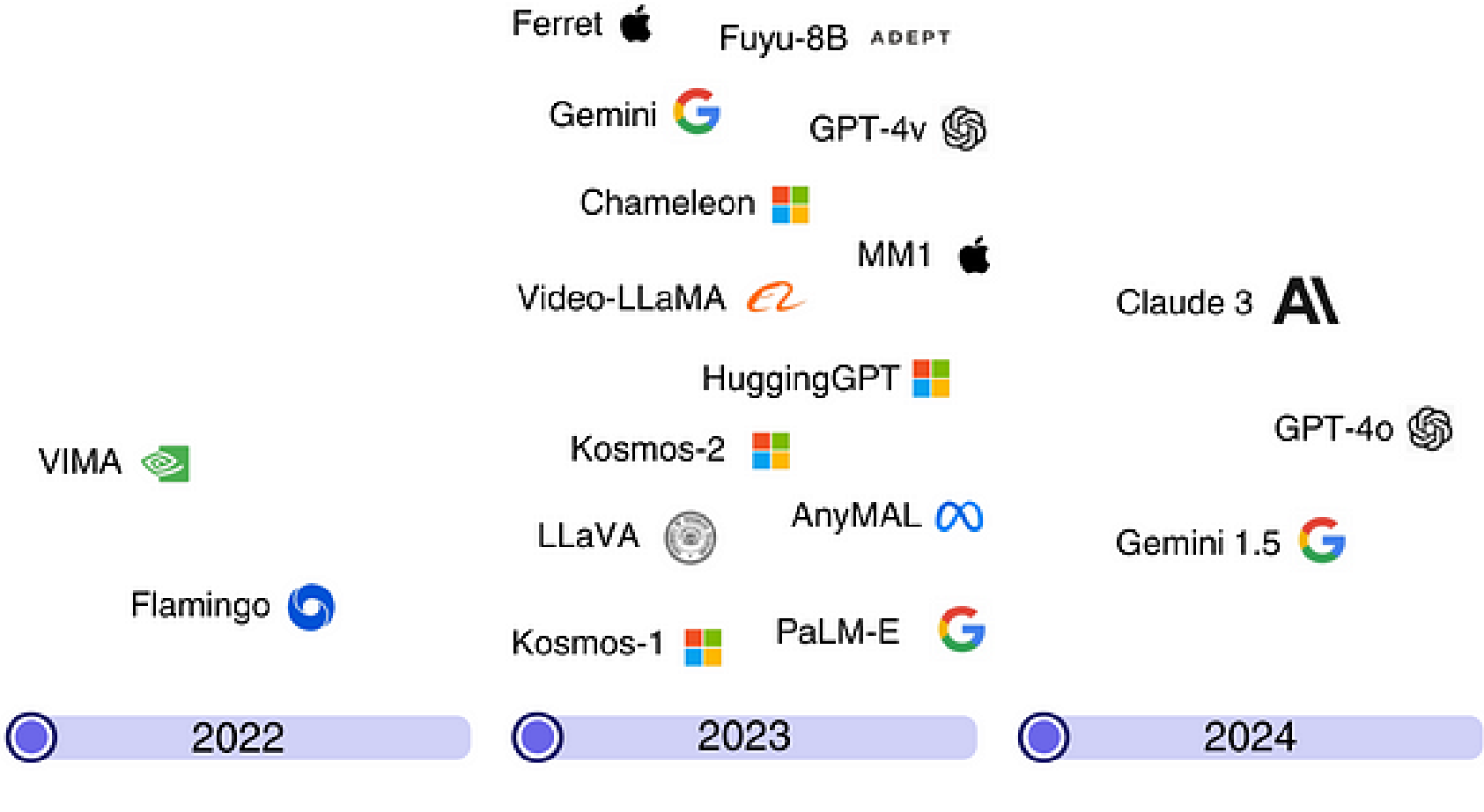




I.A. generativa multimodal

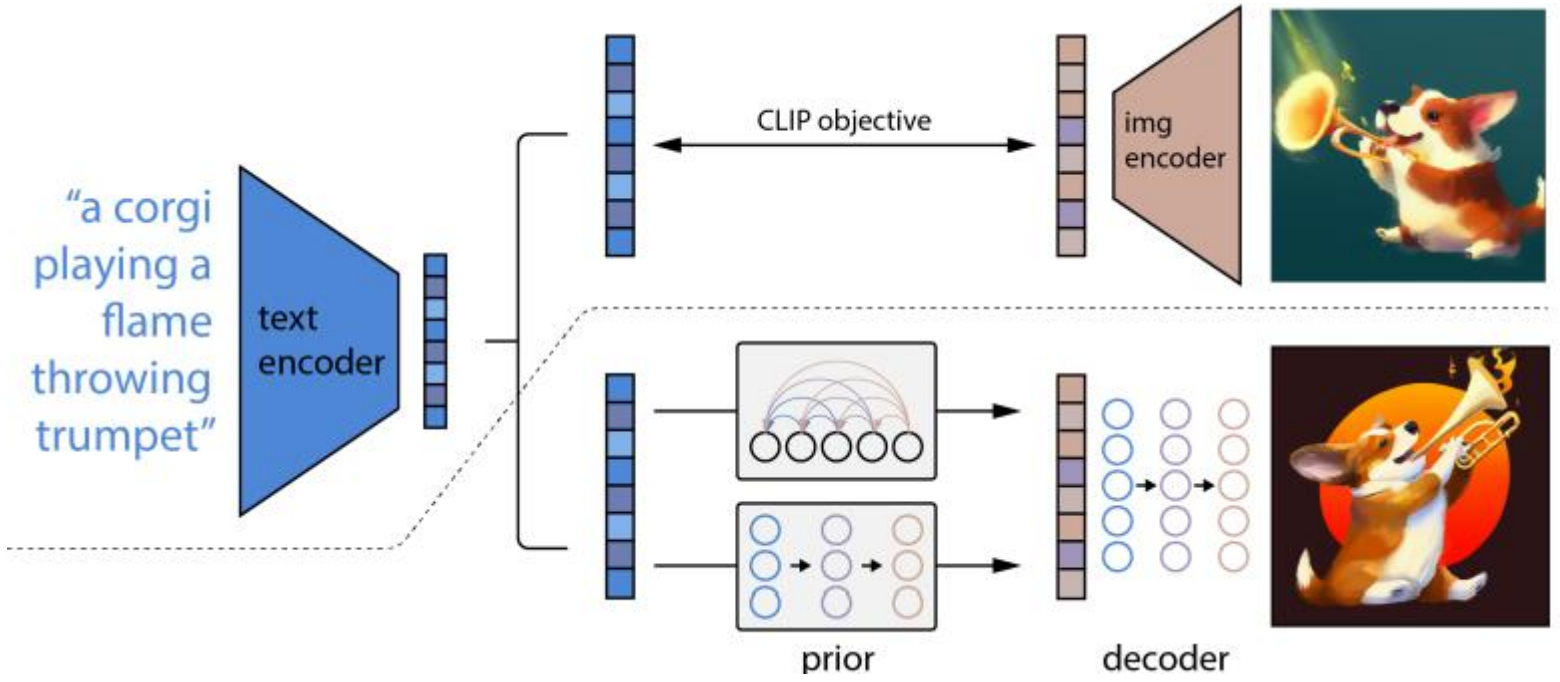


I.A. generativa multimodal



I.A. generativa multimodal – Ejemplo de modelo

Generación de imágenes condicionada por embeddings de texto.
Encoder (CLIP) – Decoder (Difussion model). **[unCLIP]**
Se utiliza el estado latente del modelo CLIP para manipular los embeddings y sacar distintos tipos de imagen



I.A. generativa multimodal – Ejemplo de modelo

Dataset de entrenamiento *(x, y)*
x = imágenes | ***y*** = descripciones
$\boldsymbol{z_i}$ = embeddings de imagen | $\boldsymbol{z_t}$ = embeddings de texto
Se utilizan dos componentes:
prior $\mathbf{P(z_i \mid y)}$ = produce los embeddings de imagenes CLIP
decoder $\mathbf{P(x \mid z_i, y)}$ = produce las imágenes *x* condicionadas a z_i e *y*
Combinando los dos componentes tenemos el modelo generativo de imágenes ***x*** dado las descripciones ***y***:

$$P(x|y) = P(x, z_i|y) = P(x|z_i, y)P(z_i|y)$$

I.A. generativa multimodal – Ejemplo de modelo

Reconstrucción de imágenes a partir de espacios latentes reducidos

Trabajo comparativo de modelos:
https://distill.pub/2021/multimodal-neurons/

Repositorio del código para utilizar:
https://github.com/Stability-AI/stablediffusion

I.A. generativa multimodal
Ejemplos

Ejemplo 1: (text2img)
https://huggingface.co/CompVis/stable-diffusion-v1-4
https://colab.research.google.com/drive/1SXnX2zlx5oE3sItz65stMYvz0nYyfEvI?usp=sharing

Ejemplo 2 (Entendimiento Multimodal)
https://huggingface.co/deepseek-ai/Janus-1.3B

Ejemplo 3 (Imagen a texto estructurado)
Ver “Codigo->Ejemplo_IMG2TXT.ipynb”

Ejercicio en clase:

- **Consigna:** Implementar una aplicación que funcione como un LLM con razonamiento, el cual recibe una pregunta compleja y utiliza diferentes agentes para resolver parcialmente y luego se compaginan todas las respuestas para ofrecer la solución.
- Además de la respuesta se debe imprimir la cantidad de tokens de entrada, salida y razonamiento (estos últimos son los que tokens utilizados en las etapas intermedias).
- **Entregables:** Link a al repositorio y video demostrativo (ídem clases anteriores).
- **Fecha límite de entrega:** Lunes 28 de Abril. No hay excepciones ya que estamos pasados de la fecha de entrega de las notas.
- Además de valorar la calidad técnica de la solución propuesta, la **prolijidad general** será un **criterio fundamental en la calificación**. Esto incluye:
  - La **claridad y redacción del texto entregado** (explicaciones, documentación, etc.).
  - La **organización del repositorio**: estructura de carpetas, nombres de archivos, uso de README, etc.
  - La **presentación general** del trabajo: que sea fácil de entender, limpio y bien presentado.

  Un buen trabajo técnico también debe ser claro, ordenado y profesional. Tener una solución funcional pero mal presentada puede afectar negativamente la nota final.