

Vision Transformers

Docentes:

Esp. Abraham Rodriguez - FIUBA

Mg. Oksana Bokhonok - FIUBA

Programa de la materia

1. Arquitectura de Transformers e imágenes como secuencias.
2. Arquitecturas de ViT y el mecanismo de Attention.
3. Ecosistema actual, Huggingface y modelos pre entrenados.
4. GPT en NLP e ImageGPT.
5. Modelos multimodales: combinación de visión y lenguaje.
6. Segmentación con SAM y herramientas de auto etiquetado multimodales.
7. **OCR y detección con modelos multimodales.**
8. Presentación de proyectos.

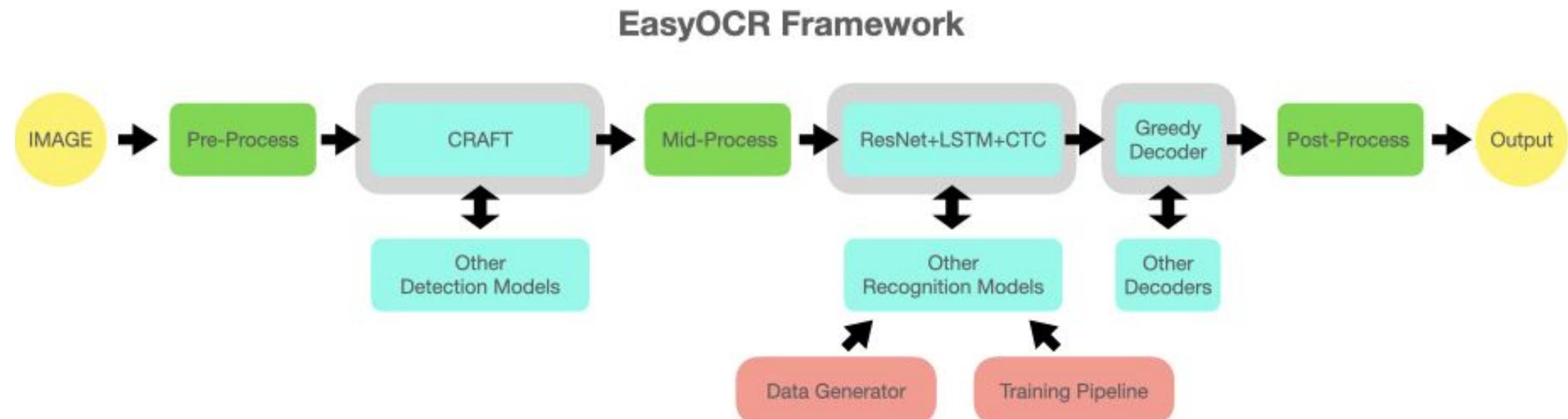
Optical Character Recognition

Optical Character Recognition (OCR)

El OCR, es un problema clásico (antes de 1980s), consiste en extraer texto de imágenes, documentos, etc. Hasta recientemente ha sido muy complejo y computacionalmente costoso. Previo a la IA, se utilizaban algoritmos complejos.

EasyOCR, es un framework que denota explícitamente el proceso tradicional de OCR.

Antes del Transformer, las CNN y LSTM servían como **backbone** para el OCR.

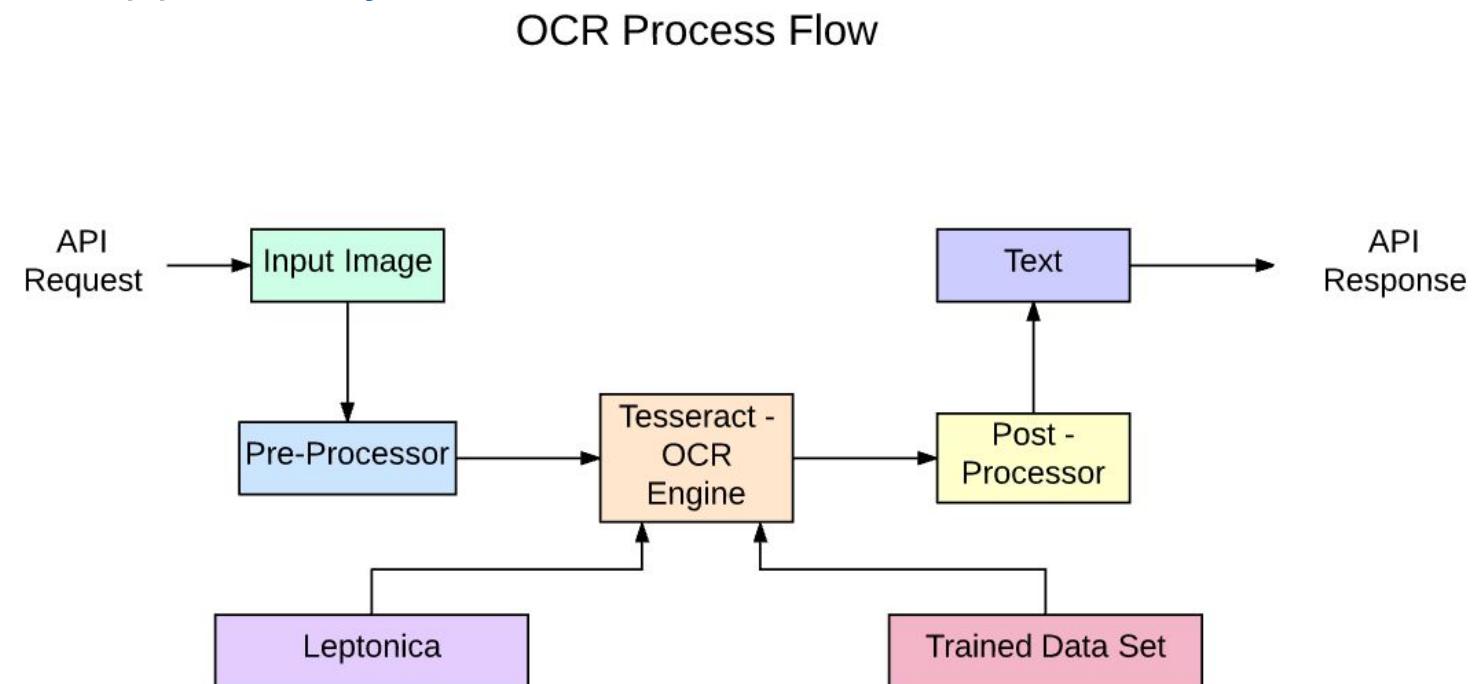


TesseractOCR

[TesseractOCR](#) es un proyecto inicialmente propietario de HP, desarrollado en los 1980s, pero finalmente fue open sourced en 2005 y patrocinado por Google en 2006.

En [2008](#), Tesseract ya utilizaba redes FeedForward, pero en 2018 se introdujo redes **LSTM como OCR engine**.

Cuenta con un wrapper de [Python](#).

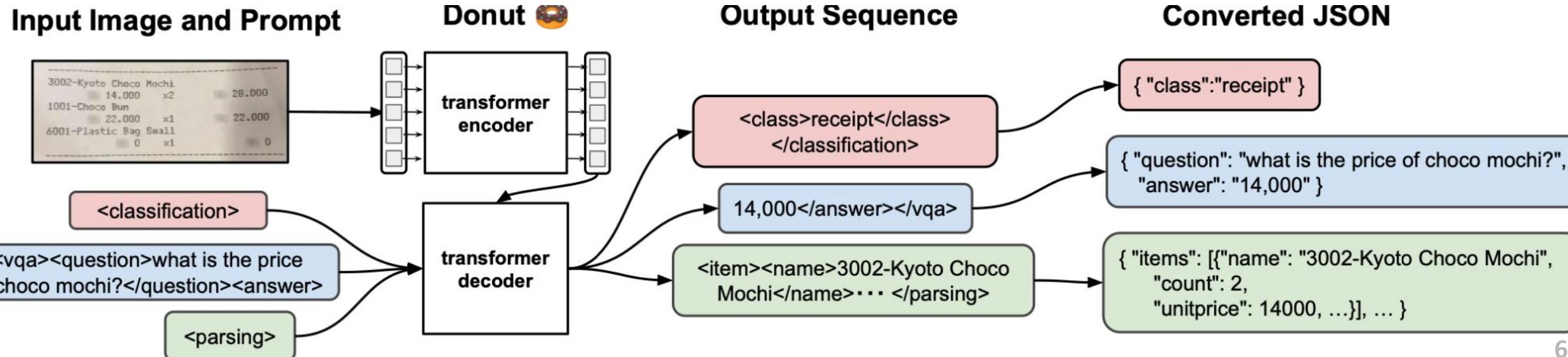
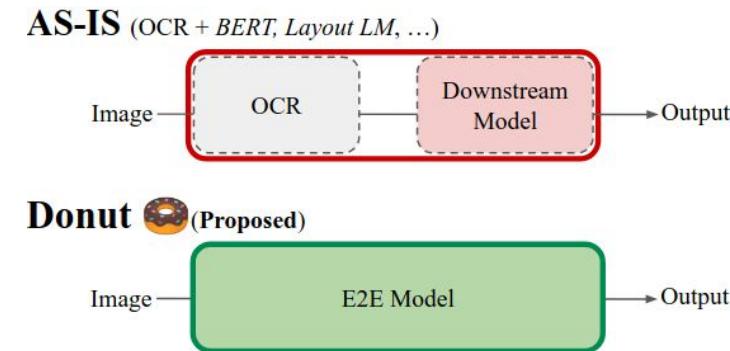


DONUT

Document Understanding Transformer (DONUT) fue presentado en 2022 en el paper “[OCR-free Document Understanding Transformer](#)”, presenta una solución de OCR **end to end** mediante un ViT.

El proceso **convencional** de OCR, consiste en: detección de texto, reconocimiento de texto y parsing.

El proceso de DONUT no es nada más que introducir una imagen y un query para realizar la extracción.

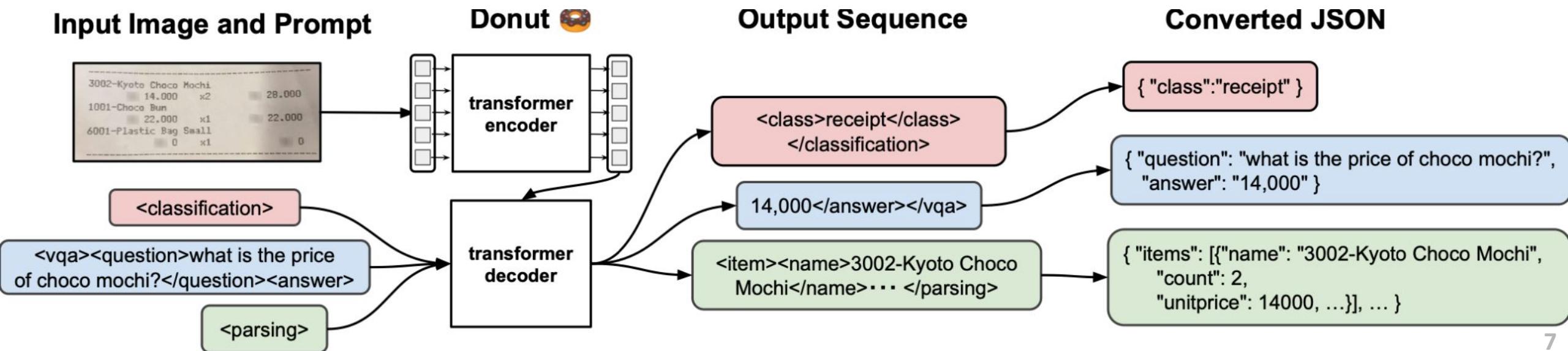


DONUT

El encoder (ViT) transforma una imagen de documento a **embeddings**, el decoder genera una secuencia de tokens que pueden ser transformados a un formato estructurado.

El encoder es un Swin-Transformer.

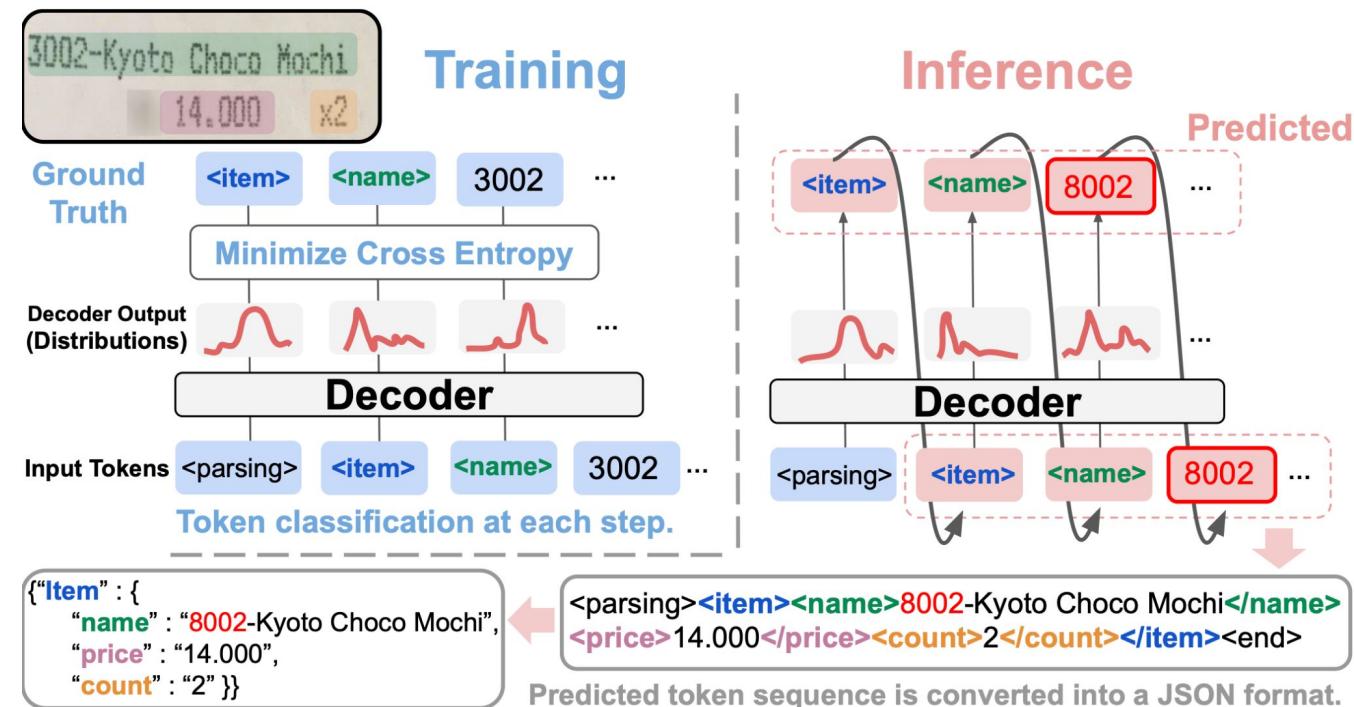
El decoder es [BART](#) preentrenado.



DONUT

[Github](#)

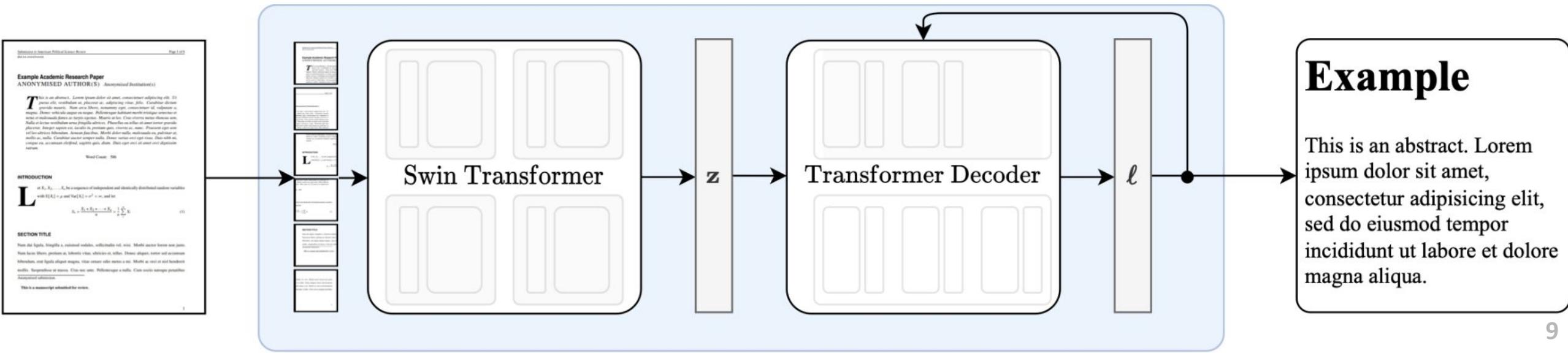
[HuggingFace](#)



NOUGAT

Presentado en el paper “[Nougat: Neural Optical Understanding for Academic Documents](#)”, consiste en el desarrollo de un modelo basado en la arquitectura de DONUT para documentos académicos. Menciona los problemas de OCR tradicional debido al nivel de granularidad linea por linea. NOUGAT realiza OCR en imágenes completas de documentos, capaz de convertir un PDF a un markup language.

Como image encoder utiliza Swin-T, como decoder utiliza mBART.



NOUGAT

[Website](#)

[HuggingFace](#)

[Github](#)

Name	Number of Pages
arXiv	7,511,745
PMC	536,319
IDL	446,777
Total	8,204,754

Table A.1: Dataset composition

Pixtral

Anteriormente vimos [Pixtral](#), sin embargo puede utilizarse para la tarea de OCR, esto conlleva a grandes beneficios sobre NOUGAT y DONUT, el cual es integrar con información más allá del contexto de la imagen gracias a ser un vision language model.

[Building OCR Systems Using Pixtral-12B](#)

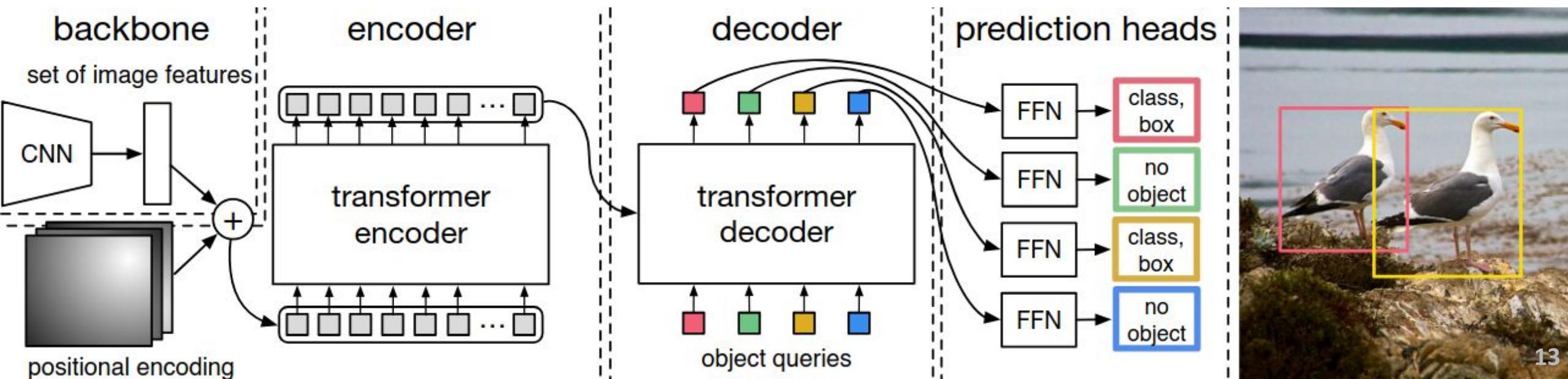
[Pixtral Vision LLM](#)

Detección

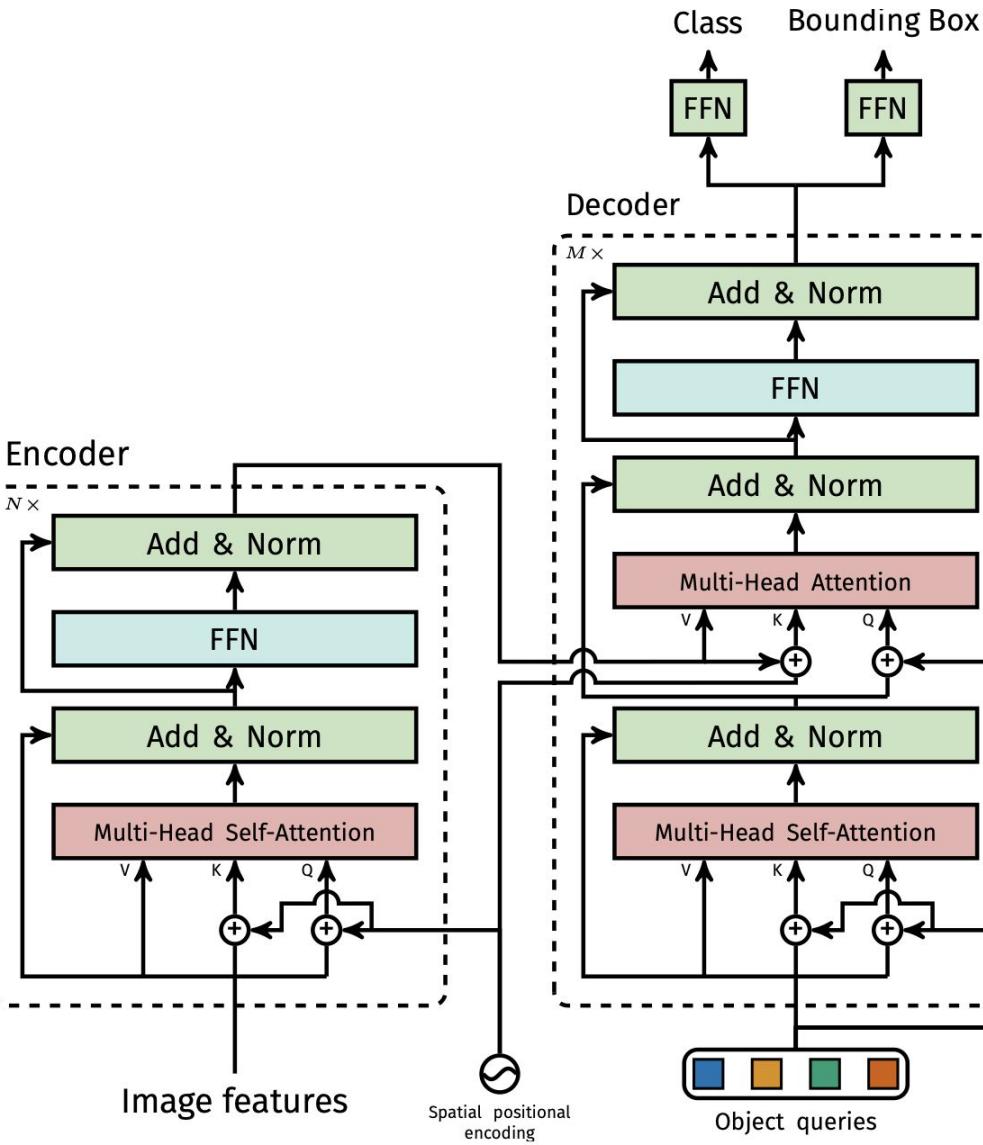
Detection Transformer (DETR)

Presentado en el paper “[End-to-End Object Detection with Transformers](#)”, DETR consiste de un Transformer encoder y decoder en conjunto con una CNN para realizar predicción de bounding boxes.

Sea y el conjunto de ground truth e $\hat{y} = \{\hat{y}_i\}_{i=1}^N$ el conjunto de N predicciones. Asumiendo que N es mayor que la cantidad de objetos en la imagen, se considera y también como un conjunto de tamaño N , rellenado con \emptyset (sin objeto).



Detection Transformer (DETR)



Para encontrar un bipartite match entre estos dos conjuntos, buscamos una permutación de N elementos $\sigma \in S_N$ con el costo más bajo:

$$\hat{\sigma} = \arg \min_{\sigma \in S_N} \sum_i^N \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$$

Donde la función de pérdida es el matching cost entre ground truth y la predicción

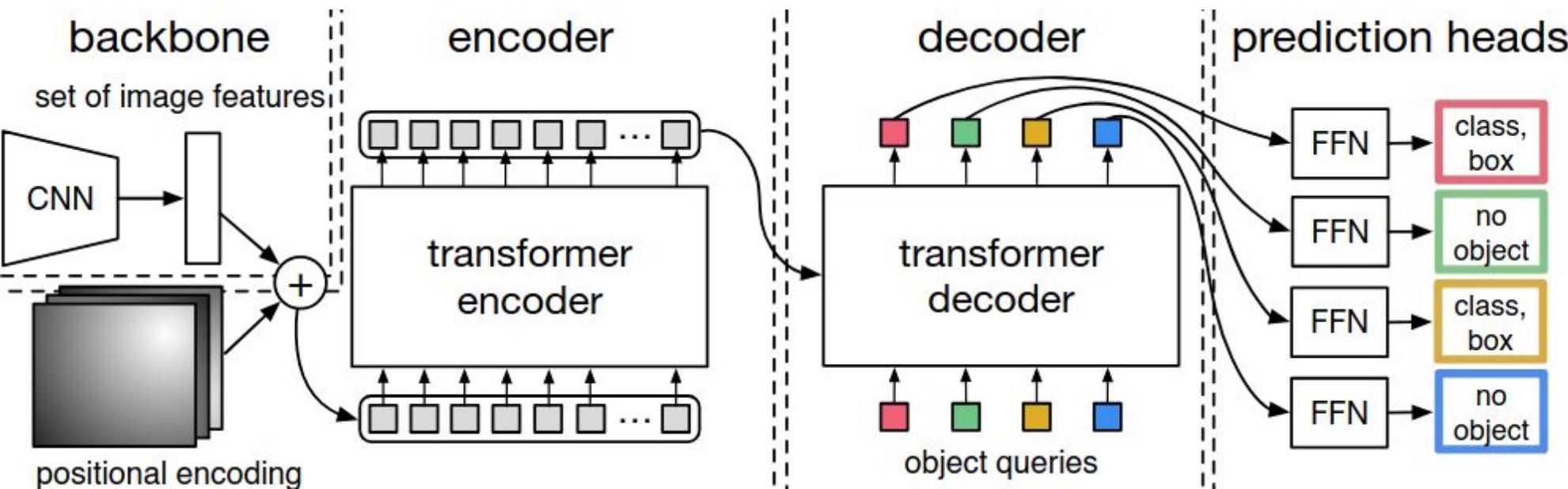
Detection Transformer (DETR)

El **backbone**, parte de una imagen inicial $x_{img} \in R^{3 \times H_0 \times W_0}$ con 3 canales de color, una CNN convencional genera un feature map de menor resolución $f \in R^{C \times H \times W}$. Los valores típicos que utiliza DETR son $C=2048$ y $H,W = H_0/32, W_0/32$. El backbone puede ser sustituido por ResNet u otros modelos convencionales.

El **encoder**, es un Transformer estándar que espera una secuencia como input, por lo tanto se transforman los features a $d \times HW$

El **decoder**, es un Transformer estándar, la única diferencia es que realiza decodificación de N objetos en paralelo.

Los **prediction heads**, son FFNs de 3 capas, donde se da como output un set de N bounding boxes, se agrega la clase especial \emptyset para representar que ningún objeto es detectado.



Detection Transformer (DETR)

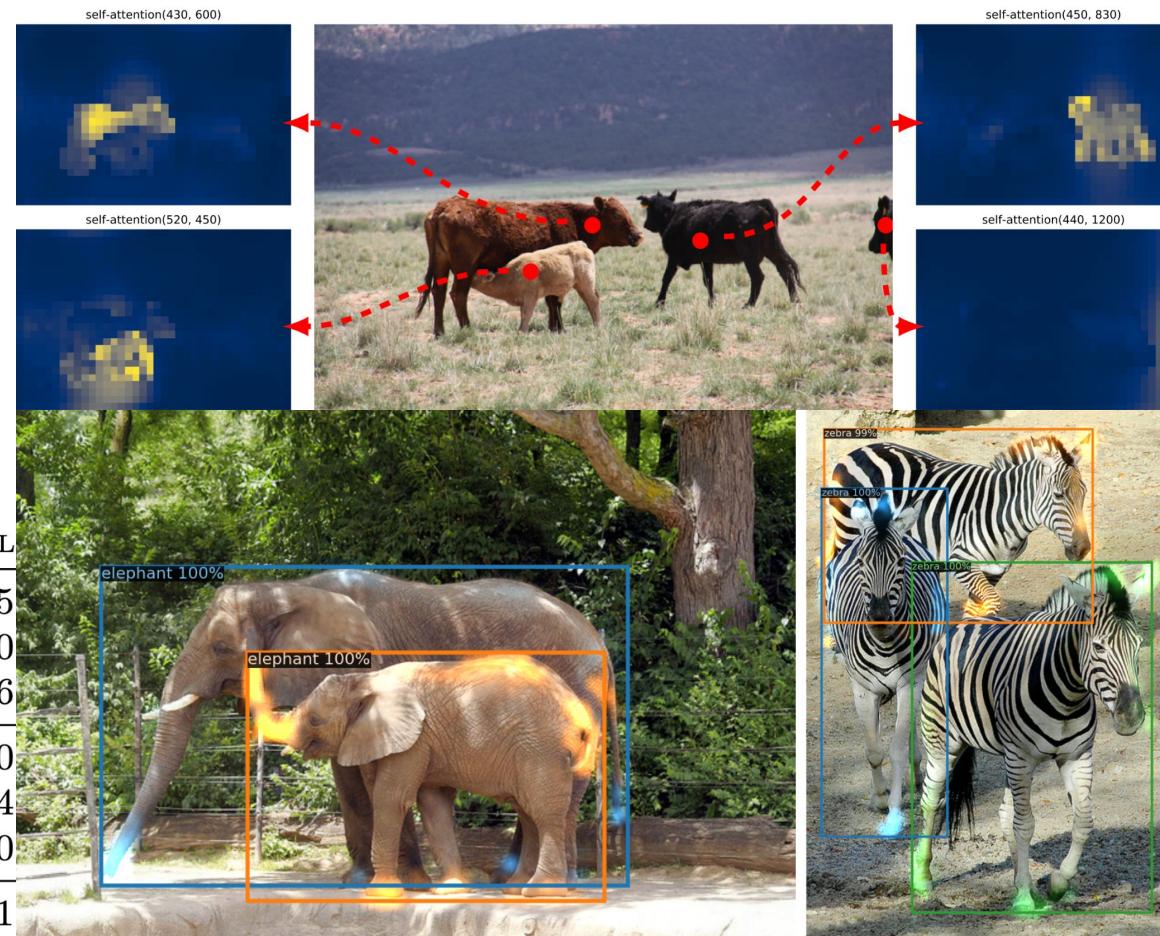
[Huggingface](#)

[Github](#)

[Roboflow: What is DETR?](#)

[DETR vs YOLO for object detection](#)

Model	GFLOPS/FPS	#params	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	47.8	27.2	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	44.9	64.7	47.7	23.7	49.5	62.3



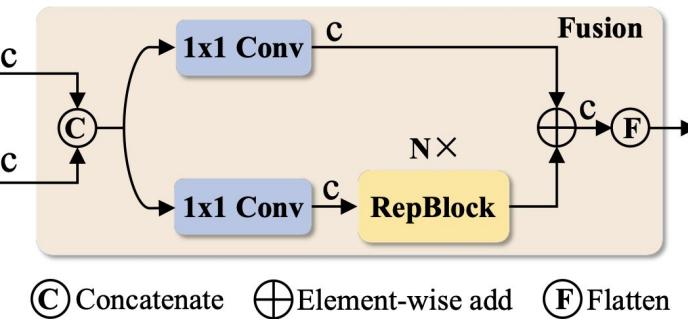
Real-Time DETR (RT-DETR)

Presentado en 2023 en el paper “[DETRs Beat YOLOs on Real-time Object Detection](#)”, mencionan los problemas que tienen YOLOs con [Non-Maximum Suppresion](#) (NMS), el cual se realiza en post-procesamiento y tiene un impacto negativo en inferencia.

Por otro lado DETR no utiliza NMS pero requiere recursos computacionales altos lo cual evita que sea utilizado en aplicaciones en tiempo real, el paper propone una modificacion de DETR para escenarios en tiempo real y superar a YOLO.

El paper menciona que el **cuello de botella se encuentra en el encoder**, para ello se introduce un encoder híbrido con la finalidad de aumentar el tiempo de procesamiento y mantener el accuracy de DETR.

Real-Time DETR (RT-DETR)

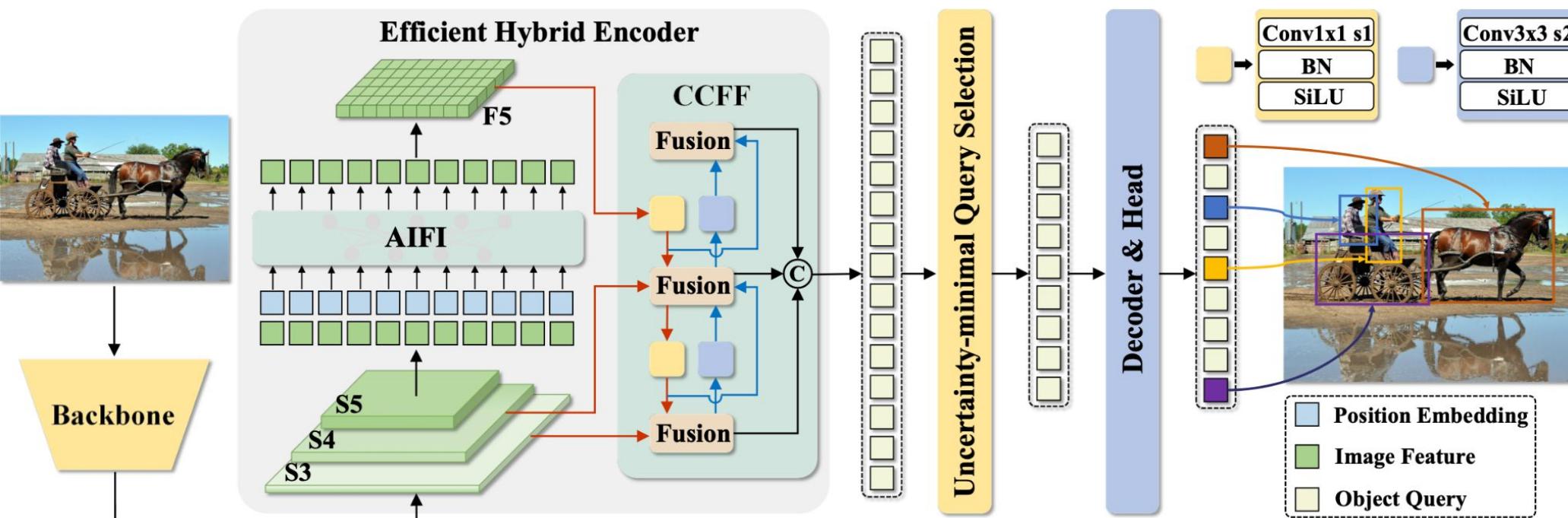


El encoder híbrido se divide en Attention-based Intra-scale Feature Interaction (AIFI) y CNN- based Cross-scale Feature Fusion (CCFF).

AIFI es un transformer al que se reduce el costo computacional al interactuar solamente con el último feature (S_5) extraído del backbone, esto genera una reducción de latencia del 35%.

CCFF, fusiona features en un nuevo feature mediante conv.

$$\begin{aligned} \mathcal{Q} &= \mathcal{K} = \mathcal{V} = \text{Flatten}(\mathcal{S}_5), \\ \mathcal{F}_5 &= \text{Reshape}(\text{AIFI}(\mathcal{Q}, \mathcal{K}, \mathcal{V})), \\ \mathcal{O} &= \text{CCFF}(\{\mathcal{S}_3, \mathcal{S}_4, \mathcal{F}_5\}), \end{aligned}$$



Real-Time DETR (RT-DETR)

Model	Backbone	#Epochs	#Params (M)	GFLOPs	FPS _{bs=1}	AP ^{val}	AP ₅₀ ^{val}	AP ₇₅ ^{val}	AP _S ^{val}	AP _M ^{val}	AP _L ^{val}
<i>Real-time Object Detectors</i>											
YOLOv5-L [11]	-	300	46	109	54	49.0	67.3	-	-	-	-
YOLOv5-X [11]	-	300	86	205	43	50.7	68.9	-	-	-	-
PPYOLOE-L [40]	-	300	52	110	94	51.4	68.9	55.6	31.4	55.3	66.1
PPYOLOE-X [40]	-	300	98	206	60	52.3	69.9	56.5	33.3	56.3	66.4
YOLOv6-L [16]	-	300	59	150	99	52.8	70.3	57.7	34.4	58.1	70.1
YOLOv7-L [38]	-	300	36	104	55	51.2	69.7	55.5	35.2	55.9	66.7
YOLOv7-X [38]	-	300	71	189	45	52.9	71.1	57.4	36.9	57.7	68.6
YOLOv8-L [12]	-	-	43	165	71	52.9	69.8	57.5	35.3	58.3	69.8
YOLOv8-X [12]	-	-	68	257	50	53.9	71.0	58.7	35.7	59.3	70.7
<i>End-to-end Object Detectors</i>											
DETR-DC5 [4]	R50	500	41	187	-	43.3	63.1	45.9	22.5	47.3	61.1
DETR-DC5 [4]	R101	500	60	253	-	44.9	64.7	47.7	23.7	49.5	62.3
Anchor-DETR-DC5 [39]	R50	50	39	172	-	44.2	64.7	47.5	24.7	48.2	60.6
Anchor-DETR-DC5 [39]	R101	50	-	-	-	45.1	65.7	48.8	25.8	49.4	61.6
Conditional-DETR-DC5 [27]	R50	108	44	195	-	45.1	65.4	48.5	25.3	49.0	62.2
Conditional-DETR-DC5 [27]	R101	108	63	262	-	45.9	66.8	49.5	27.2	50.3	63.3
Efficient-DETR [42]	R50	36	35	210	-	45.1	63.1	49.1	28.3	48.4	59.0
Efficient-DETR [42]	R101	36	54	289	-	45.7	64.1	49.5	28.2	49.1	60.2
SMCA-DETR [9]	R50	108	40	152	-	45.6	65.5	49.1	25.9	49.3	62.6
SMCA-DETR [9]	R101	108	58	218	-	46.3	66.6	50.2	27.2	50.5	63.2
Deformable-DETR [45]	R50	50	40	173	-	46.2	65.2	50.0	28.8	49.2	61.7
DAB-Deformable-DETR [23]	R50	50	48	195	-	46.9	66.0	50.8	30.1	50.4	62.5
DAB-Deformable-DETR++ [23]	R50	50	47	-	-	48.7	67.2	53.0	31.4	51.6	63.9
DN-Deformable-DETR [17]	R50	50	48	195	-	48.6	67.4	52.7	31.0	52.0	63.7
DN-Deformable-DETR++ [17]	R50	50	47	-	-	49.5	67.6	53.8	31.3	52.6	65.4
DINO-Deformable-DETR [44]	R50	36	47	279	5	50.9	69.0	55.3	34.6	54.1	64.6
<i>Real-time End-to-end Object Detector (ours)</i>											
RT-DETR	R50	72	42	136	108	53.1	71.3	57.7	34.8	58.0	70.0
RT-DETR	R101	72	76	259	74	54.3	72.7	58.6	36.0	58.8	72.1

[Github](#)

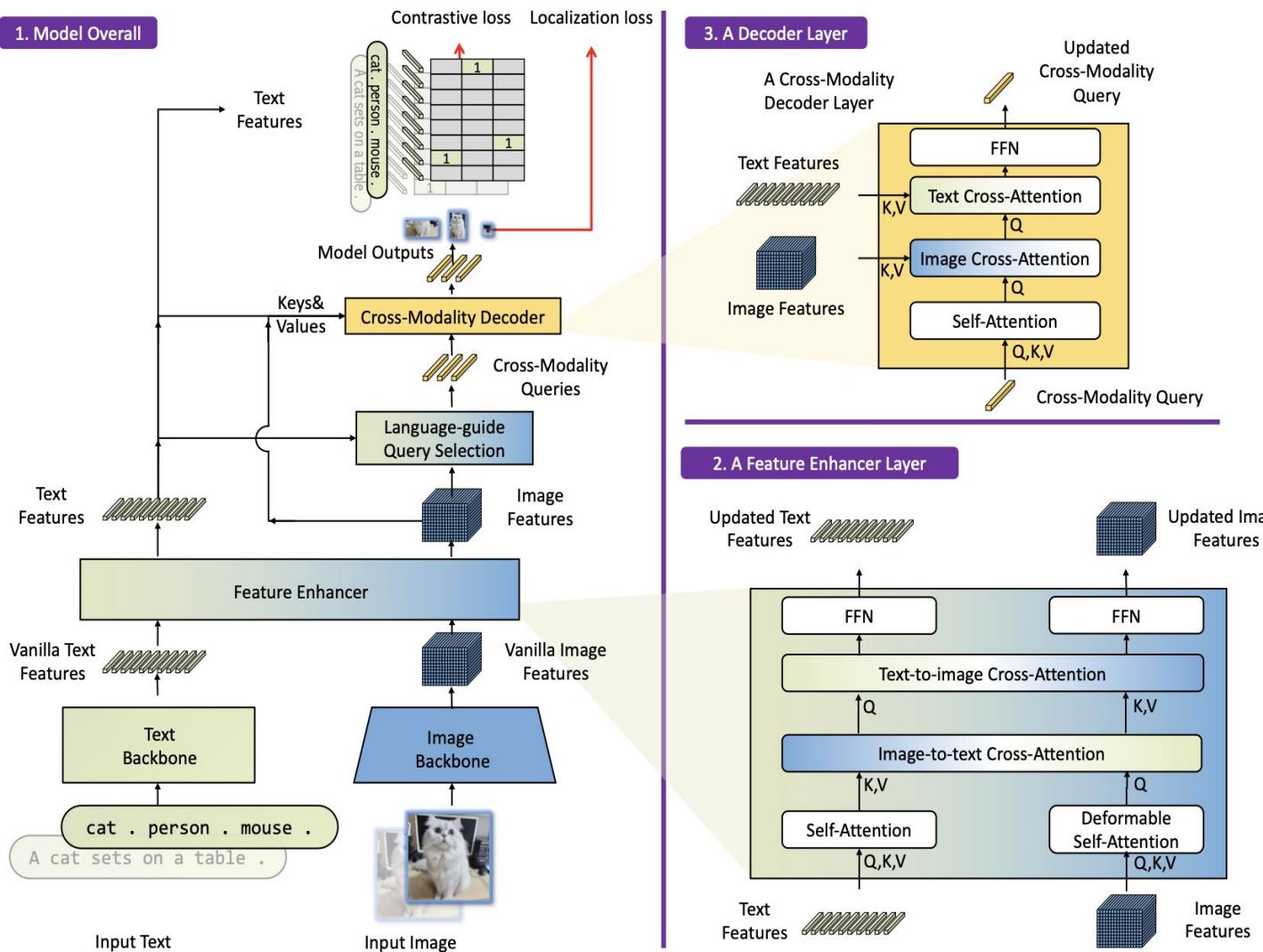
[Website](#)

[Huggingface](#)

[Ultralytics RT-DETR](#)

Actualmente se encuentra el paper corto de [RT-DETRv2](#), el cual es idéntico al original, introduce mejoras de muestreo, data augmentation

Grounding DINO



Presentado en 2024 en el paper “[Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection](#)”. Grounding DINO es basado en [DINO](#) (variante de DETR) el cual produce múltiples pares de cajas de objetos y frases nominales para un par dado de (Imagen, Texto).

Los image features son extraídos mediante un image backbone como Swin-T y los text features mediante BERT.

Grounding Dino realiza la tarea de detectar objetos en una imagen mediante un text input.

Grounding DINO

Model	Backbone	Pre-Training Data	Zero-Shot 2017val	Fine-tuning 2017val/test-dev
Faster R-CNN	RN50-FPN	-	-	40.2 / -
Faster R-CNN	RN101-FPN	-	-	42.0 / -
DyHead-T [5]	Swin-T	-	-	49.7 / -
DyHead-L [5]	Swin-L	-	-	58.4 / 58.7
DyHead-L [5]	Swin-L	O365,ImageNet21K	-	60.3 / 60.6
Soft Teacher [50]	Swin-L	O365,SS-COCO	-	60.7 / 61.3
DINO(Swin-L) [57]	Swin-L	O365	-	62.5 / -
DyHead-T† [5]	Swin-T	O365	43.6	53.3 / -
GLIP-T (B) [25]	Swin-T	O365	44.9	53.8 / -
GLIP-T (C) [25]	Swin-T	O365,GoldG	46.7	55.1 / -
GLIP-L [25]	Swin-L	FourODs,GoldG,Cap24M	49.8	60.8 / 61.0
DINO(Swin-T)† [57]	Swin-T	O365	46.2	56.9 / -
Grounding DINO T (Ours)	Swin-T	O365	46.7	56.9 / -
Grounding DINO T (Ours)	Swin-T	O365,GoldG	48.1	57.1 / -
Grounding DINO T (Ours)	Swin-T	O365,GoldG,Cap4M	48.4	57.2 / -
Grounding DINO L (Ours)	Swin-L	O365,OI [19],GoldG	52.5	62.6 / 62.7 (63.0 / 63.0)*
Grounding DINO L (Ours)	Swin-L	O365,OI,GoldG,Cap4M,COCO,RefC	60.7‡	62.6 / -

[Roboflow: Grounding DINO](#)

[Notebook: Object Detection with Grounding DINO](#)

[Using Text Prompts for Image Annotation with Grounding DINO and Label Studio](#)

Foundation models (FMs)

El Center for Research on Foundation Models (CRFM), parte del Stanford Institute for Human-Centered Artificial Intelligence (HAI), fue el primero en acuñar formalmente el término **“foundation model”** en agosto de **2021**.

Escalabilidad: entrenados con enormes cantidades de datos y cómputo, lo que permite capturar patrones complejos y generalizables.

Versatilidad: pueden adaptarse a una amplia variedad de tareas sin necesidad de rediseñar el modelo (vía fine-tuning o prompting).

Emergencia: desarrollan capacidades no anticipadas ni programadas explícitamente, que surgen del aprendizaje a gran escala.

Homogeneización: tienden a convertirse en una base común para muchos sistemas, lo que estandariza herramientas pero también centraliza riesgos (sesgos, fallas y errores se replican ampliamente).

Foundation models

- 2021: Se definen Foundation Model (FM).
- 2022: Aparecen los primeros modelos de tipo Interactive FMs (IFM)
- 2022–2023: Aparecen modelos multimodales con propiedades emergentes tipo WFM. Entre ellos GPT-4V (OpenAI, 2023), Kosmos-2 (Microsoft, 2023), PaLI-X / PaLM-E (Google, 2023), Gato (DeepMind, 2022)
- 2024–2025: Aparición del concepto de “World Foundation Models”. El concepto de World Foundation Model se cristalizó con los anuncios de NVIDIA y colaboradores a finales de 2024 e inicios de 2025.
- Los WFs son una extensión física y visual de los FMs, aplicando los mismos principios (escalabilidad, emergencia, versatilidad y homogeneización) a entornos del mundo real, más allá del texto o la imagen estática.
- NVIDIA publicó el trabajo académico “Cosmos: A World Foundation Model Platform for Physical AI” (Wang et al., 2024)
- Cosmos WFM es una familia de modelos generativos de video
(se desarrollaron variantes basadas en difusión y en autoregresión con transformers)

World foundation models (WFMs)

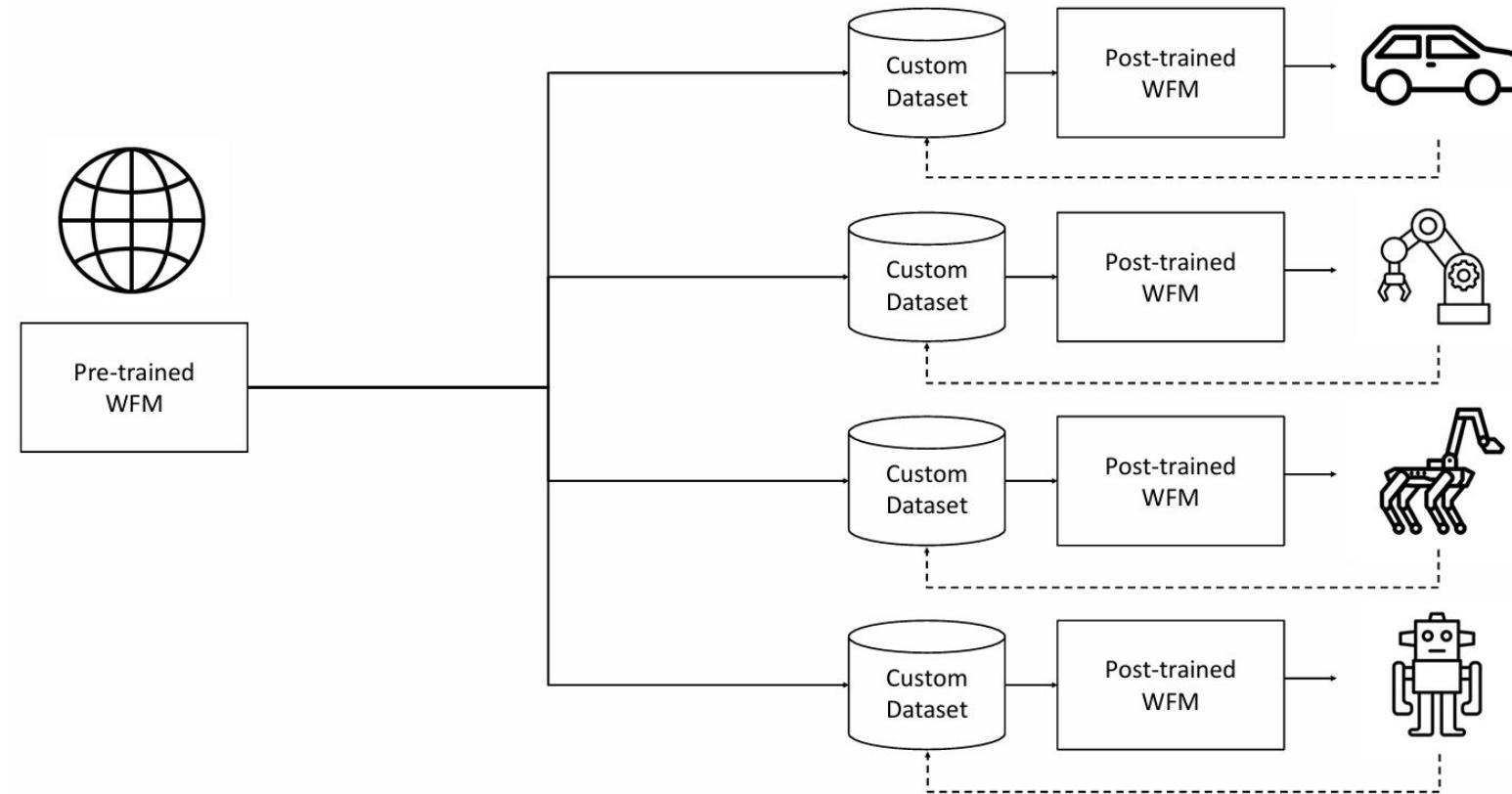
Criterio	Multimodal clásico (ej. CLIP, Flamingo)	World Foundation Model (ej. Kosmos-2, GPT-4V, Cosmos)
Usa varias modalidades	Sí	Sí
Razonamiento causal y espacial	No necesariamente <i>Ejemplo: CLIP puede decir "hay un perro", pero no sabe si está a la izquierda o qué hará después.</i>	Requisito central <i>Ejemplo: GPT-4V puede entender que "el gato saltará sobre la mesa" si ve al gato en movimiento y una mesa delante.</i>
Representación del mundo persistente	No <i>Ejemplo: Flamingo responde sobre un frame, pero no "recuerda" objetos entre preguntas.</i>	Sí <i>Ejemplo: Kosmos-2 puede seguir la trayectoria de un objeto en video y hacer tracking entre cuadros.</i>
Capacidad predictiva del entorno	No <i>Ejemplo: CLIP no predice un siguiente frame.</i>	Sí (predice dinámicas, simula) <i>En el Cosmos WFM, se genera una escena futura dada una instrucción ("el auto gira a la derecha")</i>

Evolución de los Modelos Fundacionales hacia la Acción en el Mundo Físico

Característica	Foundation Models (FM)	Interactive FMs (IFM)	World FMs (WFM)	Physical AI Systems (PAIS)
Modalidad principal	Texto, imagen, audio	Texto + acción (simulada) + visión	Multimodal + acciones + razonamiento	Sensores reales + actuadores físicos
Capacidad de actuar	No	Sí (acción en entorno simulado o abstraído)	Sí (acción razonada y contextualizada)	Sí
Comprendión del entorno	Semántica y perceptual	Contextual y reactiva	Espacial, semántica, causal, integrada	Completa (sensorimotor física)
Interactividad	No	Simulada o indirecta	Simulada y planeada	Física y continua
Output	Representaciones (embeddings, etiquetas, features)	Decisiones, comandos, trayectorias	Políticas contextualizadas, razonamiento + acción física/simulada	Movimiento físico, control directo
Conocimiento del mundo físico	No	Parcial / abstraído	Integrado desde datos simulados y reales	Directo, en tiempo real
Uso de sensores o cuerpo	Ninguno	Solo simulado o pregrabado	Simulado + grounding parcial	Sí, en ejecución física

Si un modelo entiende la palabra “manzana”, el **grounding** ocurre cuando puede asociarla a: la imagen de una manzana (visión), su peso o forma (sensorimotor), o la acción de agarrarla (interacción física).

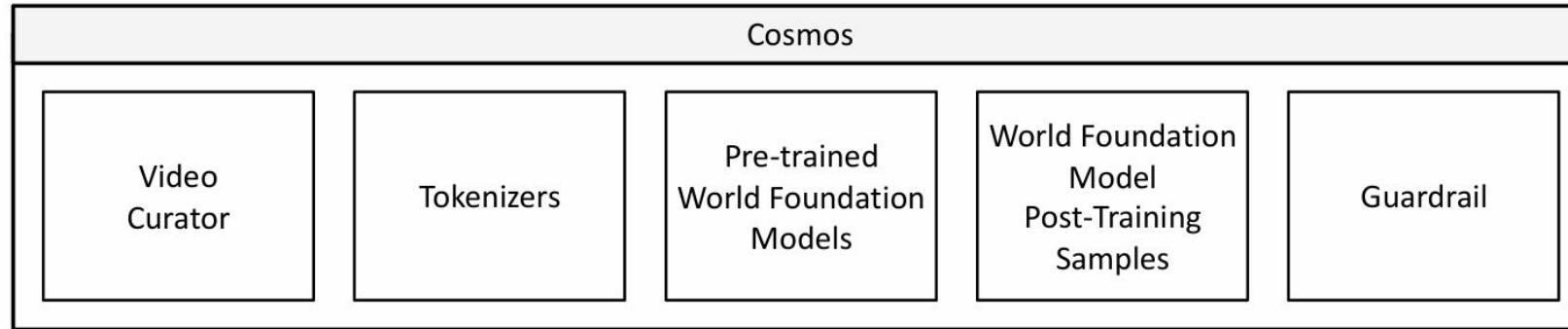
Cosmos -World foundation models (WFMs)



[Cosmos World Foundation Model Platform
for Physical AI](#)

World foundation models (WFMs)

módulos funcionales dentro de la plataforma



Video Curator: Selecciona automáticamente clips de video útiles y dinámicos para entrenar modelos, asegurando calidad y diversidad.

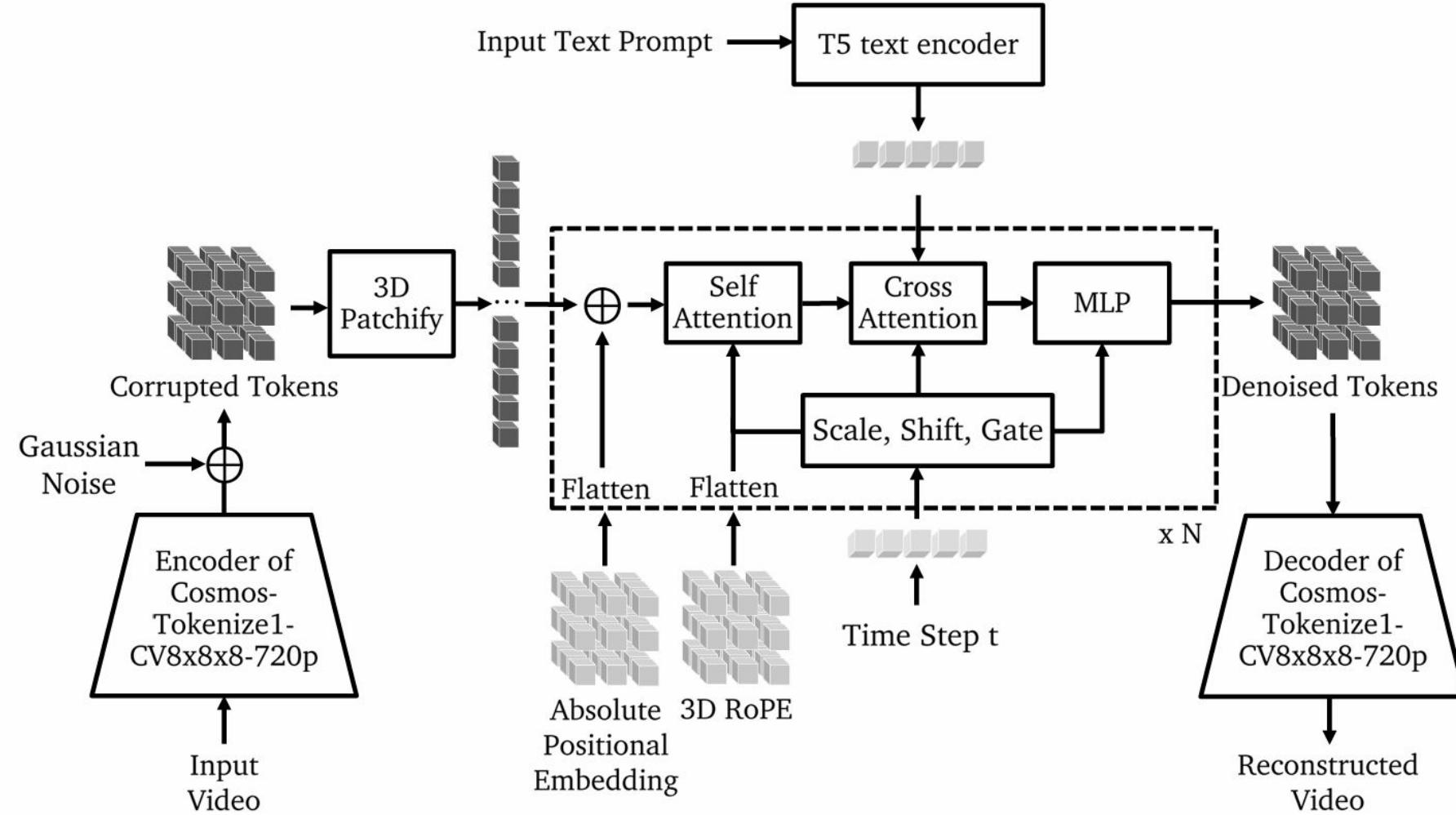
Video Tokenizer: Convierte los videos en tokens compactos que pueden ser usados por modelos. Permite entrenar con imágenes y videos de forma conjunta, respetando el orden temporal (causalidad).

WFM Pre-Training: Entrena modelos del mundo con dos enfoques: difusión y autoregresivo

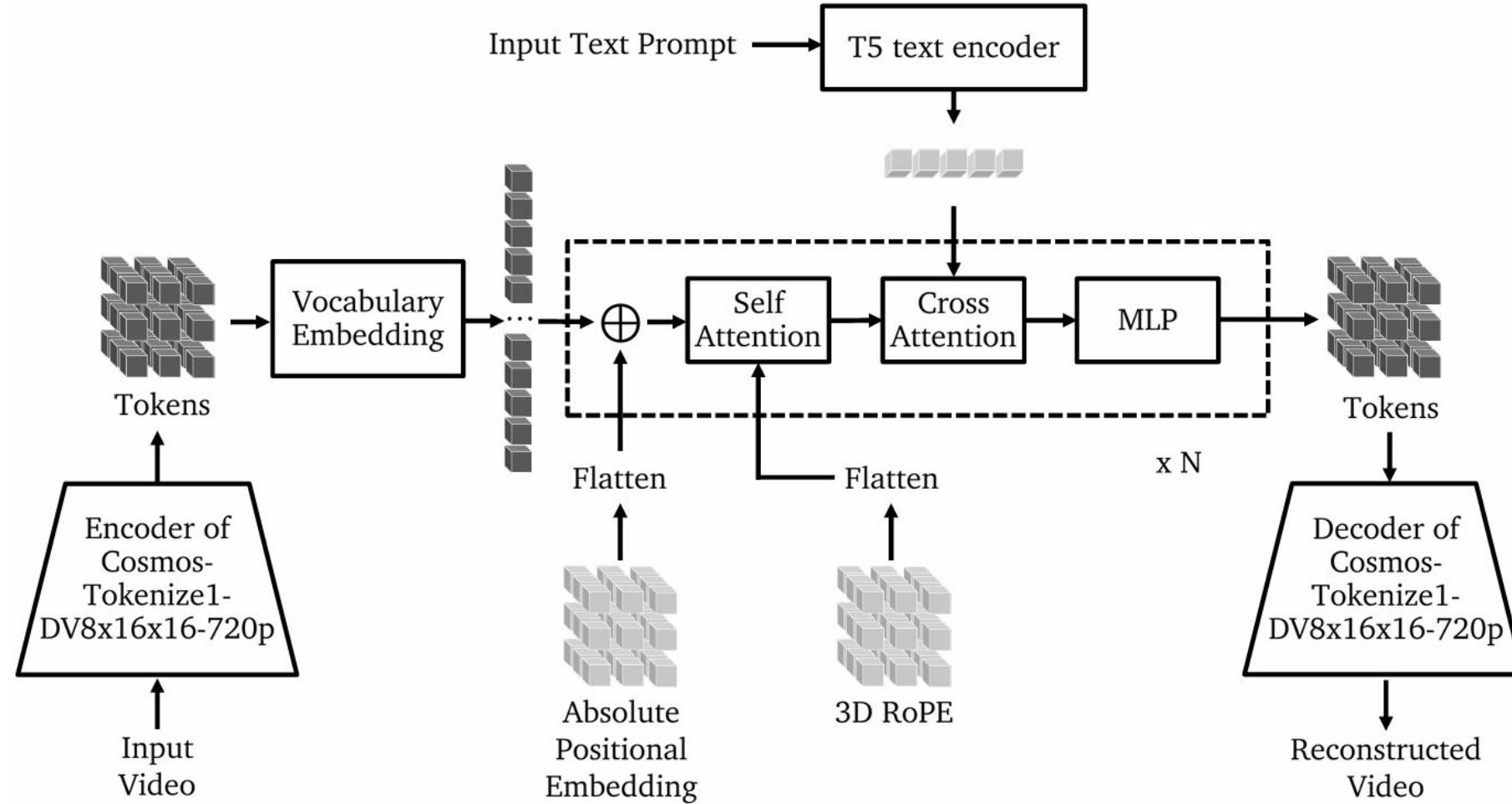
WFM Post-Training: Ajusta el modelo preentrenado a tareas específicas (navegación con cámara, robótica humanoide, conducción autónoma) Permite aplicar el WFM en IA física real.

Guardrail: Sistema de seguridad que bloquea entradas o salidas dañinas para garantizar el uso responsable del modelo.

Cosmos-Predict1: Diffusion-based World Foundation Model



Cosmos-Predict1: Autoregressive-based World Foundation Model



Vocabulary Embedding: es una tabla de embedding (NLP), donde cada token discreto del video es convertido en un vector denso (embedding).

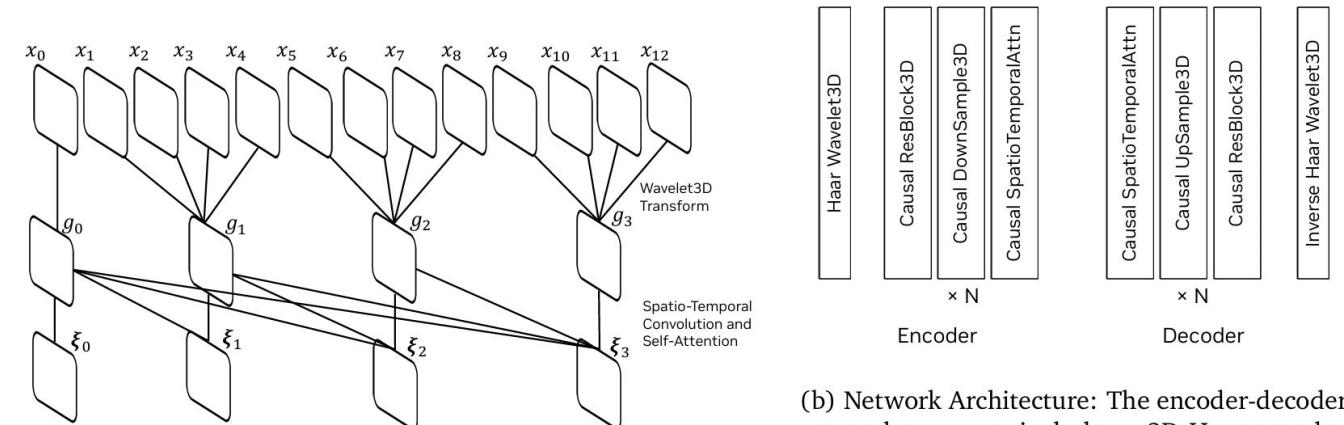
Cosmos-Tokenizer

Token Encoder:

- Recibe un video crudo.
- Aplica una serie de transformaciones (wavelet, convoluciones 3D, atención causal).
- Produce una representación comprimida: tokens latentes (pueden ser continuos o discretos).

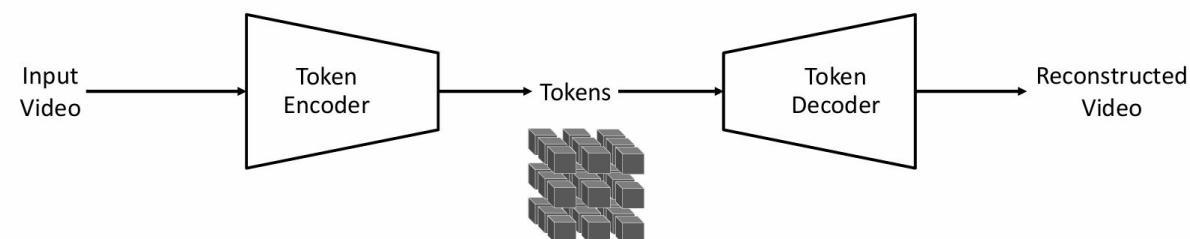
Tokens:

- Son la salida del encoder.
- Representan el contenido visual (espacial + temporal) de forma compacta.
- Se entrena para preservar la información esencial del video original.



(a) Temporal Causality: Illustration of the temporal causality mechanism, where inputs x_0, x_1, \dots, x_{12} are processed through grouped intermediate outputs g_0, g_1, \dots, g_3 , and further refined by spatio-temporal convolution and attention operations.

(b) Network Architecture: The encoder-decoder network structure includes a 3D Haar wavelet, causal residual, causal downsampling, and causal spatio-temporal attention blocks. The decoder mirrors the encoder's structure, replacing down-sampling with upsampling.



Cosmos-Tokenizer

Cosmos Tokenizer admite dos tipos de tokenización:

- Tokenización Continua: Genera embeddings latentes continuos, adecuados para modelos de difusión que operan en espacios latentes continuos. Porque los modelos de difusión trabajan con ruido Gaussiano y lo refinan progresivamente. Para eso necesitan representaciones continuas donde puedan aplicar operaciones diferenciables, interpolaciones, y optimizar con pérdida como MSE (error cuadrático medio) o score matching.
- Tokenización Discreta: Produce códigos latentes discretos mediante cuantización, ideales para modelos autoregresivos que requieren representaciones discretas para la generación secuencial.

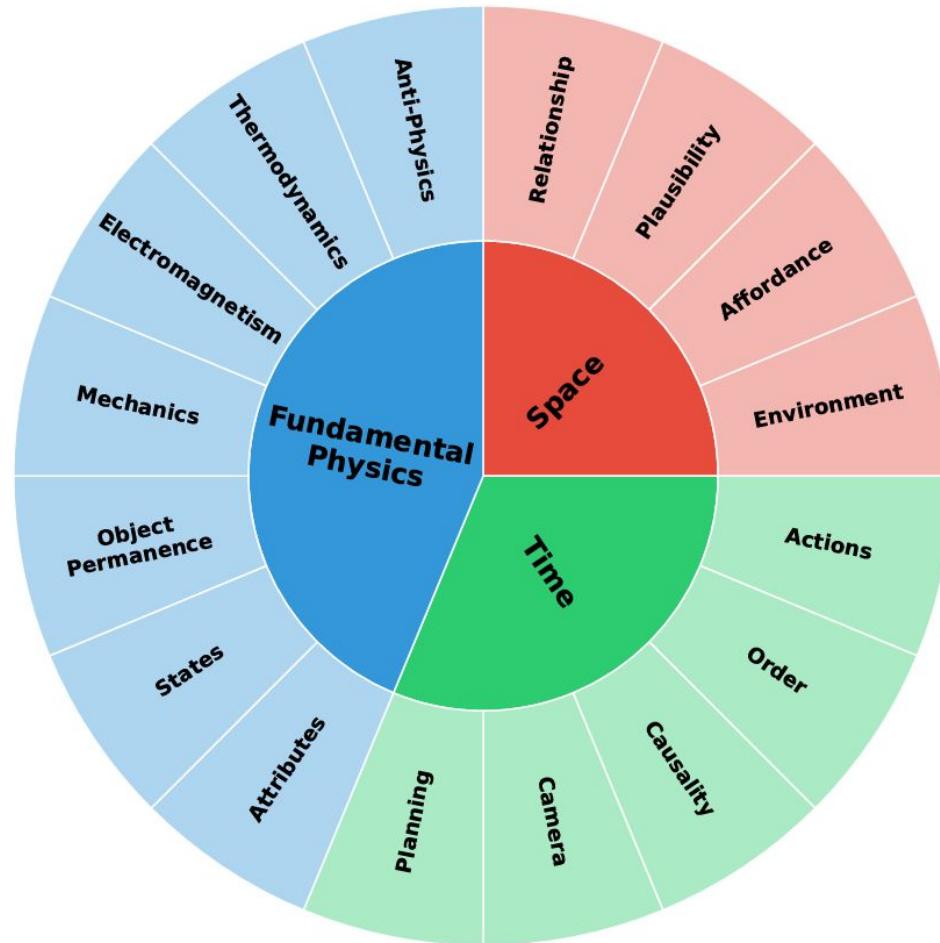
Cosmos-Tokenizer

Table 4: Comparison of different visual tokenizers and their capabilities.

Model	Causal	Image	Video	Joint	Discrete	Continuous
FLUX-Tokenizer (FLUX, 2024)	-	✓	✗	✗	✗	✓
Open-MAGVIT2-Tokenizer (Luo et al., 2024)	-	✓	✗	✗	✓	✗
LlamaGen-Tokenizer (Sun et al., 2024)	-	✓	✗	✗	✓	✗
VideoGPT-Tokenizer (Yan et al., 2021)	✗	✗	✓	✗	✓	✗
Omni-Tokenizer (Wang et al., 2024)	✗	✓	✓	✓	✓	✓
CogVideoX-Tokenizer (Yang et al., 2024)	✓	✓	✓	✓	✗	✓
Cosmos-Tokenizer	✓	✓	✓	✓	✓	✓

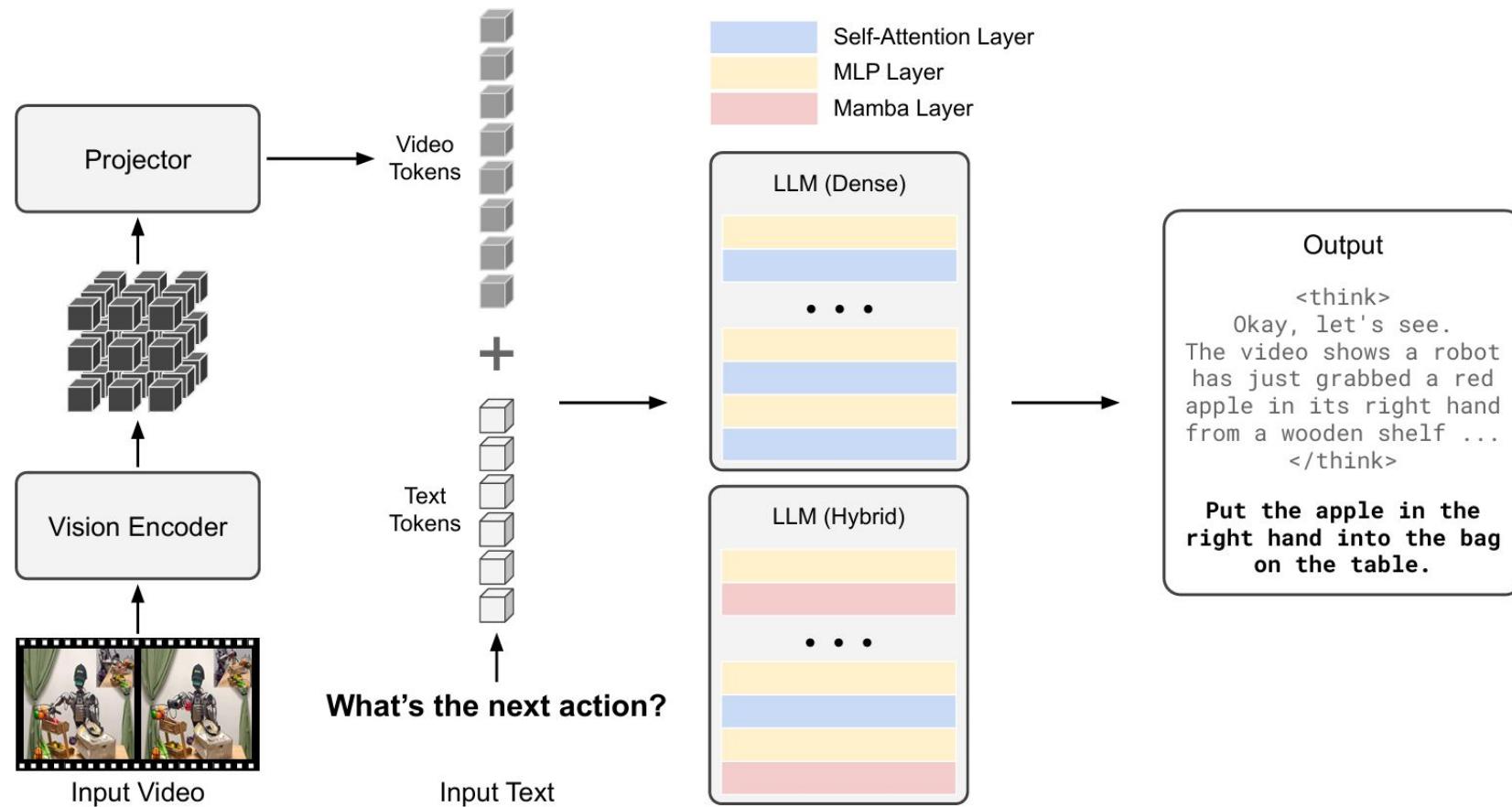
Causal	Si el tokenizador respeta la causalidad temporal (no mira el futuro).
Image	Si puede tokenizar imágenes estáticas.
Video	Si puede tokenizar videos.
Joint	Si puede tokenizar conjuntamente imágenes y videos en un espacio latente unificado.
Discrete	Si genera tokens discretos (índices enteros de un vocabulario).
Continuous	Si genera tokens continuos (vectores reales).

Physical AI systems: Cosmos-Reason1



[Cosmos World Foundation Model Platform
for Physical AI](#)

Physical AI systems: Cosmos-Reason1

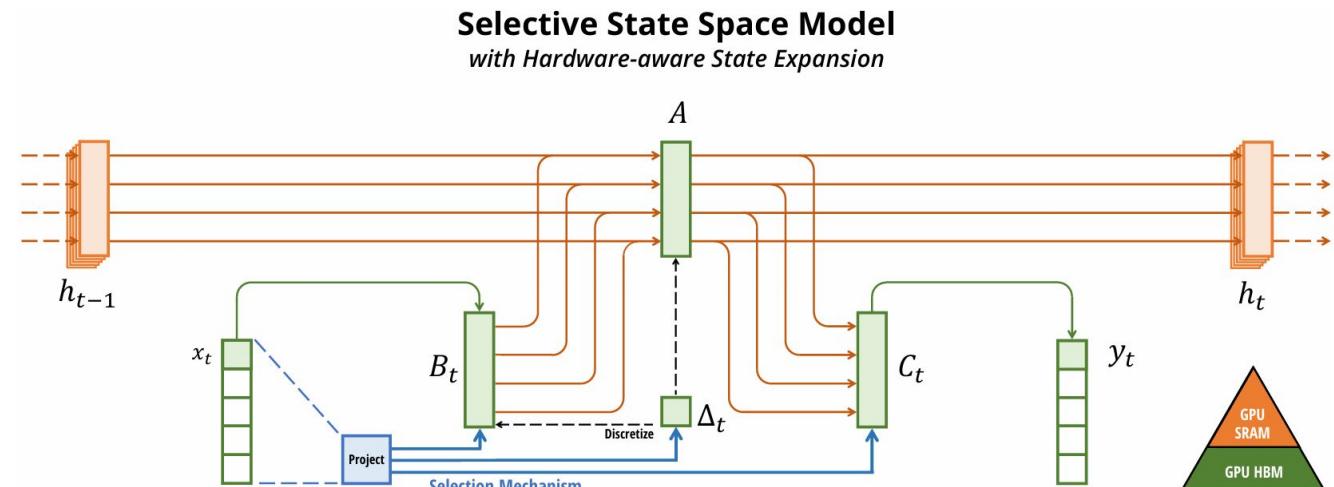


Mamba

Se basa en modelos de espacio de estados (SSMs), una estructura que permite procesar secuencias paso a paso, pero de forma **paralelizable y lineal en tiempo**.

Introduce un mecanismo selectivo y dependiente de la entrada, que le permite decidir dinámicamente qué información mantener, olvidar o actualizar.

No usa atención, pero captura dependencias a largo plazo de forma implícita y eficiente.



RRN (Recurrent Routing Network) → SSM (State Space Model) → Selective SSM → Mamba

Reemplaza la secuencia por multiplicaciones de matrices paralelas.

Permite procesamiento en paralelo sobre toda la secuencia.

Personaliza los filtros/matrizes para cada parte de la entrada

Implementa filtros selectivos y causales.

Usa operaciones tipo scan para simular dinámica recurrente con velocidad paralela.

Preguntas?