Día 10 · Tema 3 — Alternativas al Transformer: Mamba y MoE

Eduardo C. Garrido-Merchán · ecgarrido@comillas.edu Universidad Pontificia Comillas · ICADE

1. Por qué buscar alternativas

La atención escalada tiene coste $O(T^2)$ en tiempo y memoria, donde T es la longitud de la secuencia. Para $T = 10^5$ (un libro entero) la matriz $A \in \mathbb{R}^{T\times T}$ tiene 10^{10} entradas: *40 GB* en FP32. La inferencia también es cara: el KV-cache crece linealmente con T y se guarda en VRAM. Tres frentes:

- **Cuadraticidad en** $T \longrightarrow$ atención lineal, SSM.
- **Memoria por inferencia** $\longrightarrow$ *paged attention*, sliding window.
- **Capacidad vs. coste** $\longrightarrow$ MoE, modelos esparsos.

2. State Space Models y Mamba

Un SSM discreto modela una secuencia con una recurrencia lineal en el estado oculto $h_t \in \mathbb{R}^N$:

$$h_t \;=\; \bar{A}\, h_{t-1} \;+\; \bar{B}\, x_t, \qquad y_t \;=\; C\, h_t. \tag{1}$$

$\bar{A} \in \mathbb{R}^{N\times N}$	matriz de transición de estado (discretizada)
$\bar{B} \in \mathbb{R}^{N\times 1}$	cómo entra x_t al estado
$C \in \mathbb{R}^{1\times N}$	cómo se proyecta el estado a la salida
N	tamaño del estado oculto, *fijo* independiente de T

Es estructuralmente una RNN lineal. La novedad de **Mamba** (Gu & Dao, 2023) es hacer las matrices *dependientes de la entrada*: $\bar{A}_t, \bar{B}_t, C_t$ son funciones de x_t, lo que permite al modelo decidir qué guardar y qué olvidar.

2.1. Coste de Mamba

Tiempo: $O(T\,N)$ con N fijo (típicamente 16–64). Memoria: $O(N)$ porque el estado oculto no crece. Para $T = 10^6$ y $N = 16$: $1{,}6 \cdot 10^7$ operaciones, comparado con 10^{12} del Transformer denso. Compite hasta longitudes que el Transformer no puede manejar.

3. Mixture of Experts

En un MLP estándar todos los parámetros se aplican a todos los tokens. MoE introduce *esparsidad*: hay K *expertos* $E_1, \ldots, E_K$, y un *router* g elige un subconjunto pequeño por cada token:

$$\mathrm{MoE}(x) \;=\; \sum_{k=1}^{K} g_k(x)\; E_k(x), \quad g(x) = \mathrm{softmax}\big(W_g\, x\big) \text{ con } \textit{top-}r \text{ activados}. \tag{2}$$

E_k	el k-ésimo experto, un MLP independiente
$g_k(x)$	peso de *routing* hacia E_k (cero si no top-r)
K	número total de expertos (8, 16, 64, 128…)
r	cuántos expertos se activan por token (típicamente $r = 2$)

3.1. La aritmética de MoE

Parámetros totales $\sim K \cdot |E|$ pero **coste por token** $\sim r \cdot |E|$. Mixtral 8×7B: $K = 8$, $|E| = 7$ B, $r = 2$, total 56 B, activos 14 B. DeepSeek-V3: $K = 256$, $r = 8$, total 671 B, activos 37 B. *La memoria de inferencia escala con los totales* (hay que cargar todos los expertos en VRAM), *la latencia escala con los activos.*

4. Atención lineal

Otra ruta: aproximar la softmax con una factorización lineal. Para alguna *feature map* $\phi : \mathbb{R}^{d_k} \to \mathbb{R}^{d'}$,

$$\mathrm{Att}_{\mathrm{lin}}(Q,K,V) \;\approx\; \big(\phi(Q)\big)\Big(\phi(K)^{\top} V\Big), \tag{3}$$

con coste $O(T\,d\,d')$ en vez de $O(T^2 d)$. Variantes: Performer (kernels aleatorios), Linear Transformer ($\phi = \mathrm{ELU} + 1$), RetNet (decay exponencial).

5. Modelos híbridos

Lo más común en 2026 son arquitecturas mixtas: **Jamba** alterna bloques Transformer (para precisión local) con bloques Mamba (para contexto largo); **Griffin** mezcla atención local con recurrencias lineales; **StripedHyena** sustituye atención por convoluciones largas.

6. Concepto → aplicación

Familia	Caso ideal	KPI a optimizar
Transformer denso	Contexto $\leq$ 32K, calidad por defecto.	Calidad por token.
MoE	Asistente generalista, cloud con VRAM grande.	\$/token a calidad alta.
SSM (Mamba)	Genoma, registros médicos, libros, audio largo.	Memoria por petición.
Atención lineal	Streaming en edge, baja VRAM.	Tokens/seg en CPU.
Híbrido (Jamba)	Contexto $\geq$ 128K, calidad-coste balanceado.	Calidad $\times$ \$/token.

Idea clave. Las alternativas no *sustituyen* al Transformer; lo *complementan* cuando una dimensión específica (contexto, latencia, coste) deja de admitir el modelo denso.