Alternativas modernas al Transformer

Cuando $O(T^2)$ ya no es asumible.

Conceptos clave

- Tres frentes: cuadraticidad en longitud, memoria de KV-cache, capacidad vs coste.
- **State Space Models** (Mamba, Mamba-2): estado fijo $h_t \in \mathbb{R}^N$, recurrencia selectiva con $O(T)$.
- **Mixture of Experts** (Mixtral, DeepSeek-V3): router activa 2 de 8 expertos por token. Capacidad alta, coste por token bajo.
- Atención lineal (RWKV, RetNet) e híbridos (Jamba) cubren nichos específicos.

Ejemplo a mano

La aritmética de MoE. Mixtral 8×22B: 8 expertos × 22 B parámetros = **176** B totales. Pero el router solo activa **2** por token, así que el cómputo por inferencia equivale a ∼ 44 B activos.

$$\text{Factor de ahorro} = \frac{176}{44} = \mathbf{4{,}0\times}.$$

DeepSeek-V3 lleva esto al extremo: 671 B totales, 37 B activos por token (factor ≈ 18×). *Pero la memoria de inferencia crece con los totales*: MoE es para servidores con mucha VRAM.

Concepto → aplicación: cinco familias en 2026

Familia	Coste/token	Caso ideal
Transformer denso	$O(T)$ gen.	Cualquier tarea con contexto ≤ 32 K.
MoE	$O(T)$ ligero	Asistente generalista cloud.
SSM (Mamba)	$O(1)$	Genoma, libros, audio largo.
Atención lineal	$O(1)$	Edge, dispositivos con poca VRAM.
Híbrido (Jamba)	$O(T)$ local	Contexto 256 K+, balance.

Idea clave del Día 10. El Transformer denso sigue siendo la opción **por defecto**; Mamba y MoE son optimizaciones *cuando el caso lo exige* (contexto largo o coste por token bajo en cloud). Eliges familia justificando el KPI: calidad, latencia, contexto o coste.

Para llevarte: lectura Gu & Dao, *Mamba*; práctica: comparar Llama 3.3 (denso) vs. Mixtral (MoE) en latencia y calidad sobre un mismo prompt.