La consulta no puede ver la pregunta

El alcance de una convolución corta decide qué parte de una consulta condiciona la recuperación, y lo que queda afuera se vuelve una respuesta equivocada y confiada

Maximiliano Speranza
Investigador independiente, Buenos Aires, Argentina
ORCID 0009-0005-0413-8554
mrsperanza@frba.utn.edu.ar

Septiembre de 2026

Resumen

Los modelos de atención lineal y de espacio de estados arman su consulta con una convolución causal corta. El tamaño de núcleo 4 es el valor por omisión de hecho, así que la consulta de una capa resulta ser función del token actual y de tres anteriores. Mostramos que ese alcance no es un detalle de suavizado local sino un límite duro sobre *qué parte de una pregunta puede condicionar la recuperación*, y que lo que cae afuera no se degrada de a poco. Desaparece exacto, y el modelo contesta con confianza a partir de la parte que todavía alcanza a ver.

Medimos el efecto en un modelo de lenguaje chico con un archivo persistente co-entrenado que se consulta entre secuencias. La sensibilidad de la recuperación a un token de la consulta, medida como el movimiento en variación total de la distribución de lectura cuando ese token se reemplaza, vale **0,000000** apenas el token pasa el alcance de la convolución, y el corte se mueve con el núcleo. Con núcleo 3 (alcance 2) el escalón cae entre las distancias 2 y 3, y con núcleo 5 (alcance 4) entre 4 y 5. Sesenta celdas de sesenta, dos arquitecturas, tres semillas cada una, dos componentes de la consulta y cinco distancias.

La consecuencia en el comportamiento es una falla específica y frecuente. En nuestra tarea las preguntas tienen la forma *cuál es la <**relación**> de <**entidad**>*, con la entidad a distancia 1 de la posición de lectura y la relación a distancia 3. Con núcleo 3 la relación queda afuera de la ventana en el 100 % de las consultas, así que el modelo recupera solamente por la entidad. Cuando se le pregunta por una relación que nunca se enunció, sobre una entidad que sí apareció, responde con el hecho archivado equivocado en vez de abstenerse, y la abstención correcta queda en 0,5850 a 0,7349 sobre tres semillas. Ensanchar el núcleo a 5, que cuesta 1,024 parámetros sobre 865,651, la lleva a **0,9931** hasta **1,0000** sin solape entre condiciones, y la exactitud global a 0,988 hasta 0,993 contra un piso trivial de 0,4065.

Después verificamos el mecanismo afuera de nuestro propio modelo. En `mamba-130m-hf`, de 129 millones de parámetros, cambiar un token de cinco a ocho posiciones antes de la lectura mueve la salida de la capa en todas las distancias y deja la salida de la convolución en **cero exacto**, 80 celdas de 80. El estado ve la secuencia entera y la consulta que lo lee no. De paso, el tap más viejo de la convolución en ese checkpoint vale cero exacto en las 24 capas, así que su alcance efectivo es 2 y no el nominal 3. Informamos la medición y no una causa.

La profundidad cambia la ley sin derogarla, y medimos cuánto. En un modelo recurrente la combinación de dos componentes de la consulta ocurre donde los dos están disponibles, así que la distancia que decide es la que los separa entre sí. Sobre quince formas de pregunta, seis distancias y dos modelos —`mamba-130m` con 24 capas y `mamba-370m` con 48— el corte en la capa 0 es exacto y cae en el alcance medido en los dos, mientras que desde la capa 1 la recurrencia restituye la señal *atenuada*, a razón de 1,077 y 1,028 por token de distancia ($r = 0{,}978$ en los dos, curvas correlacionadas a 0,955). Tres veces los parámetros y el doble de capas dejan la tasa igual, lo que hace de la atenuación una propiedad de la arquitectura. Un control que deja fija la distancia y varía el relleno separa la distancia del contenido léxico.

Por último probamos si el acceso predice el comportamiento, y damos un negativo. Ajustar `mamba-130m` sobre la misma tarea con el componente discriminante afuera de la ventana cuesta +0,167, +0,181 y +0,141 de abstención correcta en el paso 100 —tres semillas de tres por encima de nuestro umbral pre-registrado— y **nada en absoluto para el paso** 400. Nuestro modelo chico, de cuatro bloques, no se recupera nunca; el de 24 capas se recupera rápido. **Donde no quedan capas con que pagar, la ventana pone un techo; donde quedan, pone un peaje.** La afirmación fuerte vale entonces para memoria consultada desde una capa temprana, y no como predicción de comportamiento para un modelo profundo entrenado hasta converger.

Por último preguntamos con qué frecuencia aparece esta configuración cuando nadie la está armando. Sobre cuatro corpus de preguntas —SQuAD, Natural Questions, TriviaQA y HotpotQA, 33.585 preguntas— entre el 90 % y el 100 % de las preguntas reales tienen sus partes discriminantes más separadas que el alcance medido, con las dos definiciones que se pueden calcular sin anotación, y empeora con la dificultad de la pregunta hasta llegar a **1,0000** en las multi-hop. La configuración que nuestra tarea sintética fue construida para estudiar es el caso ordinario, no un caso de borde.

Damos el diagnóstico como una receta que no necesita entrenamiento. Se cambia un token de la consulta y se mira si la recuperación se mueve. Si no se mueve, ese token es invisible para la búsqueda, y la confianza que el modelo muestre sobre él no está ganada.

1. El problema

Un modelo con memoria persistente tiene que hacer dos cosas que suelen medirse juntas y en realidad son separadas. Tiene que *encontrar* la entrada correcta, y tiene que *saber* cuándo la entrada que busca no está. La segunda es la que separa una memoria de un fabulador, y es la difícil de las dos.

Encontramos que en nuestro propio sistema la segunda falla tenía una causa puramente mecánica, aguas arriba de todo lo que el modelo aprende. La consulta con la que se busca en el archivo se arma con una convolución causal corta sobre los estados ocultos. En una pregunta de la forma *cuál es* `<art>` `<sust>` *de* `<entidad>`, la entidad queda un token antes de la posición de lectura y el sustantivo que nombra la relación queda tres tokens antes. Un núcleo de tamaño 3 alcanza dos tokens hacia atrás. La relación cae un token afuera de la ventana, de forma determinista, en todas y cada una de las consultas.

El modelo no estaba ignorando la relación. No la podía ver.

Contribuciones. Las enunciamos como afirmaciones, porque es lo que un lector tiene que poder contrastar contra la evidencia.

1. **El alcance de la convolución que forma la consulta es un corte duro, no una caída suave.** La sensibilidad de la recuperación a un token de la consulta es 0,000000 en cuanto ese token pasa el alcance, y el escalón se mueve con el kernel. Sesenta celdas de sesenta (Resultado 1), y la ablación del único tap que alcanza la relación lo reproduce (Resultado 3).

2. **Ensanchar la ventana un tap de cada lado compra abstención.** La abstención correcta en el caso difícil va de 0,5850–0,7349 a 0,9931–1,0000 sin solape en tres semillas, por 1.024 parámetros sobre 865.651 (Resultado 2).

3. **La disociación se sostiene en un modelo que no es nuestro.** En `mamba-130m` la salida de la capa se mueve a toda distancia mientras la de la convolución queda en cero exacto, 80 celdas de 80. El estado ve la secuencia entera y la consulta que lo lee no (Resultado 4).

4. **En un modelo profundo la ventana atenúa en vez de bloquear, a una tasa que no depende del tamaño.** 1,077 y 1,028 por token de distancia en modelos de 24 y 48

capas, $r = 0{,}978$ en los dos, con un control que separa la distancia del contenido léxico (Resultado 5).

5. **El acceso no implica el comportamiento, y lo informamos como negativo.** Un componente demostrablemente invisible para la consulta de una capa se aprende igual, sólo que más lento. El efecto es +0,16 en el paso 100 en tres semillas de tres y desaparece para el paso 400 (Resultado 6).

6. **La configuración es el caso común, no un caso de borde.** Sobre cuatro corpus de preguntas y 33.585 preguntas, entre el 90 y el 100 % de las reales tienen sus partes discriminantes más allá del alcance medido, y llega a 1,0000 en las multi-hop (Resultado 7).

7. **Un diagnóstico que no cuesta nada.** Reemplazar un token de la consulta y ver si la recuperación se mueve. No necesita entrenamiento, ni gradiente, ni etiquetas, y se aplica a cualquier modelo que consulte una memoria con una consulta formada localmente.

Lo que hicimos mal, y lo informamos igual. Un control que no podía dar otra cosa, y un titular que no sobrevivió al barrido de su umbral. Los dos están en el Resultado 5, con la prueba que caza cada clase de error.

2. Trabajo relacionado

Las convoluciones cortas están en todos lados y su tamaño se hereda, no se mide. Mamba-2 (Dao y Gu, 2024) usa por omisión una convolución causal depthwise de tamaño 4. En la implementación de referencia de Qwen3-Next, que usa Gated DeltaNet (Yang et al., 2025) en sus capas de atención lineal, la configuración expone `linear_conv_kernel_dim`, documentado como el tamaño de núcleo de la convolución de las capas de atención lineal y **con valor por omisión 4** (Hugging Face, 2026). Las ablaciones de esta línea informan que sacar la convolución empeora el modelo, lo cual establece que el contexto local importa. Ninguna informa qué se pierde cuando la ventana está presente pero es demasiado angosta para la consulta que hay que armar. Un alcance de tres tokens no es una configuración exótica, es lo que corre en modelos desplegados.

El resultado teórico más cercano. Li et al. (2024) estudian el recuerdo asociativo de n-gramas y prueban, por construcción, que alcanza con un filtro causal de largo N, igualando el filtro al n-grama. Es la misma familia de idea y la citamos como precursora. Tres cosas la separan de lo que informamos acá. Su consulta es un n-grama *contiguo*, los últimos N tokens sin nada irrelevante en el medio, mientras que lo que manda en nuestro caso es la *distancia* de un componente, con relleno entre las dos partes que importan. Su Teorema 1 es un resultado de existencia y no corren ningún experimento con el filtro demasiado corto. Y en la práctica informan lo contrario, al decir que la convolución sobre consultas y claves ayuda en las construcciones teóricas pero no da beneficios apreciables en los experimentos reales. Nosotros medimos un efecto grande en un régimen que ellos no prueban, que es una memoria persistente *entre secuencias* con una consulta compuesta.

Por qué el campo no lo pudo ver. La tarea canónica para esta capacidad, MQAR (Arora et al., 2024), tiene una consulta de *un solo token*. Se busca la clave y se devuelve el valor. Por construcción esa consulta no puede tener una parte adentro de la ventana y otra afuera. Zoology atribuye el 82 % de la brecha de recuerdo a que la convolución aplica un filtro fijo e independiente de la entrada, que es una afirmación sobre *adaptabilidad*. La nuestra es sobre *alcance*, y las dos son independientes. Nuestro modelo de núcleo 5 tiene exactamente la misma independencia de la entrada y la falla desaparece.

Del lado de la atención. La atención multi-token (Golovneva et al., 2025) convoluciona sobre los pesos de atención para que la atención pueda condicionarse en varios tokens a la vez, motivada por la observación de que un solo token no alcanza para localizar una coincidencia. Va en la misma dirección y confirma la nuestra por contraste. Nunca se pregunta qué parte de la consulta queda ciega, y no conecta la pregunta con la abstención.

3. Montaje experimental

El sustrato es un modelo de lenguaje de 3,5 MB, 865,651 parámetros, $d = 128$, cuatro bloques, entrenado desde cero sobre un idioma sintético cerrado de 242 tokens con un archivo persistente co-entrenado. Los hechos se enuncian en una secuencia, se revisan en otra y se consultan en una tercera, **con el estado recurrente reseteado entre secuencias**, así que la recuperación no se puede resolver arrastrando activaciones hacia adelante.

Las consultas se leen en el último token de la pregunta. La lectura se inyecta en el bloque 0 **antes del mixer**, y eso importa para el mecanismo. En ese punto el estado oculto todavía es la embedding del token, sin ninguna mezcla recurrente, así que la ventana de la convolución es literalmente todo lo que la consulta puede ver. Eso también explica por qué el efecto de abajo puede ser cero exacto y no apenas chico.

Se puntúan por separado cuatro categorías de respuesta. `vigente` y `anterior` son preguntas con respuesta en el archivo. `nose_ent` pregunta por una entidad que nunca se mencionó, que es la ausencia fácil. `nose_rel` pregunta por una relación que nunca se enunció *sobre una entidad que sí apareció*, que es el caso que se parece a una alucinación real, porque la recuperación encuentra un hecho vecino y genuino.

4. Resultado 1. La ventana es una función escalón, y afuera vale cero exacto

Medimos la sensibilidad reemplazando un componente de la pregunta por otro del mismo tipo y registrando la distancia en variación total entre las distribuciones de lectura. Para variar la distancia sin tocar la pregunta, la posición de lectura se corre r lugares sobre el relleno que ya sigue al signo de pregunta, que es exactamente equivalente a agregar r rellenos.

modelo	núcleo (alcance)	d=1	d=2	d=3	d=4	d=5	d=6
`v3` (3 semillas)	3 (2)	0,71–0,74	0,15–0,25	**0,000000**	**0,000000**	**0,000000**	**0,000000**
`kq3` (3 semillas)	5 (4)	0,36–0,46	0,10–0,15	0,05–0,35	0,07–0,33	**0,000000**	**0,000000**

Sesenta celdas de sesenta coinciden con la predicción hecha antes de correr, que la sensibilidad es positiva mientras $d \leq$ alcance y cero más allá. Una celda daba al principio 0,0106 donde iba un cero. La causa era que la formulación por el valor anterior es dos tokens más larga, así que correr la posición de lectura la recortaba contra el borde del tensor, y recortar *acorta* la distancia que se está midiendo. Esas muestras ahora se descartan en vez de recortarse y la celda pasó a 0,000000.

5. Resultado 2. Ensanchar la ventana compra abstención

unidad	núcleo	`vigente`	`nose_ent`	`nose_rel`	abstención falsa
`kq3_s0`	5	1,0000	0,9538	**0,9931**	0,0000
`kq3_s1`	5	0,9964	0,9726	**1,0000**	0,0030
`kq3_s2`	5	1,0000	0,9356	**1,0000**	0,0000
`v3_s0`	3	1,0000	0,9851	0,6090	0,0000
`v3_s1`	3	1,0000	1,0000	0,5850	0,0000
`v3_s2`	3	0,9927	0,9414	0,7349	0,0064

No hay solape entre condiciones en `nose_rel`. La exactitud global, contando una abstención correcta como acierto, va de 0,988 a 0,993 con núcleo 5 contra un piso trivial de 0,4065, que es lo que saca un modelo que se abstiene en todo.

El costo existe y lo declaramos. `nose_ent`, el caso fácil, baja un poco, de 0,941–1,000 a 0,936–0,973. Una lectura que informe solamente un número agregado de abstención esconde ese intercambio.

No es capacidad. El modelo de núcleo 5 agrega 1,024 parámetros, el 0,12 %, y el control de núcleo 3 ya recupera la entrada correcta con probabilidad 1,0000. Lo que cambia es el acceso, y el Resultado 1 lo mide directo.

6. Resultado 3. Apagar el tap que alcanza la relación

Apagar de a un tap por vez en el modelo de núcleo 5 ya entrenado, sin reentrenar nada, daña `vigente` en 0,483, 0,582 y 0,530 para el tap que alcanza la relación, contra 0,372, 0,013 y 0,195 para el tap que alcanza una palabra funcional y 0,218, 0,093 y 0,157 para el que alcanza el artículo. El tap de la relación tiene la magnitud aprendida *más chica* de los cinco, así que por unidad de peso cuesta cerca del doble que cualquiera de los dos controles.

Un criterio preregistrado formulado sobre la métrica de abstención falló, e informamos por qué. Apagar el tap saca la relación de *todas* las consultas, así que el modelo se abstiene más, y una métrica que premia abstenerse no puede bajar. El daño aparece en la exactitud y en la abstención falsa.

7. Resultado 4. La disociación se sostiene en un modelo real

Todo lo anterior está medido en un modelo chico, así que verificamos el mecanismo en uno que no es nuestro. `state-spaces/mamba-130m-hf`, 129,135,360 parámetros, en CPU y sin entrenar nada. En Mamba la convolución corta se aplica a x *antes* de calcular B, C y Δ, y C_t es el análogo de la consulta de lectura.

Cambiamos un solo token a distancia d de una posición de lectura y medimos, en la capa 0, el movimiento de la salida de la convolución y el de la salida de la capa. Diez posiciones de lectura, dos textos.

	d=1	d=2	d=3	d=5	d=7	d=8
salida de `conv1d`	0,8–3,7	0,5–1,6	**0,0**	**0,0**	**0,0**	**0,0**
salida de la capa, mediana	0,164	0,043	0,022	0,018	0,040	0,012
rango en las 10 celdas	0,09–0,96	0,02–0,06	0,01–0,05	0,01–0,04	0,01–0,10	0,01–0,02

La salida de la capa se mueve en *todas* las distancias, 80 celdas de 80, y el movimiento más chico de las ochenta es $6{,}3 \times 10^{-3}$, así que el estado sí ve la secuencia entera. La salida de la

convolución vale cero exacto más allá de su alcance. **Eso es la disociación.** El modelo ve el contexto y la consulta que lee su estado no lo ve.

Un hallazgo incidental, informado con su incertidumbre. Habíamos predicho movimiento en $d = 3$, porque el núcleo es 4. Vale cero exacto. La razón es que **el tap más viejo de la convolución vale cero exacto en las** 24 **capas**, lo que confirmamos por los pesos y también por intervención, ya que cambiar un token mueve la salida de la convolución en exactamente tres posiciones consecutivas y no cuatro. La ventana efectiva de este checkpoint es 3 y no 4, y el alcance es 2. No sabemos la causa. Un peso que vale cero exacto en 24×1536 entradas no sale de un gradiente, así que un artefacto de conversión es plausible, pero no lo verificamos y no lo afirmamos. El hecho se reproduce en tres líneas. Su consecuencia práctica, si generaliza, es que el núcleo nominal es una cota superior del alcance y conviene medirlo en vez de suponerlo.

8. Resultado 5. En un modelo profundo la ventana atenúa en vez de bloquear, y la tasa es la misma en dos tamaños

El Resultado 4 midió el acceso en la capa 0 de un modelo real. La objeción obvia es que un modelo recurrente profundo tiene muchas capas para mover la información hacia adelante, así que un cero duro en una capa no tiene por qué importar. Lo medimos de frente, y la respuesta es más útil que cualquiera de los dos extremos.

Qué se mide, y por qué acá la distancia es entre las dos *partes*. En nuestro modelo chico la consulta se forma en el último token y lee un archivo externo, así que la distancia que decide es la de cada componente a la posición de lectura. Un modelo de lenguaje recurrente no tiene archivo externo: la memoria es el estado, se actualiza en *cada* posición, y cada token tiene su propio turno de condicionar la búsqueda cuando entra. Para responder hay que *combinar* la entidad y la relación, y esa combinación puede ocurrir en cualquier posición donde las dos estén disponibles. La distancia que decide es entonces la que separa la relación de la entidad, y medimos el movimiento de la salida de `conv1d` *en la posición de la entidad* al reemplazar el token de la relación.

Montaje. Quince formas de pregunta que abarcan seis distancias relación–entidad, ocho contextos al azar de cuatro hechos cada uno, todas las capas, en `state-spaces/mamba-130m-hf` (24 capas) y `state-spaces/mamba-370m-hf` (48 capas). El alcance medido de la convolución es 2 en los dos. La atenuación se informa contra el promedio de las formas con d=2.

	d=2	d=3	d=4	d=5	d=6	d=7
capa 0 (los dos modelos)	se mueve	**0,0**	**0,0**	**0,0**	**0,0**	**0,0**
atenuación en la capa 1, `130m`	1,10	2,31	3,06	4,64	4,65	6,92
atenuación en la capa 1, `370m`	1,04	3,21	3,17	4,47	5,72	6,47

- **En la capa** 0 **el corte es exacto y cae en el alcance medido**, en los dos modelos: d=2 se mueve y $d \geq 3$ vale cero exacto. Es el Resultado 1 reproducido en un modelo que no es nuestro y a otra profundidad.
- **Desde la capa** 1 **ya no es cero.** La recurrencia sí transporta la relación hacia adelante. Pero la señal llega atenuada, y la atenuación crece con la distancia: el ajuste lineal da 1,077 por token (r = 0,9784) en `130m` y 1,028 por token (r = 0,9777) en `370m`. Las dos curvas correlacionan a 0,9549.
- **Tres veces los parámetros y el doble de capas no cambian la tasa.** Eso hace de la atenuación una propiedad de la arquitectura y no de un checkpoint particular.

El control que adjudica, y uno que no. Si la atenuación dependiera de qué palabras hay en el medio en vez de cuántas, el efecto sería léxico y no arquitectónico. Por eso dejamos fija la distancia y variamos el relleno: con d=5, cuatro rellenos distintos (*of the person named*, *of that other person*, *in the file for*, *of our dear friend*) se dispersan por un factor de 1,98, contra un rango de 1,10 a 6,92 entre distancias. La distancia decide; el relleno no.

Informamos también un control que falló, por una razón instructiva. Primero probamos dejar d=2 y alargar la pregunta *después* de la entidad. Devolvió valores idénticos a la condición base hasta el último decimal en las 24 capas, y tenía que hacerlo: nada que venga después de una posición puede afectar a esa posición. Era causalidad disfrazada de control, y vale la pena dejar escrita la prueba que caza esta clase de error antes de que lleguen los datos — *si la hipótesis fuera falsa, ¿este control podría dar distinto?*

Un titular que descartamos. Leer la capa en la que la atenuación baja por primera vez de 1,5 da $0, 9, 9, 13, 19, 21$ para $d = 2 \ldots 7$, o sea unas cuatro capas por token de distancia, con $r = 0{,}971$. Al barrer el umbral de 1,2 a 2,5 esa pendiente va de 3,97 a 1,54 y el orden se rompe, así que la cantidad es un artefacto del umbral y no se informa. La atenuación en la capa 1 no depende de ningún umbral, y es lo que dice el párrafo de arriba.

Qué cambia esto sobre la afirmación. La forma fuerte de la ley — cero exacto fuera del alcance — vale *por capa*, y vale en la capa donde se inyecta una lectura. En un modelo profundo se convierte en una ley de atenuación: la ventana no decide qué puede llegar a combinar el modelo, decide cuánto cuesta esa combinación. Ésta es la versión cuantitativa de la limitación que en el primer borrador declaramos sin medir, y precisa dónde se aplica la afirmación fuerte: a la memoria consultada desde una capa temprana, donde no quedan capas con que pagar.

9. Resultado 6. La ley de acceso no se vuelve una falla de comportamiento cuando el modelo tiene capas con que pagar

Los Resultados 1 a 5 miden *acceso*. Éste pregunta si el acceso predice el *comportamiento*, y la respuesta es un negativo limpio que nos parece la parte más útil del trabajo, porque acota la afirmación en vez de extenderla.

Montaje, pre-registrado antes de correr. Ajustamos `state-spaces/mamba-130m-hf` sobre la misma tarea, cuatro hechos en el contexto, la respuesta de un solo token, y evaluamos en la forma con la que el modelo se entrenó. Tres condiciones, tres semillas cada una, 800 pasos, 512 ejemplos por evaluación: `cerca` entrena y evalúa en d=2, donde relación y entidad comparten ventana en la capa 0; `lejos` en d=5, donde el movimiento en la capa 0 es cero exacto; `lejos+cerca` mezcla las dos. El criterio bloqueante era `vigente` $\geq 0{,}90$ y el principal `cerca` − `lejos` $\geq 0{,}10$ en ≥ 2 de 3 semillas.

El resultado, como curva, porque el punto final solo lo escondería.

paso	`cerca`	`lejos`	`cerca` − `lejos`, por semilla		
100	0,9850	0,8221	**+0,167**	**+0,181**	**+0,141**
200	0,9881	0,9209	+0,095	+0,106	+0,000
300	0,9881	0,9489	+0,095	+0,011	+0,012
400	0,9881	0,9886	−0,012	+0,011	+0,000
800	0,9485	0,9921	−0,131	+0,000	+0,000

En el paso 100 el efecto está, es grande y es unánime: tres semillas de tres por encima del umbral pre-registrado, con una media de +0,163. Para el paso 400 ya no está. **La ventana**

decide cuánto tiene que gastar el modelo para aprender la discriminación, no si puede aprenderla.

Cómo se leen los criterios, y se leen tal como están escritos. En el horizonte de 800 pasos las tres semillas de `lejos` terminan en $\geq 0{,}95$, así que la guarda de techo que declaramos por adelantado se dispara y el criterio principal es **no evaluable**: no queda margen para que `cerca` sea mejor. El criterio de velocidad, un área bajo la curva declarada post-hoc después de un piloto pero antes de esta campaña, da +0,022, +0,040, +0,021 — tres signos positivos de tres, y todos bastante por debajo del 0,10 que pedía. **No cumple**. Mezclar formas no compró nada acá (+0,013, +0,009, −0,003), y el criterio de no daño sí cumple (falsa abstención $\leq 0{,}0063$).

El área bajo la curva entera era el estadístico equivocado para un efecto transitorio: promedia ocho puntos de los cuales sólo los dos primeros traen señal, así que diluye +0,16 en +0,03. Lo informamos como lección y no como rescate — el criterio queda como está escrito y no cumple.

Por qué esto es coherente con el Resultado 5 y no una refutación. El Resultado 5 midió que el corte de la ventana es exacto en la capa 0 y que la recurrencia restituye la señal en las capas siguientes, atenuada pero presente. Un modelo de 24 capas tiene entonces con qué pagar el peaje, y lo que observamos es exactamente el pago: un arranque más lento y después paridad. El contraste con nuestro modelo chico es el punto, y es un contraste controlado, porque los dos vehículos difieren en profundidad:

	profundidad	efecto de dejar la parte discriminante afuera de la ventana
modelo chico, 26.000 pasos	4 bloques	**permanente**: 0,544, 0,678, 0,506 contra 0,993, 0,980, 1,000
`mamba-130m`, 800 pasos	24 capas	**transitorio**: +0,16 en el paso 100, cero desde el 400

Donde no quedan capas con que pagar, la ventana pone un techo. Donde quedan, pone un peaje. La afirmación fuerte de este trabajo vale entonces para memoria consultada desde una capa temprana — lecturas aumentadas por recuperación inyectadas antes del mixer, adaptadores poco profundos, cabezas de memoria de una sola capa — y no vale, como predicción de comportamiento, para un modelo profundo entrenado hasta converger en la tarea.

Lo que esto no nos autoriza a concluir. Que el efecto sea despreciable en general. Nuestra tarea tiene cuatro hechos y satura, el horizonte es de 800 pasos, y la distancia que pudimos sondear llega hasta $d{=}7$ porque la atenuación se mide sobre una plantilla fija. Una distancia mayor, una tarea más difícil o un modelo menos profundo podrían mover esto. Lo que el negativo sí establece es que *medir el acceso no alcanza*: un componente demostrablemente invisible para la consulta de una capa puede igual gobernar la respuesta, y cualquier afirmación de que un modelo «no puede usar» ese componente hay que probarla en el comportamiento y no deducirla de la convolución.

10. Resultado 7. En preguntas reales, las partes discriminantes casi siempre caen afuera de la ventana

Todo lo anterior se midió sobre un idioma sintético cerrado donde el generador fija la distancia. La pregunta obvia es cómo se ve esa distancia cuando nadie la fija. La medimos.

Qué se mide. La distancia X entre las partes de una pregunta que hay que combinar, en tokens del BPE de Mamba. Esa distancia hay que *definirla*, y la definición decide el número, así que informamos tres y creemos sólo lo que sobrevive a las tres:

- X_1, el tramo de la primera a la última palabra de contenido;

- X_2, de la última palabra de contenido al final de la pregunta;
- X_3, del interrogativo al último token capitalizado, como aproximación a la entidad.

Las palabras de contenido son todo lo que queda fuera de una lista cerrada de palabras funcionales. Cuatro corpus, elegidos por diferir en quién escribió las preguntas y para qué.

corpus	quién escribió las preguntas	$P(X_1 > 2)$	$P(X_3 > 2)$
SQuAD	anotadores mirando un párrafo	0,9625	0,9045
Natural Questions	**búsquedas reales de Google**	**0,9978**	—
TriviaQA	trivia escrita por humanos	0,9943	0,9104
HotpotQA	multi-hop, dos saltos inferenciales	**1,0000**	0,9624

Los n de X_1 son 10.570, 3.610, 12.000 y 7.405 preguntas. Los de X_3 son 6.610, —, 8.326 y 5.269, o sea entre el 62 % y el 71 % de cada corpus, porque X_3 queda definida sólo cuando la pregunta tiene un interrogativo y después un token con mayúscula; en Natural Questions, que está todo en minúscula, no queda definida nunca. El umbral es el alcance que *medimos* en Mamba, que es 2. Con el alcance nominal de 3 los valores de X_1 son 0,9144, 0,9900, 0,9837 y 0,9997.

Tres cosas que esto dice. Replica. Entre el 90 % y el 100 % de las preguntas reales tienen sus partes discriminantes más separadas que el alcance de la convolución que forma la consulta, en cuatro corpus escritos con propósitos distintos. Natural Questions es el que más pesa, porque son consultas que la gente efectivamente escribió en un buscador y no ítems producidos para un benchmark.

Empeora con la dificultad. 0,9625 → 0,9943 → 0,9978 → 1,0000, de SQuAD a HotpotQA. De 7.405 preguntas multi-hop, ni una tiene sus partes de contenido dentro de dos tokens. El régimen donde uno más querría que el modelo razone con cuidado es aquel donde su consulta ve menos.

X_2 confirma el mecanismo en vez de contradecirlo. La última palabra de contenido queda a 0,18 tokens del final en Natural Questions y a 1,25–1,34 en los otros, así que la pieza final casi siempre está *adentro* de la ventana y todo lo demás afuera. Ésa es la geometría de nuestra tarea sintética —entidad adentro, relación afuera— apareciendo en lenguaje que nadie escribió para este experimento.

La definición acota el número por arriba, y decimos por cuánto. X_1 supone que las dos partes que hay que combinar son las palabras de contenido de los extremos, y eso la vuelve una cota superior. El supuesto contrario, que son las dos *últimas* palabras de contenido, es una cota inferior, y da 0,186, 0,211, 0,295 y 0,264. Esa versión hay que calcularla sobre *palabras* de contenido y no sobre piezas del BPE: en tres de los cuatro corpus los dos últimos *tokens* de contenido pertenecen a la misma palabra entre el 90 % y el 94 % de las veces, así que una versión en sub-tokens de la misma cantidad mide el ancho de una sola palabra y no acota nada. Cuál de los dos números es el correcto depende de qué dos partes discriminan de verdad, y nuestros corpus no anotan eso. X_3, la distancia del interrogativo a la entidad, es el proxy más cercano a la cantidad que queremos, y queda entre 0,90 y 0,96. Informamos el intervalo y no la punta que más nos conviene.

No hay forma cerrada, y importa hacia qué lado falla. Es tentador ajustar una normal y leer $1 - \Phi\left(\frac{\text{alcance}-\mu}{\sigma}\right)$. Ninguna de las doce distribuciones es normal: la asimetría va de +0,93 a +3,71 y el exceso de curtosis llega a +20,95, todas con cola pesada a la derecha, como corresponde a una distancia, que está acotada por abajo y no por arriba. La cola derecha es

precisamente la región $X >$ alcance de la que trata la cantidad, así que una normal ajustada al centro subestimaría justo la parte que decide. La distribución empírica no necesita ningún supuesto: los valores de arriba son conteos.

Lo que esto no muestra. Son aproximaciones sintácticas y no las verdaderas partes semánticas de una pregunta, y en Natural Questions X_3 no se pudo calcular porque las consultas de búsqueda no llevan mayúsculas. Más importante, esto mide la *geometría del idioma* y no el comportamiento de un modelo: el Resultado 6 estableció que estar afuera de la ventana demora el aprendizaje en vez de impedirlo. Lo que sí establece es que la configuración que nuestra tarea sintética fue construida para estudiar no es un caso de borde. Es el caso ordinario.

11. El diagnóstico, como receta

La medición del Resultado 1 no necesita entrenamiento ni gradientes. Para cualquier modelo que consulte una memoria con una consulta armada desde una ventana local.

1. Tomar una consulta con sus partes separadas, de modo que algún componente quede a distancia d de la posición de lectura.
2. Reemplazar ese componente por otro del mismo tipo.
3. Comparar las distribuciones de recuperación.

Si son idénticas, ese componente es invisible para la búsqueda, y la confianza que el modelo muestre sobre él no está ganada. Con un núcleo de 4, que es el valor por omisión en las capas de atención lineal desplegadas, el alcance es de tres tokens, así que cualquier parte de una pregunta que quede más atrás no entra en la consulta de esa capa.

Código y datos. Todo el código, los pre-registros congelados con su SHA256 y los JSON crudos de los que sale cada número de este trabajo están en `https://github.com/SperanzaMax/delta-archivo`. Los Resultados 4 y 5 no necesitan entrenamiento ni GPU: descargan un checkpoint público y lo sondean, y se reproducen en minutos en una notebook.

12. Limitaciones

El sustrato es un modelo chico sobre un idioma sintético cerrado, y la distancia entre la relación y la posición de lectura la fija el generador, así que el efecto acá es determinista donde en texto natural sería una distribución. La medición es de *acceso* y no de uso. Un token adentro de la ventana no necesariamente se usa, y de hecho el tap del artículo está adentro y aporta poco. Por último, la lectura en nuestro modelo se inyecta antes del mixer, que es lo que hace que el cero sea exacto, y por eso la afirmación fuerte se hace para memoria consultada desde capas tempranas. El Resultado 5 mide qué pasa cuando no es así: la recurrencia transporta, pero la señal llega atenuada a razón de un factor $\approx 1{,}05$ por token de distancia, replicado en dos profundidades. Lo que esa medición *no* dice es si la señal atenuada alcanza para la tarea. Es una medición de magnitud, no de utilidad, y en las capas altas la atenuación tiende a 1 sin que sepamos si eso significa que la información se recuperó o que la representación pasó a estar dominada por otra cosa.

13. Conclusión

Una convolución corta decide qué parte de una pregunta puede condicionar una lectura de memoria, y lo que cae afuera no llega débil. No llega. Es una propiedad de la arquitectura,

no cuesta nada comprobarlo, y es invisible desde afuera porque el modelo contesta con total confianza desde la parte que sí puede ver.

El resto del trabajo es sobre hasta dónde llega esa afirmación. Sobrevive salir de nuestro propio modelo, en dos checkpoints públicos de distinta profundidad, donde el corte de la primera capa es exacto y cae en el alcance medido y no en el nominal. Sobrevive cambiar el tamaño del modelo, porque la atenuación posterior a la primera capa corre a prácticamente la misma tasa con 24 y con 48 capas. Y se detiene, de una forma que podemos precisar, cuando el modelo tiene profundidad para gastar: un componente afuera de la ventana se aprende igual, sólo que más tarde.

Ese último resultado es el que conservaríamos si sólo pudiéramos conservar uno. Convierte una afirmación sobre acceso en una afirmación sobre costo, y el costo es algo que un arquitecto puede negociar. Donde no quedan capas con que pagar, la ventana pone un techo; donde quedan, pone un peaje. La memoria leída desde una capa temprana, un adaptador o una única cabeza de recuperación cae del primer lado de esa línea, y son exactamente los lugares donde hoy se le está atornillando recuperación a los modelos de lenguaje.

Nos parece que lo más útil de acá no es el arreglo sino la medición, porque la medición es gratis. Se cambia un token de la consulta, se mira si la búsqueda se mueve, y uno se entera de si el modelo tiene derecho a la confianza que está por expresar. Nos sorprendió cuántas veces la respuesta era que no, y nos sorprendió igual, después, que no poder ver algo resultara no ser lo mismo que no poder aprenderlo.

Referencias

Dao, T., y Gu, A. (2024). Transformers are SSMs. Generalized Models and Efficient Algorithms Through Structured State Space Duality. ICML 2024.

Yang, S., Kautz, J., y Hatamizadeh, A. (2025). Gated Delta Networks. Improving Mamba2 with Delta Rule. ICLR 2025. arXiv:2412.06464.

Documentación de Hugging Face Transformers, `Qwen3NextConfig`, parámetro `linear_conv_kernel_dim`, valor por omisión 4. Consultada el 2 de septiembre de 2026.

Li, M., Zhang, X., Huang, Y., y Oymak, S. (2024). On the Power of Convolution Augmented Transformer. arXiv:2407.05591.

Arora, S., Eyuboglu, S., Timalsina, A., Johnson, I., Poli, M., Zou, J., Rudra, A., y Ré, C. (2024). Zoology. Measuring and Improving Recall in Efficient Language Models. ICLR 2024. arXiv:2312.04927.

Golovneva, O., Wang, T., Weston, J., y Sukhbaatar, S. (2025). Multi-Token Attention. arXiv:2504.00927.