Dónde vive la abstención
Comparación pre-registrada de cuatro interfaces de abstención en un modelo de lenguaje chico con memoria versionada

Maximiliano Speranza
Investigador independiente, Buenos Aires, Argentina
ORCID 0009-0005-0413-8554
maximiliano.speranza@gmail.com

Agosto de 2026

Nota. Esta es la versión completa en español del trabajo, provista como material suplementario. El manuscrito principal está en inglés.

Resumen

Enseñarle a un modelo de lenguaje a decir «no sé» tiene al menos tres implementaciones publicadas, y nunca se comparan entre ellas. La incertidumbre puede ser un token del vocabulario (`[IDK]`), una cabeza binaria separada (SelectiveNet), o una entrada de la memoria que el modelo consulta (pointer sentinel, el puntaje de no-respuesta de SQuAD 2.0). Cada una nació en otra arquitectura, sobre otra tarea y contra otra línea de base. Nadie hizo la pregunta obvia, que es si manteniendo fijos el modelo, los datos, las semillas y el presupuesto de entrenamiento, *importa dónde vive la decisión de abstenerse*.

La contestamos en un modelo de lenguaje de 863.730 parámetros entrenado desde cero, con un archivo persistente co-entrenado y hechos versionados, donde la respuesta es un solo token y la métrica es por lo tanto exacta. Se comparan cuatro interfaces de a pares en 27 unidades de entrenamiento, con cada predicción congelada por hash antes de que existieran los datos correspondientes.

Importa, y el orden es estable. Una cabeza binaria separada, de **129 parámetros, el** 0,015 % **del modelo**, reduce la tasa de falsa abstención en un factor de 2 a 2,8× contra el token del vocabulario a presupuesto igualado en las tres semillas, y corre hacia abajo en $\approx$ 0,10 de margen de acierto la frontera donde la abstención se vuelve aprendible. O sea que con una cabeza propia, un modelo que recupera 10 puntos peor todavía puede aprender a callarse. Renormalizar el vector del token, que es la explicación que el propio paper de `[IDK]` da para su fallo en modelos chicos, **no ayuda**, y deja una brecha de +0,1465 de falsa abstención contra la cabeza. El mecanismo no es la norma del vector sino que dos decisiones de naturaleza distinta compiten por un mismo softmax.

El slot de memoria falla, y falla de forma informativa. Su masa de atención converge a 0,4074, 0,4046 y 0,4020 contra una tasa base empírica de preguntas sin respuesta de 0,4048, o sea que aprendió el prior y no la pertenencia. Un barrido de 400 umbrales no rescata nada, con J de Youden entre +0,038 y +0,078. Leído junto con un puntaje de matcheo del archivo que queda en el azar (AUC 0,4984), esto dice algo más filoso que «el slot no funcionó», y es que **en una memoria co-entrenada leída por softmax la ausencia no tiene representación, y darle un lugar donde vivir no alcanza para crearla**.

Decimos explícitamente qué *no* habilitan estos resultados. Lo que se sostiene es que *separar la cabeza hace aprendible la abstención en este régimen*, no que *el modelo sabe cuándo no sabe*. El techo que queda es de calibración y no de capacidad (AUC 0,777 a 0,998), y siete

intentos pre-registrados de obtener la decisión sin etiquetas cayeron todos en AUC 0,50 a 0,67.

Palabras clave. Predicción selectiva; abstención; alucinación; memoria en modelos de lenguaje; conocimiento versionado; pre-registro; resultados negativos

1. La pregunta que nadie hizo

Un modelo que contesta todo es un modelo que contesta mal cuando tendría que haberse callado. Las respuestas del campo a este problema caen en tres familias arquitectónicas, cada una con su cita canónica.

- **En el vocabulario.** Se agrega un token que significa «no sé» y la propia entropía cruzada empuja masa de probabilidad hacia él. Es el token `[IDK]` (Cohen et al., 2024), que hereda gratis toda la maquinaria de entrenamiento.
- **En una cabeza separada.** Se agrega una salida cuyo único trabajo es decidir si se contesta. Es la predicción selectiva. SelectiveNet (Geifman y El-Yaniv, 2019) entrena de punta a punta tres cabezas, la de predicción, la de selección y la auxiliar.
- **En la memoria.** Si el modelo lee de un almacén recuperable, se le da al almacén una entrada que significa «acá no hay nada». Son el pointer sentinel (Merity et al., 2016) y el puntaje de no-respuesta de SQuAD 2.0 (Rajpurkar et al., 2018).

No son tres versiones de un mismo experimento. Viven en arquitecturas distintas, atacan tareas distintas, y cada una se compara contra una línea de base sin abstención en vez de compararse entre sí. La comparación que le diría a alguien *cuál construir* no se corrió nunca.

Hay una razón por la que es difícil correrla en un modelo grande. Para atribuirle una diferencia a la interfaz hay que dejar todo lo demás fijo, o sea los mismos pesos preentrenados, el mismo orden de datos, el mismo estado del optimizador, el mismo presupuesto y varias semillas, y en un modelo donde una sola corrida es cara esa grilla no se puede pagar. Nosotros la corremos en un modelo lo bastante chico como para que la grilla se pague y las tripas se puedan mirar, y decimos con todas las letras qué compra eso y qué cuesta (§7).

Aportes.

1. La comparación pareada de cuatro vías en sí misma, con semillas y presupuesto igualados, sobre 27 unidades de entrenamiento, y con todos los veredictos emitidos por scripts congelados antes de que existieran los datos.
2. Una refutación *medida* de la explicación que Cohen et al. (2024) da para su propio fallo en modelos chicos. Ellos lo atribuyen a la inicialización del embedding de `[IDK]`, y nosotros implementamos la versión más generosa de esa hipótesis y falla exactamente igual que el token sin tocar.
3. La ubicación de la frontera de entrenamiento donde la abstención se vuelve aprendible, y cuánto la corre la arquitectura, que son $\approx 0{,}10$ de margen de acierto.
4. Un negativo sobre el slot de memoria con mecanismo identificado, la convergencia al prior, y la observación que lo acompaña, que es que el puntaje de matcheo del archivo no codifica presencia en absoluto.
5. Una declaración explícita del techo. Lo que queda es calibración y no capacidad, y siete vías sin etiquetas están cerradas.

2. Trabajo relacionado

Predicción selectiva. La opción de rechazo es vieja (Chow, 1970) y su forma profunda moderna es SelectiveNet (Geifman y El-Yaniv, 2019), que separa «¿contesto?» de «¿qué contesto?» en cabezas distintas entrenadas juntas. **La separación en sí no es nueva y este trabajo no la reclama.** Lo que SelectiveNet no es, es un modelo de lenguaje, ni tiene softmax de vocabulario, ni tiene memoria.

Abstención dentro del vocabulario. Cohen et al. (2024) agregan un token `[IDK]` y desplazan masa de probabilidad hacia él dentro de la misma entropía cruzada. Tres propiedades de ese trabajo importan acá. Nunca compara contra una cabeza binaria separada. No discute la competencia por la masa de probabilidad, aunque la mitiga de refilón con un hiperparámetro Π que capa en 0,5 la masa desplazable a `[IDK]` para que el token de oro «siga siendo competitivo», o sea que la competencia está pero se parchea sin nombrarla. Y reporta que el método *falla* en modelos muy chicos (pythia-70m, pythia-160m), y lo atribuye a «problemas de precisión numérica con la inicialización de `[IDK]`». Nuestro modelo vive justo en ese régimen, lo que vuelve contrastable la atribución (§5.2).

Abstención dentro de una interfaz de recuperación. El pointer sentinel (Merity et al., 2016) agrega un centinela que compite con la distribución de copia, SQuAD 2.0 (Rajpurkar et al., 2018) puntúa la no-respuesta contra los spans, y la detección de fuera de distribución por energía (Liu et al., 2020) pone umbral sobre el puntaje. **El slot nulo no es una invención de este trabajo.** Lo nuevo es correrlo como tercer brazo de un contraste controlado dentro de un modelo cuya memoria es co-entrenada en vez de externa.

Por qué la memoria lo hace más difícil y no más fácil. Joren et al. (2024) reportan que la recuperación aumentada *empeora* la abstención, porque el contexto agregado sube la confianza del modelo y produce más alucinación en vez de más «no sé». Un modelo con archivo más abstención trabaja contra ese efecto, no a favor.

Hechos versionados. Wang et al. (2025) documentan interferencia proactiva cuando una clave se actualiza varias veces en el contexto, y el atajo de recencia que aparece. No proponen ningún arreglo arquitectónico, evalúan sólo modelos grandes preentrenados y, lo que más nos importa, nunca piden el valor *superado*. Un banco que pide sólo el valor vigente no puede separar «prefiere lo último» de «usa el orden», porque el atajo contesta las dos.

3. El banco

Modelo. Un modelo de lenguaje entrenado *desde cero*, sin ningún componente preentrenado, de 863.730 parámetros, 3,5 MB, con un vocabulario cerrado de 242 tokens legibles por humanos. La arquitectura es una recurrencia con regla delta más un **archivo persistente co-entrenado** con sello de orden aprendido en la clave, leído por softmax e inyectado en el bloque 0. El archivo persiste entre sesiones mientras el estado recurrente se resetea, así que es el único puente posible de una sesión a otra.

Tarea. Los hechos se dicen en una sesión, se corrigen en otra, incluso con correcciones elípticas del tipo «no, es beto» que no nombran la entidad, y se preguntan en una tercera.

```
sesión 1  USUARIO el director de norte es ana
sesión 2  USUARIO no , el director de norte es beto
sesión 7  USUARIO quien dirige norte ?
          MODELO  beto
```

La respuesta es un **solo token**, lo que hace la métrica exacta y determinista, sin juez LLM ni parser interpretativo. Al modelo se le pide el valor vigente, el valor superado (`anterior`) y hechos que no están en el archivo, caso en el cual la conducta correcta es abstenerse. Durante el entrenamiento de abstención, el 40 % de las preguntas no tienen respuesta ($p_{\text{nose}} = 0{,}4$).

Métricas y compuerta. `vigente` es el acierto sobre el valor actual, `nose` es la tasa de abstención en preguntas sin respuesta, y `falsa_abst` es la tasa de abstención en preguntas que *sí* la tienen. La compuerta pre-registrada, fijada antes de cualquiera de las campañas que se reportan acá y nunca ajustada, es la siguiente.

Compuerta. `nose` $\geq 0{,}50$ **y** `falsa_abst` $\leq 0{,}10$

Las dos mitades se piden juntas, porque un modelo que contesta `NOSE` a todo saca `nose` = 1,000 y tiene que verse como el fracaso que es.

El atajo, y el margen sobre él. Con $p_{\text{nose}} = 0{,}4$, la política de *no abstenerse nunca* vale 0,5906 en este banco. Reportamos entonces el **margen sobre el atajo**, que es el `vigente` del modelo al cerrar la fase base menos 0,5906. Esa variable resulta gobernar si la abstención es aprendible siquiera (§5.4).

Protocolo. Cada campaña tiene un pre-registro congelado con su SHA antes de implementar, un archivo de desviaciones, y reporte por unidad. **Los resultados nunca se reportan sólo como media**, porque la bimodalidad entre semillas es una propiedad medida de este banco y un promedio taparía justamente lo que interesa.

Uso de un modelo de lenguaje. Todo el código de los experimentos, las campañas de entrenamiento y la generación de informes de este trabajo se hicieron con asistencia de un modelo grande de lenguaje. El diseño del estudio, los pre-registros y sus hashes congelados, la elección de los controles y la verificación de cada número reportado son del autor. El modelo no es autor y no responde por el trabajo.

4. Las cuatro interfaces

Las cuatro parten del *mismo* checkpoint base y reinicializan Adam, así que la fase de abstención está pareada.

condición	dónde vive la abstención	contraparte publicada
`token`	una entrada del softmax de valores	`[IDK]` (Cohen et al., 2024)
`escala`	ídem, con el vector renormalizado	la explicación del propio paper de `[IDK]`
`cabeza`	salida binaria separada (+129 params)	SelectiveNet (Geifman y El-Yaniv, 2019)
`slot`	entrada de la memoria (+256 params)	pointer sentinel (Merity et al., 2016)

`escala` merece una aclaración, porque es el control que podía ahorrar la arquitectura entera. Una inspección de pesos mostró a `NOSE` con una norma de embedding de entrada de 0,367 contra 1,011 y 1,028 de los tokens de valor, o sea compitiendo en el mismo softmax con un vector tres veces más corto. `escala` lo renormaliza a la norma media de los vectores de valor en entrada y en salida. **Si el problema fuera la norma, `escala` lo arreglaría y no haría falta arquitectura nueva.**

Una asimetría declarada. `slot` estrena 256 parámetros y `cabeza` estrena 129. La diferencia es de 127 parámetros sobre 863.730, el 0,015 %, y se declara acá en vez de descubrirse después, aunque como `slot` pierde, la objeción de que ganó por tener más parámetros nunca llega a plantearse.

5. Resultados

5.1. Vocabulario contra cabeza

Veintiuna unidades, 7 configuraciones por 3 condiciones, a 14.000 pasos. Sobre las cinco unidades *difíciles*, que son aquellas donde el modelo base tiene margen bajo sobre el atajo.

condición	pasa la compuerta
`token`	**0 de 5**
`escala`	**0 de 5**
`cabeza`	**4 de 5**

La única que falla, la de margen más bajo de la serie, se queda en `falsa_abst` = 0,1189 contra el 0,10 exigido. `nose` **nunca fue el problema** en ninguna condición, con valores de 0,54 a 0,93, o sea que todas detectan la ausencia. Lo que falla siempre es `falsa_abst`, que es callarse de más.

Un control de sanidad pre-registrado como prueba de no inferioridad *falló hacia arriba*. Se le pedía a `cabeza` mantener `vigente` dentro de ±0,05 de `token` y se pasa en 3 de 7 unidades (+0,0633, +0,0668, +0,0928), siempre recuperando *mejor*. Registramos el fallo como tal en vez de mover el umbral después de ver el número.

5.2. La norma no es el mecanismo

`escala` promedia 0,2394 de falsa abstención en las cinco unidades difíciles contra 0,0929 de `cabeza`, o sea una brecha de **+0,1465**, casi tres veces el 0,05 que pedía el pre-registro.

El negativo es limpio y no un fallo de implementación. La renormalización sobrevive al entrenamiento, porque entra en 1,000 por construcción y cierra en 0,899 de entrada y 1,131 de salida. La norma se quedó donde se la puso y aun así no alcanzó. Y una sonda de pesos explica *por qué* la premisa estaba mal de entrada. El 0,367 que motivó la condición es de la matriz de **entrada**, mientras que el softmax que decide la respuesta usa la de **salida**, y ahí `NOSE` ya era 1,77 a 2,05 veces *más largo* que un valor promedio en las cinco unidades difíciles, y 0,825 veces en las fáciles. `escala` entonces *subía* la norma en las unidades fáciles y la *bajaba* en las difíciles, o sea la misma intervención haciendo cosas opuestas según la unidad.

> **Esto es un desacuerdo directo y medido con Cohen et al. (2024).** Ellos atribuyen el fallo de `[IDK]` en modelos chicos a la inicialización de su embedding. En este banco la inicialización sí explica de dónde *sale* la asimetría, porque `NOSE` nunca se entrena durante la fase base y queda donde lo pusieron, pero no explica el *fallo*, porque corregirla explícitamente no cambia nada. Lo que queda es que dos decisiones comparten un mismo softmax.

5.3. Presupuesto, o qué estaba midiendo la compuerta y qué sobrevive

El resultado anterior se midió con todas las unidades a 14.000 pasos, y un seguimiento pre-registrado preguntó si `token` simplemente llega más tarde. En parte llega. Extendidas a 20.000 pasos, dentro del horizonte original de la tasa de aprendizaje y por lo tanto sin tocar el schedule, cuatro de las cinco unidades que fallaban cruzan la compuerta. La falsa abstención baja en las 5 de 5 y el Spearman contra el paso es negativo en las 5 de 5.

Retiramos entonces la formulación, con las palabras del propio pre-registro. La frase «`cabeza` pasa la compuerta donde las otras fallan» deja de ser cierta apenas las otras reciben 6.000 pasos más. Lo que sobrevive es la comparación pareada a presupuesto igualado.

`falsa_abst` a 20.000	`cabeza`	`token`	razón
semilla 0	**0,0633**	0,1296	2,0×
semilla 1	**0,0421**	0,0713	1,7×
semilla 2	**0,0166**	0,0458	2,8×

> **El presupuesto explica el cruce de la compuerta, no la ventaja.** Una compuerta es un umbral, y un umbral convierte una diferencia continua en un sí o no que depende de dónde esté puesto. Lo que no depende del umbral es que la diferencia sigue ahí, con el mismo signo, en las 3 de 3 semillas.

No hubo intercambio. `vigente` sube en las cinco unidades durante la misma extensión, hasta +0,32. El modelo no compró abstención a costa de contestar peor.

5.4. La frontera, y cuánto la corre la arquitectura

Que la abstención sea aprendible o no lo gobierna el margen sobre el atajo, y esto importa en la práctica, porque **si hay que entrenar un modelo casi hasta la perfección antes de poder enseñarle a decir «no sé», el método no sirve para nada real**, ya que ningún modelo útil llega a acierto 1,0 en su dominio.

Trece unidades por condición, que cubren el rango de margen cortando corridas base *por valor de* `vigente` en vez de por número de paso, y entrenando 2.000 pasos de abstención desde cada corte.

	el corte cae entre	Spearman(margen, `falsa_abst`)
`token`	+0,2358 y +0,2826	−0,8033
`cabeza`	+0,1489 y +0,1672	−0,8886
la cabeza corre la frontera ≈ 0,10 de margen hacia abajo		

Trece unidades ordenadas por margen **sin una sola inversión**, y la transición es un salto y no una pendiente. Entre dos puntos separados por 0,047 de margen, `falsa_abst` cae de 0,2237 a 0,0855, un factor de 2,6×. En la banda de +0,1672 a +0,2358, `token` falla en 4 de 4 y `cabeza` pasa en 4 de 4.

El número operativo. Con cabeza propia alcanza con entrenar hasta un margen de ≈ +0,17, y con el token del vocabulario hay que llegar a ≈ +0,26. **Con** 129 **parámetros se le puede enseñar a callarse a un modelo que recupera** 10 **puntos peor.**

Un confound, y la celda que lo resolvió. Margen y grado de entrenamiento quedan entrelazados en el rango muestreado, porque todos los puntos de margen alto vienen de corridas truncadas y todos los de margen bajo de corridas completas. La celda que falta es un modelo entrenado a fondo que igual quede trabado en margen bajo. Una semilla no llegó a producir cortes y entregó exactamente esa celda, o sea una corrida de tarea fácil, entrenada a fondo, con margen +0,2124, y un pre-registro congeló dos predicciones excluyentes antes de medirla. `token` falló (0,1885) y `cabeza` pasó (0,0902), así que **el margen no era un proxy de la dificultad de la tarea**. Con una salvedad que decimos en vez de enterrar, y es que el veredicto binario de `cabeza` depende del último tick. Lo robusto no es ese binario sino el contraste pareado, con 7 de 7 puntos válidos en la misma dirección (prueba de signos, $p = 0{,}0078$, medias 0,2718 contra 0,1430).

5.5. El slot de memoria converge al prior

Se agregan dos vectores al archivo, $k_\varnothing$ y $v_\varnothing$, que compiten como una columna más de la lectura. Siguiendo el diseño, al nulo *nunca* se le exige ganar la lectura, porque está medido que el modelo contesta bien con la entrada correcta en el puesto 2, sino sólo llevar masa relativa. Tres semillas, 26.000 pasos, pareadas contra controles con `cabeza`.

	semilla 0	semilla 1	semilla 2
acierto (`cabeza` → `slot`)	0,9705 → 0,8991	0,7769 → 0,7859	0,8351 → 0,8999
`nose` (`cabeza` → `slot`)	0,9119 → **0,0000**	0,5416 → **0,0000**	0,7298 → **0,0000**

El modelo entrena, porque el criterio bloqueante de sanidad pasa 3 de 3, y después no se abstiene nunca, en ninguna semilla. Un cero limpio es la firma de un artefacto, y acá lo fue la primera vez, porque un script de evaluación probaba `==` donde necesitaba `in` y no podía emitir abstención en absoluto. El arreglo *sube* el acierto en las tres unidades (0,7572 a 0,8991, 0,4307 a 0,7859, 0,7744 a 0,8999), o sea favorece al slot, y `nose` sigue dando cero. Un control de regresión reprodujo bit a bit todas las unidades ya publicadas.

El mecanismo se ve en la propia masa de atención.

masa sobre el slot nulo	semilla 0	semilla 1	semilla 2
preguntas *con* respuesta	0,4074	0,4046	0,4020
preguntas *sin* respuesta	0,4086	0,4065	0,4037
AUC (sin contra con)	0,5190	0,5313	0,5182
tasa base empírica de preguntas sin respuesta, 829/2048 = **0,4048**			

> **El slot aprendió el prior.** 0,4074, 0,4046 y 0,4020 contra una tasa base de 0,4048 es el óptimo exacto de la entropía cruzada binaria para un predictor sin señal utilizable. Y como el prior queda por debajo de cualquier umbral de decisión razonable, que nunca se abstenga no es un corte mal elegido sino la *consecuencia necesaria* de haber aprendido la tasa base en vez de la pertenencia.

Probamos igual la objeción del umbral, barriendo 400 cortes sobre la masa. Ningún umbral pasa la compuerta en ninguna semilla, y el mejor punto alcanzable abstiene en ≈ 80 % de las preguntas sin respuesta y en ≈ 75 % de las que sí la tienen, con J de Youden entre +0,038 y +0,078, que es ruido. Las dos poblaciones no se separan, y por eso ningún corte puede separarlas.

Dos observaciones que vale registrar. La primera es un riesgo declarado que *no* se materializó. El slot se lleva ≈ 40 % de la masa de lectura en todas las consultas y el acierto se sostiene en 0,90, 0,79 y 0,90, sin que el acierto sobre el valor superado se caiga, e incluso subiendo +0,111 en una semilla. La lectura tolera perder cuatro décimos de su masa a un vector constante. La segunda es que $v_\varnothing$ partió de cero exacto y terminó con norma 1,34 a 1,71, o sea que el modelo no lo usó como compuerta pura sino que le aprendió un valor, inyectando un vector fijo en toda respuesta, lo que funciona como un sesgo aprendido de la lectura y no como una señal de ausencia.

5.6. El resultado que enmarca el fallo del slot

El negativo del slot no es una decepción aislada. De forma independiente y anterior, el propio puntaje de matcheo del archivo, que es el máximo del match de la consulta contra las claves guardadas, separa «la respuesta está» de «la respuesta no está» con AUC 0,4984 y 0,5022, o sea el azar exacto a dos decimales. El margen top-2 y el logsumexp dan lo mismo. Tres controles

capaces de fallar no fallaron, ya que los logits reconstruidos con el puntaje extraído coinciden bit a bit con los del modelo, el puntaje varía, y el archivo se usa. Re-medido en la posición de *máximo foco*, donde la lectura concentra hasta el 65 % de su masa en una sola entrada, sigue en el azar (0,5007 y 0,5077).

> **El modelo selecciona una entrada con fuerza, y la fuerza de la selección no codifica si lo que buscaba está. Siempre encuentra algo, y encuentra con la misma convicción cuando no hay nada que encontrar.**

Es la consecuencia esperable del mecanismo. Una lectura por softmax suma uno, así que la mejor entrada gana con masa comparable haya o no haya un buen candidato, y nada en el entrenamiento presionó jamás para que la magnitud del match signifique «está». **Un recuperador externo tiene gratis el conjunto vacío, y meter el índice adentro de la red lo destruye**, y como muestra la §5.5, darle a la ausencia un lugar donde vivir no alcanza para restituirlo.

6. Lo que no afirmamos

Tres frenos, escritos porque los resultados invitan a excederse.

Esto es calibración, no conocimiento. Lo que se sostiene es que *separar la cabeza hace aprendible la abstención en este régimen.* El techo que queda es de calibración, ya que la AUC del logit de la cabeza va de 0,777 a 0,998, o sea que la información para decidir está y lo que está mal puesto es el corte. En la única unidad que falla la compuerta, un umbral elegido en una mitad de los datos y juzgado en la otra la pasa. La frase *«el modelo sabe cuándo no sabe»* no la sostiene nada de lo que hay acá.

Todo lo utilizable pasa por etiquetas. Siete intentos pre-registrados de obtener el corte sin supervisión, entre ellos una mezcla de gaussianas sobre el logit, un umbral de dispersión sin calibrar, un monitor de desacuerdo interno bajo enmascarado y un detector de empate de claves, cayeron todos en AUC 0,50 a 0,67, contra 0,777 a 0,998 de la cabeza supervisada. Dos de ellos fallaron por una razón que vale reportar, porque acota lo que un detector futuro tiene que medir. En este banco el modelo **nunca fabrica contenido**, ya que las respuestas fuera del archivo ocurren con tasa 0,0000 en los cuatro niveles de dificultad, y toda respuesta errada es un valor real del archivo puesto en la entidad equivocada. Entonces, cuando una pregunta no tiene respuesta, el modelo no adivina en el vacío sino que se ancla en *otra entrada real*, y su respuesta es tan estable ante el enmascarado como una correcta. El monitor de desacuerdo confirmó que el modelo lee el archivo, porque el 98 a 99 % de las respuestas correctas cambian al tapar las entradas que las sostienen, y aun así no pudo discriminar, porque **el desacuerdo mide si la respuesta vino del archivo, no si vino de la entrada *correcta***. Cualquier detector sin etiquetas futuro tiene que separar esas dos cosas.

Acá la alucinación es mala atribución, no invención. El párrafo anterior tiene una consecuencia general para la detección. En un modelo con archivo, el modo de fallo es un valor verdadero puesto en la entidad equivocada. Eso es *más* difícil de cazar que la invención, porque cualquier verificación del tipo «¿este dato existe?» lo da por bueno. Declaramos el alcance, ya que con vocabulario cerrado la fabricación libre está excluida por construcción, así que lo que se mide es que la salida queda confinada al contenido archivado.

7. Límites

- **Escala.** Son 863.730 parámetros y un idioma sintético de 242 tokens. Nada de esto transfiere solo. Cohen et al. (2024) funciona en modelos grandes y *falla* en pythia-70m y pythia-160m,

que es justamente la asimetría que hace que un resultado en modelos chicos no prediga uno grande. La pregunta de la transferencia queda abierta en las dos direcciones.

- **El idioma es la interfaz de la prueba, no el objeto de estudio.** Una respuesta de un token es lo que hace la métrica exacta sin juez LLM. No se afirma nada sobre lenguaje natural abierto.
- **Semillas y bimodalidad.** La bimodalidad entre semillas es una propiedad medida de este banco. Reportamos por unidad y nunca sólo como media, y con dos unidades en una celda no se puede separar dificultad de no convergencia. Dos condiciones tienen una sola semilla por celda.
- **El presupuesto de la fase de abstención lo eligió el autor**, y la §5.3 muestra que importó, ya que una formulación del resultado principal no sobrevivió a extenderlo. La comparación pareada a presupuesto igualado sí.
- **Un detalle de implementación contradice a su propio diseño.** El umbral de decisión del slot se heredó como masa mayor a 0,5, que es más estricto que lo que el diseño estipulaba, que era masa relativa. El barrido de umbrales muestra que esto no explica el resultado, pero hay que arreglarlo si se vuelve a tocar la condición.
- **Búsqueda dirigida de literatura, no revisión sistemática.** Un trabajo que use otro vocabulario para la misma idea, como «rejection head», «versioned memory» o «fact supersession», puede habérsenos pasado.

8. Conclusión

Dónde vive la decisión de abstenerse cambia si se puede aprender siquiera, y el orden es estable entre semillas, presupuestos y regímenes de margen. Una cabeza binaria separada que cuesta el 0,015 % de los parámetros reduce a la mitad, y casi a un tercio, la tasa de falsa abstención contra un token del vocabulario a presupuesto igualado, y baja en $\approx 0{,}10$ de margen de acierto el punto en el que un modelo se vuelve enseñable para callarse. La explicación no es la norma del vector del token, ya que se implementó la versión más generosa de esa hipótesis y falló, sino que dos decisiones de naturaleza distinta competían por un mismo softmax.

El slot de memoria, que sobre el papel es el más elegante de los tres porque vuelve «de ningún lado» una respuesta expresable en el mismo espacio donde el modelo ya opera, converge a la tasa base y no se abstiene nunca. Junto con un puntaje de matcheo del archivo clavado en el azar, esto ubica una propiedad estructural de la memoria co-entrenada, y es que **ahí la ausencia no tiene representación, y darle un lugar no alcanza para crearla**. Un recuperador externo puede devolver el conjunto vacío, un índice interno leído por softmax no puede, y la diferencia no se repara agregando una entrada nula.

Lo que queda abierto es la parte que más importa y que ninguno de estos resultados alcanza. Toda vía hacia la decisión que no pase por etiquetas ha fallado, y ha fallado por una razón específica y medida, que es que el modelo, cuando no sabe, se ancla en otro recuerdo real, y eso desde afuera es indistinguible de saber.

Disponibilidad de datos y código

Todos los pre-registros con sus hashes SHA, los archivos de desviaciones, los scripts de análisis, los resultados crudos por semilla y los informes generados por máquina están archivados en el repositorio del proyecto. Cada veredicto que se reporta acá lo emite un script congelado antes de que existieran los datos correspondientes.

Declaraciones

Financiamiento. Ninguno. **Conflictos de interés.** El autor declara no tener conflictos de interés. **Contribuciones.** Autor único, responsable del diseño, la implementación, el análisis y la escritura. **Ética.** No aplica, ya que no hay sujetos humanos ni animales y todos los datos son sintéticos. **Nota del autor.** Este trabajo se hizo sin auspicio, supervisión ni financiamiento institucional, y se reporta a título personal.

Referencias

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1), 41–46, 1970.

Roi Cohen, Konstantin Dobler, Eden Biran y Gerard de Melo. I don't know. Explicit modeling of uncertainty with an [IDK] token. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv 2412.06676.

Yonatan Geifman y Ran El-Yaniv. SelectiveNet. A deep neural network with an integrated reject option. *International Conference on Machine Learning (ICML)*, 2019. arXiv 1901.09192.

Hailey Joren et al. Sufficient context. A new lens on retrieval augmented generation systems. arXiv 2411.06037, 2024.

Weitang Liu, Xiaoyun Wang, John Owens y Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Stephen Merity, Caiming Xiong, James Bradbury y Richard Socher. Pointer sentinel mixture models. arXiv 1609.07843, 2016.

Pranav Rajpurkar, Robin Jia y Percy Liang. Know what you don't know. Unanswerable questions for SQuAD. *Association for Computational Linguistics (ACL)*, 2018.

Chupei Wang et al. Unable to forget. Proactive interference reveals working memory limits in LLMs beyond context length. arXiv 2506.08184, 2025.