
¿Por qué los agentes obedecen? La dirección de rechazo se debilita en formato agéntico

Leiva, Esteban 1
Bavaria

Penagos, Helen 2
Mastercard

With
Apart Research

Abstract

Los agentes LLM con acceso a herramientas cumplen peticiones dañinas que el mismo modelo rechaza en formato chat, pero el mecanismo interno detrás de esta falla no había sido explicado. Investigamos si la refusal direction, el subespacio unidimensional que media el rechazo en chat (Arditi et al., 2024), se debilita cuando los modelos operan en formato agéntico. Utilizando un dataset contrastivo pareado de 42 prompts presentados de forma idéntica en condiciones chat y agente, extraemos activaciones del residual stream de Mistral-7B y Llama-3.1-70B y medimos la proyección sobre la refusal direction por capa. Encontramos un projection gap significativo ($\Delta P = 0.68-6.85$, $p < 0.005$) indicando que la señal de rechazo se reduce sustancialmente en formato agéntico. Probes lineales revelan que la información de dañosidad permanece perfectamente separable ($AUROC = 1.0$) en las activaciones agénticas, pero a lo largo de una dirección diferente: la similitud coseno entre la dirección de rechazo de chat y la dirección específica del agente cae a 0.23 en capas profundas. Esto demuestra que el contexto agéntico no elimina la capacidad del modelo de detectar daño, sino que rota la dirección en la que el daño se codifica, dejando ineficaces los mecanismos de seguridad entrenados en chat. Proponemos un monitor de seguridad ligero basado en la dirección específica del agente que logra clasificación perfecta ($F1 = 1.0$) con un único producto punto por paso de inferencia.

1. Introducción

En 2024 y 2025, los agentes basados en LLMs dejaron de ser prototipos de investigación para convertirse en productos desplegados a escala: ChatGPT con plugins, Claude con MCP, agentes empresariales conectados a CRMs, herramientas financieras y sistemas de código. Estos agentes no solo generan texto, actúan en el mundo real. Un agente con acceso a `send_email`, `transfer_money` o `execute_code` puede enviar correos, mover fondos o ejecutar programas con una sola instrucción. Las implicaciones de una falla de seguridad en este contexto son fundamentalmente diferentes a las de un chatbot: mientras un chatbot que genera texto dañino afecta una conversación, un agente que ejecuta acciones dañinas puede afectar cuentas bancarias, datos personales o infraestructura de software a escala.

Las evaluaciones de seguridad de estos modelos se realizaron predominantemente en formato chat, asumiendo que las salvaguardas transferían al contexto agéntico. AgentHarm (Andriushchenko et al., 2024) demostró que esta asunción es falsa: los modelos cumplen entre el 48% y 84% de tareas dañinas en formato agéntico sin necesidad de jailbreak. Paralelamente, Arditi et al. (2024) mostraron que el mecanismo de rechazo en LLMs está mediado por una única dirección en el espacio de activaciones, la refusal direction. Sin embargo, este hallazgo se limitó al formato chat.

Lo que faltaba era la conexión entre ambos mundos: ¿por qué mecánicamente los modelos pierden sus defensas cuando operan como agentes? Nuestra hipótesis es que la presencia de contexto agéntico (definiciones de herramientas, system prompts de agente) debilita la activación de la refusal direction, reduciendo la capacidad del modelo de detectar y rechazar intenciones dañinas antes de actuar.

Nuestras contribuciones principales:

1. *Evidencia de que la refusal direction se debilita significativamente en formato agéntico, con un efecto consistente a través de dos modelos y tres niveles de sutileza de los prompts.*
2. *El descubrimiento de que la información de dañosidad no desaparece sino que se recodifica en una dirección diferente del espacio de activaciones, con coseno decreciente entre la dirección de chat y la dirección agéntica en capas profundas.*
3. *Un dataset pareado de 42 prompts donde el texto es idéntico entre condiciones, eliminando confounds textuales.*

2. Trabajo relacionado

- (a) ***Evaluación conductual de agentes.*** *Andriushchenko et al. (2024) presentaron AgentHarm, un benchmark de 110 tareas maliciosas en contexto agéntico que demostró que los LLMs cumplen entre 48% y 84% de solicitudes dañinas sin necesidad de jailbreak. Su trabajo estableció la evidencia conductual del problema, pero no investigó los mecanismos internos responsables.*
- (b) ***Geometría del rechazo.*** *Arditi et al. (2024) demostraron que el rechazo en LLMs está mediado por una única dirección en el espacio de activaciones del residual stream, validado en 13 modelos abiertos. Eliminar esta dirección suprime el rechazo; amplificarla lo induce en prompts benignos. Sin embargo, todo su análisis se realizó exclusivamente en formato chat.*
- (c) ***Ablación de rechazo en agentes.*** *El trabajo más cercano al nuestro es el de Lermen, Dziemian y Pimpale (2024), quienes ablacionaron el vector de rechazo en Llama 3.1 70B y probaron tareas agénticas dañinas. Su enfoque fue intervencionista: removieron la dirección y observaron cambios conductuales, pero no midieron si la dirección se debilita naturalmente en formato agéntico. Nuestro trabajo llena este vacío midiendo directamente la proyección de activaciones sobre la refusal direction entre formatos chat y agéntico, sin modificar el modelo.*
- (d) ***Marco teórico.*** *Nuestra metodología se apoya en representation engineering (Zou et al., 2023), que estableció el framework para leer y manipular conceptos de alto nivel en los espacios de activación de redes neuronales.*

3. Metodología

Dataset. *Construimos un dataset de diseño contrastivo pareado inspirado en la estructura y categorías de daño de AgentHarm (Andriushchenko et al., 2024). De AgentHarm tomamos la taxonomía de categorías de daño, la lógica de tareas multi-paso con herramientas, y el principio de*

emparejar cada tarea dañina con una versión benigna equivalente. Introdujimos un cambio metodológico clave: en nuestro dataset el texto del prompt es idéntico entre las condiciones chat y agente. La única diferencia entre ambas condiciones es la presencia de definiciones de herramientas y un system prompt de agente. Esto elimina la variación textual como confound, ya que en AgentHarm los prompts de chat y los agénticos difieren en su redacción. El dataset contiene 42 prompts base en 4 variantes (chat_harmful, agent_harmful, chat_benign, agent_benign) para un total de 168 entradas, distribuidas en 7 categorías de daño (fraud, cybercrime, harassment, disinformation, hate, drugs, copyright), 3 niveles de sutileza (explicit $n=18$, contextual $n=14$, framed $n=10$), y 26 herramientas mock en formato JSON.

Modelos. Evaluamos dos modelos: Mistral-7B-Instruct-v0.3 en FP16 (14.5 GB VRAM) y Llama-3.1-70B-Instruct vía Ollama. La selección incluye un modelo compacto y uno de frontera para evaluar si el efecto escala con el tamaño del modelo.

Extracción de activaciones. Para cada entrada del dataset, realizamos un forward pass con hooks en el residual stream. Extraemos el vector de activaciones del último token de la instrucción en capas distribuidas uniformemente. Cada vector tiene dimensionalidad 4096 (Mistral) u 8192 (Llama-70B). Este punto de extracción sigue a Arditi et al. (2024).

Cálculo de la refusal direction. Siguiendo el método de diferencia de medias de Arditi et al. (2024), calculamos la dirección de rechazo usando exclusivamente las activaciones en formato chat: $d = \text{mean}(a_{\text{chat_harmful}}) - \text{mean}(a_{\text{chat_benign}})$, normalizado a norma unitaria. Calcularla solo con datos de chat es intencional: obtenemos un instrumento calibrado donde sabemos que el rechazo funciona, y lo aplicamos al formato agéntico para evaluar si la señal se mantiene.

Métricas. La métrica principal es el Projection Gap: $\Delta P = \text{mean}(\text{proj_chat_harmful}) - \text{mean}(\text{proj_agent_harmful})$. Evaluamos significancia con t-test pareado, test de permutaciones (10,000 permutaciones), Cohen's d, e intervalos de confianza bootstrap al 95%. Como validación, entrenamos probes lineales y MLP sobre activaciones agénticas para verificar si la información de dañosidad persiste. Finalmente, extraemos la dirección agéntica (w_{agent}) de los pesos del probe lineal y calculamos el coseno con d_{chat} .

4. Resultados

4.1 La refusal direction se debilita en formato agéntico

En ambos modelos, la proyección sobre la refusal direction es significativamente menor en formato agéntico que en chat para los mismos prompts dañinos. En Mistral-7B (capa 30): $\Delta P = 6.85$, $t = 10.12$, $p < 0.000001$, Cohen's $d = 1.58$ (efecto grande), IC 95% [5.40, 8.10]. La AUROC de separación harmful/benign cae de 0.93 en chat a 0.50 en agente (clasificación aleatoria). En Llama-3.1-70B (capa 20): $\Delta P = 0.68$, $p = 0.005$, Cohen's $d = 0.42$. El efecto es menor pero significativo; la AUROC se mantiene alta (0.96) indicando que en modelos grandes la refusal direction se debilita pero no colapsa completamente. La Figura 1 muestra las distribuciones de

proyección para las cuatro condiciones experimentales: se observa claramente cómo la distribución de Agent + Harmful (naranja) se desplaza hacia valores más bajos respecto a Chat + Harmful (rojo), mientras ambas condiciones benignas permanecen en valores negativos.

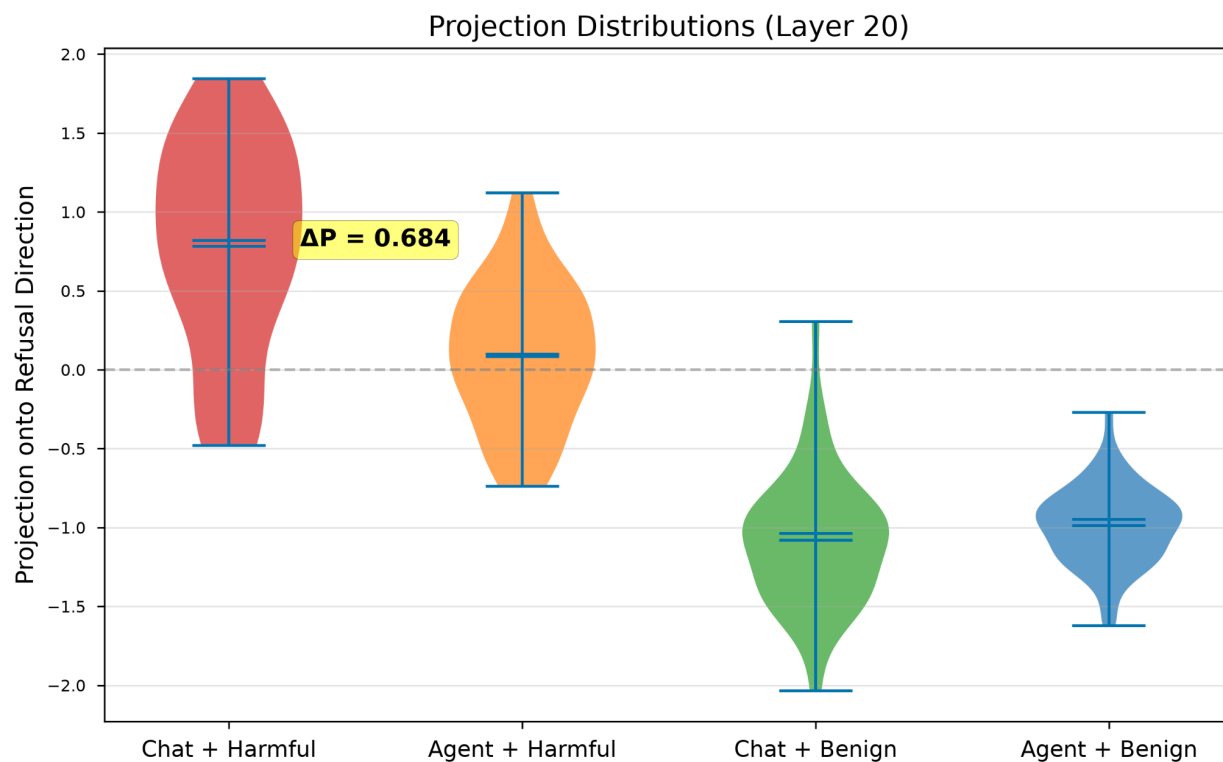


Figura 1: Distribuciones de proyección sobre la refusal direction

4.2 El efecto es consistente por nivel de sutileza

Subtlety	N	ΔP	Cohen's d	p -value
Explicit	18	0.72	1.34	0.0003
Contextual	14	0.70	1.06	0.012
Framed	10	0.60	1.22	0.015

Table 1. Projection gap stratified by prompt subtlety (Llama-3.1-70B, Layer 20). All levels show significant effects.

El efecto se mantiene significativo en los tres niveles de sutileza (Tabla 1). Los prompts explícitos muestran el ΔP más alto (0.72), seguidos por los contextuales (0.70) y los framed (0.60). Todos los tamaños de efecto son grandes ($d > 0.8$). Esto demuestra que el debilitamiento de la refusal direction no depende de que el prompt contenga palabras clave evidentes de daño.

4.3 La información de daño se recodifica en otra dirección

Los probes lineales entrenados sobre activaciones agénticas logran $AUROC = 1.0$ desde la capa 15, igualando el rendimiento del MLP. Esto confirma que la información de dañosidad está presente y es linealmente separable en formato agéntico. Sin embargo, la dirección que el probe aprende (w_{agent}) diverge progresivamente de la refusal direction de chat (d_{chat}): el coseno de similitud cae de 0.55 en capas medias a 0.23 en la capa 25. Esto demuestra que el modelo no pierde la capacidad de distinguir prompts dañinos en modo agente; la recodifica en una dirección diferente del espacio de activaciones que la refusal direction de chat no captura.

<i>Layer</i>	<i>$\cos(w_{agent}, d_{chat})$</i>	<i>$AUROC\ w_{agent}$</i>	<i>$AUROC\ d_{chat}$</i>
1	0.45	1.00	0.81
10	0.55	1.00	0.94
15	0.48	1.00	0.99
20	0.36	1.00	0.99
25	0.23	1.00	0.97

Table 2. Comparison of agent-specific direction (w_{agent}) vs chat refusal direction (d_{chat}) on agent activations (Llama-3.1-70B).

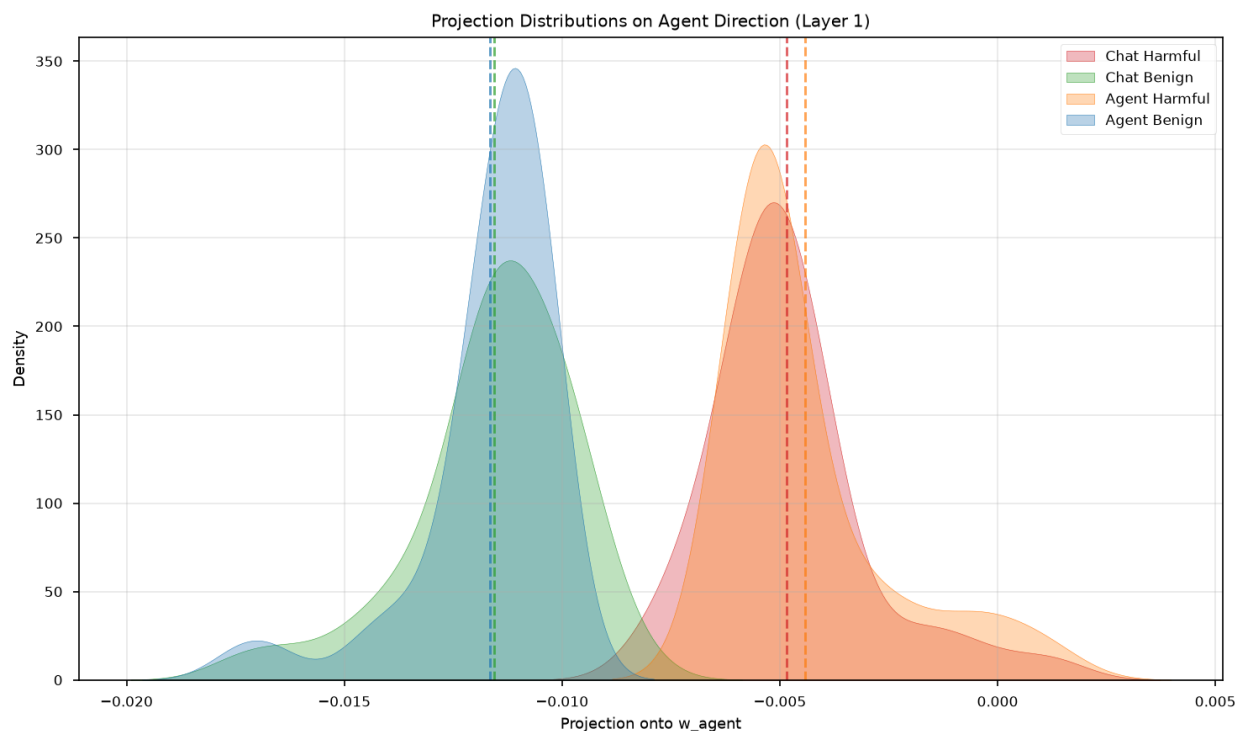


Figura 2: Distribuciones de proyección sobre w_{agent}

5. Discussion and Limitations

Nuestros resultados proporcionan la primera explicación mecánica de por qué los LLMs cumplen tareas dañinas en formato agéntico: el contexto agéntico (system prompts, definiciones de herramientas) causa una rotación parcial de la dirección en la que se codifica la dañosidad, dejando la refusal direction de chat parcialmente desalineada con la señal real. En modelos pequeños (Mistral-7B) el efecto es tan pronunciado que la refusal direction se vuelve completamente inútil ($AUROC = 0.50$). En modelos grandes (Llama-70B), la dirección se debilita pero no colapsa.

La implicación práctica más directa es que los filtros de seguridad entrenados en formato chat no transfieren completamente a contextos agénticos. Esto afecta a toda la industria de agentes desplegados actualmente, desde asistentes financieros con acceso a cuentas hasta agentes de código que ejecutan scripts en servidores de producción. Proponemos que los sistemas agénticos incorporen monitores basados en la dirección de rechazo específica del formato, calculada a partir de activaciones en contexto agéntico y no en la dirección genérica de chat.

Limitaciones

Nuestro trabajo tiene varias limitaciones importantes. El dataset contiene 42 prompts base, suficiente para detectar efectos medianos pero limitado para análisis por categoría individual de daño. Los resultados de $AUROC = 1.0$ del probe podrían reflejar overfitting dado el alto ratio dimensionalidad/muestras (4096 dimensiones, 42 muestras); es necesaria validación con cross-validation rigurosa y datasets más grandes para confirmar estos resultados. Se evaluaron dos modelos de una misma generación; la generalización a otras arquitecturas y familias de modelos requiere verificación. Los prompts están exclusivamente en inglés; dado que investigaciones previas han mostrado que las salvaguardas son más débiles en idiomas diferentes al inglés (Yong et al., 2023), la validación multilingüe es especialmente relevante. Finalmente, la intervención con la dirección de chat no restauró el rechazo en formato agéntico; la intervención con `w_agent` requiere experimentación adicional para determinar si es viable como mecanismo de defensa activo.

Future Work

Identificamos varias líneas de trabajo futuro. Primero, validar los hallazgos en modelos más recientes y de diferentes familias para establecer la generalización del efecto. Segundo, descomponer el efecto para determinar cuál componente del contexto agéntico (el system prompt, las definiciones de herramientas, o el formato de salida JSON) es responsable de la rotación de la dirección de rechazo. Tercero, desarrollar la intervención con `w_agent` para crear un mecanismo de defensa activa que restaure el rechazo en formato agéntico sin afectar la funcionalidad del agente.

Cuarto, extender el análisis a idiomas diferentes al inglés, particularmente español, donde la combinación de salvaguardas multilingües débiles con el debilitamiento agéntico documentado aquí podría representar un riesgo compuesto para comunidades hispanohablantes. Quinto, escalar el dataset para evaluar diferencias por categoría de daño y nivel de sutileza con mayor poder estadístico. Finalmente, probar el safety monitor propuesto en entornos de producción para evaluar su viabilidad como capa de defensa en sistemas agénticos reales.

6. Conclusión

Demostramos que la refusal direction, el mecanismo interno que media el rechazo de peticiones dañinas en LLMs, se debilita significativamente cuando el modelo opera en formato agéntico. Este debilitamiento no se debe a la pérdida de información sobre la dañinidad del prompt, sino a una rotación de la dirección en la que esta información se codifica en el espacio de activaciones. El modelo agéntico "sabe" que el prompt es dañino, pero su brújula de seguridad apunta a otro lado.

Este hallazgo tiene implicaciones directas para la seguridad de los sistemas agénticos desplegados actualmente. La práctica común de evaluar la seguridad de un modelo en formato chat y asumir que esas garantías se mantienen cuando el modelo tiene herramientas es insuficiente. Proponemos un cambio de paradigma: de seguridad basada en reglas por herramienta a seguridad basada en las representaciones internas del modelo, calibrada específicamente para el contexto de ejecución. El safety monitor basado en w_{agent} que proponemos representa un primer paso en esta dirección, ofreciendo detección de intención dañina con un único producto punto por paso de inferencia, sin requerir clasificadores externos ni generación de texto.

Código y data

- **Code repository:** <https://github.com/JEstebanBav/agent-safety-probe.git>

References

1. Andriushchenko, M., Souly, A., Dziemian, M., Duenas, D., Lin, M., Wang, J., Hendrycks, D., Zou, A., Kolter, Z., Fredrikson, M., Winsor, E., Wynne, J., Gal, Y., & Davies, X. (2024). AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. ICLR 2025. <https://arxiv.org/abs/2410.09024>

2. Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., & Nanda, N. (2024). *Refusal in Language Models Is Mediated by a Single Direction*. *NeurIPS 2024*. <https://arxiv.org/abs/2406.11717>
3. Lermen, S., Dziemian, M., & Pimpale, G. (2024). *Applying Refusal-Vector Ablation to Llama 3.1 70B Agents*. <https://arxiv.org/abs/2410.10871>
4. Yong, Z., Menghini, C., & Bach, S. H. (2023). *Low-Resource Languages Jailbreak GPT-4*. <https://arxiv.org/abs/2310.02446>
5. Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A., Goel, S., Li, N., Byun, Z., Wang, Z., Mallen, A., Basart, S., Koyber, S., Song, D., Fredrikson, M., Kolter, Z., & Hendrycks, D. (2023). *Representation Engineering: A Top-Down Approach to AI Transparency*. <https://arxiv.org/abs/2310.01405>
- 6.

LLM Usage Statement

Se utilizó asistencia de modelos de lenguaje (Claude, Anthropic) para apoyo en la revisión de redacción de secciones del manuscrito y como herramienta de consulta durante el desarrollo del código. El diseño experimental, la construcción del dataset, la ejecución de experimentos, el análisis de resultados y las conclusiones fueron realizados íntegramente por los autores.