Análisis Experimental de Mecanismos de Seguridad en Aprendizaje por Refuerzo aplicado a Ms. Pac-Man

Diego Hernández Suárez[1], **Carmen Gallardo Martín**[1], **Carlos Sánchez Arroyo**[1]
[1]Universidad Carlos III de Madrid
Máster en Inteligencia Artificial Aplicada
100472809, 100567491, 100451282

Resumen

La exploración no restringida en el Aprendizaje por Refuerzo (RL) plantea riesgos en entornos críticos, donde los fallos del sistema pueden tener consecuencias catastróficas. Este trabajo aborda la problemática del RL Seguro en el dominio estocástico de *Ms. Pac-Man*, proponiendo y evaluando una arquitectura híbrida que integra Deep Q-Networks (DQN) con tres mecanismos de seguridad complementarios: *Shielding* reactivo basado en distancia, *Reward Shaping* con penalización por proximidad, y Aprendizaje por Imitación guiado por demostraciones humanas.

1 Introducción

El aprendizaje por refuerzo es un área que ha demostrado grandes resultados en dominios secuenciales complejos, como videojuegos y control autónomo. Estos escenarios complejos pueden enfrentar problemas en el caso de experimentar fallos en el sistema, por ejemplo en el caso de tratarse de un dron automatizado que llega a estrellarse. Este tipo de fallos pueden llevar a daños no deseados, y es por ello que en este trabajo se analizará el enfoque del aprendizaje por refuerzo seguro, que tiene como objetivo evitar estos incidentes.

El trabajo se encuentra desarrollado en el dominio del Pac-Man. Se implementará un agente DQN como línea base y posteriormente se introducirán distintos mecanismos prácticos de seguridad con el fin de evitar comportamientos que conduzcan a estados catastróficos, en este caso que el agente sea comido por los fantasmas. Finalmente realizaremos un análisis experimental del impacto de dichos mecanismos en el compromiso entre rendimiento y seguridad.

Todos los recursos utilizados durante este proyecto pueden ser encontrados en el siguiente repositorio de GitHub: `https://github.com/Carlossanarr/RL-practica`.

2 Contexto del aprendizaje por refuerzo

El *Aprendizaje por Refuerzo* (Reinforcement Learning, RL) es un paradigma del *Machine Learning* en el que un agente aprende a tomar una secuencia de decisiones interactuando con un entorno con el objetivo de maximizar una noción de recompensa acumulada a largo plazo, donde el agente no dispone de ejemplos etiquetados. En su lugar, debe descubrir mediante prueba y error qué acciones conducen a secuencias favorables.

2.1 Procesos de Decisión de Markov

El RL se formaliza mediante los *Procesos de Decisión de Markov* (MDP). Un MDP se define mediante la tupla:

$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, T, R\},$$

donde:

- $\mathcal{S}$ es el conjunto de estados.
- $\mathcal{A}$ es el conjunto de acciones disponibles para el agente.
- $T(s, a, s') = \mathbb{P}(s_{t+1} = s' \mid s_t = s, a_t = a)$ es la función de transición.
- $R(s, a)$ es la función de recompensa al realizar una acción a en un determinado estado s.

Una política $\pi(a|s)$ define la probabilidad de seleccionar la acción a estando en el estado s. El objetivo del RL es encontrar la *política óptima* π^* que maximiza el retorno esperado total:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right], \qquad (2.1)$$

donde $\gamma \in [0, 1)$ es el factor de descuento.

2.2 Limitaciones del RL clásico en entornos de gran dimensionalidad

En tareas con espacios de estados de gran dimensionalidad, como entornos basados en imágenes, los métodos tabulares como Q-Learning se vuelven inoperables debido a la imposibilidad de representar explícitamente la función $Q(s, a)$ para todos los estados posibles. Este problema de escalabilidad llevó al desarrollo de métodos de *Deep Reinforcement Learning*, donde se emplean redes neuronales profundas como aproximadores de función, permitiendo tratar espacios continuos o extremadamente grandes, un claro ejemplo de esto es *DQN*, la cual es la versión profunda de Q-Learning, donde en vez de usar una tabla se usa una red neuronal que aproxima $Q(s, a)$.

Aunque es cierto que *DQN* permite escalar Q-learning a entornos de mayor complejidad como videojuegos basados en imágenes, sigue sufriendo los mismos problemas funda-

mentales del RL clásico, en especial aquellos relacionados con la exploración insegura durante el entrenamiento, un problema crítico en dominios donde los errores pueden implicar consecuencias elevadas.

2.3 Exploración Insegura en RL

El RL tradicional prioriza la eficiencia (maximizar la recompensa) sobre la seguridad, ya que la política, y por ende el comportamiento que se busca es el de maximizar las recompensas obtenidas. Para que el agente sea capaz de encontrar comportamientos óptimos, debe explorar el entorno, habitualmente mediante estrategias como ε-greedy, que introducen acciones aleatorias con el fin de descubrir nuevos estados y trayectorias, aunque buscando un balance con las acciones y políticas previamente probadas.

Sin embargo, en entornos reales o críticos (como robótica, sistemas autónomos, vehículos o control industrial), una exploración que introduce un factor aleatorio puede resultar inaceptable, ya que puede conducir a:

- Daños físicos al agente.
- Daños al entorno u otros actores.
- Incumplimiento de restricciones de seguridad.
- Consecuencias catastróficas o irreversibles.

En el contexto de *Pac-Man*, aunque el entorno es simulado, la similitud es clara: el problema es que la política π cometerá errores durante el entrenamiento, resultando en ser capturado por los fantasmas repetidamente. El coste de estos errores acumulados hace inviable el despliegue directo de un RL tradicional en sistemas donde el error es costoso.

Estas limitaciones han impulsado el estudio de algoritmos de *Safe Reinforcement Learning*, los cuales introducen restricciones de seguridad, funciones de coste asociadas al riesgo o supervisores externos capaces de bloquear acciones inseguras. Safe RL busca garantizar que el agente no solo aprenda una política óptima en términos de recompensa, sino también una política que cumpla criterios de seguridad durante todo el proceso de aprendizaje. Este extiende el RL tradicional para mitigar los riesgos, asegurando que el sistema mantenga un rendimiento aceptable y respete las restricciones de seguridad durante el proceso de aprendizaje y/o el despliegue.

3 Estado del Arte

3.1 Procesos de Decisión de Markov Restringidos (CMDPs)

La herramienta formal clave para incorporar restricciones de seguridad en RL son los *Constrained Markov Decision Processes* (CMDPs). Un CMDP es una extensión de los MDPs tradicionales, donde se introduce una función de coste asociada a eventos no deseados [3] [6]. Formalmente, un CMDP se define de como un MDP tradicional pero añadiéndole la función de coste de seguridad y el umbral de coste máximo permitido:

$$\mathcal{M}_C = \{\mathcal{S}, \mathcal{A}, T, R, C, d\},$$

donde:

- $C(s,a) \geq 0$ es la *función de coste de seguridad*,
- d es el *umbral máximo permitido* para el coste acumulado.

La función de coste C mide la presencia de comportamientos inseguros o no deseados. En nuestro problema, por ejemplo, $C(s,a)$ toma valor 1 cuando el agente es comido (pierde una vida) y 0 en caso contrario. El objetivo es encontrar una política que no solo maximice la recompensa, sino que además mantenga este coste por debajo de un límite aceptable. En este trabajo utilizamos los CMDPs como marco conceptual para definir la noción de seguridad, sin resolver explícitamente el problema de optimización dual asociado.

3.1.1. Optimización en un CMDP

El problema de optimización se modifica a un problema de maximización sujeta a una restricción de coste acumulado:

$$\begin{aligned} \max_{\pi} \quad & J_R(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right] \\ \text{s.t.} \quad & J_C(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t C(s_t, a_t) \right] \leq d \end{aligned} \tag{3.1}$$

Por ende esta restricción permite modelar el requisito de que el agente mantenga un comportamiento seguro durante el aprendizaje. Este tipo de medidas de seguridad son las conocidas como soft en este ámbito, ya que no se corrigen directamente las acciones que se pretenden tomar, pero si que otorga una información de peligro en el procesod e selección de acciones.

En nuestro caso, la política debe maximizar la puntuación máxima obtenida en el juego Pac-Man y a su vez mantener el número de veces que el agente es capturado (coste) por debajo del límite establecido d. De esta manera, se puede garantizar una seguridad integrada en la optimización de la política, con el fin de guiar al agente hacia comportamientos tanto eficaces como seguros.

3.2 Deep Q-Networks (DQN)

El algoritmo *Deep Q-Network* (DQN) [7] marcó un hito en el aprendizaje por refuerzo al combinar Q-learning con redes neuronales profundas para aproximar la función de acción-valor $Q(s,a)$. En lugar de mantener una tabla Q explícita, DQN utiliza una red para generalizar sobre espacios de estados continuos o de alta dimensión, lo que lo hace adecuado para entornos complejos como videojuegos. Este enfoque permitió alcanzar rendimiento humano en juegos Atari, estableciendo la base para muchas variantes posteriores.

3.3 Shielding, safety layers y filtrado de acciones

Existen otros tipos de métodos los cuales separan el aprendizaje del mecanismo de seguridad como tal, donde se introduce una capa adicional externa la cual se encarga

de supervisar y corregir acciones del agente en caso de que tome una decisión incorrecta [1]. El *shielding* sustituye acciones cuando estas ponen en riesgo la integridad del agente, lo que permite entrenar políticas estándar de RL pero bajo restricciones explícitas. Este tipo de seguridad es la que se conoce en el safety RL actualmente como tipo hard, ya que se analizan las acciones que se pretenden tomar, para modeificarlas si es necesario.

En entornos discretos, el *action masking* [8] hace posible eliminar acciones potencialmente peligrosas antes de la selección final. Este tipo de técnicas destacan sobre todo cuando es posible definir reglas de seguridad simples y fácilmente interpretables, además de que su integración con arquitecturas DQN es directa.

3.4 Aprendizaje con demostraciones, teacher y feedback humano

Otra línea de trabajo que es capaz de reducir el riesgo mediante el conocimiento previo. Un ejemplo de esto puede ser Deep Q-learning from Demonstrations (DQfD) [4], el cual combina aprendizaje por refuerzo con demostraciones expertas previas con el objetivo de mejorar el rendimiento inicial, donde es más posible acabar en estados catastróficos.

Además, el interactive reinforcement learning emplea feedback humano para guiar el aprendizaje [5] [2], ya sea mediante señales directas de refuerzo o a través de preferencias entre trayectorias. Estos enfoques son especialmente útiles cuando la seguridad resulta difícil de formalizar de manera explícita.

3.5 Relación con nuestro trabajo

Los distintos trabajos existentes sobre aprendizaje por refuerzo seguro incluyen la optimización de restricciones explícitas, mecanismos de supervisión externa y el uso de experiencias previas. Para este proyecto partiremos de un DQN como línea base, para luego explorar distintos enfoques como la incorporación de un teacher externo o un safety layer para evitar comportamientos inseguros. Pac-Man se empleará como banco de pruebas controlado para analizar el balance entre rendimiento y seguridad introducido por este tipo de prácticas.

4 Entorno Experimental: Ms. Pac-Man

Para la validación de los algoritmos de aprendizaje por refuerzo seguro, se ha utilizado el entorno `ALE/MsPacman-v5` proporcionado por la librería *Gymnasium* y el *Arcade Learning Environment* (ALE). A diferencia de entornos simplificados basados en matrices de posiciones, este entorno obliga al agente a aprender directamente desde los píxeles de la pantalla, simulando la complejidad de visión computacional.

El entorno es estocástico y parcialmente observable en cuanto a la velocidad y dirección futura de los fantasmas, lo que justifica la necesidad de técnicas avanzadas de preprocesamiento y seguridad.

4.1 Espacio de Estados

El estado crudo del juego es una imagen RGB de 210×160 píxeles. Para hacer el entrenamiento computacionalmente viable y facilitar la convergencia de la Red Neuronal Convolucional (CNN), se ha aplicado el siguiente flujo de preprocesamiento:

- **Escala de Grises:** Conversión de RGB a un solo canal de luminancia, ya que el color no es determinante para la detección de obstáculos o enemigos.
- **Redimensionado:** La imagen se reduce a una matriz de 84×84 píxeles.
- **Frame Stacking (Apilamiento):** Dado que una sola imagen estática no permite percibir la velocidad ni la dirección de los fantasmas, el estado S_t se define como una pila de los 4 últimos frames consecutivos ($84 \times 84 \times 4$). Esto permite al agente inferir la dinámica temporal del entorno.

4.2 Espacio de Acciones

El espacio de acciones es discreto y consiste en 5 acciones efectivas para la navegación en el laberinto: *NOOP* (sin operación), *UP* (arriba), *RIGHT* (derecha), *LEFT* (izquierda) y *DOWN* (abajo).

5 Metodología

5.1 Arquitectura y entrenamiento del agente DQN

Como agente base se ha empleado un Deep Q-Network (DQN), implementándolo a través de la librería Stable-Baselines3, empleando la política `CnnPolicy`. El agente aprende una aproximación de la función de valor-acción $Q(s, a)$ del estado descrito en la sección *Espacio de Estados* (ver subsection 4.1), usando observaciones visuales preprocesadas del entorno ALE/MsPacman-v5.

El proceso de entrenamiento sigue un esquema DQN estándar con *experience replay*. Las transiciones (s_t, a_t, r_t, s_{t+1}) son almacenadas en un *replay buffer*, de este se extraen lotes para actualizar los parámetros de la red neuronal. La exploración del entorno se realiza mediante una estrategia ϵ-greedy, donde se establece una fracción inicial de exploración del 20 %.

5.2 Entrenamiento Asistido (Teacher)

Para mejorar la eficiencia inicial y la seguridad del agente, se ha implementado una fase de *Imitation Learning* o Çalentamiento"guiado por un humano, que actúa como el "Profesor"(*Teacher*).

Antes de iniciar el proceso de optimización de la red neuronal, el humano controla al agente durante un número determinado de pasos ($N = 7,500$). Estas transiciones, que representan un comportamiento experto y seguro, se almacenan directamente en el *Replay Buffer*. De esta manera, cuando el agente (el alumno) comienza su entrenamiento autónomo, no parte de una exploración aleatoria ciega, sino que dispone de una memoria llena de ejemplos de alta calidad sobre cómo navegar y sobrevivir.

5.3 Escudo de Seguridad Basado en Distancia

Para proteger al agente durante la fase de exploración autónoma, se ha integrado un mecanismo reactivo denominado **Escudo de Seguridad** (*Shield*). Este sistema supervisa las acciones propuestas por la red neuronal antes de su ejecución en el entorno.

El funcionamiento del escudo se basa en un Monitor de Seguridad accediendo a la memoria RAM del emulador Atari (ALE), donde en cada paso temporal obtiene las coordenadas (x, y) de tanto el agente como de los cuatro fantasmas. A partir de estas posiciones, se calcula la distancia Manhattan del Pac-Man respecto a los fantasmas. Se establece un umbral de seguridad δ_S píxeles:

- Si la distancia a todos los fantasmas es mayor a δ_S, el estado se considera **Seguro** y el escudo permite pasar la acción original del agente DQN sin modificación.
- Si algún fantasma se encuentra a una distancia menor a δ_S (Zona de Peligro), el escudo **interviene**: bloquea la acción insegura y la sustituye por una acción evasiva óptima calculada heurísticamente.

La acción evasiva se determina identificando al fantasma más cercano y se selecciona aquel movimiento que incremente la separación con este fantasma. Esto se consigue comparando la diferencia tanto horizontal como vertical entre el agente y el fantasma más próximo, dando prioridad a aquella acción donde la distancia sea mayor. De esta manera el escudo selecciona la acción que mueve al agente en dirección opuesta al fantasma, intentando evitar una potencial captura. Todo esto con el objetivo de que el Pac-Man vaya aprendiendo pero estableciendo unas "Zonas de Peligro"las cuales se eviten a toda costa que el agente entre.

5.4 Modelado de Recompensa (Reward Shaping)

Aunque el Shield evita la muerte del agente, es crucial que el agente aprenda a evitar situaciones de peligro por sí mismo. Para ello, se ha modificado la función de recompensa original del juego mediante *Reward Shaping*:

$$R_{total} = R_{juego} + R_{seguridad} \tag{5.1}$$

Donde $R_{seguridad}$ aplica una penalización negativa máxima fija (-5 puntos) denominada λ en cada instante temporal en el que el Monitor de Seguridad detecta que el agente se encuentra dentro de la "Zona de Peligro". Dicha penalización se define de forma lineal en función de la distancia d al fantasma más cercano, de acuerdo con:

$$R_{\text{seguridad}}(d) = \begin{cases} \text{-}\lambda \left(1 - \frac{d}{\delta_R}\right), & \text{si } d < \delta_R, \\ 0, & \text{en otro caso.} \end{cases} \tag{5.2}$$

Esto desincentiva al agente de depender del escudo y fomenta el aprendizaje de una política inherentemente segura.

5.5 Métricas de Evaluación

Para cuantificar el desempeño y la seguridad de los distintos agentes (DQN Base vs. DQN + Imitation + Shield), se han definido las siguientes métricas:

- **Recompensa Media:** Puntuación total acumulada por episodio.
- **Eficiencia (η):** Relación entre puntos obtenidos y pasos sobrevividos ($\eta = \frac{Score}{Steps}$). Una eficiencia alta indica que el agente recolecta activamente, mientras que una baja indica un comportamiento pasivo o de miedo.
- **Puntos por muerte (PPM)**: Número de puntos obtenidos dividido por el número de muertes.
- **Ratio de Intervenciones del Escudo:** Ratio de veces que el escudo de seguridad tuvo que corregir al agente.
- **Ratio de Pasos Inseguros (UnSteps):** Proporción de pasos en los que el agente se encuentra en un estado considerado inseguro según el criterio de distancia (δ_R).

5.6 Configuraciones y Experimentos Propuestos

A continuación se detallarán los experimentos que han sido desarrollados para analizar los resultados obtenidos a partir de las métricas previamente propuestas. Cada experimento se define por la presencia o ausencia de tres elementos de seguridad: una fase de entrenamiento asistido o "*Teacher*"(T), un escudo de seguridad (S) basado en distancia, y un modelado de recompensa (R) que aplica una penalización si el agente se encuentra demasiado cerca de algún fantasma. Por ende, estos mecanismos se pueden entender como variables binarias, pudiendo estar activas (1) o inactivas (0).

Entre las configuraciones de seguridad, Las situaciones peligrosas se detectan utilizando un criterio basado en la distancia derivado directamente de la RAM de Atari. Por cada paso, la distancia Manhattan entre el Pac-Man y el fantasma más cercano es calculada a partir de su posición en la memoria. Un estado se considera inseguro si esta distancia es menor que un umbral denominado δ_R para la modelado de recompensa o δ_s para la intervención del escudo de seguridad. Esta definición de peligro es consistente para todos los experimentos, pero es importante destacar que este valor será evaluado bajo distintos valores para ver su impacto e influencia en los agentes.

5.6.1. Parámetros Experimentales Comunes

Para evaluar la convergencia y estabilidad de cada configuración, y con el fin de garantizar una comparación justa entre las configuraciones evaluadas, se ha decidido que todos los experimentos compartan un conjunto de parámetros comunes de entrenamiento y evaluación.

Para el entrenamiento se ha fijado un horizonte temporal de entrenamiento de 400,000 pasos, hemos considerado que no era necesario utilizar más pasos porque el fin de este proyecto es cuantificar el impacto de los mecanismos de seguridad y evitar estados catastróficos durante el entrenamiento, no entrenar lo suficiente a un agente para que a base de prueba y error pueda jugar mejor que un humano. Además, en aquellos experimentos con calentamiento por Imitación (T=1), se han utilizado 7,500 pasos de demostración humana previos (equivalente a unos 15 minutos de juego aproximadamente), empleados como fase previa

al entrenamiento por refuerzo, con el objetivo de guiar al agente en las fases iniciales del entrenamiento.

Respecto a la evaluación de los agentes entrenados, es realizado a lo largo de 20 episodios independientes, a partir de los cuales se calculan las métricas agregadas que se presentarán en la sección de Resultados (ver Table 2). Además, para monitorear el rendimiento durante el entrenamiento, se registra el estado del agente y y sus métricas principales en un archivo CSV cada 10,000 pasos, de esta manera se puede también realizar un análisis de la evolución de aprendizaje e identificar si realmente estos mecanismos de seguridad están aportando a evitar que el agente muera mientras aprende.

5.6.2. Diseño de Configuraciones

Se ha diseñado un conjunto incremental de configuraciones experimentales para analizar el impacto de cada agente:

1. **Agente Base (B1):** Se parte de un agente DQN estándar sin mecanismos de seguridad (S=0, R=0, T=0) que sirve como punto de referencia (*baseline*) para comparar el comportamiento y rendimiento.
2. **Mecanismos Individuales (S y R):** Se evalúan por separado el **Escudo de Seguridad** (S=1, R=0, T=0), variando el umbral δ_S, para estudiar el efecto de la intervención explícita en las acciones; y la **Seguridad por Penalización** (S=0, R=1, T=0), donde se aplican recompensas negativas variando δ_R para investigar el aprendizaje seguro sin intervención directa.
3. **Combinación Escudo-Recompensa (SR):** Se combinan ambos enfoques (S=1, R=1, T=0) estableciendo $\delta_S < \delta_R$. Esto permite que el agente reciba penalizaciones educativas al acercarse al peligro antes de que el escudo tenga que intervenir físicamente en el último momento.
4. **Aprendizaje Asistido e Híbridos (T, TS, TR, TSR):** Finalmente, se introduce el **Teacher** (T=1) mediante calentamiento por imitación para mitigar la vulnerabilidad inicial. Se prueba este mecanismo de forma aislada y posteriormente en configuraciones híbridas junto al escudo (TS), al modelado de recompensa (TR) y, por último, la combinación completa de los tres sistemas (TSR), buscando la máxima robustez y rendimiento.

5.6.3. Resumen de Configuraciones

En Table 1 se muestran resumidas las distintas configuraciones previamente mencionadas junto con sus respectivos mecanismos y características, identificadas por un ID.

ID	Configuración	δ_S	δ_R
B1	Baseline (DQN) (T=0, S=0, R=0)	–	–
S5	Shield Only (T=0, S=1, R=0)	5	–
S10		10	–
R10	Reward Only (T=0, S=0, R=1)	–	10
R20		–	20
SR5-10	Shield + Reward (T=0, S=1, R=1)	5	10
SR10-20		10	20
T1	Teacher Only (T=1, S=0, R=0)	–	–
TS5	Teacher + Shield (T=1, S=1, R=0)	5	–
TS10		10	–
TR10	Teacher + Reward (T=1, S=0, R=1)	–	10
TR20		–	20
TSR5-10	Teacher + Shield + Reward (T=1, S=1, R=1)	5	10
TSR7.5-15		7.5	15
TSR10-20		10	20

Tabla 1: Matriz de configuraciones evaluadas. (T: Teacher, S: Shield, R: Reward shaping).

6 Resultados

Debido a que el objetivo del trabajo es tanto garantizar la seguridad del agente tanto el aprendizaje como en la evaluación, hemos decidido analizar los resultados en ambas fases para entender de qué manera ha podido afectar la implementación de los distintos mecanismos de seguridad.

6.1 Fase de Entrenamiento

En cuanto a la fase de entrenamiento de todas las pruebas realizadas en Table 1, se ha obtenido la evaluación de las diferentes métricas definidas anteriormente, a lo largo de todos los pasos de entrenamiento. De todos los resultados, se han escogido las evoluciones más interesantes para mostrar en este estudio.

En cuanto a la eficiencia de los agentes entrenados mostrada en Figure 1, esta se calcula como los puntos medios obtenidos por paso. Se observa que el agente con una mejor energía durante todo el proceso es la configuración T1, pero muchas otras configuraciones consiguen una puntuación similar. Lo más destacable en este caso es el hecho de ver cómo las configuraciones con alta penalización (indicado con los números tras la "d.[en] el nombre), restringen demasiado a la hora de explorar, y es por ello que no se alcanza el mismo valor que en los casos donde la penalización es menor o nula.

Cabe destacar que en los agentes con un modelado de recompensa agresivo (como R20 o TSR10-20), observamos que la curva de recompensa comienza en valores negativos. Esto se debe a que el agente aprende rápidamente a temer la penalización por proximidad, adoptando una política excesivamente conservadora ('esconderse') antes de descubrir que la recompensa por comer píldoras compensa el riesgo. Este comportamiento restringe severamente la exploración, impidiendo que alcancen los niveles de rendimiento de las configuraciones con penalizaciones más leves o nulas.

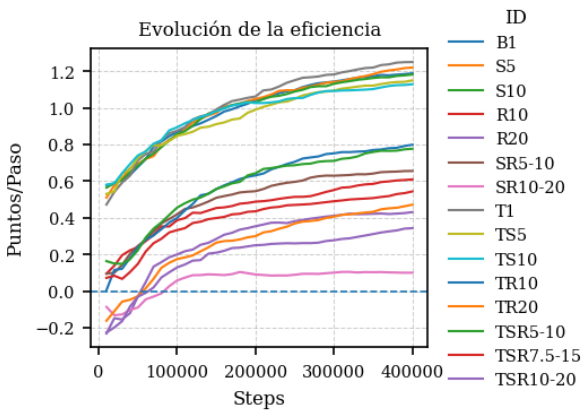

Figura 1: Evolución de la eficiencia media acumulada por paso en el entrenamiento

Si observamos los PPM obtenidos en el entrenamiento en la Figure 2, se observan muchos resultados muy variados. Sin embargo, se pueden observar tres tendencias: la primera, en la que las penalizaciones son demasiado grandes, son las que menos puntuación obtienen; si la penalización es intermedia, esta puntuación aumenta; y por último, de nuevo, las que mejor puntuación obtienen son las opciones con menos penalización.

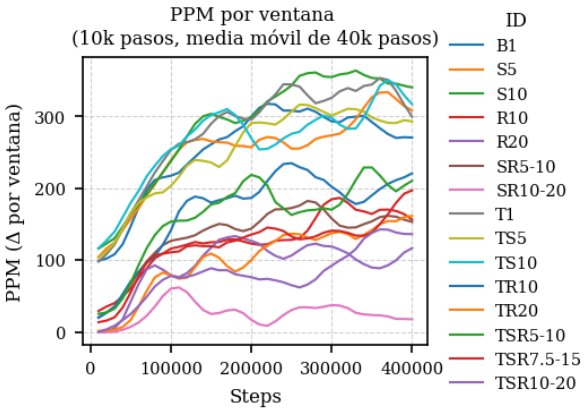

Figura 2: Evolución de los puntos por muerte obtenidos en el entrenamiento (media móvil sobre 40,000 pasos)

Contrario a lo que podría intuirse, la presencia del Escudo por sí sola no altera drásticamente la forma de la curva de aprendizaje respecto al agente base. Aunque el agente pueda conseguir menos puntos o un aprendizaje más lento debido a que el escudo bloquea acciones arriesgadas de alta recompensa, esto no representa realmente un problema, ya que el objetivo prioritario de este estudio es evitar muertes y garantizar la seguridad.

Sin embargo, el **factor determinante es la magnitud de la penalización** en el *Reward Shaping*. Mientras que el *Teacher* acelera el aprendizaje inicial, una penalización excesiva ($\delta_R \geq 20$) puede estancar el proceso en óptimos locales de "mínimo riesgo". A continuación, analizaremos qué tal se comportan estos agentes en la fase de validación.

6.2 Fase de Evaluación

En Table 2 se muestran los resultados medios obtenidos de las métricas planteadas tras 20 episodios de evaluación de los agentes previamente entrenados, con los IDs asociados a sus respectivas configuraciones, cuya configuración se puede apreciar en Table 1. Cabe destacar que métricas asociadas al escudo o al reward shaping solo son añadidas en aquellos experimentos donde se han utilizado.

ID	Rec.	Efic.	PPM	Int.	UnSteps
B1	526.00	1.02	175.33	–	–
S5	1074.00	1.37	358.00	0.09	–
S10	975.50	1.25	325.00	0.12	–
R10	629.0	0.94	209.66	–	0.11
R20	696.5	1.15	232.16	–	0.41
SR5-10	718.00	1.06	239.33	0.10	0.11
SR10-20	508.50	0.70	169.5	0.12	0.34
T1	632.00	1.12	210.67	–	–
TS5	535.50	1.22	178.50	0.13	–
TS10	792.0	0.98	264.00	0.11	–
TR10	788.5	1.24	262.83	–	0.12
TR20	815.5	1.22	271.83	–	0.35
TSR5-10	**1192.50**	**1.51**	**397.5**	**0.09**	**0.11**
TSR7.5-15	970.50	1.29	323.49	0.10	0.14
TSR10-20	260.00	0.77	86.66	0.23	0.68

Tabla 2: Resultados agregados por experimento.

De estos resultados se pueden extraer varias conclusiones sobre el impacto de los mecanismos de seguridad en el rendimiento.

6.2.1. Impacto del escudo de seguridad

El uso exclusivo del escudo de seguridad (S5 y S10) mejora notablemente el rendimiento, duplicando la recompensa media respecto al baseline. Sin embargo, como se muestra en Figure 3, una mayor frecuencia de intervención del escudo se asocia con una reducción del rendimiento, indicando que un control excesivo puede resultar perjudicial.

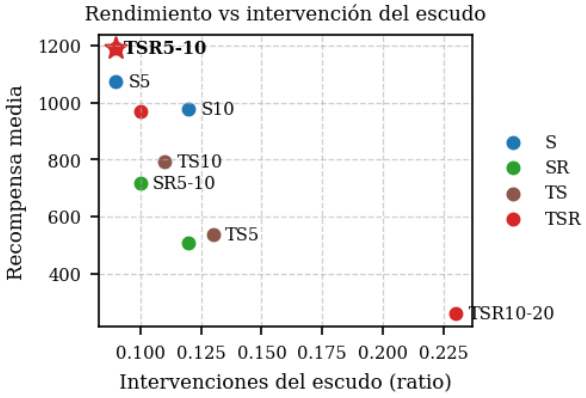

Figura 3: Gráfico de dispersión entre la recompensa contra el ratio de intervenciones del escudo de seguridad

6.2.2. Efecto del Aprendizaje por Imitación

El uso del Teacher de forma aislada (T1) mejora el rendimiento respecto al baseline (632.00 vs 526.00), aumentando la eficiencia a su vez. Sin embargo, esta mejora es modesta en comparación con el uso del escudo. Esto indica que, aunque las demostraciones humanas ayudan a arrancar el aprendizaje (superando el problema de la exploración inicial), el agente sigue siendo vulnerable a errores fatales una vez que toma el control si no tiene una red de seguridad activa.

6.2.3. Configuraciones Híbridas (*'sweet spot'*)

El hallazgo más relevante se encuentra en la combinación de los tres mecanismos (TSR).

La configuración TSR5-10 (Teacher + Shield con $\delta_S = 5$ + Reward con $\delta_R = 10$) obtiene el mejor rendimiento absoluto de todos los experimentos, con una recompensa media de 1192.5, el mayor valor de PPM (397.5) y también de eficiencia registrada (1.51).

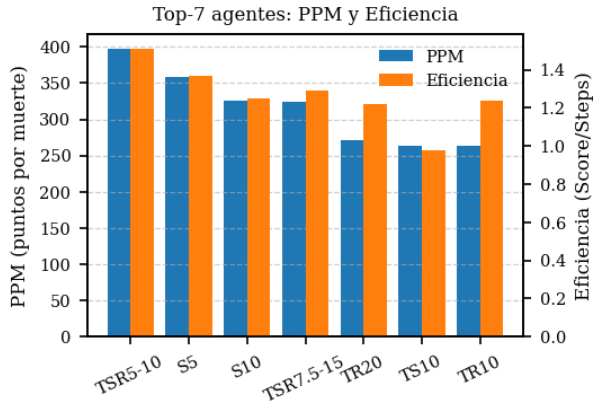

Figura 4: Gráfico de barras de los 7 mejores agentes por Puntos por Muerte (PPM) y Eficiencia

Esto confirma la hipótesis de que la combinación de un buen punto de partida (Teacher), una protección activa (Shield) y una señal de refuerzo negativa (Reward Shaping) crea una sinergia positiva, permitiendo al agente aprender conductas complejas y seguras.

6.2.4. Sensibilidad de los umbrales de distancia

Los resultados muestran una clara sensibilidad a los parámetros de distancia (δ). Distancias Cortas (5-10) favorecen el rendimiento. Al permitir que el agente se acerque más a los fantasmas antes de intervenir o penalizar, se le permite .ªrriesgar"lo suficiente para comer píldoras en zonas difíciles. En cambio, distancias Largas (10-20) tienen un efecto contraproducente drástico. Observamos que en SR10-20 y especialmente en TSR10-20, el rendimiento se desploma (260.00, muy por debajo incluso del baseline).

Esto nos lleva a suponer que Un umbral de seguridad demasiado conservador (20) genera un exceso de intervenciones (Int. altos) y penalizaciones constantes. Esto confunde al agente o lo bloquea en zonas "seguras"donde no hay puntos que ganar (correlacionado con la baja eficiencia), impidiendo la exploración efectiva del entorno.

6.2.5. La Paradoja del Maestro Experto

Un fenómeno notable se observó en el experimento con peor desempeño, **TSR10-20**. Paradójicamente, este agente fue entrenado con el conjunto de datos de imitación de mayor calidad, donde el operador humano alcanzó niveles avanzados del juego (Nivel 8). Este bajo rendimiento puede atribuirse a una disonancia cognitiva en el aprendizaje: el humano experto sobrevive asumiendo riesgos calculados (navegando muy cerca de los fantasmas), mientras que el sistema de de este experimento, configurado de forma conservadora ($\delta_R = 20$), penaliza severamente esas mismas acciones. El agente recibe señales contradictorias, lo que resulta en el colapso de la política. Además, al no haber observado errores en la demostración experta, el agente carece de ejemplos sobre cómo recuperarse de situaciones subóptimas (*distributional shift*).

6.2.6. Análisis de la Exposición al Riesgo (Métrica UnSteps)

La métrica *Unsafe* revela un comportamiento fascinante en el mejor agente obtenido. Al analizar la configuración óptima (**TSR5-10**), observamos que ha logrado la puntuación más alta manteniendo uno de los niveles de inseguridad más bajos de la tabla (**10.89**), comparable e incluso inferior a configuraciones conservadoras que obtuvieron mucho peor rendimiento (como SR5-10 con 718 puntos).

Esto contradice la intuición de que "para ganar más hay que arriesgar más". En su lugar, demuestra que el agente TSR ha alcanzado una política de seguridad eficiente: ha aprendido a navegar el entorno maximizando la recolección de puntos pero entrando en la zona de peligro únicamente cuando es estrictamente necesario y seguro, minimizando su exposición al riesgo de forma inteligente gracias a la combinación del conocimiento previo del Teacher y la guía del Reward Shaping.

Como se puede apreciar en Figure 5, nuevamente se reafirma la idea de que valores altos tanto para δ_S como δ_R supone un impacto en los PPM, esto encaja con que cuando más restrictivo son los mecanismos de seguridad, produce un empeoramiento a la hora de aprender la forma más óptima de jugar, a cambio de intentar evitar muertes durante el aprendizaje.

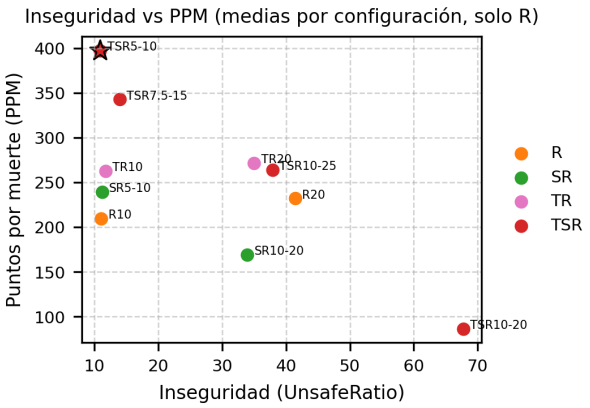

Figura 5: Gráfico de dispersión de los agentes por Puntos por Muerte (PPM) frente a ratio de Pasos Inseguros (UnSteps)

7 Conclusiones

El desarrollo de este trabajo ha permitido constatar que la integración de mecanismos de seguridad en el Aprendizaje por Refuerzo profundo (Deep RL) no solo cumple una función de protección, sino que actúa como un catalizador fundamental para la eficiencia del aprendizaje.

Aunque las técnicas de Shielding y Reward Shaping mejoran el rendimiento por separado, su combinación con el Aprendizaje por Imitación (Teacher) produce resultados superiores a la suma de sus partes. El çalentamientoïnicial proporciona al agente una política base competente, mientras que el escudo actúa como una red de seguridad que impide el catastrophic forgetting (olvido catastrófico) durante las primeras etapas de exploración autónoma.

Por otro lado, se ha demostrado que la existencia de un punto de equilibrio crítico (sweet spot) en la definición de los márgenes de seguridad. Mientras que distancias de seguridad ajustadas (5-10) potencian la capacidad del agente para maximizar la recompensa en entornos de alto riesgo, los umbrales conservadores coartan la exploración, impidiendo que el agente acceda a recompensas valiosas y degradando su rendimiento.

Por último, la intervención directa en las acciones (Shield) ha demostrado ser más efectiva que la mera penalización (Reward Shaping) para garantizar la supervivencia a corto plazo. Sin embargo, el modelado de recompensa es esencial para interiorizar el concepto de peligro a largo plazo, reduciendo la dependencia del agente respecto a los sistemas de asistencia externos.

Este proyecto valida que es posible entrenar agentes de RL más seguros y eficientes en entornos como Ms. Pac-Man sin sacrificar el rendimiento, siempre que los parámetros se ajusten para permitir una exploración efectiva.

8 Trabajo futuro

Queremos destacar algunos puntos que consideramos que podrían tenerse en cuenta para proyectos futuros con el objetivo de mejorar la calidad del trabajo.

En primer lugar, podría mejorarse a futuro la cantidad de muestras aportadas durante el calentamiento por imitación, en nuestro caso, utilizamos 7500 demostraciones, que equivalían a unos 15 minutos de juego, pero podría ampliarse para obtener mejores resultados.

Otro aspecto a tener en cuenta son los parámetros de distancia (δ), los cuales han sido estáticos durante todo el entrenamiento. Una propuesta innovadora sería implementar un enfoque donde el escudo sea muy restrictivo al inicio (para evitar muertes rápidas) y se vaya relajando progresivamente a medida que el agente aprende. Esto permitiría transicionar suavemente de un comportamiento asistido a una autonomía total.

Referencias

[1] Mohammed Alshiekh, Roderick Bloem, Ruediger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding, 2017.

[2] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2023.

[3] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.*, 16(1):1437–1480, January 2015.

[4] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Gabriel Dulac-Arnold, Ian Osband, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations, 2017.

[5] W. Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: the tamer framework. In *Proceedings of the Fifth International Conference on Knowledge Capture*, K-CAP '09, page 9–16, New York, NY, USA, 2009. Association for Computing Machinery.

[6] Ankita Kushwaha, Kiran Ravish, Preeti Lamba, and Pawan Kumar. A survey of safe reinforcement learning and constrained mdps: A technical survey on single-agent and multi-agent safety, 2025.

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[8] Van Havermaet, Stef and Khaluf, Yara and Simoens, Pieter. No more hand-tuning rewards : masked constrained policy optimization for safe reinforcement learning. In *AAMAS '21, Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1344–1352. IFAAMAS, 2021.