Universidad Internacional de Valencia

Aprendizaje Continuo en Agentes Autónomos

Titulación:
Máster Universitario en Inteligencia Artificial
Curso académico 2023-2024

Alumno/a: Ballester Gúrpide, Íñigo
D.N.I: 18965098B

Director/a de TFM: Borja González León

Convocatoria:

Adelantada

15 noviembre 2023

De:
Planeta Formación y Universidades

"Dave, my mind is going. I can feel it. I can feel it.
My mind is going."

HAL 9000 (2001: A Space Odissey)

Agradecimientos

Me gustaría agradecer a toda mi familia, amigas y amigos por su apoyo incondicional durante este proyecto, a mi director Borja González León tanto por su paciencia como por su valiosa ayuda y guía, y a la Meri y la Silvi por aguantarme cada día. Su apoyo ha sido fundamental para mi éxito.

También quiero destacar la ayuda de varios investigadores en este campo, en particular a Anirudh Goyal, Kanika Madan y Yuhun-Shi por sus valiosas sugerencias y aclaraciones.

Por último a dos palomas que llevan visitando mi balcón desde hace un tiempo y que han resultado más fáciles de entrenar por refuerzo que buena parte de los algoritmos aquí descritos.

Índice general

Índice de Figuras

Índice de Tablas

Índice de Algoritmos

Lista de Acrónimos

ACC Precisión Media o *Average Accuracy*.

AGI Inteligencia Artificial General o *Artificial General Intelligence*.

ANPyC *Adversarial Neural Pruning and Synaptic Consolidation*.

BLIP *Bit-level Information Preserving*.

BWT Transferencia hacia Atrás o *Backward Transfer*.

CF Olvido Catastrófico o *Catastrophic Forgetting*.

CL Aprendizaje Continuo o *Continual Learning*.

CLS Sistemas de Aprendizaje Complementario o *Complementary Learning Systems*.

DL Aprendizaje Profundo o *Deep Learning*.

DNN Red Neuronal Profunda o *Deep Neural Network*.

DRL Aprendizaje Profundo por Refuerzo o *Deep Reinforcement Learning*.

EWC *Elastic Weight Consolidation*.

FIM *Matriz de Información de Fisher*.

GAN Redes Generativas Antagónicas o *Generative Adversarial Network*.

HCAM *Hierarchical Chunk Attention Memory*.

IA Inteligencia Artificial.

IID Independientes e Idénticamente Distribuida/os o *Independent and Identically Distributed*.

MAML *Model-Agnostic Meta-Learning*.

MDP Proceso de Decisión Markov o *Markov Decision Processes*.

PC Consolidación Sináptica o *Policy Consolidation*.

POMDP Proceso de Decisión Markov parcialmente observable o *Partially-observable Markov Decision Process*.

PPO *Proximal Policy Optimization*.

RIMs *Recurrent Independent Mechanisms*.

RL Aprendizaje por Refuerzo o *Reinforcement Learning*.

SOTA Estado del Arte o *State of the Art*.

SPP *Soft Parameter Pruning*.

VAE Autocodificadores Variacionales o *Variational Autoencoder*.

Glosario

Actor-Critic Método de aprendizaje por refuerzo donde el "actor" elige acciones y el "crítico" evalúa su rendimiento, mejorando así la toma de decisiones y la estrategia de aprendizaje.

Aprendizaje *Few-Shot* Técnica de aprendizaje automático que permite a un modelo aprender e identificar nuevas tareas o categorías con muy pocos ejemplos de entrenamiento..

Resumen

El Olvido Catastrófico o *Catastrophic Forgetting* (CF) es un problema intrínseco asociado al aprendizaje mediante redes neuronales profundas (DNN). Se refiere a la tendencia de las redes a olvidar la información previamente aprendida cuando se las entrena en nuevas tareas y adquiere especial relevancia en el aprendizaje secuencial de tareas.

Se trata de un reto importante para el desarrollo de sistemas de inteligencia artificial (IA) que necesiten aprender y adaptarse a nueva información a lo largo del tiempo y está intrínsecamente relacionado con el Aprendizaje Continuo o *Continual Learning* (CL) y la consecución de la Inteligencia Artificial General o *Artificial General Intelligence* (AGI).

Este problema ha sido ampliamente documentado en contextos de aprendizaje supervisado y ha demostrado ser un desafío significativo en la conservación y adaptabilidad de habilidades adquiridas. En el ámbito del Aprendizaje por Refuerzo o *Reinforcement Learning* (RL)), donde un agente busca maximizar una recompensa acumulada a través de interacciones con un entorno, el olvido catastrófico presenta implicaciones aún más complejas y menos comprendidas, en buena parte debido a la propia naturaleza dinámica y en continua evolución de los entornos habituales en dicho ámbito.

Este trabajo de fin de máster explora la cuestión del CL en el Aprendizaje Profundo o *Deep Learning* (DL), con especial atención a su impacto en el ámbito del Aprendizaje Profundo por Refuerzo o *Deep Reinforcement Learning* (DRL). En general, este trabajo pretende contribuir tanto a la comprensión de los mecanismos tras el olvido catastrófico como a los diferentes enfoques a fin de mitigar sus efectos en el aprendizaje profundo por refuerzo, proporcionando una revisión detallada del estado del arte en estrategias de mitigación existentes. Inspirándose en las soluciones propuestas para el aprendizaje supervisado y en los desafíos únicos del RL, se proponen nuevos enfoques que adaptan y combinan estrategias previas para enfrentar este fenómeno.

Se espera que este trabajo contribuya a una comprensión más amplia del olvido catastrófico en el dominio del RL y a la identificación de diferentes vías para su mitigación con el fin de lograr sistemas DRL más robustos y adaptativos en el aprendizaje secuencial.

Keywords: Catastrophic Forgetting, Deep Reinforcement Learning, Continual Learning, Knowledge Distillation, Meta-Learning, Synaptic Consolidation

Introducción 1

El modo o mecanismo por el cual los humanos aprendemos y adquirimos conocimientos y habilidades de manera continua a lo largo de nuestras vidas, sin olvidar las experiencias pasadas, sigue siendo un fascinante campo de estudio dentro del ámbito de la neurociencia. La plasticidad de las sinapsis neuronales es una característica esencial del cerebro que facilita cambios físicos en la estructura neuronal y nos permite aprender, recordar y adaptarnos a entornos dinámicos. Asimismo el cerebro lleva también a cabo tareas complementarias como son generalizar a través de las experiencias y retener recuerdos específicos de acontecimientos de tipo episódico. Aunque los mecanismos neuronales exactos siguen siendo poco conocidos (Parisi et al., 2019), si existen teorías que en cierta medida nos ayudan a describirlos, como pueden ser la teoría hebbiana sobre la plasticidad (Hebb, 1950), o la teoría de Sistemas de Aprendizaje Complementario o *Complementary Learning Systems* (CLS) (McClelland y O'Reilly, 1995). Esta facultad, que nos resulta algo intrínseco y natural, es en cambio uno de los retos más importantes a los que se enfrenta la investigación en Inteligencia Artificial, y de hecho gran parte de dicha investigación bebe y se complementa de los avances dentro del campo de la neurociencia. El objetivo es, en definitiva, que arquitecturas de aprendizaje automático como las redes neuronales puedan adquirir la capacidad de aprendizaje permanente y continuo (Chen y Liu, 2018).

El fenómeno del Olvido Catastrófico o *Catastrophic Forgetting* tiene lugar en sistemas de aprendizaje basados en redes neuronales artificiales. Este fenómeno, también referido como interferencia catastrófica, hace referencia a la tendencia de las redes a olvidar información previamente aprendida cuando se las entrena de modo secuencial en nuevas tareas, o dicho de otro modo, el entrenamiento en un nuevo conjunto de ítems puede alterar drásticamente el rendimiento en los ítems aprendidos previamente. Es un problema intrínseco asociado al propio proceso de *backpropagation*, tal como fue descrito por (McCloskey y Cohen, 1989). Dicho problema es en gran medida una manifestación de la interferencia entre tareas. Cuando una red es entrenada en una tarea y posteriormente en otra, la información aprendida en la primera tarea puede verse degradada o completamente olvidada. Esto suele implicar que una nueva tarea probablemente anulará los pesos (*weights*) que se han aprendido en el pasado y, por tanto, degradará el rendimiento del modelo para las tareas anteriores. Los parámetros aprendidos previamente se reutilizan para la nueva tarea, y el modelo olvida las tareas anteriores a medida que dichos parámetros se mueven en el espacio de búsqueda (fig. 1.1) durante la fase de entrenamiento para la nueva tarea (Fayek, 2019). Si no se soluciona este problema, una

red neuronal no será capaz de adaptarse a un escenario de aprendizaje continuo, al olvidar la información/conocimiento existente cuando aprende nuevas tareas.

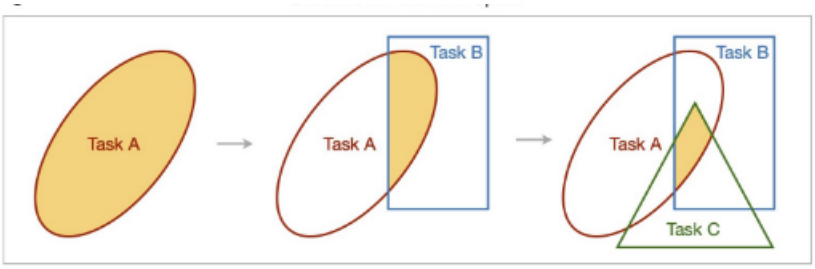


Figura 1.1: Espacio de parámetros en secuencias de tareas (Wang et al., 2023)

A fin de evitar o minimizar este problema, el agente debe por un lado ser capaz de adquirir nuevo conocimiento y refinar el existente, y por otro lado evitar en la medida de lo posible que este nuevo conocimiento interfiera negativamente con el conocimiento ya adquirido. El grado en que un sistema puede integrar nueva información de tal manera que no interfiera con el conocimiento consolidado es lo que se conoce como el dilema entre estabilidad-plasticidad (French, 1999), ampliamente estudiado tanto en sistemas biológicos como computacionales (Parisi et al., 2019). Si un modelo es demasiado estable, le será muy difícil incorporar nueva información de los futuros datos o tareas nuevas de entrenamiento. Por otro lado, un modelo que presente mucha plasticidad sufre grandes cambios de peso y olvida las representaciones aprendidas previamente (Chen y Liu, 2016).

Como se ha recalcado, el concepto de olvido catastrófico y el paradigma del aprendizaje continuo están íntimamente relacionados, siendo imposible abordar este último sin abordar las posibles causas del primero. A lo largo de los años, se han investigado y desarrollado diversos enfoques y estrategias para abordar este problema. Hay cierta diversidad de opiniones a la hora de agrupar estos enfoques, y no siempre es posible una organización estricta en estos grupos, lo cual da lugar a que, a lo largo de la literatura, estas categorizaciones sean un tanto difusas y en ocasiones muy ligadas a ámbitos específicos como puedan ser el aprendizaje supervisado y la clasificación visual, el procesado natural del lenguaje o al RL en particular (Khetarpal et al., 2020). Aun así, es posible trazar una taxonomía general que en gran medida permite englobar buena parte de los enfoques y estrategias desde la perspectiva conceptual del Aprendizaje Continuo (Wang et al., 2023)

- **Enfoques reguladores:** Estos métodos añaden un término de penalización a la función de pérdida durante el entrenamiento que ayude a la consolidación del conocimiento mientras se aprenden tareas o distribuciones de datos posteriores, intentando alcanzar un compromiso entre la estabilidad y la plasticidad. En dicha categoría se incluirían por ejemplo métodos de regularización sináptica que se centrarían en reducir el cambio en parámetros importantes que se aprendieron en tareas anteriores.

- **Enfoques de ensayo/repetición:** Estos métodos implican conservar algunos datos de tareas anteriores y utilizarlos durante el entrenamiento en nuevas tareas. Esto puede ha-

cerse almacenando directamente algunos de los datos anteriores (repetición) o también utilizando modelos generativos para producir datos similares a los de tareas anteriores (pseudoensayo).

- **Enfoques Basados en la Arquitectura:** Esta categoría engloba métodos que buscan mitigar el CF y facilitar el CL a través de la configuración arquitectónica inherente del modelo. Por ejemplo, mediante la designación de subespacios de parámetros exclusivos para cada tarea a lo largo de la red, con una arquitectura que puede ser fija o dinámica. Otros métodos en esta categoría se enfocan en desglosar los modelos en componentes compartidos entre tareas y específicos de cada tarea. Finalmente, también se pueden considerar enfoques modulares que emplean subredes o submódulos paralelos para abordar tareas incrementales de manera distinta, sin elementos preestablecidos para tareas compartidas o específicas.

- **Enfoques basados en la optimización:** Los enfoques basados en la optimización en el CL están diseñados para manipular directamente el proceso de optimización con el fin de evitar el CF y facilitar la retención y transferencia de conocimientos a medida que los modelos aprenden con el tiempo. Se incluirían métodos que introducirían modificaciones en el proceso de descenso de gradiente para que el proceso de optimización favorezca más el CF, o estrategias basadas en el Meta-Aprendizaje, por el que los agentes aprenden cómo aprender nuevas tareas de manera más eficiente, optimizando el proceso de aprendizaje en sí mismo en lugar de solo para una tarea específica.

Cabe señalar, tal y como indican varios de los autores, que muchos de estos métodos y estrategias pueden estar estrechamente relacionados entre sí y ser sinérgicos en cierta medida (Wang et al., 2023), y que por otro lado existen varias categorías híbridas que incluirían trabajos que combinarían varios de los enfoques entre sí o que harían uso de estrategias adicionales (Lomonaco, 2019). Asimismo, la propia naturaleza de las tareas, el tipo de datos disponibles, la eficiencia de las muestras y muchos otros factores afectan en buena medida al modo de aprendizaje; y cada ámbito (aprendizaje supervisado, aprendizaje por refuerzo, etc.) presenta sus propios retos específicos. Conviene remarcar también, que en líneas generales los autores parten de marcos conceptuales diferenciados. Por una parte es común el estudio del CF y sus causas desde una perspectiva enfocada en los propios mecanismos por los que la red neuronal almacena la información en sus pesos o parámetros (especialmente en el campo del aprendizaje supervisado y la clasificación visual), y como ello afecta directamente al desempeño del modelo en un escenario de CL; y por otro lado, enfoques desarrollados desde el marco teórico del CL (más comunes en investigaciones dentro del ámbito del RL), desde donde se estudiarían estrategias que permitan que un sistema pueda aprender eficientemente de manera continua, lo cual por definición implicaría una mitigación de-facto del CF.

En particular, el ámbito del Aprendizaje Profundo por Refuerzo o *Deep Reinforcement Learning* es en esencia un problema de aprendizaje continuo debido precisamente a su forma de aprender explorando durante el proceso de entrenamiento y es particularmente sensible al ol-

vido catastrófico. En el aprendizaje supervisado, al aprender con un conjunto de datos fijo, el fenómeno del olvido suele darse al intentar aprender un nuevo conjunto de datos. En el aprendizaje por refuerzo el proceso de aprendizaje es esencialmente dinámico, por el que se persigue un objetivo en marcha. El conjunto de datos se actualiza durante todo el periodo y las muestras obtenidas mediante la interacción con el entorno siguen cambiando junto con el proceso de exploración del agente. El agente recibe un flujo de experiencias no Independientes e Idénticamente Distribuida/os o *Independent and Identically Distributed* (IID) debido a cambios en la política (*policy*), la distribución de estados y la dinámica del entorno, y esto conduce a una baja eficiencia de la muestra. Los nuevos datos suelen hacer que la red entrenada cambie mucho para adaptarse a ellos, pero olvida lo que ha aprendido en el proceso de entrenamiento anterior aunque sea útil. Esta es una limitación de la aplicación de redes neuronales como funciones de aproximación en los métodos de aprendizaje por refuerzo (Dong et al., 2020). En este sentido, el ámbito del aprendizaje profundo por refuerzo es por su propia definición y formulación, un banco de pruebas ideal para el estudio e investigación de la problemática del olvido catastrófico y su influencia directa en el Aprendizaje Continuo o *Continual Learning*.

1.1. Motivación

En una era donde la IA juega un papel central en la tecnología y en la vida diaria, garantizar que los sistemas puedan aprender y adaptarse de manera continua es esencial. Sin embargo, el olvido catastrófico en Red Neuronal Profunda o *Deep Neural Network* (DNN) representa un obstáculo importante. Resolver este desafío podría catalizar el desarrollo de sistemas de IA más avanzados y adaptativos, especialmente en aplicaciones donde la capacidad de aprender secuencialmente es crítica. Uno de los objetivos supremos en el campo de la IA es la consecución de la Inteligencia Artificial General o *Artificial General Intelligence*. Para lograrlo, es fundamental que los sistemas puedan aprender de manera continua y adaptativa. Abordar el Olvido Catastrófico en DRL es, por lo tanto, un paso crucial en el camino hacia la realización de la AGI.

Como se ha indicado en la sección anterior, es la propia naturaleza dinámica del DRL, donde los agentes interactúan y evolucionan con entornos en constante cambio, la que nos proporciona un excelente marco en el que estudiar dicho fenómeno. Esto crea un escenario y campo de investigación que es a la vez más complejo y menos explorado respecto al olvido catastrófico. Mientras que en el aprendizaje supervisado se han hecho avances significativos con métodos y estrategias SOTA para mitigar este olvido durante el aprendizaje secuencial de tareas, el DRL ofrece por su parte muchos aspectos sobre los que explorar dicho fenómeno, más teniendo en cuenta que la formulación sobre la que se fundamenta el DRL nos lleva directamente a confrontarnos con el paradigma del aprendizaje continuo o permanente.

La motivación de este trabajo no es simplemente una exploración académica de estrategias SOTA existentes, veo también un vasto campo de oportunidades para adaptar, experimentar y optimizar estas soluciones SOTA en el contexto de DRL. Es por ello que me propongo también estudiar su particular relevancia en entornos DRL en donde algunas de estos métodos no han

sido estudiados en profundidad, así como investigar nuevos enfoques basados en hibridación o adaptaciones específicas a DRL de dichas estrategias y su comparación con métodos previos y líneas de base claramente definidas.

Cada avance que logremos en la comprensión y mitigación del olvido catastrófico en DRL tiene el potencial de transformar cómo construimos sistemas de IA en el futuro. Sistemas más resilientes, más adaptativos y más inteligentes. Mi esperanza, al final de esta investigación, es realizar un aporte al campo del DRL, ofreciendo un entendimiento más profundo del Olvido Catastrófico y proponiendo en la medida de lo posible soluciones y vías de exploración de dicho fenómeno.

1.2. Planteamiento del problema

Son varias las preguntas que se pretenden investigar y responder en la medida de lo posible en este trabajo y que definen su alcance: ¿En qué medida está relacionado el olvido catastrófico con el aprendizaje continuo?, ¿Qué soluciones y enfoques se han propuesto hasta la fecha para abordar el olvido catastrófico en el aprendizaje supervisado?, ¿Cuáles son los mecanismos subyacentes al olvido catastrófico en el aprendizaje profundo por refuerzo?, ¿Cuál es la eficacia de las diferentes técnicas de mitigación del olvido catastrófico en el aprendizaje profundo por refuerzo? ¿Cómo se adaptan las soluciones propuestas en el aprendizaje supervisado al dominio del aprendizaje por refuerzo?

Se propone realizar una revisión bibliográfica general sobre el estado del arte de los diversos enfoques existentes para la mitigación del olvido catastrófico y la facilitación del aprendizaje continuo. El objetivo de dicha revisión es no solo profundizar en los aspectos teóricos detrás de dichos enfoques, sino también contribuir a contextualizar el problema dentro del marco específico del aprendizaje por refuerzo y facilitar las posteriores labores de desarrollo e implementación de modelos, sentando unos criterios base y delimitando el alcance del trabajo de investigación que se realizará.

Se establecerá una metodología de experimentación debidamente justificada, tanto para las líneas de base como para los enfoques finalmente seleccionados y los enfoques híbridos propuestos dentro del marco de investigación. A grandes rasgos, dicha metodología se construirá sobre los siguientes componentes:

- Una línea de base construida sobre el algoritmo de aprendizaje por refuerzo *Proximal Policy Optimization* (PPO) (Schulman et al., 2017), siendo este algoritmo un estándar en muchas investigaciones de aprendizaje por refuerzo profundo debido a su estabilidad de entrenamiento, eficiencia en muestreo y relativa simplicidad en su implementación y/o adaptación

- Los diversos enfoques de mitigación de CF se implementarán sobre dicho algoritmo, a fin de asegurar la consistencia tanto en los entrenamientos como en los resultados obtenidos.

- Para todos los experimentos se propone un entorno EL común basado en Minigrid. Este entorno ha sido diseñado específicamente para la investigación en inteligencia artificial y aprendizaje profundo y presenta varias ventajas: se trata de un entorno 2D en cuadrícula simplificada y acciones discretas, es eficiente, ligero y versátil, pudiéndose configurar fácilmente para una gran variedad de tareas. Como se ha indicado, está orientado a la investigación, permitiendo probar de manera sencilla nuevas ideas y algoritmos en un entorno controlado antes de transferirlas a entornos más complejos (Chevalier-Boisvert et al., 2023).

- En la evaluación de resultados, se propone el uso de una serie de métricas de evaluación enfocadas al aprendizaje secuencial de tareas y de uso común en escenarios de aprendizaje continuo (Lopez-Paz y Ranzato, 2017), como son por ejemplo la Transferencia hacia Atrás o *Backward Transfer* (BWT) y la Precisión Media o *Average Accuracy* (ACC) sobre la secuencia total de tareas. Dichas métricas nos permitirán evaluar de una manera consistente y objetiva el desempeño de los diferentes enfoques ante una batería de secuencias de tareas predefinida.

Es fundamental recalcar que el desafío que supone el olvido catastrófico, al mismo tiempo que pone a prueba la capacidad de adaptabilidad y aprendizaje de los modelos, ofrece una oportunidad única de adentrarnos en la esencia de cómo las redes neuronales procesan y retienen la información. Esta investigación, por lo tanto, no solo busca aportar soluciones prácticas al reto, sino también obtener una perspectiva más profunda sobre las dinámicas internas de los sistemas de aprendizaje automático. Al hacerlo, esperamos no solo contribuir al campo del aprendizaje profundo por refuerzo, sino también allanar el camino hacia la creación de sistemas más resilientes y autónomos, capaces de aprender de manera más natural y duradera.

1.3. Objetivos

Los principales objetivos de este trabajo son, por un lado, el estudio del fenómeno del Olvido Catastrófico y su profunda interrelación con el paradigma del Aprendizaje Continuo o Permanente, y sus implicaciones en el campo del Aprendizaje por Refuerzo y los actuales enfoques SOTA actuales. Más concretamente, se estudiará experimentalmente el fenómeno en varias arquitecturas propuestas, se propondrán nuevos enfoques combinados y se evaluará su rendimiento a la hora de abordar dicho fenómeno.

1. **Objetivo parcial 1.** Presentar una visión general del paradigma del Aprendizaje Continuo y su papel clave hacia la consecución de una Inteligencia Artificial General y los retos a los que se enfrenta, con especial atención a la mitigación del Olvido Catastrófico como uno de los principales peldaños a superar en la persecución de este objetivo, pero también su implicación directa en la eficiencia y mejora de los actuales sistemas basados en IA y Aprendizaje Automático.

2. **Objetivo parcial 2.** Presentar un estudio más detallado sobre el fenómeno del Olvido Catastrófico y sus causas, al tiempo que se revisan las líneas de trabajo y estrategias actuales y del SOTA para abordar esta problemática, con especial atención a los enfoques empleados en el ámbito del Aprendizaje por Refuerzo.

3. **Objetivo parcial 3.** Proponer enfoques novedosos basados en la combinación y/o integración de arquitecturas que apenas se han probado o no se han probado antes de esa manera, e implementar y evaluar estos enfoques frente a líneas de base en varios escenarios experimentales de Aprendizaje por Refuerzo.

1.4. Estructura de la memoria

Este Trabajo Fin de Máster está estructurado de la siguiente manera:

- **Resumen**: Breve síntesis del trabajo.
- **Introducción**: Incluye la motivación, planteamiento del problema, objetivos y la estructura de la memoria.
- **Contexto y Estado del Arte**: Aborda una visión general del Aprendizaje por Refuerzo, el Aprendizaje Continuo y el Olvido Catastrófico, y diversos enfoques orientados al Aprendizaje Continuo en RL, como estrategias de regularización, ensayo y repetición, enfoques basados en la arquitectura, y meta-aprendizaje. Termina con conclusiones sobre estos temas.
- **Metodología**: Detalla los enfoques propuestos y su justificación, abarcando enfoques basados en regulación y consolidación (como EWC y BLIP), enfoques de tipo arquitectónico (como el meta-aprendizaje con arquitecturas modulares), enfoques híbridos, y la implementación, que incluye el entorno, el escenario y las tareas, el procedimiento experimental y las métricas de evaluación.
- **Resultados y Discusión**: Presenta los resultados de los enfoques BLIP y EWC, enfoques híbridos, *Policy Consolidation*, Meta-aprendizaje con RIMs, HCAM, y una comparación y discusión de los resultados.
- **Conclusiones**: Resume los hallazgos y contribuciones clave del trabajo.
- **Limitaciones y Perspectivas de Futuro**: Expone las limitaciones identificadas en la investigación y sugiere direcciones para trabajos futuros.

Contexto y Estado del Arte

2

2.1. Visión general del Aprendizaje por Refuerzo

El Aprendizaje por Refuerzo o *Reinforcement Learning* (RL) representa un paradigma de aprendizaje automático distinto a los enfoques supervisados y no supervisados, centrado en la toma de decisiones secuenciales y la interacción con un entorno. En el RL, el agente aprende a actuar dentro de un entorno más o menos complejo, recibiendo retroalimentación en forma de recompensas o penalizaciones que utiliza para actualizar su estrategia de toma de decisiones y mejorar su rendimiento a lo largo del tiempo. A diferencia del aprendizaje supervisado y no supervisado, donde los agentes se entrenan con datos etiquetados o buscan patrones en datos sin etiquetar respectivamente, en el RL el proceso de aprendizaje involucra la exploración del entorno, donde el agente realiza acciones y observa los resultados en forma de estados y recompensas. Estas experiencias se utilizan para actualizar las políticas del agente, es decir, su estrategia de toma de decisiones, que determina qué acción realizar en cada estado.

El RL se estructura en torno a la interacción entre el agente y el entorno, que se modela comúnmente como un Proceso de Decisión Markov parcialmente observable o *Partially-observable Markov Decision Process* (POMDP). Un POMDP proporciona un marco matemático $(\mathcal{S}, \mathcal{A}, p_s, r, \gamma)$ que describe cómo el entorno responde a las acciones del agente y cómo se asignan las recompensas (Sutton y Barto, 2018). Aquí, $\mathcal{S}$ y $\mathcal{A}$ denotan los espacios de estado y de acción respectivamente (que pueden ser discretos o continuos). La función p_s define la probabilidad (o función de densidad en el caso de valores continuos) de transición al estado $s_{t+1} \in \mathcal{S}$ condicionada a que el agente tome la acción $a_t \in \mathcal{A}$ en el estado $s_t \in \mathcal{S}$. El factor de descuento o *discount factor* γ determina la importancia relativa de las recompensas inmediatas versus las futuras. Finalmente, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ define una función que mapea cada transición (s_t, a_t) a una recompensa escalar (Figura 2.1).

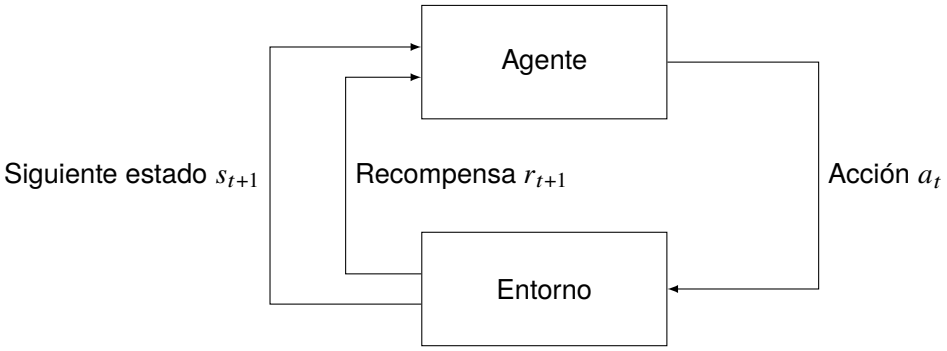


Figura 2.1: Diagrama del paradigma del Aprendizaje por Refuerzo

En este contexto, los componentes clave del RL son:

- **Agente:** La entidad de aprendizaje o decisión.
- **Entorno:** El mundo externo con el que el agente interactúa.
- **Estado:** Una representación del entorno en un momento dado.
- **Acción:** Una decisión tomada por el agente que afecta al estado.
- **Recompensa:** Una señal inmediata que evalúa la acción tomada.
- **Política o *policy*:** Una estrategia que guía la selección de acciones del agente.
- **Función de valor o *value*:** Una estimación de la recompensa futura esperada.
- **Modelo:** Una representación del entorno que puede predecir cómo cambiarán los estados y qué recompensas se recibirán en respuesta a las acciones del agente.

El objetivo del agente RL es determinar una política, definida como una función de probabilidad sobre acciones dado el estado $\pi(a_t|s_t)$, que maximice la suma esperada de recompensas futuras o retorno:

$$\pi^* = \arg\max_{\pi} \sum_{t} \mathbb{E}_{\pi}[r(s_t, a_t)] \tag{2.1}$$

En el RL, los métodos se pueden clasificar ampliamente en dos categorías según cómo abordan la tarea de aprender la política óptima:

- Métodos basados en la función *value*, que estima cuánto de buena es cada acción o estado en términos de la cantidad de recompensa esperada que se puede obtener comenzando desde ese estado o después de tomar esa acción. Con esta información, la política se puede derivar indirectamente escogiendo siempre la acción con el mayor valor esperado.
- Métodos basados en el gradiente de la *policy*, que se centran directamente en aprender la *policy* que el agente debe seguir. La función *policy* asignaría estados (o pares de estado-acción) a probabilidades de seleccionar cada acción posible. El objetivo es encontrar los parámetros que maximicen la recompensa esperada ajustando estos parámetros en la dirección del gradiente de esta recompensa esperada.

Los métodos Actor-Critic son una combinación de estos dos enfoques, que intentaría aprovechar la ventaja de ambos. La arquitectura Actor-Critic consta de dos componentes principales:

- *Actor:* Propone acciones según la *policy* actual que se está aprendiendo.
- *Critic:* Evalúa las acciones tomadas por el actor utilizando una función de *value*.

El *actor* actualiza la *policy* en dirección a lo que parece ser una recompensa mayor, mientras que el *critic* evalúa las acciones del actor y ajusta la estimación de la función de *value*. Este enfoque permite al agente aprender más eficientemente, ya que la *policy* y el *value* se actualizan de manera concurrente.

En aprendizaje por refuerzo profundo (DRL), tanto la *policy* como la función de *value* se obtienen mediante estimadores paramétricos (redes neuronales); con θ y ϕ como sus respectivos parámetros. La *policy* se actualiza mediante el *policy gradient*, mientras que el *value* suele actualizarse mediante la diferencia temporal o simulaciones de Monte Carlo. En la práctica, para una secuencia de transiciones $\{s_t, a_t, r_t, s_{t+1}\}_{t=0,N}$, se utiliza la siguiente función de pérdida de *policy gradient* (que incluye además un término de entropía H comúnmente usado para promover la exploración del agente):

$$L_{PG} = -\frac{1}{N}\sum_{t'=t}^{t+N}\left(A_{t'}\log\pi(a_{t'}|s_{t'},\theta) + \alpha H(\pi(s_{t'},\theta))\right), \tag{2.2}$$

donde α es el coeficiente de entropía y A_t es el estimador de ventaja generalizada, función que mide cuánto mejor (o peor) es tomar una acción específica en comparación con el promedio de todas las acciones posibles en un estado dado. Se favorecen así acciones con mayores retornos esperados frente aquellas con menores retornos que el promedio.

Uno de los algoritmos Actor-Critic más relevantes es el conocido como *Proximal Policy Optimization* (PPO) (Schulman et al., 2017), el cual es usado comúnmente como línea de base principal en la experimentación RL y que a tal efecto se adoptará en la experimentación realizada en este trabajo. PPO optimiza las *policies* de una manera que evita grandes cambios en ellas, lo cual es beneficioso para la estabilidad del entrenamiento. Al igual que otros algoritmos Actor-Critic, hace uso de una función de ventaja para estimar cuánto mejor es una acción en comparación con la *policy* actual, pero en este caso aplica un recorte (*clip*) en la actualización de la *policy* a fin de mantener los pasos de actualización dentro de límites razonables y prevenir cambios demasiado grandes. Esto ayuda a mantener las actualizaciones de la *policy* dentro de un rango que asegura la convergencia y evita la inestabilidad en el entrenamiento. La función de pérdida a optimizar se define como:

$$L(s,a,\theta_k) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}A^{\pi_{\theta_k}}(s,a),\ g(\epsilon, A^{\pi_{\theta_k}}(s,a))\right), \tag{2.3}$$

donde g es la función que implementa el mecanismo de *clipping*

$$g(\epsilon, A) = \begin{cases}(1+\epsilon)A & \text{if } A \geq 0\\ (1-\epsilon)A & \text{if } A < 0.\end{cases} \tag{2.4}$$

La función objetivo o de pérdida de PPO se modifica de tal manera que si el *ratio* entre la nueva *policy* π_θ y la antigua π_{θ_k} cae fuera del rango [1-ϵ, 1+ϵ], la función de pérdida no seguirá

facilitando un incremento en la actualización de la *policy* en esa dirección.

PPO es menos sensible a los hiperparámetros que otros algoritmos y ofrece resultados sólidos en una variedad de entornos, lo que lo hace adecuado para las comparaciones. Por otro lado, PPO es relativamente simple de implementar y escala bien a problemas complejos, lo que permite usarlo como un punto de referencia estándar para evaluar nuevos enfoques o arquitecturas en PPO.

El Annexo A incluye una descripción más detallada del pseudocódigo correspondiente al algoritmo 1 PPO

2.2. Aprendizaje Continuo y Olvido Catastrófico

El aprendizaje continuo (CL) es un paradigma de aprendizaje automático cuyo objetivo es aprender de forma adaptativa a lo largo del tiempo aprovechando las tareas aprendidas previamente para mejorar la generalización en tareas futuras. Por lo tanto, el CL estudia el problema del aprendizaje secuencial a partir de un flujo continuo de datos, extraídos de una distribución potencialmente no estacionaria, y la reutilización de los conocimientos adquiridos a lo largo de la vida evitando el CF (Biesialska et al., 2020). En una definición más general, el CL sería el desarrollo constante e incremental de comportamientos cada vez más complejos. Esto incluye el proceso de construir comportamientos complicados sobre los ya desarrollados al tiempo que se es capaz de aplicar, adaptar y generalizar a nuevas situaciones las habilidades previamente aprendidas. También está estrechamente relacionado con paradigmas como el aprendizaje permanente, el aprendizaje *online*, el aprendizaje *lifelong* o el aprendizaje sin fin *never-ending u open-ended* (Mundt et al., 2020).

El agente de aprendizaje continuo debe ser capaz de transferir y adaptar lo que ha aprendido de experiencias, datos o tareas anteriores a situaciones nuevas y hacer uso de experiencias más recientes para mejorar el rendimiento de las capacidades aprendidas anteriormente. La Tabla 2.1, sin ser exhaustiva, resume algunas de la desiderata de propiedades esperada en CL (Biesialska et al., 2020). En la práctica, los sistemas actuales de CL suelen relajar al menos uno de los requisitos enumerados en la tabla. La mayoría de los métodos siguen el paradigma del aprendizaje *offline*, especialmente en el ámbito del Aprendizaje Supervisado, siendo el aprendizaje *online* algo intrínseco al RL debido a su interacción continua con el entorno.

Propiedad	**Definición**
Retención de conocimiento	El modelo no es propenso al olvido catastrófico.
Forward transfer	El modelo aprende una nueva tarea mientras reutiliza conocimientos adquiridos en tareas previas.
Backward transfer	El modelo mantiene o mejora el rendimiento en tareas previas después de aprender una nueva tarea.
Aprendizaje *online*	El modelo aprende de un flujo de datos continuo.
Sin delimitación entre tareas	El modelo aprende sin requerir delimitación entre tareas.
Capacidad fija del modelo	El tamaño de la memoria es constante independientemente del número de tareas.

Tabla 2.1: Desiderata del Aprendizaje Continuo

.

De una manera más formal, el paradigma del CL en el contexto del RL podría definirse de la siguiente manera:

1. **Secuencia de Tareas**: Al agente se le presentan una secuencia de tareas $\{T_1, T_2, \ldots, T_n\}$, donde cada tarea T_i está caracterizada por un Proceso de Decisión de Markov (MDP) $(S_i, A_i, P_i, R_i, \gamma_i)$.

2. **Red Neuronal Única**: Se emplea una única red neuronal f con parámetros θ para aproximar la *policy* π_θ o las funciones de *value* $V_\theta(s)$ o $Q_\theta(s, a)$ en todas las tareas.

3. **Objetivo de Aprendizaje Continuo**: El objetivo del agente es optimizar los parámetros θ para maximizar el retorno esperado para la tarea actual T_i, entendiendo implícitamente que el proceso de aprendizaje no debe interrumpir significativamente el rendimiento en las tareas previamente aprendidas $T_1, \ldots, T_{i-1}$. Formalmente, el agente busca:

$$\max_\theta \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma_i^t R_i(s_t, \pi_\theta(s_t))\right] \tag{2.5}$$

   mientras se esfuerza por mantener o mejorar el rendimiento en las tareas anteriores a través de una retención eficaz del conocimiento.

Tal y como fue mencionado en la Introducción 1, el olvido o interferencia catastrófica es uno de los principales retos para muchos sistemas de aprendizaje automático basados en redes neuronales. El DRL es esencialmente un problema de CL debido a su modo de aprendizaje a partir de un flujo de datos no estacionario, explorando mientras se aprende, y es por ello particularmente vulnerable al CF.

La razón subyacente al CF en DRL es la generalización global y la representación superpuesta de las redes neuronales profundas (DNN). Las DNN suelen asumir que los datos de entrenamiento se distribuyen de forma IID. Sin embargo, en DRL, un agente recibe un flujo de experiencia no IID debido a cambios en la *policy*, la distribución de estados y la dinámica del entorno, o simplemente debido a la estructura inherente del entorno. Los datos de entrenamiento suelen estar correlacionados en el tiempo y la *policy* del agente cambia gradualmente a medida que aprende. Esto hace que la distribución de los datos de entrenamiento sea no estacionaria, lo que puede provocar el olvido CF. Por otro lado, por definición, los métodos de RL se basan en gran medida en el *bootstrapping*, por el que el agente RL utiliza su propia función de *value* estimada como objetivo, lo que hace que las *output* objetivo tampoco sean estacionarias. Cabe también destacar que los búferes de repetición de experiencias, de uso común en RL, pueden acentuar la no estacionariedad de los datos de entrenamiento al muestrear experiencias con mayores errores de diferencia temporal (Zhang et al., 2021).

Una vez que la distribución de los datos de entrenamiento encuentra una deriva notable, es probable que se produzca una interferencia catastrófica y una reacción en cadena, lo que resultará en un deterioro repentino del rendimiento del entrenamiento

2.3. Enfoques orientados al Aprendizaje Continuo en RL

2.3.1. Estrategias de Regularización

Los enfoques de aprendizaje continuo basados en la regularización intentan mantener un balance: por un lado, proteger lo que el modelo ya ha aprendido, y por otro lado, permitirle adaptarse para incluir nueva información. Suelen inspirarse en modelos teóricos de neurociencia y se basan principalmente en el concepto del dilema de estabilidad-plasticidad (capítulo 1). A priori, este balance sería más factible cuanto más amplio sea el espacio multidimensional de las representaciones compartido entre las tareas.

Desde una perspectiva computacional, esto se modela generalmente añadiendo términos de regularización adicionales en la función de pérdida a fin de consolidar el conocimiento previo mientras se aprende en nuevas tareas. Dichos términos de regularización incluirían el cálculo de penalizaciones y la destilación de conocimientos. Teniendo esto en cuenta, varios autores subdividen los enfoques reguladores en enfoques basados en la preservación de parámetros o estructurales y enfoques basados en la destilación de conocimiento o funcionales (Khetarpal et al., 2020; Mundt et al., 2020; Zhang et al., 2021; (Wang et al., 2023)):

- En los enfoques estructurales, la regularización estaría destinada a proteger explícitamente los parámetros de las tareas aprendidas, e impondrían restricciones a los cambios en los pesos de la arquitectura de un modelo durante el aprendizaje de nuevas tareas a fin de aliviar el CF.
- Los enfoques de regularización funcional se inspiran generalmente en la destilación del conocimiento (Hinton et al., 2015), un enfoque propuesto originalmente para la compresión de modelos. La destilación de conocimientos se refiere al proceso de utilizar una red neuronal como *soft target* para otra (en lugar de una etiqueta definida). Se introduciría pues una pérdida de destilación almacenando la predicción de una muestra de datos o de una tarea previa para su uso futuro como *target*. La destilación puede utilizarse para aumentar las experiencias de entrenamiento de una red, proporcionando un nuevo objetivo auxiliar para que coincida con la red actual que se está entrenando. En el contexto de la RL, este *target* puede referirse a una *policy* o a una función de *value*. En el contexto del CL, la destilación es una estrategia popular para asegurar que el agente preserve el conocimiento importante de tareas pasadas siempre que sea posible.

Estos enfoques suelen inspirarse en modelos teóricos de neurociencia que sugieren que el conocimiento consolidado puede protegerse del olvido mediante sinapsis con una cascada de estados que producen distintos niveles de plasticidad (Benna y Fusi, 2016).

2.3.2. Ensayo y Repetición

Los métodos de *ensayo o repetición* se basan en reusar datos o experiencias de tareas previas durante el entrenamiento de una nueva tarea como estrategia para mitigar el CL. Aunque en teoría se podría solucionar el aprendizaje continuo almacenando y repitiendo todos los

datos vistos, esta estrategia es obviamente impracticable debido a los altos costos de memoria y procesamiento que implicaría (Mundt et al., 2020). Por lo tanto, se prefiere seleccionar y conservar un conjunto representativo de muestras previas. Estas muestras se almacenan en un pequeño búfer de memoria y se utilizan para aproximar la distribución completa de los datos observados, reproduciéndolas de manera estratégica con el fin de intentar preservar lo aprendido en tareas previas hasta ese momento. Debido al espacio de almacenamiento limitado, los principales retos consisten en cómo implementar y cómo explotar eficazmente este búfer de memoria. En cuanto a la implementación, las muestras de entrenamiento previo conservadas deben seleccionarse, comprimirse, aumentarse y actualizarse cuidadosamente para recuperar de forma adaptativa la información pasada. (Wang et al., 2023)

Un enfoque derivado de las técnicas de ensayo es el *pseudoensayo o pseudorehearsal*, también conocido como *repetición generativa*. En lugar de almacenar muestras reales de tareas anteriores, esta estrategia emplea un modelo generativo para crear nuevas muestras que son estadísticamente similares a los datos antiguos (Khetarpal et al., 2020). Al intercalar estas muestras generadas con datos de tareas nuevas durante el entrenamiento, el modelo refuerza continuamente los conocimientos antiguos a la vez que da cabida a información nueva. Arquitecturas generativas como las Redes Generativas Antagónicas o *Generative Adversarial Network*s (GANs) y los Autocodificadores Variacionales o *Variational Autoencoder*s (VAEs) se utilizan a menudo para este propósito. El enfoque de repetición generativa aborda directamente las limitaciones de memoria de los métodos de repetición tradicionales. En lugar de ampliar el almacenamiento de memoria para dar cabida a nuevas muestras de ejemplo, limita el coste de memoria al tamaño del modelo generativo. Sin embargo, hay que tener en cuenta que el propio modelo generativo puede sufrir a su vez CL durante su entrenamiento (Cossu et al., 2021)

En conclusión, tanto las estrategias de *ensayo/repetición* como las de *pseudoensayo/generación* proporcionan vías para el aprendizaje continuo al tiempo que mitigan el olvido catastrófico. Las técnicas de ensayo se centran en la selección y gestión de un búfer de memoria que contenga las muestras clave, mientras que las estrategias generativas emplean modelos para simular distribuciones pasadas, limitando así el uso de memoria.

2.3.3. Enfoques basados en la Arquitectura:

Los enfoques basados en la arquitectura se basan en la adaptación de la estructura de la red neuronal de forma que se conserven los conocimientos antiguos al tiempo que se acomoda la nueva información. Dada la amplitud de esta premisa, es quizás la categoría que más diversidad de criterios presenta en la literatura, por lo que la siguiente clasificación responde más al propósito particular de este trabajo.

En una primera clasificación, dichos enfoques pueden dividirse en aislamiento de parámetros y arquitecturas dinámicas (Wang et al., 2023). Esta división se basa en si la arquitectura de la red es estática o capaz de expandirse. En el aislamiento de parámetros, ciertas regiones de la red neuronal se reservan para tareas específicas, protegiéndolas de las actualizaciones que puedan producirse durante el aprendizaje de nuevas tareas. Esto se consigue a menudo

mediante máscaras binarias que congelan efectivamente el aprendizaje en partes seleccionadas de la red, minimizando la interferencia que pudiera haber entre el nuevo aprendizaje y el conocimiento existente. En el segundo enfoque, a menudo se inyectan nuevas capas de forma dinámica para aumentar un modelo con módulos adicionales que den cabida a nuevas tareas. El principal inconveniente de estas estrategias es el número cada vez mayor de parámetros, que aumenta la complejidad del modelo.

Por otro lado, es posible afrontar el problema desde la perspectiva de modularidad de la arquitectura. Las redes modulares adoptan el concepto de subredes o submódulos paralelos. Estas redes permitirían la reutilización del conocimiento mediante la combinación estratégica de estas subredes. Dichas redes modulares no predefinirían los componentes como tareas compartidas o tareas específicas, sino que se basan en una configuración dinámica que fomenta el aprendizaje de tareas específicas mediante el ensamblaje de estos módulos. Para mejorar la composicionalidad, la red puede organizarse en módulos especializados en distintos aspectos de una tarea. Esta modularización no solo reforzaría la capacidad de la red para componer conocimientos de distintas áreas, sino que en principio ayudaría también a facilitar el CL. Este enfoque modular otorgaría a la red una capacidad amplificada tanto de abstracción como de especialización (Khetarpal et al., 2020).

2.3.4. Meta-Aprendizaje:

El meta-aprendizaje, o aprender a aprender (*learning to learn*), surge como un enfoque significativo dentro del RL para permitir a los agentes adaptarse rápidamente a nuevas tareas o entornos. Esta capacidad es crucial en escenarios de CL, en los que se espera que los agentes actúen en una serie de tareas no estacionarias y aprendan continuamente, acumulando los conocimientos aprendidos en tareas anteriores y aplicándolos a tareas más recientes. En el meta-aprendizaje se intenta que el modelo obtenga sesgos inductivos (*inductive bias*) válidos para varios escenarios a partir a partir de los propios datos, en lugar de depender de un sesgo inductivo impuesto en su diseño. En principio, esto permitiría lograr sistemas de aprendizaje más flexibles y generalizables, capaces de ajustar sus sesgos en función de las tareas a las que se enfrenten (Wang et al., 2023).

A diferencia del enfoque tradicional, que aborda cada nueva tarea de manera aislada, el meta-aprendizaje introduce un modelo de aprendizaje dual: el aprendizaje directo de la tarea, que se considera un ciclo de aprendizaje interno, y el aprendizaje de cómo mejorar este proceso de aprendizaje de tareas, que se define como un ciclo externo. Ambos ciclos de aprendizaje se optimizan simultánea o iterativamente, permitiendo al agente actualizar y perfeccionar su capacidad de aprender con cada nueva tarea que enfrenta (Dong et al., 2020).

En el meta-aprendizaje existen diferentes estrategias que son especialmente relevantes para el RL y el CL: la *detección del contexto*, el *aprendizaje para adaptarse* y el *aprendizaje para explorar*.

- *La detección del contexto* en el meta-aprendizaje se centra en la identificación de diferentes tareas o cambios en el entorno que pueden requerir que el agente adapte su

comportamiento. Al reconocer el contexto, un agente puede recuperar los conocimientos o habilidades más relevantes que haya aprendido previamente y aplicarlos a la situación actual. Esto puede ser crucial, ya que los agentes RL a menudo se enfrentan a entornos no estacionarios en los que la dinámica subyacente puede cambiar con el tiempo. Un agente capaz de detectar el contexto puede discernir estos cambios y ajustar su política en consecuencia. Los sistemas de aprendizaje continuo se benefician de la detección del contexto al reconocer eficazmente cuándo la tarea ha cambiado y las antiguas políticas pueden haber dejado de ser eficaces. Esto minimiza el CF y maximiza la transferencia positiva o FWT.

- *El aprendizaje para adaptarse* implica el entrenamiento de modelos en una variedad de tareas para que puedan aprender la estructura de cómo varían las tareas y la mejor manera de adaptarse a nuevas tareas rápidamente con pocos ejemplos (Aprendizaje *Few-Shot*). Para un agente RL, aprender a adaptarse significa ser capaz de modificar su *policy* rápidamente en respuesta a nuevas tareas o cambios en la distribución de tareas. Esto se consigue a menudo mediante algoritmos como *Model-Agnostic Meta-Learning* (MAML) (Finn et al., 2017), que optimizan una representación θ que puede adaptarse rápidamente a nuevas tareas en escenarios de pocos ejemplos. La idea básica tras MAML y otros algoritmos dentro de esta familia como *Reptile* (Nichol et al., 2018), es proporcionar una mejor inicialización de los pesos para cada nueva tarea entrenando los parámetros de los modelos utilizando un conjunto de datos diferente. Cuando se utiliza para una nueva tarea, el modelo ofrece un mejor rendimiento al utilizar parámetros ya inicializados para afinar la arquitectura mediante uno o varios pasos de gradiente. Aprender a adaptarse es fundamental para la CL porque permite al agente mantener su rendimiento en una serie de tareas sin necesidad de volver a entrenarse desde cero.

- Finalmente, *las estrategias de aprendizaje* para explorar intentan paliar las dificultades de exploración en entornos con pocas recompensas. El meta-aprendizaje puede mejorar la capacidad de exploración de un agente identificando qué tipos de estrategias de exploración funcionan mejor en diferentes contextos, acelerando así el proceso de aprendizaje. En un escenario de CL, un agente no solo debe aprender de las nuevas experiencias, sino también explorar de forma que no interrumpa los conocimientos adquiridos en tareas anteriores, explorando selectivamente zonas del espacio de acción que estén poco desarrolladas.

Estas estrategias no se excluyen mutuamente y a menudo se combinan para crear sistemas de meta-aprendizaje robustos que puedan aprender y adaptarse eficazmente a una amplia gama de entornos y tareas. El meta-aprendizaje, más que una técnica o enfoque como tal, se trata de una forma de abordar el problema del CL y ello permite que pueda ser combinado o integrado con varios de los diferentes enfoques descritos en secciones anteriores. El meta-aprendizaje, más que una técnica o enfoque como tal, se trataría de una perspectiva a la hora de abordar el problema del CL y ello permite que pueda ser combinado o integrado con varios de los diferentes anteriormente descritos.

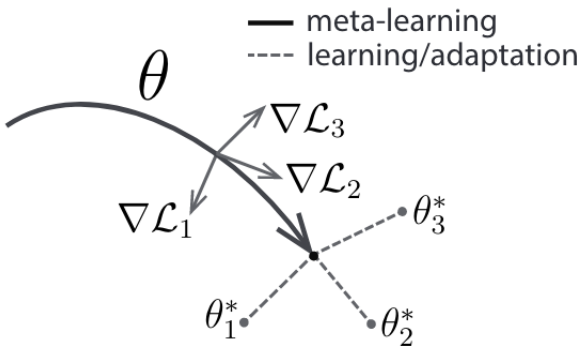


Figura 2.2: Algoritmo *Model-Agnostic Meta-Learning* (MAML) (Finn et al., 2017)

Metodología 3

3.1. Enfoques propuestos y Justificación

Las técnicas y enfoques seleccionados para este estudio han sido escogidos por su alineación con los principios fundamentales del CL (Kaushik et al., 2021). Entre ellos, destacan enfoques reguladores como EWC (*Elastic Weight Consolidation*), BLIP y diversas hibridaciones propuestas. Estas metodologías se prefirieron debido a su fiel cumplimiento de los requisitos del CL, evitando estrategias como el *replay*, que pueden resultar excesivas en el uso de memoria, y arquitecturas dinámicas, las cuales tienden a incrementar el número de parámetros en juego, complicando así el proceso de entrenamiento. Por otro lado, este trabajo también explora el desempeño de arquitecturas diseñadas para CL, como HCAM y meta-RIMs, evaluando su capacidad para mitigar el olvido catastrófico (CF). Este enfoque detallado y considerado permite entender mejor cómo diferentes estrategias y arquitecturas pueden abordar eficazmente los retos asociados al Aprendizaje Continuo.

Dentro de las diversas hipótesis y cuestiones iniciales que se pretenden investigar se incluyen:

- ¿Se beneficia el efecto granular de BLIP sobre partes de un parámetro, con el uso posterior de una máscara que permita liberar algunos parámetros para tareas posteriores?
- ¿Es posible aprovechar dicho efecto granular de BLIP en conjunción con la regularización que EWC efectúa sobre todo el parámetro?
- El sesgo inductivo en arquitecturas como RIMs/meta-RIMS favorecen la identificación de la modularidad en las tareas aprendidas y han proporcionado algunas mejoras en escenarios de aprendizaje *few-shot*, ¿podrían entonces hacer frente mejor al CF?

3.2. Enfoques basados en regulación y consolidación

3.2.1. Enfoque: EWC Elastic Weight Consolidation

El enfoque *Elastic Weight Consolidation* (EWC) (Kirkpatrick et al., 2016) se basa en el concepto de consolidación sináptica observado en los cerebros biológicos. Las sinapsis involucradas en retener las tareas aprendidas tienen menos probabilidades de alterarse, lo que permite

conservar los conocimientos antiguos al tiempo que se adquiere nueva información. El mecanismo clave de EWC consiste en identificar la importancia de los parámetros de la red (pesos) para las tareas aprendidas previamente y, a continuación, aplicar una penalización a los cambios en estos parámetros importantes mientras se aprenden nuevas tareas. Para ello hace uso de la *Matriz de Información de Fisher* (FIM), una medida que cuantifica la importancia de cada parámetro en términos de su contribución al rendimiento en las tareas aprendidas. Cuando la red se entrena en una nueva tarea, el EWC añade un término de regularización a la función de pérdida. Este término penaliza los cambios en los pesos identificados como importantes para tareas anteriores, de forma proporcional a su importancia determinada por el FIM. Esta penalización impide que la red altere significativamente esos pesos cruciales, preservando así el conocimiento de las tareas anteriores. En esencia, EWC restringe los cambios significativos en los pesos más cruciales de la red (sinapsis en el contexto de la red neuronal), preservando el conocimiento adquirido en tareas anteriores.

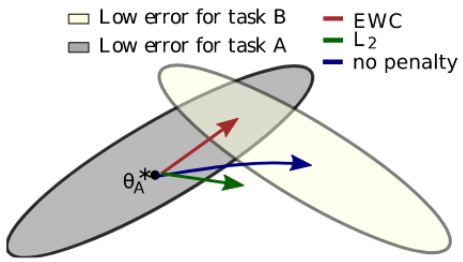


Figura 3.1: Solapamiento de espacios de solución

EWC emplea una estrategia bayesiana para estimar los parámetros θ de la red. Parte del concepto de que los parámetros θ^* óptimos para una tarea pueden encontrarse en un espacio de soluciones con errores aceptables. Cuando se aprenden varias tareas, el objetivo de la red es encontrar un conjunto de parámetros que se encuentre en la región de solapamiento de estos espacios de solución 3.1. Cuando una red neuronal aprende una tarea, podemos representar la probabilidad de sus parámetros (pesos), θ, dados los datos $\mathcal{D}$, utilizando el teorema de Bayes:

$$P(\theta|\mathcal{D}) \propto P(\mathcal{D}|\theta)P(\theta) \tag{3.1}$$

Aquí, $P(\theta|\mathcal{D})$ es la probabilidad posterior de los parámetros dados los datos, $P(\mathcal{D}|\theta)$ es la probabilidad de los datos dados los parámetros, y $P(\theta)$ es la probabilidad a priori de los parámetros.

Dada la complejidad de calcular la distribución posterior exacta después de aprender una tarea, el EWC emplea la aproximación de Laplace. Esta aproximación trata la distribución posterior como una gaussiana centrada en los valores de los parámetros θ^* obtenidos después de aprender la tarea:

$$P(\theta|\mathcal{D}) \approx \mathcal{N}(\theta^*, \mathbf{F}^{-1}) \tag{3.2}$$

Aquí, $\mathbf{F}$ es la *Matriz de Información de Fisher* (FIM), y $\mathcal{N}(\theta^*, \mathbf{F}^{-1})$ representa la distribución gaussiana con media θ^* y covarianza $\mathbf{F}^{-1}$. La FIM desempeña un papel crucial en la determinación de la importancia de cada peso en la red y en esta transformación se utiliza como aproximación de la diagonal de la matriz hessiana, que mide la curvatura de la curva en el espacio de soluciones y serviría para cuantificar cuánto influye el cambio en un parámetro en la predicción de salida. Sin embargo, la hessiana implica derivadas parciales de segundo orden de la función de pérdida con respecto a los parámetros y su cálculo puede ser muy laborioso. En cambio, el FIM lo aproxima considerando la varianza de los gradientes (derivadas de primer orden) de la *log-likelihood*, lo que facilita su cálculo e implementación.

$$F_i = \mathbb{E}\left[\left(\frac{\partial \log P(\mathcal{D}|\theta)}{\partial \theta_i}\right)^2\right] \tag{3.3}$$

A continuación, EWC modifica la función de pérdida añadiendo un término de regularización que penaliza los cambios en los parámetros importantes. La función de pérdida EWC puede escribirse como:

$$\mathcal{L}(\theta) = \mathcal{L}_{new}(\theta) + \sum_i \lambda \cdot F_i \cdot (\theta_i - \theta_i^*)^2 \tag{3.4}$$

donde $\mathcal{L}_{new}(\theta)$ es la pérdida para la nueva tarea, λ es un hiperparámetro que controla la fuerza de la regularización, F_i es la información de Fisher para el parámetro θ_i, y $(\theta_i - \theta_i^*)^2$ es la diferencia al cuadrado entre los parámetros óptimos de la tarea actual y la anterior, penalizando las desviaciones significativas de los parámetros importantes.

Este término de regularización se implementa como una penalización cuadrática y, por tanto, puede imaginarse como un muelle que ancla los parámetros a la solución anterior, de ahí el nombre de elástico. Es importante destacar que la rigidez de este muelle no debe ser la misma para todos los parámetros, sino que debe ser mayor para aquellos parámetros que más influyen en el rendimiento durante la tarea A. El término de regularización de la función de pérdida impide que la red introduzca grandes cambios en estos parámetros críticos, preservando así los conocimientos adquiridos en tareas anteriores y permitiendo al mismo tiempo cierta flexibilidad para aprender nueva información.

La regularización EWC es de uso relativamente común dentro tanto dentro del aprendizaje supervisado como del RL. A efectos de este trabajo, se ha adaptado el código existente en PyTorch integrando la regularización EWC en el algoritmo PPO para su experimentación en entornos y tareas MiniGrid.

El A incluye una descripción en pseudocódigo del algoritmo EWC 2

3.2.2. Enfoque: BLIP Bit Level Information Preserving

El algoritmo BLIP (*Bit-level Information Preserving*) (Shi et al., 2022) se basa en la teoría de la información y está motivado por el concepto de ganancia de información en el contexto del CL, con el objetivo de preservar la información obtenida por los parámetros del modelo en cada tarea. Según este método, el CF en un modelo se debe a la pérdida de información

sobre los parámetros de tareas anteriores cuando se aprende una nueva tarea. BLIP funciona preservando la información ganada en los parámetros del modelo mediante la actualización de estos parámetros en términos de bits. Cuando una red se entrena en una nueva tarea, gana información que se refleja en los cambios de sus parámetros. BLIP considera un parámetro del modelo como una serie de bits binarios. Inicialmente, todos los bits son libres de cambiar de estado, pero a medida que el modelo aprende una tarea, algunos bits se vuelven seguros (es decir, dejan de cambiar). La idea central de BLIP es identificar y congelar estos bits seguros para evitar olvidar información de las tareas anteriores.

Para ello, el modelo se entrena primero en una nueva tarea con los pesos de la red cuantizados. Tras el entrenamiento, BLIP estima la ganancia de información que cada tarea proporciona a cada parámetro. Esta estimación orienta sobre qué bits del parámetro deben congelarse (es decir, hacerse inmutables) para evitar el olvido. BLIP tiene una sobrecarga de memoria constante, ya que solo necesita hacer un seguimiento del número total de bits congelados para cada parámetro. De manera más detallada, los pasos que sigue BLIP son los siguientes 3.2:

1. *Cuantización de parámetros*: Inicialmente, los parámetros del modelo se cuantizan en N bits (por ejemplo, 20 bits). Esta cuantificación es crucial, ya que constituye la base para los pasos posteriores de estimación de la ganancia de información y congelación de bits.

2. *Entrenamiento con parámetros cuantizados*: Cuando se introduce una nueva tarea, el modelo se entrena con estos parámetros cuantizados. Durante esta fase, ciertos bits de los parámetros, que se han determinado como significativos a partir de tareas anteriores, se congelan o se mantienen constantes para evitar el olvido.

3. *Estimación de la ganancia de información*: Después del entrenamiento, el método consiste en estimar la ganancia de información sobre los parámetros que aporta la nueva tarea. Para ello, se evalúa la reducción de la *entropía de Shannon* de los parámetros cuantizados. La ganancia de información sugiere cuántos bits congelar.

4. *Congelación de bits en función de la ganancia de información*: El núcleo de BLIP es congelar los bits que han ganado certeza a través del aprendizaje de la nueva tarea. Esto se hace desde las posiciones de bits más altas (más significativas) a las más bajas (menos significativas). Estos bits congelados se convierten en "seguros" ya no deberían cambiar de estado en el aprendizaje posterior. Este paso garantiza que no se pierda la información de las tareas anteriores, mitigando el problema del CF.

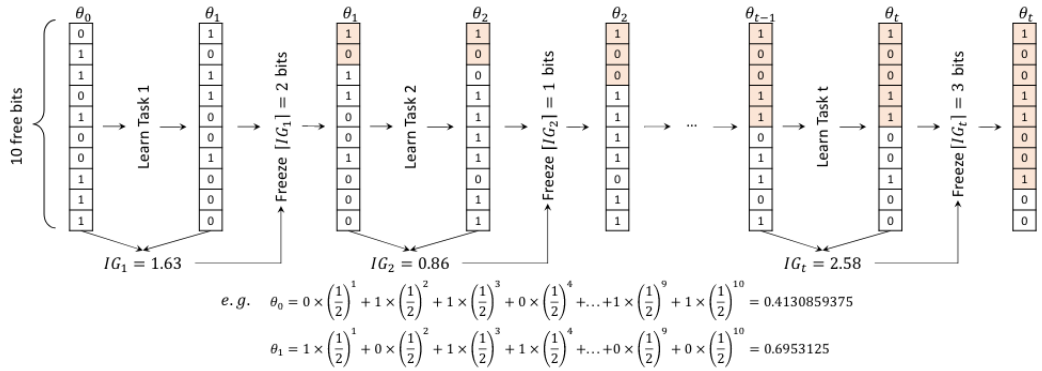


Figura 3.2: Bit-level Information Preserving (Shi et al., 2022)

De una manera más formal, la ganancia de información se definiría como la reducción de la *entropía de Shannon* sobre $Q(\theta, N)$ tras el entrenamiento en una tarea o conjunto de datos $\mathcal{D}_t$:

$$\text{IG}(Q(\theta N), \mathcal{D}_t) = H(Q(\theta_{0:t-1}, N)) - H(Q(\theta_{0:t}, N)) \tag{3.5}$$

Para calcular esto, se requiere el conocimiento de las distribuciones posteriores $p(\theta_{0:t-1})$ y $p(\theta_{0:t})$ antes y después de aprender sobre $\mathcal{D}_t$, respectivamente. Sin embargo, la distribución posterior $p(\theta)$ es intratable y necesita ser aproximada. En BLIP, la *matriz de información de Fisher* se utiliza para aproximar la distribución posterior de los parámetros del modelo θ después de aprender sobre un conjunto de datos $\mathcal{D}$. La matriz de información de Fisher $F_{\mathcal{D}}(\theta)$ se calcula como sigue:

$$F_{\mathcal{D}}(\theta) = \mathbb{E}_{x\sim X, y\sim p_\theta(x)}\left[\left(\frac{\partial \ln p_\theta(y|x)}{\partial \theta}\right)^2\right] \tag{3.6}$$

Aquí, $\mathbb{E}$ denota la expectativa sobre el conjunto de datos $\mathcal{D}$ (representado como X, Y), $p_\theta(y|x)$ es la probabilidad de predicción del modelo, y $\frac{\partial \ln p_\theta(y|x)}{\partial \theta}$ es la derivada de la *log-likelihood* con respecto a los parámetros θ.

La matriz de información de Fisher, por tanto, proporciona una forma de cuantificar cuánto contribuye cada parámetro a las predicciones del modelo sobre el conjunto de datos $\mathcal{D}$. Esta aproximación es crucial para estimar la entropía de Shannon, que a su vez se utiliza para calcular la ganancia de información (IG) requerida.

A efectos de los experimentos de este trabajo, el algoritmo BLIP se ha implementado sobre PPO y se ha adaptado para que sea compatible con los entornos discretos basados en cuadrículas que proporciona MiniGrid. El Anexo A A incluye una descripción en pseudocódigo del algoritmo BLIP 4

3.2.3. Enfoque: Policy Consolidation en RL

La idea fundamental del algoritmo de Consolidación Sináptica o *Policy Consolidation* (PC) (Kaplanis et al., 2019) se inspira en la neurociencia, concretamente en el modelo sináptico de

(Benna y Fusi, 2016), que hace uso de una serie de ecuaciones diferenciales para modelizar los cambios sinápticos en neuronas a lo largo del tiempo. Este modelo conceptualiza el proceso de consolidación sináptica mediante una cadena de vasos comunicantes que representarían diferentes escalas temporales de almacenamiento de la memoria. En esta analogía, el nivel de líquido de cada vaso representa el peso sináptico utilizado en el cálculo neuronal. El flujo entre los vasos representa la transferencia y regulación de los pesos sinápticos a lo largo de diferentes escalas de tiempo, en las que los vasos más profundos almacenan información durante periodos más largos.

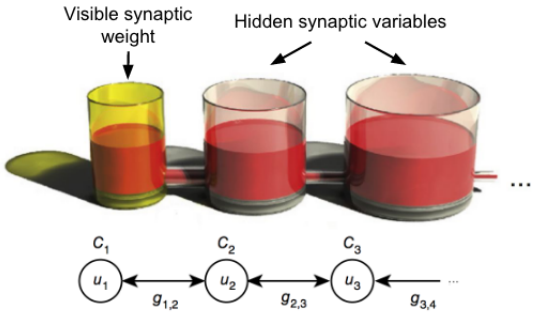


Figura 3.3: Modelo de Benna-Fusi de Consolidación Sináptica

El modelo PC extiende el concepto de consolidación sináptica al ámbito del RL. Incorpora una red con una *policy* visible que interactúa con una cascada de redes con *policies* ocultas, cada una de las cuales representa diferentes escalas temporales. Estas redes ocultas cumplen dos funciones principales

1. Recordar la *policy* del agente en varias escalas temporales.

2. Regularizar la *policy* actual basándose en el histórico de *policies*, lo que aumenta la estabilidad del aprendizaje del agente y evita desviaciones significativas de las *policies* anteriores.

El modelo PC utiliza métodos de *policy gradient*, concretamente una versión del algoritmo PPO 1. En el PPO estándar, la función objetivo ayuda a evitar grandes cambios de *policy* que podrían degradar el rendimiento. Mientras que PPO limita los cambios de *policy* para evitar cambios drásticos, la PC amplía esta función garantizando que la *policy* se mantenga cerca de sus estados históricos en cada momento. Para ello, se adapta la versión de divergencia fija de *Kullback-Leibler* (KL) de PPO. El modelo PC introduce términos adicionales de divergencia KL entre *policies* adyacentes en la cascada, con coeficientes que aumentan exponencialmente para políticas más profundas (es decir, más antiguas). Esta configuración en principio garantiza que las *policies* de los distintos niveles de la cascada evolucionen a escalas de tiempo diferentes.

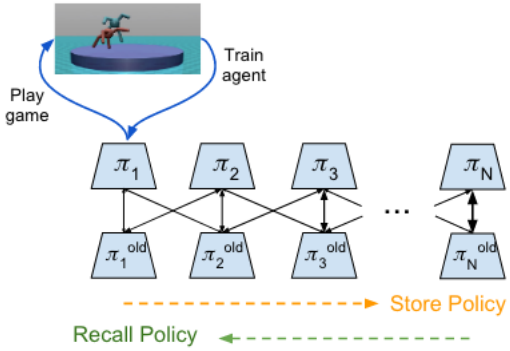


Figura 3.4: Consolidación Sináptica o *Policy Consolidation*

La función objetivo final del modelo PC, combina el gradiente de la *policy* L^{PG}, las regularizaciones propias de PPO L^{PPO} y un término que captura la divergencia KL entre *policies* adyacentes en la cascada L^{CASC}. En principio, esta combinación garantiza que cada *policy* π de la cascada se acerque a sus versiones anteriores, estabilizando el proceso de aprendizaje en diferentes escalas temporales y haciendo que las *policies* recientes sean más flexibles mientras que las más antiguas sean más estable, reflejando el aprendizaje a más largo plazo

$$L^{PC}(\pi) = L^{PG}(\pi_1) + L^{PPO}(\pi) + L^{CASC}(\pi) \tag{3.7}$$

Para la ejecución de los experimentos, se ha implementado este algoritmo partiendo de la versión *clipped* de PPO en lugar de la versión original PPO-KL citada en el estudio. PPO-*clipped* suele preferirse por su sencillez, estabilidad y solidez, especialmente cuando la facilidad de aplicación es una prioridad. Sustituyendo los términos correspondientes a la divergencia KL en las ecuaciones originales del algoritmo PC, se ha llegado a esta formulación que es la que finalmente ha sido traducida a código, adaptando a *Pytorch* el código original en *TensorFlow*, donde π representa las diferentes *policies* en cascada, ω controla el grado de regularización de las *policies*, y A es la función de ventaja habitual en PPO:

$$L^{PG}(\pi_1) \text{ es el gradiente de la } \textit{policy}\ \mathbb{E}_t\left[\frac{\pi_1}{\pi_{old}}\hat{A}_t\right] \tag{3.8}$$

$$L^{PPO}(\pi) = \mathbb{E}_t\left[\sum_{k=1}^{N}\omega^{k-1}\,\text{mín}\left(\frac{\pi_k}{\pi_{old}}\hat{A}_t^{\pi_{old}}, clip(\frac{\pi_k}{\pi_{old}}, 1-\epsilon, 1+\epsilon)\hat{A}_t^{\pi_{old}}\right)\right] \tag{3.9}$$

$$\begin{aligned} L^{CASC}(\pi) = \mathbb{E}_t\Bigg[&\omega_{1,2}\,\text{mín}\left(\frac{\pi_1}{\pi_{2_{\text{old}}}}\hat{A}_t^{\pi_{2_{\text{old}}}}, clip\left(\frac{\pi_1}{\pi_{2_{\text{old}}}}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t^{\pi_{2_{\text{old}}}}\right) \\ &+\sum_{k=2}^{N}\Bigg[\omega\,\text{mín}\left(\frac{\pi_k}{\pi_{\text{k-1}_{\text{old}}}}\hat{A}_t^{\pi_{\text{k-1}_{\text{old}}}}, clip\left(\frac{\pi_k}{\pi_{\text{k-1}_{\text{old}}}}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t^{\pi_{\text{k-1}_{\text{old}}}}\right) \\ &+\text{mín}\left(\frac{\pi_k}{\pi_{\text{k+1}_{\text{old}}}}\hat{A}_t^{\pi_{\text{k+1}_{\text{old}}}}, clip\left(\frac{\pi_k}{\pi_{\text{k+1}_{\text{old}}}}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t^{\pi_{\text{k+1}_{\text{old}}}}\right)\Bigg]\Bigg] \end{aligned} \tag{3.10}$$

Finalmente, el algoritmo se ha entrenado sobre una serie de tareas bajo el entorno Minigrid 3.5.1, caracterizado por acciones discretas. Este nuevo campo de pruebas para el algoritmo presentaba un reto distinto al de las tareas de control continuo exploradas inicialmente en el estudio original. En los espacios de acciones discretas, la dinámica de aprendizaje difiere, ya que la *policy* debe seleccionar entre un conjunto de acciones distintas. Esto requiere centrarse en elegir la acción correcta de un conjunto limitado, a diferencia del ajuste fino de los valores en las acciones continuas.

El Annexo A incluye una descripción más detallada del pseudocódigo correspondiente al algoritmo PC 3

3.3. Enfoques de tipo arquitectónico

3.3.1. Enfoque: Meta-aprendizaje con arquitecturas modulares

El concepto central de este enfoque (Madan et al., 2021), gira en torno al uso de *Recurrent Independent Mechanisms* (RIMs). Los RMIs son esencialmente un conjunto de módulos dentro de una red neuronal, donde cada módulo funciona de forma independiente e interactúa levemente con otros módulos a través de mecanismos de atención. Esta arquitectura modular permite la activación dinámica y selectiva de módulos en función de la relevancia de la entrada, fomentando así la especialización entre los módulos. Este diseño pretende mitigar las limitaciones de las arquitecturas de red monolíticas tradicionales, que a menudo tienen problemas de adaptación y CF cuando se enfrentan a nuevas tareas o entornos cambiantes.

Una característica fundamental de los RIMs es el uso de mecanismos de atención para controlar el flujo de información dentro del modelo. Se emplean dos tipos de atención:

- Atención de entrada (iA): Este mecanismo se utiliza para la activación selectiva de diferentes módulos. Cada módulo genera una consulta en *step* t, y un mecanismo de atención basado en valores clave selecciona los k módulos que prestan más atención a la entrada x_t formando un conjunto activo A_t.

- Atención a la comunicación (cA): Este mecanismo facilita la comunicación dispersa entre módulos. Solo los módulos activados A_t pueden leer información contextual relevante de todos los demás módulos, utilizando un proceso de atención similar basado en clave-valor (*key-value*).

Tanto el mecanismo de atención de entrada como el de comunicación se basan en *multi-head soft-attention*, calculada como $Attention(Q,K,V) = softmax(QK^T/\sqrt{d_K})V$, donde Q, K y V representan la consulta o *query*, la clave o *key* y el valor o *value*, respectivamente, y d_K es el tamaño del tensor de claves. Los valores de k (número de módulos que se activan en cada *step*) y N (número total de módulos) son hiperparámetros. Permanecen constantes durante el aprendizaje de una tarea concreta y en diferentes entornos.

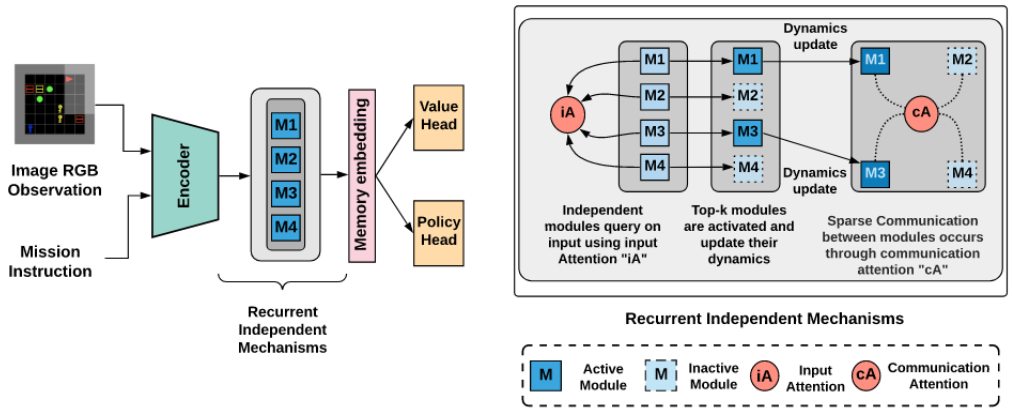


Figura 3.5: *Recurrent Independent Mechanisms* (Madan et al., 2021)

Un aspecto clave de este enfoque es el uso del meta-aprendizaje junto con redes de atención para controlar el flujo de información dentro del modelo. En la configuración Meta-RIMs, los componentes de la red se entrenan en diferentes escalas temporales. Algunos parámetros se adaptan rápidamente a las condiciones cambiantes, mientras que otros cambian más lentamente, lo que permite a la red equilibrar la adaptación rápida con la estabilidad. El algoritmo introduce un mecanismo de aprendizaje dual con bucles de actualización rápidos y lentos:

- *Bucle interno rápido*: Este bucle actualiza rápidamente los parámetros de los módulos pertinentes, lo que permite una adaptación rápida a los cambios en la distribución de tareas.

- *Bucle externo lento*: Aquí, los parámetros de los mecanismos de atención se actualizan con menos frecuencia, capturando aspectos más estables de la distribución de tareas y manteniendo un intercambio de información eficiente entre módulos.

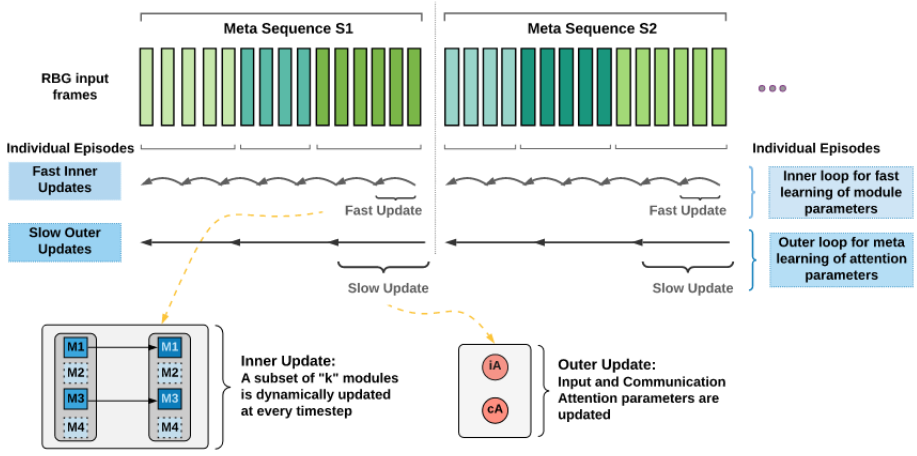


Figura 3.6: Meta-learning RIMs (Madan et al., 2021)

El diseño de la red modular, junto con el enfoque de aprendizaje de doble escala temporal, en principio aumentaría la eficacia del aprendizaje a partir de un número limitado de ejemplos (*few-shot* o *one-shot*), mejorando la generalización de las *policies*. Al permitir una adaptación rápida a nuevas tareas mediante actualizaciones rápidas de los parámetros de los módulos y un aprendizaje estable de los mecanismos de atención, la red puede generalizarse mejor a través de distintas distribuciones de tareas. Esto mejoraría el rendimiento en entornos en los que las condiciones de las tareas cambian dinámicamente.

Para este trabajo en particular, el algoritmo se ha implementado desde cero en PyTorch y entrenado en diversas tareas sobre entornos MiniGrid (ver sección **??**). El Anexo A A incluye una descripción en pseudocódigo del algoritmo meta-RIMs 5

3.3.2. Enfoque: Aprendizaje RL combinado con memoria inspirada en modelos Transformer

Los agentes de RL tradicionales tienen dificultades para recordar en detalle durante periodos prolongados, especialmente tras retrasos o tareas de distracción. Las arquitecturas de memoria existentes, como las LSTM o los transformadores, son ineficaces a la hora de recordar con detalle o razonar sobre la memoria durante periodos prolongados.

HCAM o *Hierarchical Chunk Attention Memory* (Lampinen et al., 2021) consiste en una arquitectura de memoria inspirada en *transformers* que divide el pasado en fragmentos distintos, almacenando cada fragmento en detalle junto con una única clave de resumen para ese fragmento, y los recupera jerárquicamente. Este proceso en dos pasos consiste en atender inicialmente a resúmenes generales de los trozos y, a continuación, atender en detalle a los trozos más relevantes (Figura 3.7). HCAM se inspira en la capacidad humana de "viajar mentalmente en el tiempo", es decir, la capacidad de recordar acontecimientos pasados con todo detalle sin distraerse por los acontecimientos intermedios. Esta memoria episódica, similar a la humana, permitiría un recuerdo detallado, a la vez disperso y minucioso, centrado en fragmentos relevantes de experiencias pasadas mejorando la capacidad de aprendizaje y retención de conocimiento en un agente RL.

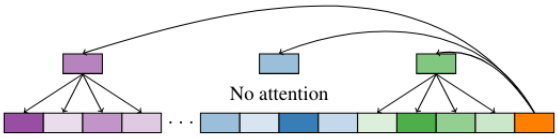


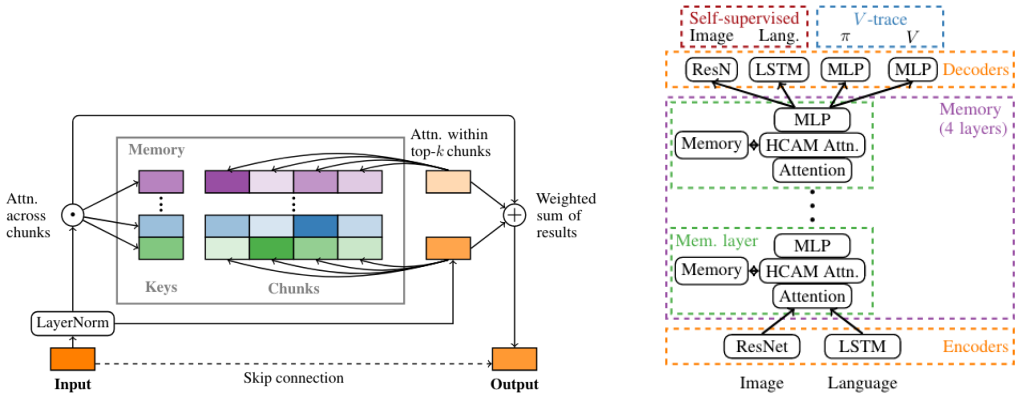
Figura 3.7: Atención jerárquica (Lampinen et al., 2021)

Este mecanismo de atención jerárquica presenta las siguientes características:

- Una estructura de memoria basada en trozos o *chunks*: El HCAM divide las experiencias pasadas del agente en trozos distintos, almacenando cada trozo en detalle junto con una única clave de resumen para ese trozo.
- Mecanismo de atención a dos niveles: El modelo atiende primero a los resúmenes de

estos trozos para identificar cuáles son los más relevantes en el contexto actual. Para calcular la relevancia se aplica una función *softmax* al producto de una proyección de la capa de consulta de las claves de entrada y de resumen. A continuación, presta atención más en detalle a los fragmentos más relevantes identificados en el paso anterior. Esta atención más detallada dentro de los trozos elegidos se realiza utilizando atención *multi-head*, asegurando que el agente se centra solo en la información pasada más pertinente.

A diferencia de los *transformers* estándar, que atienden a todos los elementos de una secuencia, el mecanismo de atención de HCAM es más disperso y focalizado, y solo se centra en fragmentos y eventos específicos dentro de esos fragmentos. Este enfoque reduciría a priori la complejidad computacional y mejoraría el recuerdo de eventos pasados relevantes.



(a) Bloque HCAM (b) Integración de bloques HCAM en un agente RL

Figura 3.8: Arquitectura HCAM (Lampinen et al., 2021)

HCAM modifica la arquitectura *transformer*, en particular el *Gated TransformerXL* (Parisotto et al., 2019), a fin de acomodar la estructura de memoria jerárquica basada en trozos. Esto implica ajustar los mecanismos de atención para que funcionen en dos etapas: una atención más amplia a los trozos almacenados y una atención más fina dentro de aquellos trozos seleccionados (Figura 3.8a). La arquitectura modificada sería más eficiente en el manejo de datos dispersos, un reto habitual en entornos de RL en los que solo unos pocos sucesos pasados pueden ser relevantes para la decisión actual.

La Figura 3.8b describe a alto nivel la integración de bloques HCAM en agentes de RL:

- Los agentes están equipados con codificadores de entrada para procesar entradas visuales y/o lingüísticas, una capa de memoria basada en HCAM y capas de salida para la estimación de políticas y valores.
- Los bloques de memoria HCAM se inserta en la capa de memoria del agente, concretamente entre los bloques de atención local y *feed-forward*, sustituyendo a la memoria estándar en el caso de agentes que integran el *Gated TransformerXL*.

Para la experimentación en este trabajo con el enfoque HCAM, a diferencia del artículo original en el que HCAM se integraba con un agente *IMPALA/V-trace* (un algoritmo Actor-Critic

off-policy), se ha implementado la arquitectura en Pytorch adaptando el código original de los autores e integrándola sobre un agente PPO.

El Anexo A incluye el pseudocódigo del algoritmo HCAM 7 como referencia.

3.4. Enfoques híbridos

3.4.1. Enfoque híbrido: BLIP combinado con máscara de poda suave (*Soft Parameter Pruning*)

En este trabajo, se presenta un enfoque experimental híbrido innovador, que combina el enfoque existente *Bit-level Information Preserving* (BLIP) (Shi et al., 2022) (ver 3.2.2) con algunos conceptos inspirados en *Soft Parameter Pruning* (SPP) y *Adversarial Neural Pruning and Synaptic Consolidation* (ANPyC) (Peng et al., 2018; Peng et al., 2019). La base de esta combinación es la implementación de máscaras binarias, una técnica central en SPP y ANPyC, para mejorar la precisión del método BLIP en la preservación de información. Esta integración permite podar de manera selectiva aquellos parámetros que son cruciales durante el aprendizaje de tareas, minimizando así su alteración. El objetivo de esta amalgama es explorar si la aplicación de máscaras binarias puede potenciar la efectividad del enfoque BLIP, el cual actúa a un nivel granular de bit, para lograr un equilibrio entre la retención de información esencial y la eficiencia en el procesamiento. Mientras que el uso de la máscara gestiona la optimización estructural a nivel macro (poda selectiva y refuerzo de parámetros), BLIP puede garantizar que la información detallada a nivel de bits se siga conservando de forma eficiente. Esta sinergia podría dar lugar a un modelo más robusto, capaz de mantener la relevancia de la tarea a alto nivel, preservando al mismo tiempo la información esencial a un nivel inferior.

La poda neuronal mediante máscaras en ANPyC es una estrategia clave para gestionar el espacio de parámetros de la red neuronal. Consiste en podar de forma iterativa los parámetros irrelevantes para la tarea con el fin de centrarse en los cruciales para la misma. Este proceso se guía por una medida de saliencia o importancia (Ω) de los parámetros, que se determina utilizando la matriz de información diagonal de Fisher de forma similar a *Elastic Weight Consolidation* (EWC) (Kirkpatrick et al., 2016) (ver 3.2.1):

$$\Omega = \text{máx}\left(\left(\left(\frac{\partial \mathcal{H}(q)}{\partial W}\right)^T \delta W + \frac{1}{2}\delta W^T H \delta W\right), 0\right) \tag{3.11}$$

En la anterior expresión para el cálculo de la importancia (Ω), $\mathcal{H}(q)$ es la entropía, W son los parámetros que se entrenan, y H es la hessiana. Como se ha indicado anteriormente, en la implementación se hace uso de la matriz de Fisher como aproximación de la hessiana, a fin de reducir el costo computacional.

La importancia Ω de los parámetros se va sumando a las importancias de las tareas anteriores para obtener los valores acumulativos tras el entrenamiento de cada tarea:

$$\Omega_{i,j}^{1:t} = \Omega_{i,j}^{1:t-1} + \Omega_{i,j}^{t} \tag{3.12}$$

El valor de Ω cumulativo se usaría por un lado para añadir un término extra de regularización en la función de pérdida, donde w son los pesos del modelo para la tarea actual, w' los pesos del modelo tras aprender la anterior tarea, y λ un hiperparámetro:

$$L = L_{\text{new}} + \lambda \sum_{i,j} \Omega^{t}_{i,j} (w_{i,j} - w'_{i,j})^2 \tag{3.13}$$

Por otro lado, la máscara binaria se va igualmente calculando tras completar el entrenamiento de cada tarea, en la que estos parámetros salientes se ponen a ceros para evitar que se actualicen en la tarea actual según su importancia Ω y un umbral predeterminado β (un hiperparámetro) a partir del cual se podarían. La poda es suave (*soft*)), en el sentido en que el parámetro podado sigue estando presente, pero se previene su actualización usando un valor de máscara a cero a fin de preservar su peso. Esta máscara serviría para determinar los parámetros disponibles para la tarea en curso. Para evitar la poda de parámetros de tareas anteriores, la nueva máscara de poda para la nueva tarea es la intersección con las máscaras de tareas anteriores:

$$M^{t+1} = M^{t} \cap M^{t+1} \tag{3.14}$$

Este mecanismo se ha integrado en el algoritmo BLIP al completar el entrenamiento de cada tarea. Un aspecto crucial es intentar mantener el equilibrio entre la estabilidad y la plasticidad, fundamentales en el proceso de aprendizaje de un modelo. Para lograr esto, se ha optado por ajustar y relajar algunos de los parámetros reguladores del enfoque BLIP. Esta modificación tendría como intención el asegurar que la incorporación de la máscara binaria y el término regulador no comprometan excesivamente la plasticidad del modelo. Para ello se ha optado por una solución simple, que consistiría en relajar el número de bits cuantizados que se congelan en los parámetros con mayor ganancia de información (IG). Originalmente esté número se calculaba mediante un redondeo hacia arriba (*ceil*) de la magnitud de la IG calculada por BLIP. En la modificación se ha optado por redondear hacia abajo (*floor*), lo que práctica implicaría que no se congelarían tantos bits como en el algoritmo BLIP original. Esta modificación tendría como intención dotar al modelo de cierto margen de plasticidad, teniendo en cuenta que la inclusión de la máscara binaria de poda puede asimismo afectar a dicha plasticidad. Este ajuste en los parámetros busca encontrar un punto medio óptimo entre la preservación de la información esencial y la capacidad adaptativa del modelo.

Mientras que el uso de máscaras inspirado en ANPyC gestiona la optimización estructural a nivel macro (poda selectiva), BLIP puede garantizar que la información granular a nivel de bits se conserve de forma eficiente. Esta sinergia podría dar lugar a un modelo más robusto, capaz de mantener la relevancia de la tarea a alto nivel, preservando al mismo tiempo la información esencial a un nivel inferior.

El pseudocódigo del algoritmo ANPyC original 6, del cual se tomado el uso de la máscara de poda, se detalla como referencia en el Annexo A.

3.4.2. Enfoque: BLIP combinado con EWC

En la búsqueda de soluciones eficaces para el olvido catastrófico en redes neuronales, se propone un enfoque híbrido que integraría las técnicas anteriormente descritas *Bit-level Information Preserving* (BLIP) y *Elastic Weight Consolidation* (EWC) (subsecciones 3.2.2 y 3.2.1). Esta combinación tiene como objetivo aprovechar las fortalezas de ambos algoritmos para lograr un balance entre la plasticidad y la estabilidad del aprendizaje en entornos de CL.

BLIP funciona a un nivel granular, congelando bits específicos de los parámetros cuantizados en función de la ganancia de información al aprender nuevas tareas. Esta técnica permite preservar conocimientos previos a nivel de bit, lo que resulta en una retención más eficaz de la información relevante. No obstante, en esta propuesta, ciertas condiciones de BLIP se relajarán para facilitar una mejor integración con EWC. En particular se permitirá un mayor grado de flexibilidad en el mecanismo de congelación de bits que implementa BLIP, de forma idéntica a la descrita en la subsección anterior 3.4.1.

EWC, por su parte, introduce una regularización sobre los parámetros críticos para tareas específicas, limitando su variabilidad durante el aprendizaje de nuevas tareas. Esta estrategia ayuda a mantener el rendimiento en tareas anteriores sin comprometer el aprendizaje de nuevas. En este enfoque híbrido, se ajustará el coeficiente de regulación de EWC para permitir cierta flexibilidad, buscando un equilibrio adecuado entre la retención de conocimientos de tareas previas y la adquisición de nuevos.

La integración de BLIP y EWC se centra en encontrar un punto medio que maximice los beneficios de ambos métodos. Con la relajación de ciertas restricciones en BLIP y en el coeficiente de regulación de EWC se busca una sinergia que permita una mayor adaptabilidad. Esta sinergia se ve favorecida también por el uso compartido de la matriz de información de Fisher para evaluar la importancia de los parámetros en ambos métodos tras el aprendizaje de cada tarea, lo que elimina la necesidad de cálculos redundantes y optimiza el proceso.

3.5. Implementación

En el desarrollo del presente trabajo, la implementación de los experimentos se ha llevado a cabo utilizando la librería de código abierto PyTorch, ampliamente reconocida por su flexibilidad y eficiencia en la definición de modelos de aprendizaje profundo. Asimismo, para la implementación específica de los agentes RL tanto en la línea de base como en los diversos enfoques usados, se ha partido de las implementaciones Actor-Critic desarrolladas en PyTorch por (Kostrikov, 2018).

En las fases iniciales del proyecto, se realizaron pruebas de depuración y validación preliminar de los algoritmos y arquitecturas propuestas empleando un entorno de desarrollo local compuesto por un MacBook Pro equipado con un procesador Intel i7 de 2.8 GHz. Esta etapa preliminar fue crucial para garantizar el correcto funcionamiento de los códigos y para el ajuste fino de los parámetros iniciales de los modelos.

Una vez superada la etapa de depuración, se procedió a la ejecución de baterías de expe-

rimentos a mayor escala. Para ello, se utilizó la herramienta Ansible, que facilitó la automatización y el despliegue eficiente de los procesos en la infraestructura de computación en la nube de Google Cloud Platform. En concreto, se seleccionó la instancia c2-standard-16, dotada de 16 vCPUs (CPUs virtuales), que proporcionó la capacidad de cómputo necesaria para llevar a cabo entrenamientos intensivos de los modelos de aprendizaje automático sin las limitaciones de hardware de un entorno local.

Para los experimentos específicos que involucraron arquitecturas basadas en mecanismos de atención o con componentes inspirados en modelos Transformer (secciones 3.3.1 y 3.3.2), y que por tanto se benefician significativamente de la aceleración por hardware específico, se optó por la instancia n1-standard-8 de Google Cloud, que contó con el refuerzo de una GPU Nvidia Tesla T4. Esta configuración se eligió para aprovechar las ventajas del procesamiento paralelo y la eficiencia energética que estas GPUs ofrecen, permitiendo así una reducción considerable en los tiempos de entrenamiento y un aumento en la capacidad de procesamiento de datos en paralelo.

En resumen, la infraestructura y las herramientas seleccionadas para el desarrollo y ejecución de los experimentos han permitido abordar los retos computacionales del proyecto de manera eficiente y escalable, asegurando resultados fiables y reproducibles.

3.5.1. Entorno, escenario y tareas

El entrenamiento y evaluación de los diversos agentes RL ha tenido lugar sobre entornos basados en Minigrid. Esta librería ofrece una suite de entornos de aprendizaje por refuerzo 2D sencillos y configurables, y fue diseñada para facilitar la investigación en una amplia gama de tareas orientadas a objetivos. Los entornos siguen la API estándar de Gymnasium enfocada a investigación en DRL (*fork* vivo de la librería Gym de OpenAI, que ha dejado de estar mantenida) y están diseñados para ser ligeros, rápidos y fácilmente personalizables. Esta biblioteca fue desarrollada específicamente con un paradigma de diseño minimalista para permitir a los usuarios desarrollar rápidamente nuevos entornos para necesidades específicas de investigación. La adopción a gran escala por parte de la comunidad RL ha facilitado la investigación en una amplia gama de áreas.

Los entornos de la librería son Procesos de Decisión Markov parcialmente observables (POMDP). Estos entornos pueden ser descritos matemáticamente por la tupla $(\mathcal{X}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{R}, \Omega, \gamma)$ donde $\mathcal{X}$ representa el espacio de estados, $\mathcal{A}$ el espacio de acciones, $\mathcal{O}$ el espacio de observaciones, $\mathcal{T} : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$ la función de transición, $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ la función de recompensa, $\Omega : \mathcal{X} \to \mathcal{O}$ la función de observación, y $\gamma \in [0, 1]$ el factor de descuento (Chevalier-Boisvert et al., 2023).

Los agentes tienen un espacio de acción discreto ($\mathcal{A}$) de siete opciones representando `["turn left", "turn right", "move forward", "pickup", "drop", "toggle", "done"]`. La función de recompensa por defecto ($\mathcal{R}$) para el entorno es dispersa, de modo que la recompensa solo es distinta de cero cuando se cumple la misión. Para el agente, el espacio de observación ($\mathcal{O}$) es, por defecto, una imagen RGB de tamaño 80 × 60 desde la perspectiva

del mundo del agente. Si bien es posible usar directamente las observaciones como imágenes RGB, MiniGrid ofrece la opción de codificar la visión parcial del agente como un tensor de tamaño 7x7x3. Lo que significa que cada baldosa está representada por tres valores enteros, que representan categorías: tipo de objeto, color y atributo abierto/cerrado. El tipo de objeto puede ser suelo, pared, puerta, etc. La meta del episodio y las paredes exteriores se consideran también se consideran objetos.

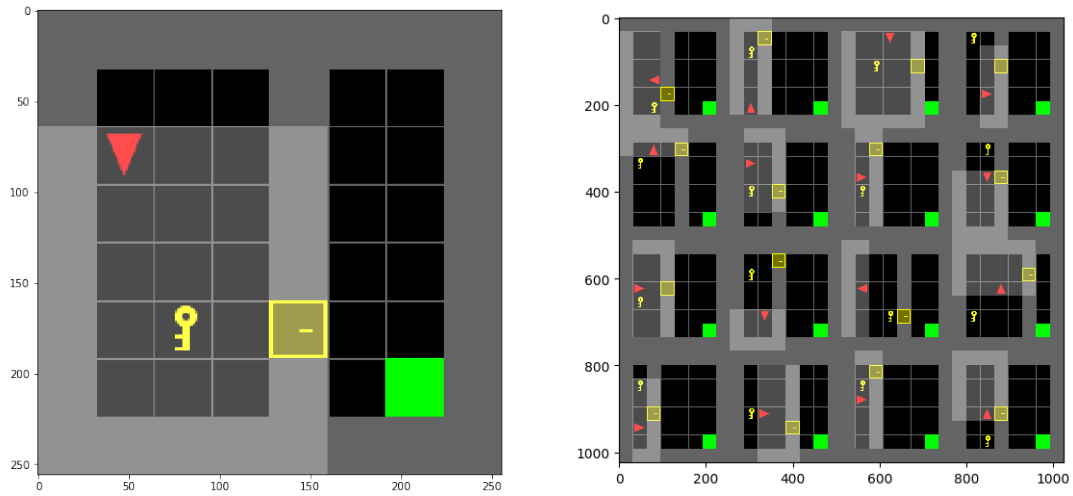


Figura 3.9: Ejemplo de entorno Minigrid 8x8g y ejemplo de entorno multiproceso

La Figura 3.9 (izq.) muestra un entorno discreto (`MiniGrid-DoorKey-8x8-v0`) basado en una cuadrícula con un agente (triángulo rojo) que tiene una campo de visión parcial (aunque puede ser configurado para visón completa de todo el entorno), que se muestra como una transparencia gris claro en la imagen.

Para los entrenamientos de los diferentes modelos, se ha hecho uso de la capacidad multiproceso proporcionada por la librería RL de (Kostrikov, 2018), que permite definir el número de procesos para así realizar un entrenamiento multi-agente en paralelo. En tal caso, es posible configurar MiniGrid para que en cada proceso, el agente entrene en una configuración aleatoria válida de la tarea a aprender, facilitando con ello la exploración y la generalización de la política aprendida. La Figura 3.9 (dcha.) muestra un ejemplo de dicha configuración estocástica para el entorno `MiniGrid-DoorKey-8x8-v0` bajo un total de 16 procesos.

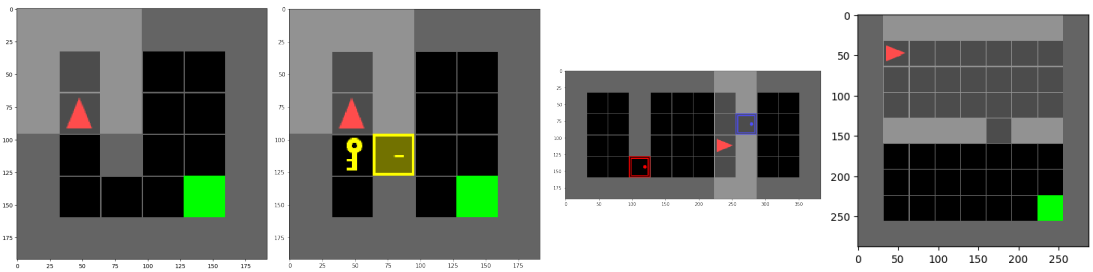


Figura 3.10: Secuencia de 4 tareas

Para la realización de los experimentos en el presente trabajo, se ha seleccionado una secuencia estandarizada de cuatro tareas en función de su diversidad y combinando diferentes grados de complejidad. Se ha establecido un orden arbitrario para estas tareas con el objetivo de examinar la capacidad de generalización de los algoritmos empleados. Cada tarea se ha entrenado a lo largo de 500,000 pasos, buscando un equilibrio entre la profundidad del aprendizaje y la factibilidad en términos de tiempo y recursos. Por razones de eficiencia, se ha optado por entornos de 6x6, lo que permite simplificar la carga computacional manteniendo al mismo tiempo la variedad y el reto de las tareas propuestas. Las tareas escogidas se detallan a continuación, y la Figura 3.10 ilustra visualmente la secuencia usada.

• `MiniGrid-WallGapS6-v0`: Este entorno no existe en la distribución original de MiniGrid y ha sido creado exprofeso adaptando el código fuente de MiniGrid. Consiste simplemente en un muro con un estrecho hueco que el agente debe de atravesar para llegar a la meta.

• `MiniGrid-DoorKey-6x6-v0`: Este entorno tiene una llave que el agente debe recoger para abrir una puerta y llegar a la casilla verde de la meta.

• `MiniGrid-RedBlueDoors-6x6-v0`: El agente se sitúa aleatoriamente dentro de una habitación con una puerta roja y otra azul orientadas en direcciones opuestas. El agente tiene que abrir la puerta roja y después la azul, en ese orden.

• `MiniGrid-SimpleCrossingS9N1-v0`: Entorno parecido a `MiniGrid-WallGapS6-v0`, pero con una complejidad y variabilidad ligeramente superiores debido al incremento en el tamaño de la rejilla (9x9). El objetivo del agente es llegar a la casilla de meta, de color verde, localizada en la esquina contraria de la habitación.

Como se ha indicado, la recompensa solo se calcula en caso de que el agente complete la tarea con éxito. Cada tarea conlleva predefinido un número máximo de pasos `step_count` que se calcula a partir de un coeficiente específico a cada tarea en virtud de su dificultad, y el tamaño de la rejilla usada (5x5, 6x6, 8x8, etc.). En caso de completar la tarea dentro de ese número máximo de pasos asignado, la recompensa $\mathcal{R}$ se obtiene teniendo en cuenta el número de pasos que tardó el agente en resolver la tarea `step_count`:

$$\mathcal{R} = \begin{cases} 1 - 0{,}9 \cdot \frac{\texttt{step\_count}}{\texttt{max\_steps}}, & \text{en caso de éxito} \\ 0, & \text{en caso de fallo} \end{cases} \tag{3.15}$$

3.5.2. Procedimiento experimental

Como línea de base principal para la experimentación, se ha adoptado el algoritmo PPO 'vanilla', de tipo Actor-Critic. Tal y como se ha descrito en la sección 2.1, PPO es comúnmente utilizado como línea de base en la experimentación de PPO debido a su balance entre eficiencia y efectividad, y la relativa facilidad tanto en su implementación como en su modificación.

Para la experimentación con los enfoques híbridos especialmente desarrollados en este trabajo, se ha considerado como líneas de base adicionales aquellas que incorporan métodos ya existentes y anteriormente detallados en la sección 2.3 como *Elastic Weight Consolidation*

(EWC), que se basa en la regularización, y *Bit-level Information Preserving* (BLIP) (subsecciones 3.2.1 y 3.2.2), que emplea la congelación selectiva de bits de parámetros. Asimismo, estas técnicas subyacen en parte de los enfoques híbridos evaluados tal y como fue explicado.

Por otro lado, se ha investigado específicamente el enfoque de Consolidación Sináptica o *Policy Consolidation*, que aplica destilación y consolidación de conocimiento, así como las arquitecturas *Hierarchical Chunk Attention Memory* (HCAM) y *Recurrent Independent Mechanisms* (RIMs) (subsecciones 3.2.3, 3.3.2 y 3.3.1), que se fundamentan en estructuras modulares y el uso de meta-aprendizaje. Estos enfoques no se habían adaptado previamente al entorno de RL de MiniGrid, y en el caso de arquitecturas modulares como HCAM y RIMs orientadas al CL, no se había evaluado específicamente su desempeño a la hora de mitigar el CF.

Para cada enfoque, se han efectuado tres entrenamientos utilizando tres semillas distintas para garantizar la robustez de los resultados. En la fase de evaluación, se han ejecutado 30 episodios deterministas por modelo entrenado, a fin de eliminar la aleatoriedad en la selección de acciones y en la respuesta del entorno. Posteriormente, se ha calculado la mediana y la desviación estándar de la recompensa sobre las tres semillas para cada tarea de la secuencia de entrenamiento.

Las métricas de evaluación, que se detallarán en la siguiente subsección, han sido estandarizadas a lo largo de todos los experimentos para facilitar la comparación entre los distintos enfoques y las secuencias de tareas utilizadas.

3.5.3. Métricas de evaluación

A fin de evaluar el rendimiento de los métodos y enfoques implementados, y su susceptibilidad al CF tras ser entrenados en la secuencia estandarizada de cuatro tareas, se emplearán una serie de métricas específicas de uso común en CF y aprendizaje secuencial (Lopez-Paz y Ranzato, 2017).

Para ello, una vez que el modelo termina de aprender una tarea t_i dada, se evalúa su rendimiento sobre todas las tareas T de la secuencia. Al hacerlo, se va construyendo la matriz $R \in \mathbb{R}^{T \times T}$, donde $R_{i,j}$ es la mediana de la recompensa $\mathcal{R}$ obtenida por el modelo sobre la tarea t_j justo después de aprender la tarea t_i. Tras construir esta matriz, las métricas de interés se definen de la siguiente manera:

1. **Accuracy Cumulative (ACC):** Esta métrica mide el rendimiento promedio de los modelos en todas las tareas aprendidas hasta el momento. Se calcula sumando las precisiones del modelo en cada una de las tareas y dividiéndolas por el número de tareas T, proporcionando una visión global del aprendizaje acumulativo.

$$\text{Average Accuracy: ACC} = \frac{1}{T}\sum_{i=1}^{T} R_{T,i} \tag{3.16}$$

2. **Backward Transfer (BWT):** La BWT evalúa el impacto que tiene el aprendizaje de una tarea nueva en el rendimiento de las tareas previas. Se define como la diferencia en el rendimiento en una tarea previa k después de haber aprendido una tarea subsiguiente t, donde

$k \prec t$. Si hay una mejora en el rendimiento de las tareas previas después de aprender nuevas tareas, esto se considera como transferencia hacia atrás positiva, lo que implica que el aprendizaje de nuevas tareas contribuye positivamente al conocimiento previo. Por el contrario, una transferencia hacia atrás negativa indica que el aprendizaje de nuevas tareas degrada el rendimiento en tareas anteriores, lo cual es una manifestación del CF.

$$\text{Backward Transfer: BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1} R_{T,i} - R_{i,i} \tag{3.17}$$

3. En cuanto a la **Forward Transfer (FWT)**, esta mediría cómo el aprendizaje de una tarea t puede influir en el rendimiento en tareas futuras k, donde $k \succ t$. Una transferencia hacia delante positiva se da cuando el conocimiento adquirido en la tarea t ayuda a mejorar el rendimiento en las tareas futuras, incluso antes de que el modelo sea entrenado explícitamente en esas tareas futuras (aprendizaje *zero-shot*). Sin embargo, la FWT no se utilizará en este estudio, ya que el aprendizaje *zero-shot* está fuera del alcance de este trabajo. En este caso, $\overline{b}_i$ es el rendimiento o recompensa media obtenida para cada tarea tras una inicialización aleatoria.

$$\text{Forward Transfer: FWT} = \frac{1}{T-1}\sum_{i=2}^{T} R_{i-1,i} - \overline{b}_i. \tag{3.18}$$

Como se ha indicado, para el ámbito de estudio de este trabajo, se hará uso únicamente de las métricas ACC y FWT, teniendo en cuenta que un algoritmo ideal de CL debería lograr la máxima ACC con la menor BWT negativa (o incluso positiva). Estas métricas serán aplicadas consistentemente en la evaluación de todos los modelos y enfoques para asegurar una comparación equitativa y para entender mejor cómo diferentes estrategias pueden mitigar o exacerbar el CF en entornos de CL o aprendizaje secuencial.

Resultados y Discusión

4

En este capítulo se presentan y discuten los resultados experimentales obtenidos de una serie de enfoques y arquitecturas. Estos han sido diseñados con el doble objetivo de mitigar el CF y facilitar el CL. Se ha realizado un análisis exhaustivo tanto de enfoques existentes, adaptados a las especificidades del entorno RL de MiniGrid, como de desarrollos híbridos concebidos específicamente para este estudio. Una contribución notable de esta investigación es la evaluación del CF en el enfoque de tipo modular, un aspecto que no había sido abordado explícitamente en investigaciones previas. Para garantizar la consistencia y la comparabilidad de los resultados, todos los modelos se han entrenado utilizando la misma secuencia de cuatro tareas MiniGrid, manteniendo idéntica la duración en términos de *steps*. Los resultados se han analizado utilizando las métricas comunes definidas previamente en la subsección 3.5.3, lo que ha permitido una evaluación y comparación rigurosa de los resultados obtenidos con los distintos enfoques estudiados.

4.1. Enfoques BLIP y EWC

Los siguientes enfoques se presentan evaluados de manera conjunta al corresponder con estrategias ya existentes específicamente dirigidas a la mitigación del CF, bien a través de la regularización como pueda ser EWC, o a través de alteraciones más o menos directas sobre la estructura de los propios pesos sinápticos y sus gradientes, ya sea de manera granular a nivel de bits como BLIP o mediante el uso de máscaras como en la contribución inspirada en ANPyC. El enfoque EWC es usado principalmente en el ámbito de aprendizaje supervisado, por lo que resulta interesante comprobar su rendimiento en el ámbito del RL, y en particular en entornos como Minigrid donde el agente necesita explorar. Por otro lado, BLIP es un enfoque mucho más reciente y pese a que ha sido probado con relativo éxito en entornos RL de tipo Atari, más ricos en recompensas, en este trabajo se pretende evaluar su desempeño en entornos más *sparse reward*. Por otro lado todos ellos han sido implementados sobre un código base común (Kostrikov, 2018), lo que facilita tanto su entrenamiento como evaluación conjunta.

En relación con los modelos experimentados, se han configurado siguiendo los hiperparámetros indicados en la tabla B.1 del Anexo B. Cabe destacar que no todos los experimentos realizados se incluyen en los resultados. Se llevó a cabo un barrido simple sobre los diversos hiperparámetros a fin de seleccionar aquellas configuraciones que dieron lugar a entrenamientos más estables de manera general y que son las que se presentan en este trabajo.

Como línea de base en esta batería de experimentos, se utilizó la versión estándar de PPO-clipped. Como se ha indicado, los enfoques EWC y BLIP se adaptaron específicamente para funcionar con secuencias de tareas Minigrid, sin alterar significativamente la estructura del algoritmo subyacente.

La tabla 4.1 muestra las métricas de *ACC* y *BWT* obtenidas para cada enfoque, incluyendo la línea de base PPO. Tal y como se ha descrito en la sub-sección 3.5.3, la métrica *ACC* refleja la precisión media sobre el conjunto de tareas, mientras que la *BWT* proporciona una indicación del CF, evaluando el rendimiento del modelo en sobre cada tarea anterior, incluida la última tarea, después de un entrenamiento continuo y secuencial sobre el total de tareas.

Enfoque	**ACC**	**BWT**
PPO-clip	$0{,}29201 \pm 0{,}098$	$-0{,}56598 \pm 0{,}13079$
EWC	$0{,}61541 \pm 0{,}11525$	$0{,}17525 \pm 0{,}16087$
BLIP	$0{,}46271 \pm 0{,}16051$	$0{,}0 \pm 0{,}21832$

Tabla 4.1: Resultados de métricas *ACC* y *BWT*

.

Resultados destacados:

- La línea de base **PPO-clipped**, sin mecanismos de mitigación del CF, mostró los peores resultados tanto en *ACC* como en *BWT*, como era de esperar. El valor de la *BWT* es negativo, lo cual es indicador claro de CF.
- **BLIP** y **EWC** mostraron resultados similares entre sí, aunque los valores *ACC* (precisión media sobre toda la serie de tares) es definitivamente más alto para EWC. Con relación al *BWT*, ambos enfoques muestran resultados más cercanos, con ligera ventaja de EWC sobre BLIP (el valor de *BWT* es ligeramente positivo).

Con respecto la evaluación de los enfoques arriba descritos, la Figura 4.1 muestra de manera más detallada el rendimiento de cada modelo para cada una de las tareas tras haber sido entrenados en toda la secuencia de cuatro tareas (subsección 3.5.1). Como se ha indicado, cada tarea se entrena un número fijo de $500,000$ *steps*, resultando en un total de $2,000,000$ de *steps* para toda la serie.

- En la primera tarea (`tarea 0`), pese a ser sencilla, se observa claramente como a medida que el aprendizaje avanza y se entrena en nuevas tareas, su rendimiento en la línea de base PPO se degrada de manera muy rápida (prácticamente tras completar el aprendizaje de la segunda tarea (`tarea 1`). En cambio parece conservarse prácticamente intacta con el resto de enfoques.
- Con relación a la segunda tarea (`tarea 1`), se observa una situación similar en lo que respecta a la línea de base PPO, una rápida degradación en su rendimiento. En el resto de enfoques, pese a cierta inestabilidad, dicho conocimiento parece conservarse hasta el final tras todo el entrenamiento en la serie

- La tercera tarea (`tarea 2`) conlleva un mayor grado de exploración del entorno y resultó ser la más compleja, de hecho no se consiguió aprender con ninguno de los enfoques, lo que posiblemente sea debido a la restricción impuesta por diseño de un número máximo de $500,000$ *steps*.
- Finalmente, la cuarta tarea (`tarea 3`), similar a la `tarea 0`, pero con una rejilla de dimensiones superiores es aprendida por todos los modelos en mayor o menor grado. En este caso, al tratarse de la última tarea, el grado de CF por definición será el menor al ser la tarea más reciente. De hecho la línea de base PPO no presenta demasiados problemas en para recordarla. El resto de enfoques muestran cierta inestabilidad, posiblemente a que sus arquitecturas se vean ya afectadas por restricciones en la plasticidad y presenten dificultades en el aprendizaje dentro de los límites impuestos en el número de *steps* por tarea.

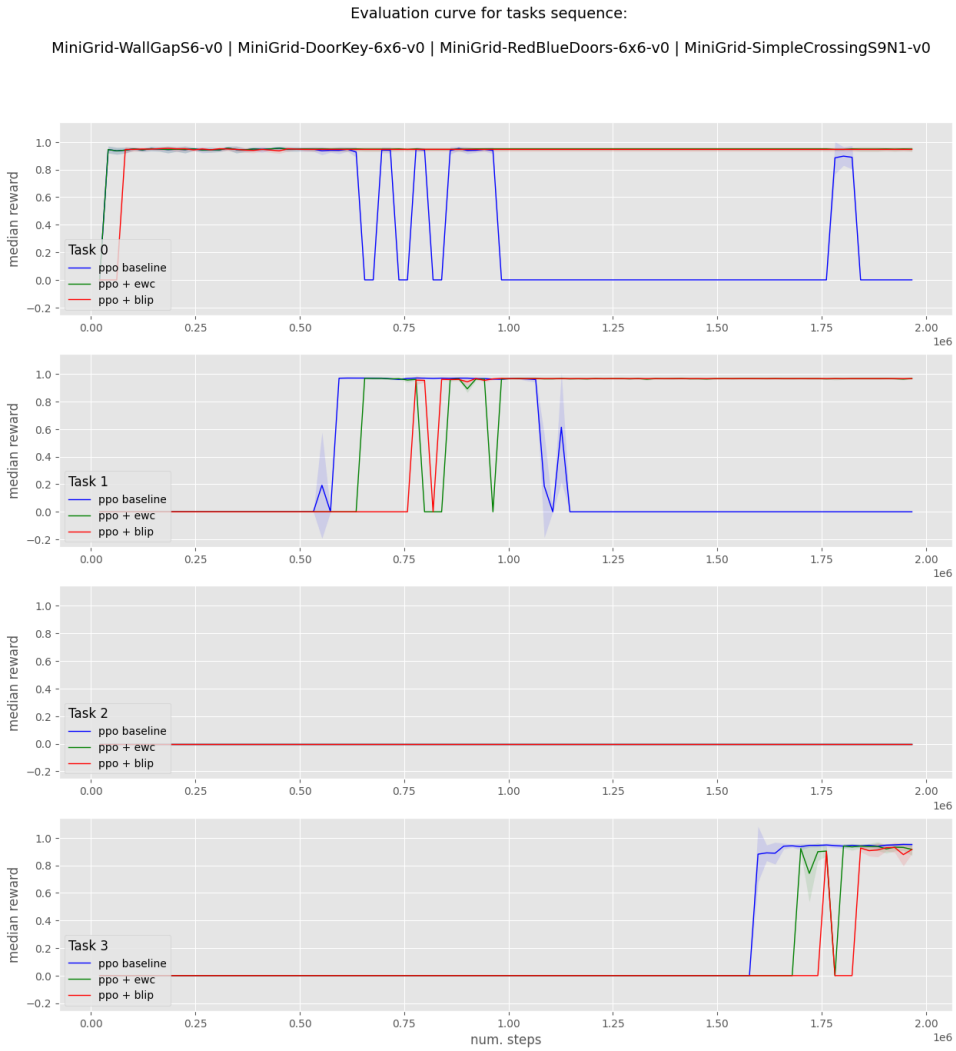


Figura 4.1: Curvas de evaluación por tareas para enfoques BLIP y EWC

Por último, la Figura 4.2 refleja las diversas curvas de aprendizaje para cada uno de los

enfoques, en las que se aprecia claramente la dificultad para aprender la tercera tarea. Cabe destacar que el aprendizaje más estable se percibe en la línea de base PPO, la cual no lleva integrada ningún mecanismo de mitigación del CF, por lo que a priori para cada tarea, todos los pesos de la red estarían disponibles para el aprendizaje (con la desventaja de que también va a ser el modelo más susceptible al CF por la propia interferencia de las nuevas tareas que se van aprendiendo). Por otro lado, se observa también la dificultad antes mencionada en el aprendizaje de la tercera tarea, a priori la más compleja.

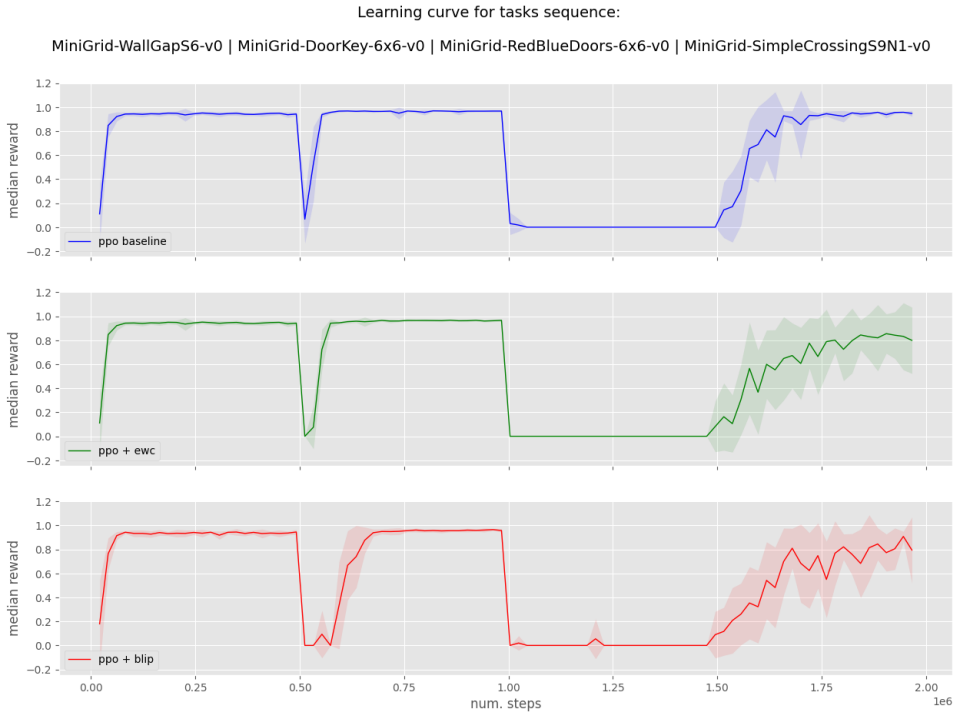


Figura 4.2: Curvas de entrenamiento para enfoques BLIP y EWC

4.2. Enfoques híbridos BLIP+EWC y BLIP+mask

Los experimentos con enfoques híbridos (sección 3.4) se han centrado por un lado en la combinación de BLIP y EWC, intentando adaptar los hiperparámetros de ambos métodos con la intención de buscar un efecto de sinergia. Se presentan dos evaluaciones en las que se ha variado el hiperparámetro λ específico de EWC que controla el grado de regularización EWC aplicado. Con respecto a BLIP+SPP (máscara de poda *suave*), este enfoque además de la mitigación propia de BLIP mediante congelación de bits en parámetros cuantizados, presenta también un componente regulador además del uso de la máscara de poda. Se ha optado por un valor relativamente bajo en el hiperparámetro λ (tabla B.1) que controla dicho grado de regularización, a fin de no restringir en demasía la plasticidad del modelo. La tabla 4.2 muestra los valores de las métricas tras la evaluación de los modelos entrenados en la secuencia de tareas:

Resultados destacados:

Enfoque	ACC	BWT
PPO-clip	$0{,}29201 \pm 0{,}098$	$-0{,}56598 \pm 0{,}13079$
BLIP+EWC (1)	$0{,}44815 \pm 0{,}16383$	$-4 \times 10^{-5} \pm 0{,}21835$
BLIP+EWC (2)	$0{,}51849 \pm 0{,}10407$	$-3 \times 10^{-5} \pm 0{,}06876$
BLIP+SPP mask	$0{,}54105 \pm 0{,}1462$	$-2 \times 10^{-5} \pm 0{,}1672$

Tabla 4.2: Resultados de métricas *ACC* y *BWT* para enfoques híbridos

.

- La línea de base **PPO-clipped**, sin mecanismos de mitigación del CF, mostró los peores resultados tanto en *ACC* como en *BWT*.

- En los enfoques híbridos, **BLIP+EWC** no se alcanzó el rendimiento esperado, mostrando una *ACC* superior a PPO pero relativamente baja. La *BWT* fue similar a la del enfoque BLIP individual.

- Por otro lado, la combinación **BLIP+SPP** muestra una ligera mejora en la *ACC*, manteniendo una mostrando *BWT* similar tanto a los otros enfoques híbridos, como alenfoque BLIP por separado, y ligeramente peor al enfoque EWC.

Con respecto a las curvas de evaluación, la situación es similar a la de los enfoques BLIP y EWC por separado. Se aprecia claramente la caída de rendimiento de la línea de base PPO (sin mecanismos de mitigación de CF) a medida que se aprenden tareas y se evalúa sobre las tareas anteriores. Con respecto a los enfoques híbridos, si mantienen el conocimiento de tareas previas (con excepción de la tercera tarea) aunque con cierta inestabilidad posiblemente asociada a la limitación impuesta en número de *steps* de entrenamiento.

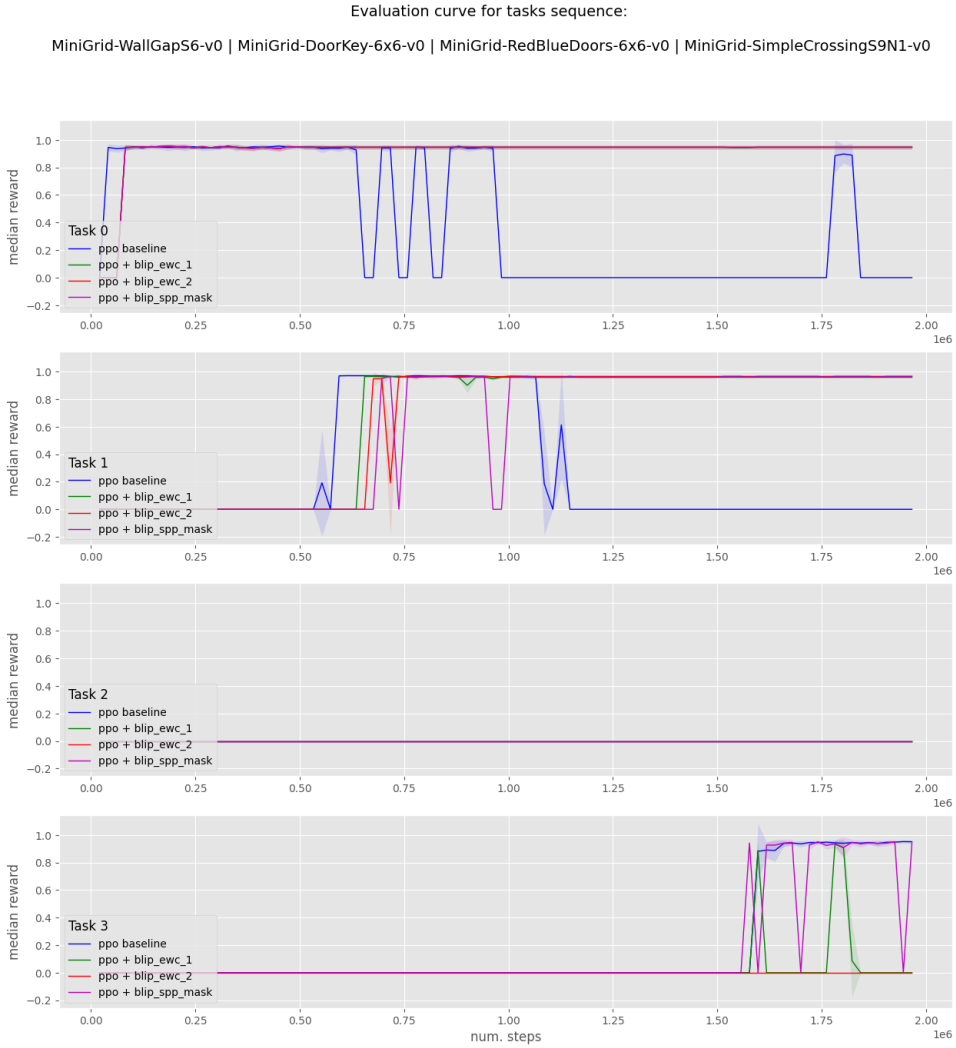


Figura 4.3: Curvas de evaluación por tareas para enfoques híbridos

Las curvas de entrenamiento son también muy similares a las de los enfoques por separado, en las que se observa cierta inestabilidad en el aprendizaje de la última tarea por parte de los enfoques híbridos debido al compromiso entre estabilidad y plasticidad. En cualquier caso, la tercera tarea (que requiere un mayor grado de exploración) no consigue ser aprendida dentro del rango de *steps* asignados.

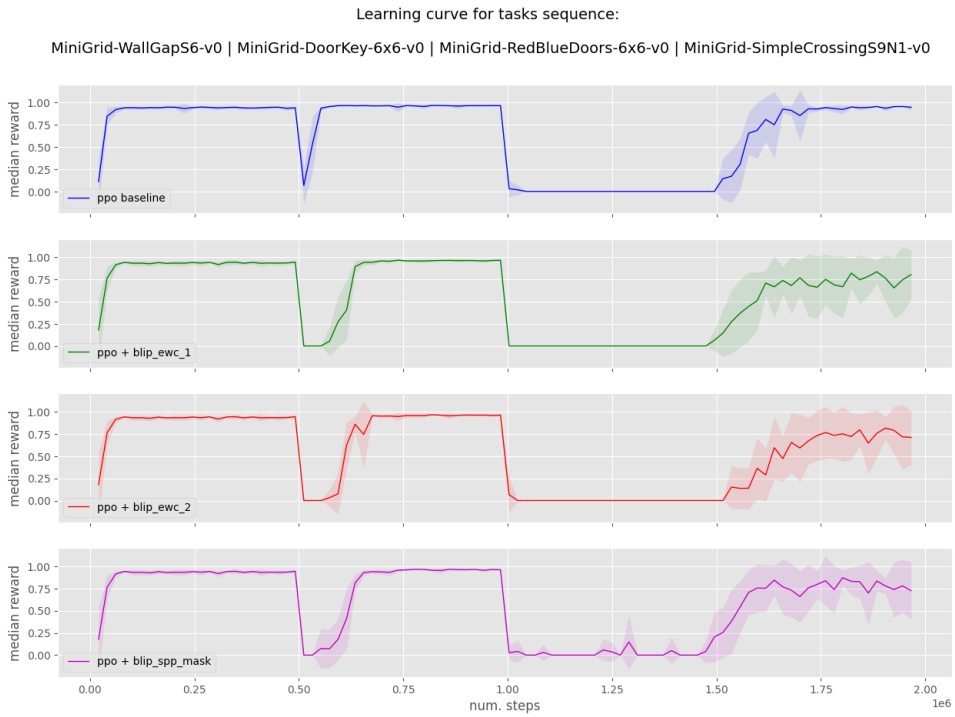


Figura 4.4: Curvas de entrenamiento para enfoques híbridos

4.3. Enfoque *Policy Consolidation*

El enfoque de Policy Consolidation ha sido implementado en PyTorch a partir del código original de los autores bajo TensorFlow, y se ha procedido a adaptarlo a entornos MiniGrid. Los hiperparámetros usados se presentan en la tabla B.2 dentro del Anexo B. Tal y como se ha descrito en la subsección 3.2.3, Policy Consolidation conlleva el uso de una *policy* visible y una serie de *policies* ocultas, inspirado en el proceso de consolidación de conocimiento que tendría lugar en las sinapsis biológicas. El uso de esta cadena de *policies*, que han de ser entrenadas durante todo el proceso de aprendizaje, conlleva a una mayor complejidad del modelo Actor-Critic, lo cual se ha traducido en unos tiempos de entrenamiento a idéntico número de *steps* ostensiblemente más dilatados que en los enfoques anteriores (tanto los enfoques individuales EWC y BLIP, como las hibridaciones propuestas). A fin de minimizar el tiempo de entrenamiento, se ha optado por entrenar modelos con 4 *policies* en cadena, en lugar de las 8 *policies* usadas en el artículo original. Tras realizar numerosas pruebas y ajustes, no ha sido posible identificar una configuración para que este enfoque sea efectivo bajo tareas Minigrid, como se verá a continuación.

La Tabla 4.3 muestra los resultados de las métricas tras completar la misma secuencia de tareas probada en los otros enfoques, con idénticos parámetros ($500,000$ *steps* para cada tarea).

- La línea de base **PPO-clipped**, sin mecanismos de mitigación del CF, mostró los peores resultados tanto en *ACC* como en *BWT*.

Enfoque	ACC	BWT
PPO-clip	$0{,}29201 \pm 0{,}098$	$-0{,}56598 \pm 0{,}13079$
PPOPC (1)	$0{,}2391 \pm 0{,}0014$	$-0{,}6458 \pm 0{,}0477$
PPOPC (2)	$0{,}2394 \pm 0{,}0008$	$-0{,}6576 \pm 0{,}0923$

Tabla 4.3: Resultados de métricas *ACC* y *BWT*

- PPOPC (1) implementa 4 *policies* y un número de 512 *frames-per-process*, superior al usado en la línea base PPO con una sola *policy* (un número por debajo no permitía al algoritmo completar los aprendizajes de las tareas por separado). En varias de las pruebas realizadas se comprobó que el algoritmo PPOPC aprendía de manera más estable al aumentar el valor de dicho hiperparámetro. Los resultados presentan unos valores poco prometedores tanto para la *ACC* como para la *BWT*, incluso por debajo de la línea de base PPO-clip.
- PPOPC (2) similar a PPOPC (1) pero con 1048 *steps-per-epoch*. Los resultados son prácticamente similares.

Las curvas de entrenamiento para las dos variaciones de PPOPC son similares a las de los enfoques anteriores, con una clara dificultad en el aprendizaje de la tercera tarea, e inestabilidad hacia el final de la secuencia de entrenamiento.

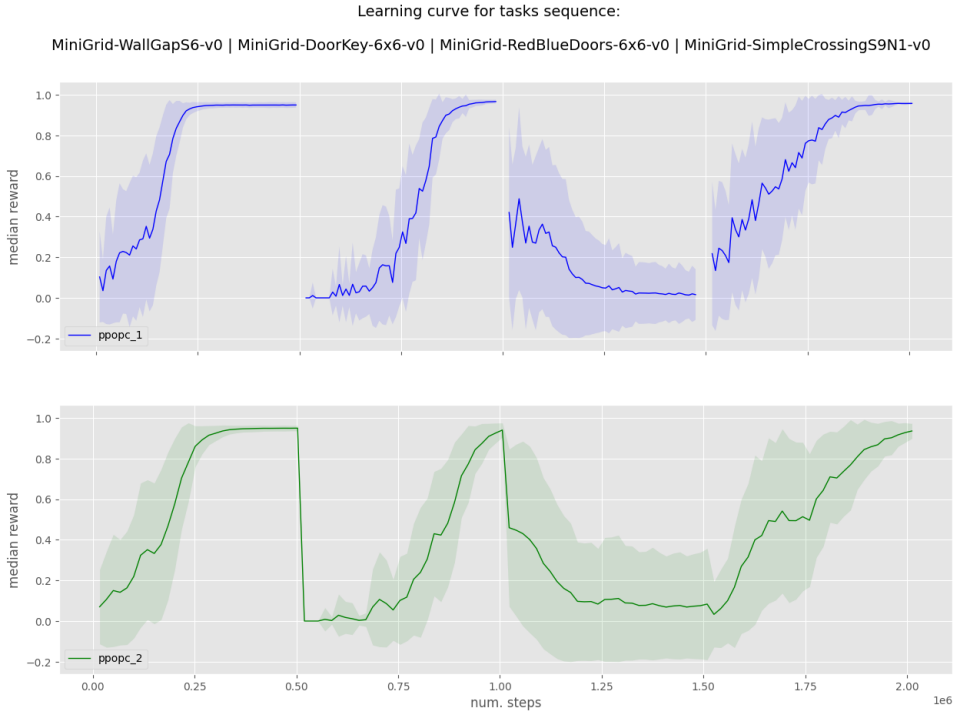


Figura 4.5: Curvas de entrenamiento para enfoque *Policy Consolidation*

Nota: Debido a un error en el *script* la evaluación simultánea al entrenamiento no fue registrada por *Tensorboard*, por lo que no se han podido incluir las curvas de evaluación. En cualquier caso las métricas *ACC* y *BWT* corresponde a la evolución del modelo a lo largo de

los diferentes *checkpoints* registrados, y son el indicativo cuantitativo de la precisión y resistencia al CF del enfoque evaluado.

4.4. Enfoque Meta-aprendizaje con RIMs

El enfoque de meta-aprendizaje con *Recurrent Independent Mechanisms* (RIMs) a priori no es un mecanismo específico para la mitigación del CF, sino orientado al CL a través de la implementación de un tipo de arquitectura modular basada en RIMs y reforzada mediante una estrategia de meta-aprendizaje (subsección 3.3.1). Dicha arquitectura aprovecharía mecanismos de atención con la intención de que la activación dinámica de los módulos permita atender diversos aspectos de las tareas y facilitar el aprendizaje de tipo *zero-shot* o *one-shot*. Se ha implementado el algoritmo sobre Pytorch directamente a partir de las indicaciones ofrecidas en el trabajo científico (Madan et al., 2021) y adaptado a los entornos Minigrid. Los hiperparámetros específicos para este modelo se muestran en la Tabla B.3 del Anexo B.

La Tabla 4.4 muestra los resultados de las métricas de evaluación tras el entrenamiento del modelo en la misma secuencia de tareas que en los enfoques anteriores:

Enfoque	**ACC**	**BWT**
PPO-clip	$0{,}29201 \pm 0{,}098$	$-0{,}56598 \pm 0{,}13079$
Meta-RIMs	$0{,}74139 \pm 0{,}10082$	$-0{,}13419 \pm 0{,}18674$

Tabla 4.4: Resultados de métricas para enfoque *Meta-learning RIMs*.

Los resultados obtenidos resultan prometedores, sobre todo teniendo en cuenta que dicho enfoque no ha sido desarrollado explícitamente para la mitigación del CF, siendo en este trabajo la primera vez que explícitamente se evalúa el grado de CF de dicha arquitectura. La *ACC* obtiene un valor relativamente alto, indicativo de que las cuatro tareas han podido ser aprendidas en buena medida. Por otro lado, la métrica *BWT*, que mediría el grado de CF, aun siendo negativa, presenta un valor relativamente bajo, por lo que aparentemente esta arquitectura retendría bien el conocimiento. Es probable que la propia filosofía de la arquitectura basada en atención, y reforzada con meta-aprendizaje haya logrado en cierta medida destilar los aspectos o habilidades más destacables para la resolución de cada tarea y haya podido conservarlos para su uso en tareas diferentes.

Las curvas de aprendizaje mostradas en la Figura 4.6 indican que en efecto el modelo ha podido completar en buena medida los aprendizajes consecutivos de cada tarea, incluida la tercera tarea (`MiniGrid-RedBlueDoors-6x6-v0`), la cual presentó grandes dificultades con los enfoques anteriores.

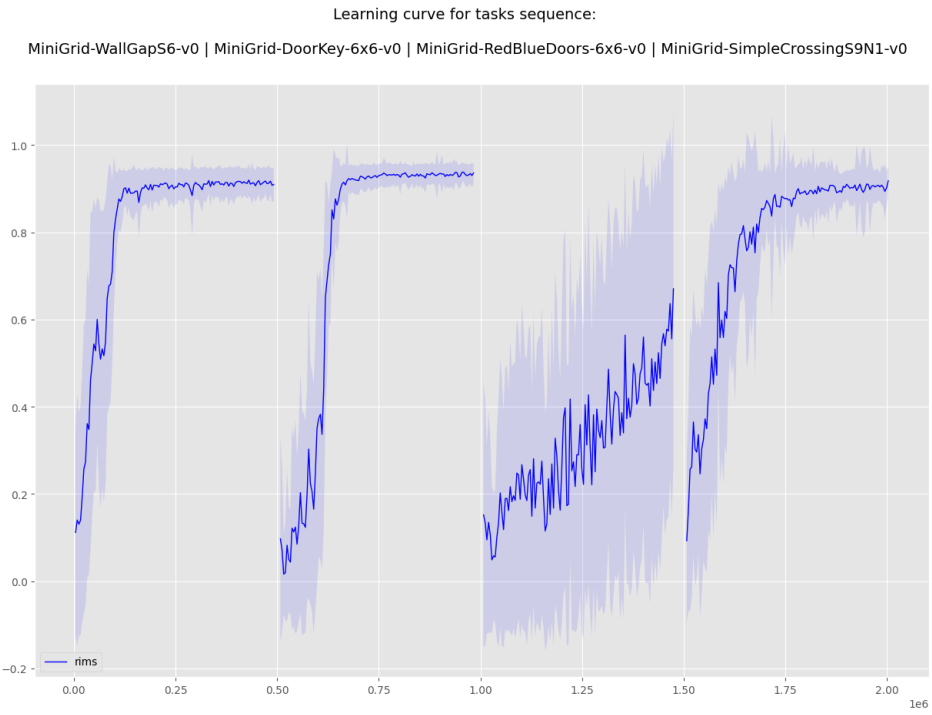


Figura 4.6: Curvas de entrenamiento para enfoque *Meta-learning RIMs*

Nota: Debido a un error en el *script* la evaluación simultánea al entrenamiento no fue registrada por *Tensorboard*, por lo que no se han podido incluir las curvas de evaluación. En cualquier caso las métricas *ACC* y *BWT* corresponde a la evolución del modelo a lo largo de los diferentes *checkpoints* registrados, y son el indicativo cuantitativo de la precisión y resistencia al CF del enfoque evaluado.

4.5. Enfoque HCAM

El enfoque *Hierarchical Chunk Attention Memory* (HCAM) (subsección 3.3.2), al igual que el Meta-aprendizaje RIMs, consiste en un tipo de arquitectura orientada al CL y el aprendizaje *zero-shot* o *one-shot*, más que un mecanismo para la mitigación del CF. El algoritmo fue implementado en PyTorch y adaptado a MiniGrid, con la intención de realizar una serie de baterías de pruebas como parte de la investigación en este trabajo sobre arquitecturas modulares. Desafortunadamente, debido a necesidades computacionales específicas no han podido ser completados todos los experimentos. La arquitectura base de HCAM está construida sobre una adaptación de modelos *Transformer* (en particular el modelo *Gated TransformerXL*). Los entrenamientos preliminares sobre CPU resultaron prohibitivos en cuanto a tiempo, por lo que inicialmente se realizaron una serie de tests sobre máquinas en la nube con acceso a tarjetas gráficas GPU, que ciertamente beneficiaron los entrenamientos proporcionando velocidades de incluso 20x con respecto a la CPU. El coste de dichos servicios, unido a su falta de disponibilidad debido a la alta demanda, ha impedido que se incluya aquí un análisis exhaustivo sobre esta interesante arquitectura.

4.6. Comparación y discusión de resultados

La Tabla 4.5 recopila las métricas evaluadas sobre todos los enfoques, incluida la línea de base PPO. A modo de resumen, una serie de consideraciones sobre los diferentes resultados:

Enfoque	ACC	BWT
PPO-clip	$0{,}29201 \pm 0{,}098$	$-0{,}56598 \pm 0{,}13079$
EWC	$0{,}61541 \pm 0{,}11525$	$0{,}17525 \pm 0{,}16087$
BLIP	$0{,}46271 \pm 0{,}16051$	$0{,}0 \pm 0{,}21832$
BLIP+EWC (1)	$0{,}44815 \pm 0{,}16383$	$-4 \times 10^{-5} \pm 0{,}21835$
BLIP+EWC (2)	$0{,}51849 \pm 0{,}10407$	$-3 \times 10^{-5} \pm 0{,}06876$
BLIP+SPP mask	$0{,}54105 \pm 0{,}1462$	$-2 \times 10^{-5} \pm 0{,}1672$
PPOPC (1)	$0{,}2391 \pm 0{,}0014$	$-0{,}6458 \pm 0{,}0477$
PPOPC (2)	$0{,}2394 \pm 0{,}0008$	$-0{,}6576 \pm 0{,}0923$
Meta-RIMs	$0{,}74139 \pm 0{,}10082$	$-0{,}13419 \pm 0{,}18674$

Tabla 4.5: Métricas de evaluación de los enfoques experimentados.

- **PPO-clip (línea de base)**: Como ya se ha comentado, los resultados entrarían de lleno en lo esperado. Si bien el algortimo PPO es bastante versátil y cuenta con una buena capacidad de generalización y aprendizaje, al no constar con ningún mecanismo específico de mitigación de CF, el rendimiento a medida que aprende las tareas disminuye claramente para las tareas previas, mostrando el peor valor de la métrica *BWT*

- **EWC**: A priori, es el mecanismo que mejor *BWT* ha mostrado, si bien con cierto detrimento en la precisión media sobre todas las tareas *ACC*. La ganancia que se observa en cuanto a estabilidad (consolidación de conocimientos previos), afectaría a la plasticidad del modelo, al enfocarse EWC en una regularización restrictiva. En cualquier caso, dicho grado de regularización puede controlarse desde un hiperparámetro y por otra parte, la relativa sencillez de su implementación e integración con otros mecanismos lo hace un buen candidato a tener en cuenta en futuros estudios.

- **BLIP**: Presenta un rendimiento algo peor que EWC, aunque el enfoque granular a nivel de bit es interesante. En cualquier caso, requiere de unas modificaciones muy específicas en los componentes del modelo Actor-Critic, a fin de cuantizar los parámetros durante el aprendizaje.

- Las combinaciones **BLIP+EWC**, en particular BLIP+EWC (2), demuestran que la relativa facilidad para la integración de dichos métodos en enfoques híbridos. Los resultados de la *BWT*, si bien algo peores, muestran también una menor desviación estándar, por lo que parece que dicho enfoque se ha comportado de manera relativamente estable ante diferentes semillas durante los entrenamientos.

- La combinación de **BLIP con una máscara de poda *suave*** (SPP o *Soft Parameter Pruning*), también ofrece resultados interesantes, con una precisión media algo superior a

BLIP+EWC, y una *BWT* similar. Abre la posibilidad de estudiar más en detalle este tipo de sinergia y en partícular el uso de máscaras de poda *suaves* (que no podan completamente un parámetro, este se sigue conservando dentro del modelo).

- En cuanto a los resultados **PPOPC (*Policy Consolidation*)**, no han sido para nada los esperados. Por otra parte se realizaron una serie de modificaciones que desviaron la implementación de su arquitectura original (por ejemplo, se usó PPO-clipped en lugar de PPO basado en divergencia KL) y por otra parte, el algoritmo original estaba muy enfocado a espacios de acciones continuas en lugar de los espacios discretos y parcialmente observables de MiniGrid.

- Finalmente la arquitectura **Meta-learning RIMs**, que como se ha indicado no estaba originalmente diseñada para la mitigación del CF, ha presentado unos resultados muy positivos, con la mayor precisión media *ACC* de todos los enfoques y con una *BWT* relativamente baja. El enfoque modular con mecanismos de atención y meta-aprendizaje parece ser ha facilitado el aprendizaje continuo en cierta medida, trayendo cierta mitigación del CF como valor añadido. Abre la posibilidad a seguir experimentando en esta dirección, en particular estudiando posibles hibridaciones de otros enfoques como BLIP o EWC como mecanismos adicionales dentro de estas arquitecturas.

Conclusiones

5

Este Trabajo de Fin de Máster se ha centrado en profundizar en el paradigma del Aprendizaje Continuo y su relación con el Olvido Catastrófico, un tema de creciente interés en el campo de la Inteligencia Artificial. Al abordar este tema, el trabajo ha buscado no solo entender estos fenómenos desde una perspectiva teórica y neurocientífica, sino también explorar y experimentar con estrategias prácticas para superar los retos que presentan. La intersección de estas áreas de estudio ha proporcionado una base sólida para el desarrollo de sistemas de IA más avanzados y ha abierto caminos hacia la realización de la Inteligencia Artificial General (AGI). En particular, y con relación a los Objetivos 1.3 iniciales, las contribuciones aportadas se pueden resumir a continuación:

1. **Contextualización del Aprendizaje Continuo y Olvido Catastrófico**: Este trabajo ha proporcionado una comprensión integral del paradigma del Aprendizaje Continuo, destacando su interrelación intrínseca con el fenómeno del olvido catastrófico. Se ha enfatizado su relevancia tanto para el avance en sistemas de Inteligencia Artificial (IA) más eficientes y robustos, como para el desarrollo potencial de la Inteligencia Artificial General (AGI). Además, se han explorado las bases neurocientíficas que subyacen al aprendizaje continuo, resaltando la interacción y el intercambio mutuo entre la neurociencia y la IA, proporcionando una perspectiva enriquecedora para entender mejor estas dinámicas.

2. **Estado del Arte en Mitigación del Olvido Catastrófico y Aprendizaje Continuo**: Se ha realizado una extensa revisión del estado actual en estrategias para mitigar el olvido catastrófico y promover el aprendizaje continuo. Esta revisión abarca una variedad de enfoques y técnicas, incluyendo aquellas enfocadas en el aprendizaje de tareas continuas y el aprendizaje a partir de pocos ejemplos. Especial atención se ha dado al aprendizaje por refuerzo, identificándolo como un marco ideal para la experimentación en este campo, dada su naturaleza y aplicabilidad en contextos de aprendizaje continuo.

3. **Contribuciones Específicas y Experimentación**: En el ámbito específico de este TFM, se han experimentado con diversos enfoques heterogéneos, tanto en la mitigación del olvido catastrófico como en el aprendizaje continuo. Este trabajo ha prestado especial atención a la aplicación de estos enfoques en el aprendizaje por refuerzo, particularmente en entornos con recompensas escasas y parcialmente observables, como es el caso del entorno MiniGrid. Si bien algunas de las hipótesis formuladas no han ofreci-

do los resultados inmediatos esperados en las experimentaciones, por otro lado se han identificado aspectos en los que profundizar dicho estudio y áreas de interés potencial para investigaciones futuras, como las arquitecturas modulares y el meta-aprendizaje, destacando su prometedor campo de investigación.

En resumen, este TFM ha contribuido significativamente al entendimiento del Aprendizaje Continuo y del olvido catastrófico, aportando tanto una revisión teórica como experimentación práctica en el campo. Las estrategias exploradas y los enfoques adoptados en este trabajo no solo subrayan la complejidad y la importancia de estos temas en la IA, sino que también señalan direcciones futuras prometedoras para la investigación. La sinergia entre los conocimientos aportados por las investigaciones en Neurociencia, Aprendizaje por Refuerzo y en particular sobre el paradigma del Aprendizaje Continuo, ofrecen el potencial para avanzar en la creación de sistemas de IA cada vez más adaptables, robustos y eficientes. Este trabajo pretende aportar su grano de arena en esta dirección y animar a futuras investigaciones que podrían llevar a innovaciones significativas en el campo de la IA.

Limitaciones y Perspectivas de Futuro

6

Una de las principales limitaciones encontradas en este trabajo, ha sido en la propia experimentación y desarrollo de los algoritmos y su implementación. Uno de los principales problemas ha sido la variabilidad y la sensibilidad de los algoritmos de RL a los hiperparámetros y a las configuraciones específicas del entorno, lo que ha hecho que la reproducción de resultados sea compleja. En parte debido a la propia sensibilidad inherente del RL donde detalles aparentemente menores pueden tener impactos significativos en los resultados, y su diagnóstico puede resultar complicado. Otro factor que ha afectado directamente a la experimentación es la necesidad de largos entrenamientos, los algoritmos de RL a menudo requieren una gran cantidad de datos de entrenamiento para lograr un rendimiento óptimo y esto tiene si cabe más impacto en el contexto del CL, donde se requiere experimentar con varias tareas más o menos complejas de manera secuencial. En particular, algunos de los experimentos (como los relacionados con arquitecturas HCAM) no han podido ser completados debido a los costes en computación. Son arquitecturas basadas en Transformers que se benefician enormemente de entrenamientos sobre GPUs, a menudo con limitada disponibilidad en los servicios de computación en la nube. Aun así, el uso de entornos ligeros 2D como MiniGrid, con multitud de diversas tareas fácilmente configurables, ha facilitado tanto la experimentación como la reproducibilidad de los resultados. Por otro lado, la investigación y experimentación en RL y en particular en CL, dada su amplitud en enfoques y técnicas, proporciona un campo muy fértil para profundizar en otras áreas dentro del ámbito del aprendizaje automático.

Las perspectivas futuras en el ámbito del Aprendizaje Continuo dentro del campo del Aprendizaje por Refuerzo son particularmente prometedoras, especialmente en lo que respecta a la investigación sobre el meta-aprendizaje, sobre el cual se ha experimentado en este trabajo, y que implica buscar algoritmos que puedan adaptarse rápidamente a nuevas tareas o entornos con un mínimo de intervención o datos adicionales. Esta capacidad de adaptación rápida es especialmente valiosa en el contexto del CL, donde el objetivo es desarrollar sistemas capaces de aprender continuamente sin olvidar conocimientos previos. Esto conlleva retos futuros en el desarrollo de algoritmos que imiten más de cerca los procesos de aprendizaje humano, como la capacidad de abstraer principios generales de experiencias pasadas y aplicarlos a situaciones nuevas.

Además, se espera que la investigación en CL y RL explore más a fondo la interacción entre diferentes módulos de aprendizaje, como la memoria, la atención y la toma de decisiones. Este enfoque holístico podría conducir a una mejor comprensión de cómo diferentes componen-

tes de aprendizaje interactúan y se influyen mutuamente en entornos dinámicos y continuos. Finalmente, se anticipa que la investigación futura se adentrará más en la comprensión y el desarrollo de arquitecturas y estrategias de optimización que sean más robustas y menos susceptibles a las fluctuaciones en los datos de entrada. Esto incluiría investigar enfoques más sofisticados para el balance entre la estabilidad y la plasticidad en los modelos de aprendizaje.

En resumen, el futuro del Aprendizaje Continuo en el ámbito del Aprendizaje por Refuerzo promete avances significativos en la creación de sistemas de IA más adaptables, eficientes y capaces de un aprendizaje genuinamente continuo y acumulativo.

Anexo A

Algoritmo 1: PPO-Clip

Require: initial policy parameters θ, initial value function parameters ϕ

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: Collect set of trajectories $D_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.

3: Compute rewards-to-go R^τ.

4: Compute advantage estimates, A^τ (using any method of advantage estimation) based on the current value function V_ϕ.

5: Update the policy by maximizing the PPO-Clip objective:

$$\theta_{\text{new}} = \arg\max_\theta \sum_{\tau \in D_k} \sum_t \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A^\tau(s_t, a_t),\right.$$
$$\left.\text{clip}(\epsilon, A^\tau(s_t, a_t))\right)$$

6: Typically via stochastic gradient ascent with Adam.

7: Fit value function by regression on mean-squared error:

8: Typically via some gradient descent algorithm.

9: **end for**

Algoritmo 2: Elastic Weight Consolidation (EWC)

Require: A neural network $\mathcal{N}$ with parameters θ, a dataset $\mathcal{D}$, a loss function $\mathcal{L}$, and a forgetting rate λ.

Ensure: A trained neural network $\mathcal{N}$ that resists catastrophic forgetting.

1: Initialize $\mathcal{N}$ and θ.

2: Set the Fisher information matrix $\mathcal{F} = \mathbf{0}$.

3: **for** each epoch e **do**

4: Sample a minibatch $\mathcal{B}$ from $\mathcal{D}$.

5: Compute the gradient of the loss function with respect to the parameters: $\nabla_\theta \mathcal{L}(\theta, \mathcal{B})$.

6: Update the Fisher information matrix: $\mathcal{F} \leftarrow \mathcal{F} + \lambda(\nabla_\theta \mathcal{L}(\theta, \mathcal{B}))^2$.

7: Update the parameters: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{B})/(1 + \lambda\mathcal{F})$.

8: **end for**

9: **return** $\mathcal{N}$

Algoritmo 3: Policy Consolidation Algorithm

Initialization:

Initialize policy networks $\pi_1, \pi_2, \ldots, \pi_N$ with parameters $\theta_1, \theta_2, \ldots, \theta_N$;

Initialize old policy networks $\pi_{1_{\text{old}}}, \pi_{2_{\text{old}}}, \ldots, \pi_{N_{\text{old}}}$;

Set hyperparameters λ, ω, α, and other relevant parameters;

Result: Trained policy networks $\pi_1, \pi_2, \ldots, \pi_N$

Training:

for each iteration **do**

  for $k = 1, 2, \ldots, N$ **do**

    Collect set of trajectories D_k by running policy π_k in the environment;

    Compute rewards and advantages based on the trajectories;

    Update policy π_k by maximizing PPO objective with added KL penalty;

    $L_{\text{PPO}}(\pi) = \mathbb{E}_t \left[\sum_{k=1}^{N} \lambda \omega^{k-1} D_{\text{KL}}(\pi_k \parallel \pi_{k_{\text{old}}}) \right]$;

    Compute KL divergence between π_k and $\pi_{(k+1)_{\text{old}}}$ and between π_k and $\pi_{(k-1)_{\text{old}}}$;

    Update policy π_k using the policy gradient with a KL penalty term;

    $L_{\text{CASC}}(\pi) =$
    $\mathbb{E}_t \left[\omega_{1,2} D_{\text{KL}}(\pi_1 \parallel \pi_{2_{\text{old}}}) + \sum_{k=2}^{N} \omega D_{\text{KL}}(\pi_k \parallel \pi_{(k-1)_{\text{old}}}) + D_{\text{KL}}(\pi_k \parallel \pi_{(k+1)_{\text{old}}}) \right]$;

    Update old policies $\pi_{k_{\text{old}}}$ to π_k for next iteration;

  end for

  Optionally, adjust λ and ω over time to control the consolidation strength;

end for

Algoritmo 4: Bit-Level Information Preserving (BLIP)

Input: Pre-defined quantization bits y, hyperparameters S_0, F_0, loss function L, learning rate α, quantization function Q

Output: Optimized parameters θ

Training:

$\theta \leftarrow$ randomly initialize parameters;
$S \leftarrow S_0, F \leftarrow F_0$;
for $t = 1$ **to** T **do**
  Obtain task & dataset $D_t = \{X, Y\}$;
  $\theta \leftarrow \text{TRAIN}(\theta, c, S, D_t, L, Q, \alpha)$;
  $IG, F_{\text{post}} \leftarrow \text{ESTIMATEINFOGAIN}(\theta, Q, F, t)$;
  $S_t \leftarrow \text{mín}(\text{máx}([IG], 0), N - S)$;
  $c \leftarrow Q(\theta, S_t), S \leftarrow S_t + m, F \leftarrow F_{\text{post}}$;
end for

Function `TRAIN`$(\theta, c, S, D, L, Q, \alpha)$**:**

  while *loss* L *not converged on* D **do**
    $\theta \leftarrow \theta - \alpha \nabla_\theta L(D, Q(c))$;
    $c \leftarrow \text{mín}(\text{máx}(\theta, c - S), c + S)$;
  end
  return θ;

Function `ESTIMATEINFOGAIN`(θ, Q, F, t)**:**

  $F_{\text{old}} \leftarrow F$;
  for *each data point* x *in* V **do**
    Sample $y \sim p_\theta(x)$;
    $R_{\text{old}} \leftarrow R_{\text{old}} + (\log p_{\theta_{\text{old}}}(y|x) - \log p_\theta(y|x))$;
  end
  $F_{\text{post}} \leftarrow tF + F_{\text{old}}$;
  $IG \leftarrow S \log y p_{\text{old}}$;
  return IG, F_{post};

Algoritmo 5: Meta-Learning Recurrent Independent Mechanisms

Require: $p(T)$: Distribution over tasks

Require: α, β: Hyperparameters for step size

Parameters: θ_A: Attention parameters, θ_M: Module parameters

Randomly initialize θ_A, θ_M;

while *not done* **do**

  Sample batch of tasks $T_i \sim p(T)$;

  Sample trajectories D_{T_i} from each task T_i;

  for *each task* T_i **do**

    $\theta_M = \theta_M - \alpha \nabla_{\theta_M} L(f_{\theta_M;\theta_A}(D_{T_i}))$; // θ_A remains unchanged

  end

  Sample trajectories τ_i from tasks $T_i \sim p(T)$ and concatenate to get $D_{meta} = [\tau_1, \tau_2, \ldots]$;

  $\theta_A = \theta_A - \beta \nabla_{\theta_A} L(f_{\theta_A;\theta_M}(D_{meta}))$; // θ_M remains unchanged

end

Algoritmo 6: Pseudo Code for ANPyC

Start with:

W_{old}: old task parameters

W: new task parameters

X, Y: training data and ground truth on the new task

tasks: total number of tasks

$H(q)$: information entropy of the output

H: Hessian matrix

η: coefficients to control the momentum

ϕ: threshold of salience of parameters for pruning

Training:

1: **for** each $t \in$ tasks **do**

2: $W_{\text{old}} \leftarrow W$; // Update the old task parameters

3: $\Omega^t_{i,j} \leftarrow \max(0, (\frac{\partial H(q)}{\partial W})^T \delta W + \frac{1}{2} \delta W^T H \delta W)$; // Calculate the importance of the parameters of the T-1 tasks

4: $\Omega^{1:t} \leftarrow \Omega^{1:t-1} + \Omega^t$; // Cumulative importance computation

5: Define: $\hat{Y} = f(X; W)$; // new task output

6: $W \leftarrow \arg\min_W L_{\text{new}}(Y, \hat{Y})$; // Update the new task parameters

7: $\nabla_\Theta \leftarrow \nabla_\Theta + \text{Momentum}$

8: $\text{Momentum} \leftarrow \lambda(-\Omega^{1:t} W + W)$; // Update the gradients

9: $\Theta \leftarrow \Phi\{\Omega > \beta\}$; // generate pruning mask

10: $M^{1:t+1} \leftarrow M^{1:t} \cap M^{t+1}$; // prune parameters

11: **end for**

Algoritmo 7: Hierarchical Chunk Attention Memory (HCAM)

Input:

$input\_sequence$: a sequence of inputs to the agent

$chunk\_size$: the fixed length of each memory chunk

num_chunks: the number of chunks to maintain in memory

Input:

$action$: the action chosen by the agent based on recalled memory

Initialization:

$memory\_chunks \leftarrow$ empty list

$chunk\_summaries \leftarrow$ empty list

$current\_chunk \leftarrow$ empty list

Main Execution Flow:

1: **for** each $input$ in $input\_sequence$ **do**
2: STORE(*input*)
3: **end for**
4: $action \leftarrow$ AGENT_ACT(*some_query*)
5: Execute $action$

Function STORE(*input*)**:**

Append input to current_chunk

if *length of current_chunk equals chunk_size* **then**

Add current_chunk to memory_chunks

Compute summary of current_chunk using AVERAGE_POOL

Add computed summary to chunk_summaries

Reset current_chunk to empty

end

Function RECALL(*query*)**:**

Identify relevant chunk indices using attention_mechanism with query and chunk_summaries

detailed_information ← empty list

foreach *index in relevant chunk indices* **do**

Retrieve chunk from memory_chunks using index

Apply attention_mechanism with query to the retrieved chunk

Append the result to detailed_information

end

return detailed_information

Function AVERAGE_POOL(*chunk*)**:**

return Compute and return the average of the elements in chunk

Function AGENT_ACT(*query*)**:**

detailed_information ← RECALL(*query*)

Use detailed_information to determine action

return action

Anexo B

B

Hiperparámetros	PPO	EWC	BLIP	BLIP+EWC (1)	BLIP+EWC (2)	BLIP+SPP
optimizer	Adam	Adam	Adam	Adam	Adam	Adam
# of processes	16	16	16	16	16	16
epochs	4	4	4	4	4	4
batch size	256	256	256	256	256	256
frames per process	128	128	128	512	1048	128
discount rate	0.99	0.99	0.99	0.99	0.99	0.99
learning rate	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.5e-4
gae lambda	0.95	0.95	0.95	0.95	0.95	0.95
entropy coefficient	0.01	0.01	0.01	0.01	0.01	0.01
value loss coefficient	0.5	0.5	0.5	0.5	0.5	0.5
max. gradient norm	0.5	0.5	0.5	0.5	0.5	0.5
optimizer eps	1e-8	1e-8	1e-8	1e-8	1e-8	1e-8
optimizer alpha	0.99	0.99	0.99	0.99	0.99	0.99
clip eps	0.2	0.2	0.2	0.2	0.2	0.2
EWC lambda	-	5000	-	1000	2500	-
EWC steps	-	20	-	20	20	-
EWC epochs	-	100	-	100	100	-
EWC online	-	True	-	True	True	-
Prior of Fisher information	-	-	5e-18	5e-18	5e-18	5e-18
SPP lambda	-	-	-	-	-	0.5

Tabla B.1: Hiperparámetros para PPO, EWC, BLIP, BLIP+EWC y BLIP+SPP

.

Hiperparámetros	**PPOPC (1)**	**PPOPC (2)**
optimizer	Adam	Adam
# of processes	16	16
epochs	4	4
batch size	256	256
frames per process	512	1048
discount rate	0.99	0.99
learning rate	2.5e-4	2.5e-4
gae lambda	0.95	0.95
entropy coefficient	0.01	0.01
value loss coefficient	0.5	0.5
max. gradient norm	0.5	0.5
optimizer eps	1e-8	1e-8
optimizer alpha	0.99	0.99
clip eps	0.2	0.2
cascade depth	4	4

Tabla B.2: Hiperparámetros para PPOPC

.

Hiperparámetros	**Meta-RIMs**
optimizer	Adam
# of processes	16
epochs	4
batch size	256
frames per process	128
discount rate	0.99
learning rate	2.5e-4
gae lambda	0.95
entropy coefficient	0.01
value loss coefficient	0.5
max. gradient norm	0.5
optimizer eps	1e-8
optimizer alpha	0.99
clip eps	0.2
num. modules (n)	6
active modules (k)	0.99
input attention heads	1
meta-learn inner recurrence	8
meta-learn outer recurrence	32

Tabla B.3: Hiperparámetros para *Meta-learning RIMs*

.

Bibliografía

Benna, M. K. y Fusi, S. (2016). Computational principles of synaptic memory consolidation. *Nature Neuroscience*, 19(12):1697–1706.

Biesialska, M., Biesialska, K., y Costa-jussà, M. R. (2020). Continual Lifelong Learning in Natural Language Processing: A Survey. https://arxiv.org/abs/2012.09823v1.

Chen, Z. y Liu, B. (2016). Lifelong Machine Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(3):1–145. https://www.morganclaypool.com/doi/abs/10.2200/S00737ED1V01Y201610AIM033.

Chen, Z. y Liu, B. (2018). *Lifelong Machine Learning*.

Chevalier-Boisvert, M., Dai, B., Towers, M., de Lazcano, R., Willems, L., Lahlou, S., Pal, S., Castro, P. S., y Terry, J. (2023). Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks. http://arxiv.org/abs/2306.13831.

Cossu, A., Carta, A., Lomonaco, V., y Bacciu, D. (2021). Continual Learning for Recurrent Neural Networks: An Empirical Evaluation. https://arxiv.org/abs/2103.07492v4.

Dong, H., Ding, Z., y Zhang, S., editors (2020). *Deep Reinforcement Learning: Fundamentals, Research and Applications*. Springer Singapore, Singapore. http://link.springer.com/10.1007/978-981-15-4095-0.

Fayek, H. M. (2019). *Continual Deep Learning via Progressive Learning - RMIT University*. PhD thesis. https://researchrepository.rmit.edu.au/esploro/outputs/doctoral/Continual-deep-learning-via-progressive-learning/9921864066401341?institution=61RMIT_INST.

Finn, C., Abbeel, P., y Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv:1703.03400 [cs]*. http://arxiv.org/abs/1703.03400.

French, R. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3:128–135.

Hebb, D. O. (1950). The Organization of Behavior; A Neuropsychological Theory. In *The American Journal of Psychology*, volume 63, page 633. https://www.jstor.org/stable/1418888?origin=crossref.

Hinton, G., Vinyals, O., y Dean, J. (2015). Distilling the Knowledge in a Neural Network. http://arxiv.org/abs/1503.02531.

Kaplanis, C., Shanahan, M., y Clopath, C. (2019). Policy Consolidation for Continual Reinforcement Learning. https://arxiv.org/abs/1902.00255v2.

Kaushik, P., Gain, A., Kortylewski, A., y Yuille, A. (2021). Understanding Catastrophic Forgetting and Remembering in Continual Learning with Optimal Relevance Mapping. https://arxiv.org/abs/2102.11343v1.

Khetarpal, K., Riemer, M., Rish, I., y Precup, D. (2020). Towards Continual Reinforcement Learning: A Review and Perspectives. https://arxiv.org/abs/2012.13490v1.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., y Hadsell, R. (2016). Overcoming catastrophic forgetting in neural networks. https://arxiv.org/abs/1612.00796v2.

Kostrikov, I. (2018). PyTorch Implementations of Reinforcement Learning Algorithms. https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail.

Lampinen, A., Chan, S., Banino, A., y Hill, F. (2021). Towards mental time travel: A hierarchical memory for reinforcement learning agents. In *Advances in Neural Information Processing Systems*, volume 34, pages 28182–28195. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2021/hash/ed519dacc89b2bead3f453b0b05a4a8b-Abstract.html.

Lomonaco, V. (2019). *Continual Learning with Deep Architectures*. PhD thesis.

Lopez-Paz, D. y Ranzato, M. (2017). Gradient Episodic Memory for Continual Learning. https://arxiv.org/abs/1706.08840v5.

Madan, K., Ke, R., Goyal, A., Schölkopf, B., y Bengio, Y. (2021). *Fast and Slow Learning of Recurrent Independent Mechanisms*. https://arxiv.org/abs/2105.08710.

McClelland, J. L. y O'Reilly, R. C. (1995). Why There Are Complementary Learning Systems in the Hippocampus and Neocortex:InsightsFrom the Successesand Failuresof Connectionist Models of Learning and Memory. *Psychological Review*, 102(3):39.

McCloskey, M. y Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem.

Mundt, M., Hong, Y. W., Pliushch, I., y Ramesh, V. (2020). A Wholistic View of Continual Learning with Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning. https://arxiv.org/abs/2009.01797v2.

Nichol, A., Achiam, J., y Schulman, J. (2018). On First-Order Meta-Learning Algorithms. *arXiv:1803.02999 [cs]*. http://arxiv.org/abs/1803.02999.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., y Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71. https://www.sciencedirect.com/science/article/pii/S0893608019300231.

Parisotto, E., Song, H. F., Rae, J. W., Pascanu, R., Gulcehre, C., Jayakumar, S. M., Jaderberg, M., Kaufman, R. L., Clark, A., Noury, S., Botvinick, M. M., Heess, N., y Hadsell, R. (2019). Stabilizing Transformers for Reinforcement Learning. http://arxiv.org/abs/1910.06764.

Peng, J., Hao, J., Li, Z., Guo, E., Wan, X., Min, D., Zhu, Q., y Li, H. (2018). Overcoming Catastrophic Forgetting by Soft Parameter Pruning. https://arxiv.org/abs/1812.01640v1.

Peng, J., Tang, B., Jiang, H., Li, Z., Lei, Y., Lin, T., y Li, H. (2019). Overcoming Long-term Catastrophic Forgetting through Adversarial Neural Pruning and Synaptic Consolidation. https://arxiv.org/abs/1912.09091v3.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., y Klimov, O. (2017). Proximal Policy Optimization Algorithms. http://arxiv.org/abs/1707.06347.

Shi, Y., Yuan, L., Chen, Y., y Feng, J. (2022). Continual Learning via Bit-Level Information Preserving. http://arxiv.org/abs/2105.04444.

Sutton, R. S. y Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, Massachusetts, second edition edition.

Wang, L., Zhang, X., Su, H., y Zhu, J. (2023). A Comprehensive Survey of Continual Learning: Theory, Method and Application. http://arxiv.org/abs/2302.00487.

Zhang, T., Wang, X., Liang, B., y Yuan, B. (2021). Catastrophic Interference in Reinforcement Learning: A Solution Based on Context Division and Knowledge Distillation. https://arxiv.org/abs/2109.00525v1.