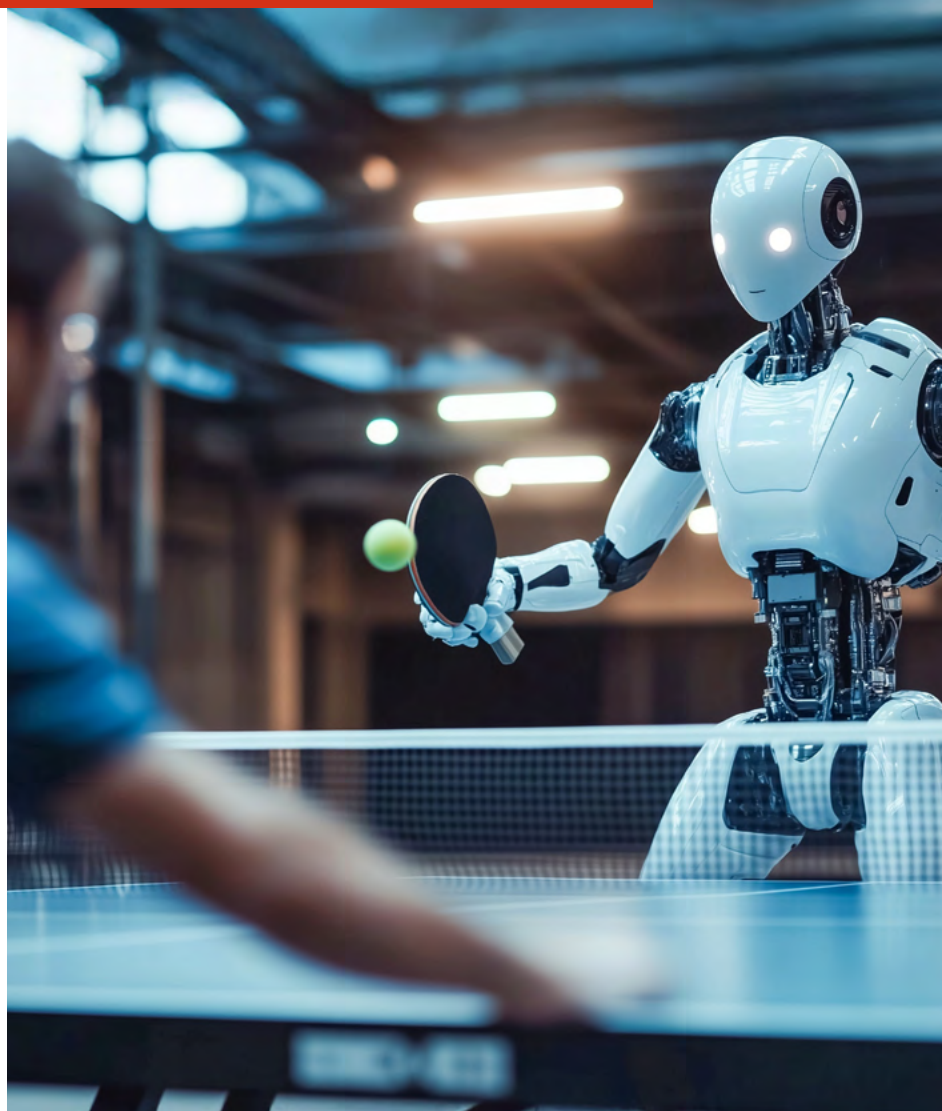


Premio Turing 2024:

Sutton y Barto, los pioneros
del aprendizaje por refuerzo



Javier Ruiz Del Solar

Doctor en Ingeniería por la Technische Universität Berlin, Alemania. Profesor Titular del Departamento de Ingeniería Eléctrica de la Universidad de Chile. Sus líneas de investigación incluyen aprendizaje de máquinas, robótica móvil y aplicaciones de robótica e inteligencia artificial en minería y ambientes industriales. Imparte los cursos Inteligencia Computacional, Procesamiento Avanzado de Imágenes y Robótica Móvil en la Universidad de Chile y es el líder de la línea de I+D de Autonomía Minera del Advanced Mining Technology Center (AMTC).

✉ jruizd@ing.uchile.cl

Resumen / En marzo de 2025, la Association for Computing Machinery (ACM) anunció que el Premio Turing 2024 —considerado el “Nobel de la Computación”— sería otorgado a Andrew G. Barto y Richard S. Sutton “por desarrollar los fundamentos conceptuales y algorítmicos del aprendizaje por refuerzo” [1]. Durante casi cinco décadas, ambos investigadores construyeron las bases de una disciplina que, a partir de ideas tomadas de la psicología animal y la teoría del control óptimo, hoy es protagonista central de los avances más espectaculares de la inteligencia artificial: desde AlphaGo y AlphaZero, pasando por robots que juegan tenis de mesa al nivel de jugadores profesionales, hasta el ajuste fino de modelos de lenguaje como ChatGPT mediante RLHF [2]. Este artículo presenta el aprendizaje por refuerzo, las contribuciones específicas de Sutton y Barto, y los principales hitos tecnológicos que su trabajo hizo posibles.

¿Qué es el aprendizaje por refuerzo?

El aprendizaje por refuerzo (en inglés, *reinforcement learning* o RL) es uno de los tres grandes paradigmas del aprendizaje de máquinas, junto al aprendizaje supervisado y al no supervisado. A diferencia del aprendizaje supervisado, en RL no se entrena al modelo con un conjunto de ejemplos y sus respuestas correctas. En lugar de eso, un **agente** interactúa con el **ambiente**: en cada instante observa el **estado** del ambiente, ejecuta una **acción**, y recibe una **recompensa** numérica que mide qué tan bueno fue lo que hizo. El objetivo del agente no es responder correctamente a una pregunta, sino aprender una **política** —es decir, una forma de elegir las acciones correctas para cada estado del ambiente— que maximice la suma esperada (futura) de recompensas a lo largo del tiempo, un concepto conocido como el **retorno** esperado (ver Figura 1).

Esta formulación, aparentemente simple, captura una clase enorme de problemas que comparten una característica común: la decisión secuencial bajo incertidumbre, donde las acciones de hoy condicionan las posibilidades de mañana. La inspiración biológica es directa. Cuando entrenamos a un perro con golosinas, cuando un niño aprende a caminar tropezándose o a andar en bicicleta cayéndose, seguimos el mismo esquema: probar, recibir retroalimentación, y ajustar el comportamiento. El psicólogo Edward Thorndike, a comienzos del siglo XX, formalizó esta idea con su “ley del efecto”: las acciones seguidas de consecuencias positivas tienden a repetirse, y las que producen consecuencias negativas tienden a evitarse [3]. El aprendizaje por refuerzo

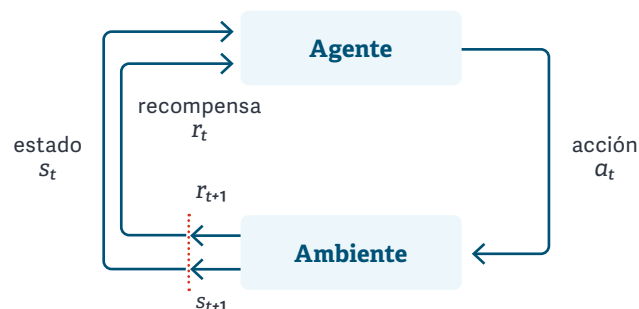


Figura 1 / Esquema del aprendizaje por refuerzo: el agente observa el estado s_t del ambiente, ejecuta una acción a_t , y recibe una recompensa r_{t+1} junto con el siguiente estado s_{t+1} . El objetivo es aprender una política $\pi(a/s)$ que maximice la recompensa acumulada esperada.

Cuando un niño aprende a caminar tropezándose o a andar en bicicleta cayéndose, seguimos el mismo esquema [que en reinforcement learning]: probar, recibir retroalimentación, y ajustar el comportamiento.

computacional, también llamado aprendizaje reforzado, es, en cierto sentido, una traducción matemática y algorítmica de este principio.

La diferencia del RL con el aprendizaje supervisado merece ser enfatizada. En el aprendizaje supervisado, el sistema recibe pares (entrada, respuesta correcta) y aprende una función que mapea las entradas en salidas para, por ejemplo, clasificar imágenes, predecir precios o traducir oraciones. En RL, en cambio, nadie le dice al agente cuál es la acción correcta; al ejecutar una acción, en un cierto estado del ambiente, el agente sólo recibe una señal escalar de recompensa, a menudo ruidosa y muchas veces *retardada*. A partir de esta recompensa debe aprender la política, es decir qué acciones elegir en cada situación/estado: la acción que realmente importó pudo haber ocurrido decenas o cientos de pasos antes de que se observe su consecuencia. Este es el famoso problema de *asignación de crédito temporal* (*temporal credit assignment problem*), uno de los desafíos centrales de RL.

¿Cuándo conviene usar RL? Típicamente, cuando se cumplen tres condiciones: (i) el problema es de toma de decisiones secuencial, con decisiones encadenadas; (ii) no se dispone de un conjunto de ejemplos etiquetados con la “respuesta correcta”, pero sí es posible interactuar con el ambiente (real o simulado) y obtener una señal de éxito o fracaso (recompensa); y (iii) se busca un comportamiento óptimo a largo plazo, no sólo una respuesta puntual. Robótica, control de procesos, juegos, gestión de recursos, vehículos autónomos, sistemas de recomendación, y trading financiero son ejemplos naturales de estas situaciones.

Las contribuciones de Sutton y Barto

Andrew Barto comenzó su carrera en la Universidad de Massachusetts Amherst en 1977. Richard Sutton llegó como estudiante de postgrado a ese mismo grupo en 1978 y obtuvo su doctorado bajo la supervisión de Barto en 1984,

con una tesis pionera sobre asignación de crédito temporal [4]. Lo que comenzó como una colaboración entre profesor y estudiante se transformó en una de las asociaciones científicas más fructíferas de la computación moderna, sostenida por más de cuarenta años.

A inicios de los ochenta, el aprendizaje por refuerzo era un área marginal, considerada por muchos como una curiosidad teórica sin mucha aplicabilidad. La inteligencia artificial dominante era la simbólica, basada en lógica y reglas; el aprendizaje conexionista comenzaba a renacer pero con foco en aprendizaje supervisado. Sutton y Barto, sin embargo, vieron en las ideas de la psicología animal y en los trabajos pioneros de Marvin Minsky, Arthur Samuel y A. Harry Klopff una vía distinta para construir agentes inteligentes. Sus contribuciones específicas pueden agruparse en tres grandes ejes.

Primero, los fundamentos conceptuales. Sutton y Barto fueron quienes formalizaron el problema de RL en los términos que hoy usamos universalmente: agente, ambiente, estado, acción, recompensa, política, función de valor. Establecieron además las conexiones profundas entre RL y otras tres disciplinas que estudiaban variantes del mismo problema sin saberlo: la programación dinámica (Bellman) en investigación de operaciones [5], el control óptimo en ingeniería, y el condicionamiento clásico e instrumental en psicología. Esta unificación no fue trivial, pero permitió importar herramientas matemáticas maduras —como las ecuaciones de Bellman y los procesos de decisión de Markov— al estudio del aprendizaje, dando rigor a un campo que hasta entonces se apoyaba en intuiciones.

Segundo, los algoritmos fundamentales. En 1983, Barto, Sutton y Anderson publicaron el trabajo seminal sobre la *arquitectura actor-crítico*, que separa la función de selección de acciones (el “actor”) de la función de evaluación del estado (el “crítico”) [6]. Esta arquitectura, aplicada al clásico problema de balancear un péndulo invertido, anticipó por décadas las técnicas que hoy entrenan modelos de lenguaje. En 1988, Sutton publicó “Learning to Predict by the Methods of Temporal Differences” [7], introduciendo formalmente el aprendizaje por diferencias temporales (*TD learning*) y el algoritmo $TD(\lambda)$. La idea es de una elegancia notable: en vez de esperar el resultado final de un episodio para corregir las predicciones, se ajustan las predicciones usando la diferencia entre la predicción actual y la siguiente. Es decir, el sistema aprende de sus propias predicciones futuras —*bootstrapping*—, lo que permite aprender de manera incremental y eficiente. A esto se sumaron contribuciones posteriores cruciales: la arquitectura Dyna (que integra aprendizaje, planificación y ejecución) [8], métodos de gradiente de política, el RL jerárquico, y los algoritmos de TD con gradiente.

Tercero, la divulgación y consolidación del campo. En 1998 Sutton y Barto publicaron “Reinforcement Learning: An Introduction”, el libro de texto que se convirtió rápidamente en la biblia indiscutida del área (ver Figura 2). Esta primera edición, junto a la segunda, de 2018 [9], expandida y actualizada, superan en conjunto las 95.000 citas en Google Scholar. El libro de Sutton y Barto no sólo enseñó RL a varias generaciones de investigadores; también fijó la notación, ordenó la taxonomía de los algoritmos, y conectó RL con neurociencia, psicología y control. Es difícil imaginar la explosión actual del área sin esa base pedagógica común.

Aplicaciones del aprendizaje por refuerzo

Como se mencionó anteriormente, el RL es una herramienta apropiada de aprendizaje para problemas donde un sistema/agente debe tomar decisiones secuenciales en ambientes inciertos. Entre los principales ámbitos de aplicación con estas características destacan:

Agentes y juegos. Los juegos —desde los clásicos de tablero hasta los videojuegos modernos— son un ambiente ideal de aplicación para el RL: reglas conocidas, recompensas bien definidas (ganar o perder), y posibilidad de simulación masiva. Es el dominio donde RL ha mostrado sus resultados más espectaculares y mediáticos.

Robótica. Aprender políticas de control para manipulación, locomoción, navegación y agarre. Aquí RL enfrenta dificultades adicionales: la interacción con el ambiente es costosa (cada intento/prueba toma tiempo y desgasta el hardware), los espacios de estados y acciones son continuos y de alta dimensión, y la transferencia de simulación a realidad (*sim-to-real*) es compleja de realizar.

Procesamiento de lenguaje natural. La revolución reciente en grandes modelos de lenguaje incorpora el RL como pieza clave del entrenamiento, no para la fase inicial de pre-entrenamiento, sino para *alinear* el comportamiento del modelo con las preferencias humanas.

Optimización industrial y de sistemas. Desde la regulación de centros de datos (Google reportó reducciones de hasta 40% en consumo energético de enfriamiento usando RL), hasta el ruteo de tráfico, la programación de plantas químicas, y el diseño automatizado de chips —área en que el propio equipo de Google usó RL para colocación de componentes en circuitos integrados.

Economía y finanzas. Estrategias de mercado, gestión de carteras, fijación dinámica de precios y subastas en línea



Figura 2 / Portada de la primera y segunda edición de Reinforcement Learning: An Introduction de Sutton y Barto (MIT Press, 1998 y 2018), considerado el texto canónico del área, con más de 95.000 citas.



Figura 3 / Portadas de revistas Nature en los que se presentan hitos importantes de la aplicación del aprendizaje por refuerzo: febrero de 2015, aplicación de RL en juegos de video Atari (izquierda), enero 2016, primera vez que un sistema de RL venció a un jugador profesional humano en el juego de Go (derecha).

son problemas naturalmente formulables como decisión secuencial bajo incertidumbre.

Salud. Diseño de protocolos de tratamiento personalizados, donde cada decisión clínica afecta el estado futuro del paciente; aquí, sin embargo, el desafío ético y de seguridad es central, porque no se puede experimentar libremente con pacientes reales.

Hitos y avances habilitados por RL

Si los fundamentos los pusieron Sutton y Barto en los ochenta y noventa, la explosión práctica del RL ocurrió en

El libro de Sutton y Barto no sólo enseñó reinforcement learning (RL) a varias generaciones de investigadores; también fijó la notación, ordenó la taxonomía de los algoritmos, y conectó RL con neurociencia, psicología y control.

los últimos quince años, gracias a la combinación de los algoritmos RL con redes neuronales profundas, paradigma conocido como *deep reinforcement learning* (DRL). La aplicación del DRL hizo posible los hitos siguientes:

De Atari a AlphaGo, AlphaZero y MuZero. Entre 2013 y 2015, DeepMind demostró que un agente de RL profundo podía aprender a jugar decenas de videojuegos de Atari utilizando únicamente la información visual de la pantalla como entrada, alcanzando o superando el nivel humano [7]. El salto cualitativo llegó en 2016, cuando AlphaGo derrotó al campeón mundial Lee Sedol en Go, un juego que se consideraba inalcanzable por su gigantesca complejidad combinatoria. Sin embargo, esta versión de AlphaGo aún dependía del conocimiento humano, ya que fue entrenada con millones de jugadas de expertos [8]. La versión siguiente, AlphaGo Zero (2017), aprendió a jugar Go partiendo desde cero, exclusivamente por self-play —jugando contra sí misma— y superó ampliamente a la versión original. AlphaZero (2018) generalizó el enfoque a ajedrez y shogi con el mismo algoritmo, sin conocimiento específico del dominio más allá de las reglas [12]. Más recientemente, MuZero aprendió a jugar incluso sin que se le entregara explícitamente el modelo del juego: lo descubre por sí mismo. El éxito de estos sistemas ha tenido impactos más allá del ámbito algorítmico, por ejemplo, están cambiando la forma en que los propios jugadores humanos de Go entienden el juego, y han sido destacados en portadas de revistas como Nature (ver Figura 3), en las que hasta ahora era impensado ver artículos sobre sistemas de aprendizaje de máquinas.

Póker y juegos con información incompleta. Mientras Go, ajedrez y shogi son juegos de información perfecta, el póker introduce incertidumbre sobre las cartas del rival. En 2019, Pluribus —desarrollado por la Universidad Carnegie Mellon y Facebook AI— derrotó a varios profesionales de élite en póker Texas Hold'em sin límite de seis jugadores, combinando *self-play* con técnicas de búsqueda y abstracción [13]. Anteriormente, Libratus había hecho lo propio en la variante de dos jugadores. Estos resultados son particularmente nota-

bles porque los juegos con información incompleta exigen razonar sobre las creencias (*beliefs*) del oponente y aleatorizar la propia estrategia para no ser predecible.

Robots que juegan tenis de mesa. Quizás ningún resultado reciente ilustra mejor el potencial del RL en robótica que el robot de tenis de mesa presentado por DeepMind en 2024: el primer sistema robótico entrenado mediante IA capaz de jugar competitivamente al nivel de un jugador humano amateur. El sistema combina aprendizaje por refuerzo en simulación, *fine-tuning* en hardware real y políticas jerárquicas de bajo y alto nivel [14]. En 2026, Ace, un robot aún más avanzado, reportó victorias contra jugadores profesionales en condiciones de competencia oficial [15]. El tenis de mesa es un banco de prueba ideal para la aplicación de RL en problemas de control en tiempo real: requiere percepción a alta velocidad, control fino y adaptación al adversario, todo en escalas de fracciones de segundo.

Locomoción y manipulación robóticas. Más allá de demostraciones espectaculares, el uso de DRL ha permitido a robots cuadrúpedos y humanoides aprender a caminar y correr en terrenos complejos, recuperarse de empujones, y manipular objetos delicados. Equipos como los de Boston Dynamics, ETH Zürich, NVIDIA y diversos laboratorios académicos han mostrado que el ciclo “entrenar masivamente en simulación usando DRL, transferir a la realidad” es hoy una metodología viable y reproducible. Para el campo de la robótica móvil, esto representa un cambio de paradigma respecto a los métodos de control clásicos.

Robots que juegan fútbol. En la liga de fútbol robótico RoboCup, participan desde hace décadas equipos conformados por robots autónomos. En los últimos años, varios equipos han incorporado RL para aprender habilidades específicas —patear, driblar, posicionarse— que son difíciles de programar a mano. Trabajos recientes de DeepMind con humanoides Robotis OP3, así como aproximaciones académicas en la liga SPL (*Standard Platform League*), han demostrado que RL permite cerrar la brecha entre el comportamiento simulado y el real.

Grandes modelos de lenguaje y RLHF. Posiblemente la aplicación de mayor impacto cultural reciente sea el uso de RL en el entrenamiento de modelos de lenguaje. La técnica conocida como *Reinforcement Learning from Human Feedback* (RLHF) fue introducida por OpenAI hacia 2017 [16] y popularizada masivamente con InstructGPT y ChatGPT [2]. La idea es la siguiente: tras el preentrenamiento con aprendizaje supervisado, el modelo se afina mediante un esquema que considera tres pasos: (i) ajuste fino supervisado usando entrenadores humanos que crean ejemplos de prompts (preguntas) y respuestas ideales; (ii) entrenamien-

to de un *modelo de recompensa* a partir de la evaluación de parte de humanos de las respuestas del modelo en base a utilidad, veracidad y seguridad; y (iii) optimización del modelo de lenguaje mediante un algoritmo de RL —típicamente PPO [17], utilizando el modelo de recompensa entrenado en el paso (ii). Es decir, los modelos que hoy usamos para conversar, programar y redactar fueron *alineados* con preferencias humanas mediante un descendiente directo de los algoritmos actor-crítico que Sutton y Barto introdujeron hace cuatro décadas. ChatGPT, Claude, Gemini y otros sistemas comerciales descansan en este paradigma.

Conclusión

El otorgamiento del Premio Turing 2024 a Sutton y Barto es un reconocimiento a su trabajo científico sistemático. Durante los años ochenta y noventa, el aprendizaje por refuerzo era visto como un nicho minoritario, especulativo, demasiado conectado con la psicología animal para ser tomado en serio por la corriente principal de la inteligencia artificial. Sutton y Barto, junto a un grupo reducido de colaboradores, sostuvieron la convicción de que un marco basado en interacción, recompensa y aprendizaje por ensayo y error era

no sólo biológicamente plausible sino computacionalmente potente. Construyeron los conceptos, los algoritmos y los textos necesarios para que el campo madurara.

Hoy, el RL se encuentra en el centro de los sistemas de inteligencia artificial más avanzados que existen: desde los agentes que dominan los juegos de estrategia, pasando por los robots que aprenden a realizar tareas complejas en el mundo físico, hasta los modelos de lenguaje que conversan con miles de millones de personas. Detrás de cada uno de estos sistemas hay, en algún lugar, una ecuación de Bellman, una regla de actualización TD, o una arquitectura actor-crítico cuyo origen puede rastrearse a un par de artículos publicados hace cuarenta años.

El mensaje de fondo trasciende el premio. Es una invitación al trabajo serio y sistemático, a profundizar en conexiones interdisciplinarias incluso cuando no están de moda, y a relevar el hecho de que los libros que recopilan sistemáticamente lo que hay que saber sobre un área son una forma de contribución científica de enorme alcance. Para quienes enseñamos inteligencia computacional y robótica, el caso de Sutton y Barto es un recordatorio doblemente útil: que los fundamentos importan, y que vale la pena enseñarlos bien. **B**

Referencias

- [1] ACM. "Andrew Barto and Richard Sutton are the recipients of the 2024 ACM A.M. Turing Award", 2025. <https://awards.acm.org/about/2024-turing>.
- [2] Ouyang, L., et al. "Training Language Models to Follow Instructions with Human Feedback" (InstructGPT). NeurIPS, 2022.
- [3] Thorndike, E. L. Animal Intelligence: Experimental Studies. Macmillan, 1911.
- [4] Sutton, R. S. "Temporal Credit Assignment in Reinforcement Learning". Tesis de Doctorado, University of Massachusetts Amherst, 1984.
- [5] Bellman, R. E. Dynamic Programming. Princeton University Press, 1957.
- [6] Barto, A. G., Sutton, R. S., y Anderson, C. W. "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems". IEEE Transactions on Systems, Man, and Cybernetics, SMC-13(5):834–846, 1983.
- [7] Sutton, R. S. "Learning to Predict by the Methods of Temporal Differences". Machine Learning, 3:9–44, 1988.
- [8] Sutton, R. S. "Dyna, an integrated architecture for learning, planning, and reacting". SIGART Bull., vol. 2, no. 4, pp. 160–163, Jul. 1991.
- [9] Sutton, R. S., y Barto, A. G. Reinforcement Learning: An Introduction. Segunda edición. MIT Press, Cambridge MA, 2018.
- [10] Mnih, V., et al. "Human-Level Control through Deep Reinforcement Learning". Nature, 518:529–533, 2015.
- [11] Silver, D., et al. "Mastering the Game of Go with Deep Neural Networks and Tree Search". Nature, 529:484–489, 2016.
- [12] Silver, D., et al. "A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go through Self-Play". Science, 362(6419):1140–1144, 2018.
- [13] Brown, N., y Sandholm, T. "Superhuman AI for Multiplayer Poker". Science, 365(6456):885–890, 2019.
- [14] D'Ambrosio, D. B., et al. "Achieving Human Level Competitive Robot Table Tennis". DeepMind. arXiv:2408.03906, 2024.
- [15] Dürr, P., El Gheche, M., Maeda, G.J. et al. "Outplaying elite table tennis players with an autonomous robot". Nature 652, 886–891 (2026).
- [16] Christiano, P., et al. "Deep Reinforcement Learning from Human Preferences". NeurIPS, 2017.
- [17] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., y Klimov, O. "Proximal Policy Optimization Algorithms". arXiv:1707.06347, 2017.